

Department of Precision and Microsystems Engineering

The use of a rigid linkage balancer with torsion springs to realize nonlinear moment-angle characteristics

Sjors van Nes

Report no : 2022.074
Coach : ir. A. Amoozandeh Nobaveh
Professor : Prof. dr. ir. J.L. Herder
Specialisation : MSD
Type of report : Master Thesis
Date : October 31, 2022

The use of a rigid linkage balancer with torsion springs to realize nonlinear moment-angle characteristics

by

Sjors van Nes

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on November 14, 2022.

Student number: 4694481
Project duration: August 16, 2021 – November 14, 2022
Thesis committee: Prof. dr. ir. J.L. Herder, TU Delft, chair
ir. A. Amoozandeh Nobaveh, TU Delft, daily supervisor
Dr. ir. C. Ayas, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

The master thesis report I hereby present concludes my Mechanical Engineering program with the High - Tech Engineering track. This thesis, done within the ShellSkeletons group, is made possible by the support of others as well. Without the intention to be complete, I will thank these others in the following.

First of all, I would like to thank Ali for his support and patience during this project. For me, our progress meetings were of great value. I would also like to thank the other members of the research group for thinking along with me. This especially holds for Giuseppe, as he was my second supervisor. My gratitude to my parents and sisters should be expressed as well. Lastly, I thank my friends for the necessary distraction and entertainment.

Sjors van Nes
Delft, October 2022

Contents

1	Introduction	1
2	Literature survey	5
3	Research paper	25
4	Discussion	39
4.1	Discussion research paper	39
4.2	Discussion application as exoskeleton	39
4.3	Recommendations for future research	40
5	Conclusion	41
A	Geometric analysis	45
B	FBD three segment balancer	47
B.1	Segment 1.	48
B.2	Segment 2.	49
B.3	Segment 3.	50
B.4	Reaction forces	50
C	FBD four segment balancer	53
C.1	Segment 4.	53
C.2	Reaction forces	54
D	Release of contact	57
E	Lagrange	59
F	SAM	61
G	Solidworks	63
G.1	Solidworks model assembly.	63
G.2	Solidworks model parts	64
H	Assembly	79
H.1	Installation axes.	79
H.2	Connection main axis with environment	80
H.3	Mounting double springs	80
H.4	Constraining upside down double springs	80
I	Spring design	83
J	Spring evaluation	85
J.1	Spring evaluation preparation	85
J.2	Spring evaluation results	87
J.3	Spring evaluation discussion	88
K	LabVIEW code	89
L	TU cluster	91
M	Optimization results	93
M.1	Optimization without contact.	93
M.2	Optimization with contact	94
M.3	Optimization with relaxed lower- and upperbounds for segment lengths	94

M.4 Optimization results elaborated.	95
M.4.1 Sine	95
M.4.2 Progressive.	96
M.4.3 Progressive-degressive.	97
M.4.4 Degressive-progressive.	98
M.4.5 Laevo	100
M.4.6 Sine, 180 degree	101
N MATLAB code	103
N.1 Spring selection.	103
N.2 Work reduction	104
N.3 Three segment balancer	104
N.4 Four segment balancer	125

1

Introduction

Much research effort has been done to design mechanisms that support the human body, either in an active or passive way. A support mechanism reduces the muscular force that is required for a certain action. This support is needed in case of muscular weakness or a muscular disease. An example of such a disease is Duchenne muscular dystrophy [1], which causes the muscular capacity to decrease as the disease progresses. Support mechanisms are also of use in industries with heavy physical labor or work with repetitive motion. As the muscle activity can be decreased, the risk on injuries and eventual incapacity for work reduce as well. State of the art support mechanisms realize, for example, the support of the human arm [2] [3], neck [4] and lower body [5].

A remarkably large part of the population, 60 to 80% of the adults, is confronted with low back disorders once in its life. Lowering and lifting activities are expected to be important potential causes of these disorders [6]. The negative effect of these lowering and lifting activities can be mitigated by the use of a back support. Examples of assistant devices that support the lower back are the exoskeletons made by Laevo [7], as depicted in figure 1.1. The latter mechanisms are designed to statically balance the human back throughout the range of motion that is expected to be repeated most of the time. A mechanism is said to be statically balanced if it is in static balance for all possible configurations in its range of motion [8]. Correspondingly, the potential energy is constant throughout the range of motion and no actuation energy is needed anymore. In the case of the human back, the center of mass (COM) of the upper part of the body experiences a displacement during forward bending. As the COM is no longer aligned with the hip, a moment is induced by the gravitational force. As the orthogonal component of the distance between the hip and the COM is described by a sine, the induced moment is a sine as well. If the human back would be statically balanced, the same sine moment is exerted in reverse direction by the exoskeleton. Otherwise, muscles in the back should provide a reaction force in order to attain equilibrium. In case of perfect balancing by the exoskeleton, theoretically no reaction force or moment is required to realize forward bending of the torso.

In most cases, static balance is achieved with use of counterweights or springs. Counterweights are frequently used in gravity balancers, where the gravitational force and induced moment of another mass are balanced. Disadvantages of systems that comprise these counterweights are increased volume, mass and inertia of the total system [9]. Statically balanced mechanisms with springs do not have these disadvantages as springs, instead of masses, are used to store and release energy. A complication of the latter mechanisms could be the dependency on zero free length springs [10] [11], which are no off-the-shelf products and thus complicate the mechanism design. Alternatively, static balance could be achieved by implementing a linear spring with nonzero initial length and a transmission between the spring and the to be balanced mechanism. The spring will thus have a linear load-displacement characteristic, but nonlinear characteristics could be obtained by plotting the spring force against the displacement at the transmission output. The Laevo exoskeletons are designed via this approach as well, as energy is stored in gas springs and a cam is used as a transmission.

Another example of a lower back support is the SPEXOR [12] [13] [14], which is an exoskeleton that stores energy in a series connection of two elastic elements. The first is embodied by a parallel connection of multiple beams that store energy during bending and release energy during the reverse motion. The compliant beams thus facilitate an energy distribution, whereas it is concentrated in a spring in the Laevo device. The second element is a helical spring that is connected with a pulley to realize a degressive relation

of the provided moment and the rotation of the hip joint. The SPEXOR exoskeleton is shown in figure 1.2a. The compliant beams are seen in the top right corner of the figure, whereas the helical spring is represented in blue at the height of the legs of the wearer. A third example of a passive exoskeleton that supports the human back is the PLAD [15] [16] [17], which is called a soft exoskeleton. This exoskeleton consists of rubber bands that store energy during forward bending and release that energy during the reverse motion. An image of the device is included in figure 1.2b.

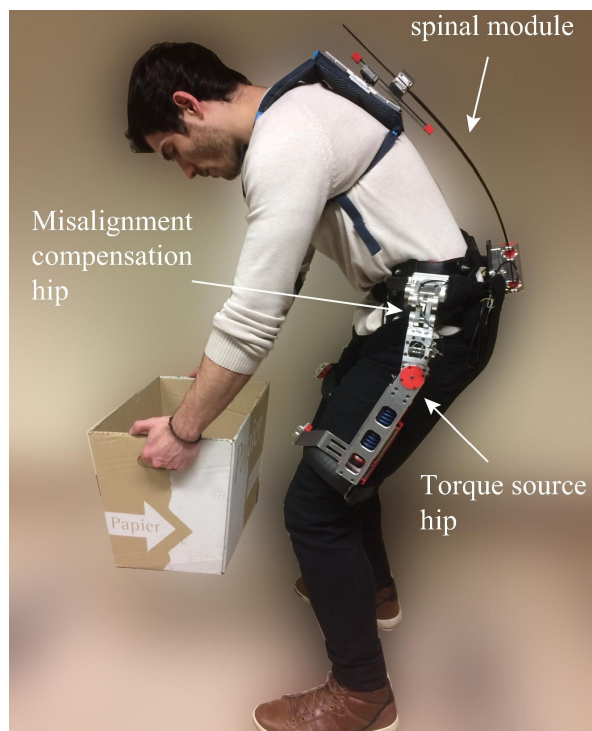


(a) Industrial application Laevo exoskeleton

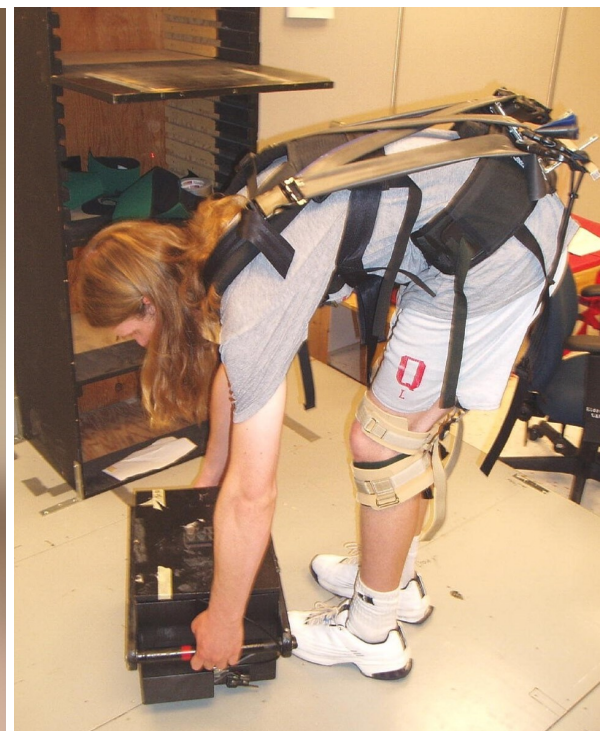


(b) Agricultural application Laevo exoskeleton

Figure 1.1: Laevo exoskeletons as examples of mechanisms that support the human back [7]



(a) SPEXOR exoskeleton [13]



(b) PLAD exoskeleton [17]

Figure 1.2: SPEXOR and PLAD as examples of exoskeletons that support the human back

Although the presented exoskeletons are able to provide support and conformability to the human body, the working principle of the Laevo and SPEXOR devices are based on a pulley that is used as a transmission of forces and displacements. The latter increases the complexity and the size of the exoskeleton. The PLAD does not have this disadvantage, but the moment reduction for larger angles of flexion-extension is reported to be only 19.5%. The SPEXOR, on the other hand, realizes a work reduction around the hip of only 18-25% [12]. The work reduction corresponding to the Laevo device could not be found in literature, but the reduction in

back muscle activity is reported to be 44% [18].

It is expected that it would be of great value to present an exoskeleton that does not require cams or a similar transmission, while the energy required for forward bending is reduced by a percentage that is close to 100%. Furthermore, it would be advantageous to be able to distribute the storage of potential energy in the exoskeleton. The latter is expected to improve the inherent safety of the device. Although the SPEXOR and PLAD exoskeletons make use of distributed energy storage in compliant beams and rubber bands, respectively, concentrated energy storage and peak stresses are likely to occur close to the attachment of the compliant segment with a rigid body. Stresses and material fatigue are thus not easily controlled. This especially holds for the SPEXOR, as the sections of the beams with high curvature will experience relatively high stresses.

In this thesis, a mechanism is proposed that could balance the human back without requiring pulleys or other transmissions. The moment induced by the mass of the torso is balanced by using the internal degree of freedom of a multi-linkage balancer. More specifically, the objective of this work is to examine the possibilities to statically balance various nonlinear moment-angle characteristics by this kinematically indeterminate rigid body balancer with torsion springs and to verify the results that are obtained by the proposed method with an experimental setup that contains a prototype of the system.

The to be evaluated system is shown in figure 1.3. Subfigure 1.3a visualises the proposed mechanism that will function as the balancer. It consists of three segments and three torsion springs that interconnect these segments. By connecting the balancer with an inverted pendulum, as shown in subfigure 1.3b, the four bar mechanism as shown in subfigure 1.3c is created. This four bar has two degrees of freedom in total. One of these DOF is an internal degree of freedom, which enables the balancer to provide balancing moments other than a linear characteristic.

A review on zero stiffness compliant path generation mechanisms is included in chapter 2. Zero stiffness mechanisms are analogous to constant force mechanisms, which require a constant actuation load throughout their range of motion. Statically balanced mechanisms have constant potential energy and require a constant actuation force that is equal to zero. Statically balanced mechanisms could thus be interpreted as a subset of zero stiffness mechanisms. The literature survey focuses on the more general principle of zero stiffness, which should hold during a predefined motion of the mechanism. Mechanisms that are designed to follow a certain path with their end effector are called *path generation mechanisms*. Zero stiffness path generation mechanisms are of particular interest for this work, as a human back has its own COM that describes a path in the sagittal plane during forward bending. In addition, this rotation should be statically balanced and should thus have zero stiffness. Although the proposed mechanism of this work is a rigid body balancer, the literature survey analyzes the state of the art on compliant solutions because of their typical advantages in terms of reduced or eliminated backlash, increased reliability and reduced maintenance [19] [20]. Subsequently, chapter 3 will provide the research paper regarding the analysis of the proposed balancer. The results will be interpreted from the perspective of the research paper and in a broader view as a potential contribution to an exoskeleton in the discussion, found in chapter 4. Gathered information that is omitted in this paper is discussed in the appendices, which will succeed the conclusion in chapter 5.

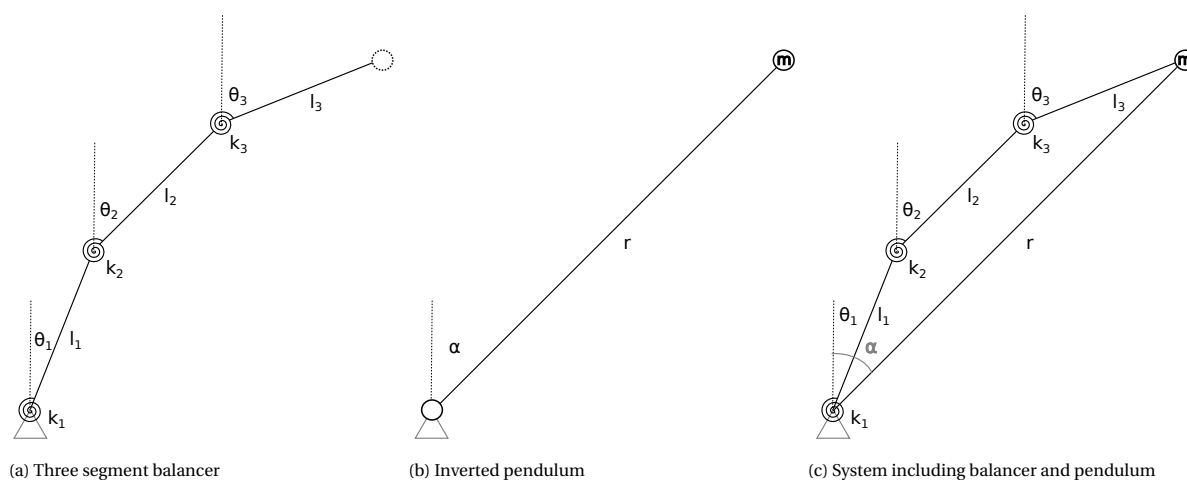


Figure 1.3: Schematic overview of the system

2

Literature survey

A review on zero stiffness compliant path generation mechanisms

Sjors van Nes

Abstract—

Although compliant mechanisms typically have several advantages compared to traditional rigid body mechanisms, a part of the input energy is used in the compliant members to enable motion by elastic deformation. As a result, more energy is applied at the input of the system than is received at the output. Static balancing could resolve this problem by providing energy to the system that could compensate for the strain energy in the compliant members. Static balancing results in a constant potential energy level, no residual forces and zero stiffness in the balanced direction. This work provides a literature review on zero stiffness compliant mechanisms that can be used to describe a path in a planar or spatial range of motion. The mechanisms are categorised based on their range of motion, the location of compensation energy and the type of compliance. The properties and performance of the examples are tabulated to facilitate a convenient comparison. Most planar examples store the required compensation energy internally, whereas the energy is stored in a partially compliant external mechanism in most spatial cases. The majority of the mechanisms have a linear force-deflection characteristic in the unbalanced configuration and demonstrate a stiffness reduction in the range of 80% - 100%.

I. INTRODUCTION

Compliant mechanisms are mechanisms that realise their displacements by elastic deformations. As a compliant mechanism does not rely on hinges, it has several advantages compared to a rigid body mechanism. Examples of these advantages are potential cost reduction because of monolithic production, increased reliability, reduced weight and reduced maintenance [1] [2]. A disadvantage of compliant mechanisms is the energy required by their elastic deformations. Although the energy is not dissipated in conservative systems, a part of the input energy of the system does not reach the output. As a result, the efficiency is decreased and the actuation force is larger than the force perceived at the output of the mechanism. Static balancing of compliant mechanisms could, however, mitigate this problem as the stored elastic energy is compensated by a certain amount of compensation energy in the compliant members [3]. Statically balanced mechanisms possess a (locally) constant potential energy level in their range of motion. Because of this constant potential energy, the system is in continuous equilibrium and has zero stiffness in the corresponding loading direction.

Several works that discuss the state of the art of statically balanced compliant mechanisms have been published. In their article “On zero stiffness”, Schenk and Guest discuss several examples of zero stiffness based on different interpretations [4]. The different interpretations are continuous equilibrium, constant potential energy, neutral stability and

zero stiffness. Although the highlighted examples have the same characteristics, each example is discussed via the most applicable interpretation.

Dunning et al. reviewed the literature on statically balanced compliant precision stages [5]. In the discussion on statically balanced compliant precision stages, an elaborate overview is made. This overview lists the characteristics of the found stages, such as: flexure type, size of the stage and the range of motion. It appeared that a statically balanced compliant 6- DOF stage does not exist yet. The stages with six degrees of freedom are either not compliant or not statically balanced.

Hogervorst classified zero stiffness compliant joints based on their working principle and the type of compliant joint [6]. The off-axis stiffness and axis drift were compared in a qualitative manner, whereas the zero stiffness error, range of motion and the volume were the quantitative performance criteria.

Linszen provided an overview of single element neutrally stable compliant mechanisms with the focus on their kinematics [7]. Only shell and ring mechanisms were found under the restrictions to be neutrally stable, compliant and single piece. These single element mechanisms were categorised based on deformation type (local or global), deformation dimension (planar or spatial), motion range (finite or infinite) and the extractable mechanism motion (translation and/or rotation).

Examples of other qualitative literature reviews on neutrally stable mechanisms are the work of Kok [8] and Dekens [9].

Kok made a division between single element and multiple element mechanisms. The multiple element mechanisms were classified as having linear opposed load curves or nonlinear opposed load curves. The single element mechanisms were also separated in two groups based on their working principle: application of prestress or application of boundary conditions. Furthermore, the subclasses of single element mechanisms were discriminated based on their range of motion (infinite or finite). Dekens used less categorisation than Kok and distinguished two-dimensional and three-dimensional zero stiffness mechanisms, also without mentioning the performance of the discussed examples.

Doornenbal et al. reviewed prestressing techniques for compliant shell mechanisms with tailored stiffness [10]. Both mechanisms with negative stiffness and mechanisms with zero stiffness were discussed. The potential and advantages of rolling, casting/ injection moulding, laser forming, tempering with gradient, curing, curing + ply-steering, chemical treatment, stretched fibres, viscoelastic fibres and combining

layers were compared in a qualitative manner.

Daynes and Weaver summarised different prestress solutions for achieving tailored stiffness [11]. A distinction is made between in-plane and out-of-plane prestressing. Furthermore, the examples are classified as “structure with prestress” or “material with prestress”. Although the categorisation made by Daynes and Weaver could be very useful in general, the scope of the review is merely on adaptive composite structures and no special attention is paid to neutrally stable structures.

Similar to Daynes and Weaver, Staats presented an overview of methods that provide controllable stiffness in structures [12]. Staats categorised the structures based on the controllable stiffness direction and the working principle. The presented examples are from different fields of research and of different phases of their development. The performance of the examples is elaborately discussed as well. Like in the work of Daynes and Weaver, obtaining neutral stability is not seen as a research objective. As a result, most of the structures are not statically balanced.

The properties of the discussed literature review papers on zero stiffness are summarised in table I. Most reviews discuss examples of zero stiffness whereas the work of Daynes and Weaver is more directed towards composites from an aerospace perspective and the review of Staats is dedicated to the tuning of a structure’s stiffness. Furthermore, it is observed that most literature surveys on zero stiffness do not include any performance evaluation. The surveys that do include a performance evaluation are focused on precision stages and rotary joints. Although the relevance of neutrally stable behaviour in rotary joints and precision stages is obvious, a more general analysis of the state of the art with a quantitative performance comparison is still missing. Such an overview would enable a designer to gain quick and thorough knowledge on different solutions and their potential.

TABLE I: State of the art of review papers on zero stiffness

Work	Performance evaluated	Zero stiffness	Focus
Schenk and Guest [4]	×	✓	Equivalent interpretations
Dunning et al. [5]	✓	✓	Precision stages
Hogervorst [6]	✓	✓	Rotary joints
Linssen [7]	×	✓	Single element, kinematics
Kok [8]	×	✓	Working principles
Doornbal[10]	Qualitative ranking	Partly	Production process
Dekens [9]	×	✓	General
Daynes and Weaver [11]	×	×	Adaptive composites
Staats [12]	✓	×	Controllable stiffness

Although the literature on statically balanced compliant mechanisms is relatively sparse, considerable attention has been paid to straight-line motion mechanisms [13] [14] [15] [16] [17] [18] [19] [20]. As table I presents a gap in review papers that do a performance evaluation of zero stiffness

mechanisms in general and relatively much attention has been paid to straight-line motion mechanisms, this work will elaborate on zero stiffness mechanisms that do not describe a straight line. Focusing on one point of interest on the mechanism, this kind of mechanism could be referred to as a compliant path generator. Furthermore, the performance in terms of the stiffness reduction is evaluated. To facilitate a convenient comparison, the mechanisms will be categorised based on their range of motion, the location of energy storage and the type of compliance. Therefore, the objective of this work is to present the state of the art on zero stiffness compliant mechanisms that could be used for path generation and to evaluate and compare their stiffness reduction.

Chapter II will elaborate on the methods that are used to realise the mentioned objective. In chapter III, the found literature on zero stiffness compliant mechanisms is discussed. To preserve readability and a proper overview, the work is categorised in sections. The results are then discussed in the discussion, chapter IV. Lastly, conclusions are given in chapter V.

II. METHODS

The discussed papers will be categorised as illustrated in figure 1. This categorisation is based on the work of Herder and Van den Berg [21]. In this work, a categorisation of statically balanced compliant mechanisms was presented. Four different categories were described: compliant elements with conventional compensation mechanisms, compliant compensation mechanisms, internal compensation energy and adaptive mechanisms. A “compensation mechanism” is a mechanism that is used to store the required compensation energy for static balancing. During elastic deformation of the members, energy is extracted from the compensation mechanism and used to compensate for the storage of strain energy in the elastic members. In the case of internal compensation energy, no dedicated energy storage mechanism is used but the energy is stored in the compliant mechanism itself. Adaptive statically balanced compliant mechanisms are mechanisms that remain statically balanced under different load conditions. Except for the latter category, the categorisation by Herder and Van den Berg will be used here as well as it provides insight into the type of compliant mechanism and the possibility of monolithic production. Moreover, this categorisation method would isolate the mechanisms with conventional compensation mechanisms from the other design solutions. This is important as friction, encountered more frequently in conventional mechanisms, could jeopardise neutral stability. Apart from the previously mentioned categorisation, the mechanisms are categorised as having a planar or spatial range of motion and lumped or distributed compliance. From a design perspective, this information is indispensable as it determines the mechanism’s practical applicability. In the following, the categorisation is summarised in the order in which the mechanisms are discussed.

The first distinction will be made based on the range of motion of the mechanism. The mechanism will be classified

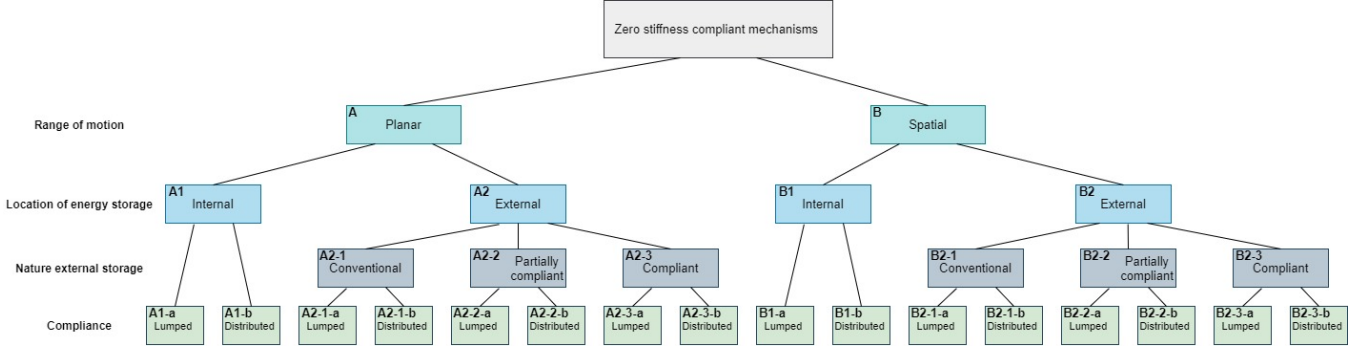


Fig. 1: Division of zero stiffness compliant mechanisms into (sub-) categories

as planar if the degrees of freedom of the point(s) of interest are in one plane, otherwise the mechanism will be considered spatial. Subsequently, the examples are grouped based on the location of the compensation energy storage. In the case of external energy storage, the distinction between compliant, partially compliant and non-compliant compensation mechanisms will be made. A partially compliant compensation mechanism realises its displacement by elastic deformation of the members but is, on the other hand, still dependent on pins or hinges and is therefore not fully compliant. The last categorisation step is based on the type of compliance. To that end, the original compliant mechanism with nonzero stiffness is analysed to evaluate if the mechanism displaces by lumped compliance or distributed compliance.

Moreover, the synthesising method, type of force-deflection characteristic, range of motion and stiffness reduction are tabulated in table II and table III. The conventional synthesising methods for compliant mechanisms are: kinematic approach, building blocks approach and structural optimisation [22] [23]. These methods are used to design statically balanced compliant mechanisms as well, albeit a modified version of the method. The force-deflection characteristic of the unbalanced examples are classified as linear or nonlinear as it appeared that the positive stiffness of the unbalanced mechanisms was constant in relatively much cases. The range of motion indicates the range of motion of the point of interest or the end effector. The stiffness reduction is the change in stiffness of the statically balanced mechanism with respect to the unbalanced mechanism. Equation 1 is used in case of a derivation of the stiffness reduction.

$$k_{red} = 100 \frac{k_{sb} - k_p}{k_p} \quad (1)$$

The stiffness of the statically balanced mechanism is represented by k_{sb} in equation 1, whereas the stiffness of the unbalanced mechanism is denoted as k_p . The reduction is expressed in percents. In case of unknown k_{sb} and/or k_p , the average stiffness can be derived by determining the average slope in the given force-displacement characteristic graph. To that end, discrete points on the graph are tabulated and a polynomial is fitted through these points in Matlab. Sequentially, the derivative of the equation of the polynomial with respect to the corresponding degree of

freedom is evaluated and its average value is calculated. By following this procedure for both the reference and the statically balanced configuration the stiffness reduction can be calculated by applying equation 1. The cells of tables II and III are coloured to obtain quick insight into the source of the data. The specifications in green cells are adopted from the corresponding paper as they are explicitly mentioned or shown in the work. The data in the orange cells is derived from other information in the paper. The red cells do not contain any data as no information about that subject is given in the publication and no reliable derivation could be made.

III. RESULTS

The table in figure 2 provides the amount of literature examples per sub-category of figure 1. It is observed that more planar mechanisms than spatial mechanisms were found. Most planar mechanisms are classified as distributed compliance mechanisms with internal compensation energy. The second largest group of planar mechanisms uses a fully compliant external compensation energy storage and deforms by distributed compliance. A relatively small amount of lumped compliance planar mechanisms was collected, whereas the class of lumped compliance mechanisms with a fully compliant storage mechanism is even empty. Furthermore, no work on spatial zero stiffness mechanisms with lumped compliance was found. Only one example utilises internal compensation energy. Although represented by only two examples, the category with partially compliant external energy storage mechanisms is the largest spatial category. In the following, the examples will be discussed per category. Each category will have its own section with a number that corresponds to the numbers provided in black in figure 1 and the table in figure 2.

A. Planar zero stiffness compliant mechanisms

Relatively much work done on zero stiffness compliant mechanisms is related to the design of grippers. Although compliant grippers offer several advantages compared to traditional grippers, as briefly touched upon earlier, the elastic energy stored in the members increases the operating effort and distorts the force feedback of the mechanism [24] [25]. A statically balanced gripper could solve these problems and would therefore be of great value in, for

Range of motion	Planar				Spatial			
	Internal	External			Internal	External		
Conventional		Partially compliant	Compliant	Compliant		Partially compliant	Conventional	
Lumped compliance	A1-a 2	A2-1-a 1	A2-2-a 1	A2-3-a 0	B1-a 0	B2-1-a 0	B2-2-a 0	B2-3-a 0
Distributed compliance	A1-b 11	A2-1-b 3	A2-2-b 2	A2-3-b 7	B1-b 1	B2-1-b 0	B2-2-b 2	B2-3-b 0

Fig. 2: Amount of literature examples per sub- category

example, the medical and agricultural sectors. As a matter of fact, a surgeon could use a statically balanced gripper to be able to sense undisturbed reaction forces of the patients tissue. This enhanced feedback could ultimately result in less damage to the tissue and qualitatively better operations. The agricultural sector would benefit from a zero stiffness gripper as the gripper can be used as a constant force mechanism to grasp delicate fruits or crops [1]. Because of this constant force, no control system is needed anymore to measure the applied force.

A1-a. Planar, internal compensation energy, lumped compliance

Soroushian et al. designed a constant force spring with pseudoelastic behaviour [26]. The spring was made of a Nickel- Titanium alloy, also called Nitinol. Nitinol is an example of a shape memory alloy: an alloy that is able to recover it's original shape when subjected to a temperature field. Upon heating a phase transition occurs: from the relatively easy deformable martensitic structure toward the stiffer austenitic phase [27]. Furthermore, Nickel- Titanium is thus also classified as a pseudoelastic alloy [28]. Without any further temperature gradient, a pseudoelastic material experiences an austenitic- martensitic phase transition when mechanically loaded [28] [29]. During this transition the material possesses an approximately constant stress plateau, as illustrated in figure 3. By using the designed Ni- Ti spring in this constant stress region, constant force behaviour is obtained. The eventual design of the spring is represented in figure 4. The design consists of six flexible parts interconnected by rigid members. These rigid members are realised by bracing the elements that should not deflect. As a result, the deflections are very localised and the material is expected to be subjected to pure bending. The annealing, quenching and aging parameters were derived by an optimization programm using response surface analysis. Although the experimental results seem to show nearly constant force behaviour, no performance details are mentioned. It should be noted that the example summarised here does not have constant potential energy behaviour as the mechanism is not statically balanced. It does, however, illustrate the realisation of a constant force region and it could therefore be used as

a zero stiffness spring.

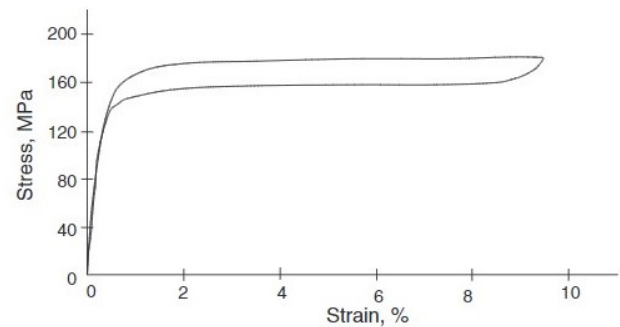


Fig. 3: Common stress- strain characteristic of a pseudoelastic material [26]

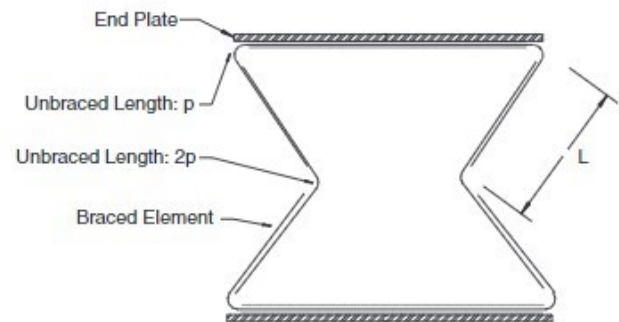


Fig. 4: Constant force spring by Soroushian et al. [26]

Merriam et al. designed a statically balanced compliant pantograph consisting of two neutrally stable four bar mechanisms [30]. The prototype of the mechanism can be seen in figure 5. The pantograph can be actuated at point A through a statically balanced domain of more than 100° of rotation. To realise the neutral stability of the mechanism, the constituent four- bar mechanisms were optimised by a genetic algorithm. The genetic algorithm was coupled to a FEM model to evaluate the performance of the possible configurations. The objective of the optimisation was to minimise the difference between the torque- deflection curve of the design and the desired constant torque- deflection curve equal to zero. The four- bar mechanisms were prestressed as they consist of two separate parts that are deflected upon mutual connection. The strain energy, which is used as compensation energy for the eventual energy storage in the compliant members, is stored in the small- length flexural pivots. The pantograph consists of two four bar mechanisms, such that the off- axial stiffness is increased.

A1-b. Planar, internal compensation energy, distributed compliance

Lan and Wang designed adjustable constant force forceps for medical applications [31]. The grasping part of the forceps is a rigid body linkage, but the constant torque mechanism providing the constant actuation force is compliant. A vi-

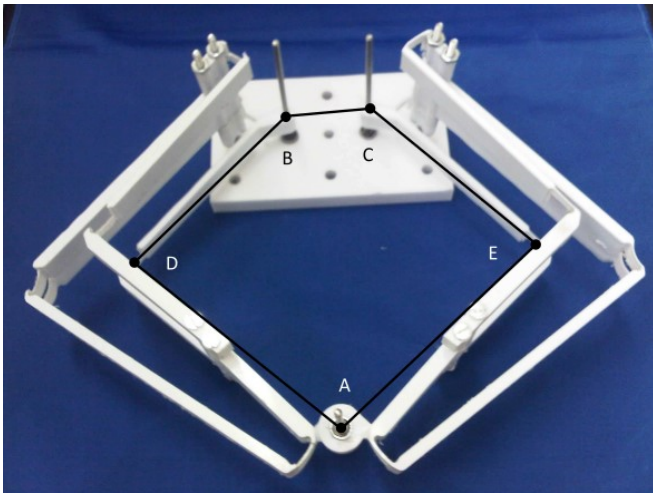


Fig. 5: Neutrally stable compliant pantograph by Merriam et al. [30]

sualisation of the concept is given in figure 6. In the right part of the figure, the constant torque mechanism can be seen. This constant torque mechanism is connected to the forceps by a wire. The connection- points of the wire are at “Slider A” and “Slider B”. As the wire is attached to the circumference of the constant torque mechanism, the force in the wire can be adjusted by manipulating the length between the attachment point at slider B and the center of the constant torque mechanism (CTM). A linear motor is employed to change this distance. A torque is applied in the center of the CTM, which is compensated by the force in the wire at slider B. Four flexible arms realise the constant torque behaviour. As the arms are identical and symmetrically positioned in the mechanism, only one flexible arm is optimised by an optimisation routine to obtain the appropriate values of the design points along the arm and thus the general shape. The range of motion with approximately constant force, defined as less than 5% deviation from the average force, is reported to be 26° .

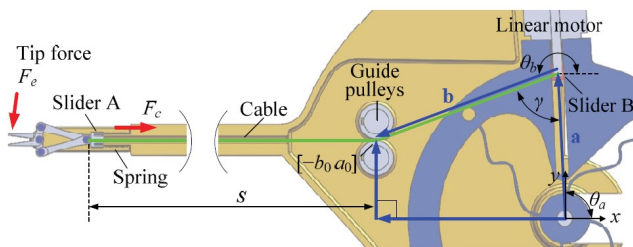


Fig. 6: Medical constant force forceps by Lan and Wang [31]

Nguyen et al. presented a statically balanced gripper for micro manipulation purposes [32]. A schematic of the gripper is given in figure 7. Two pairs of 4- bar linkages are prestressed and are thus able to provide a part of the energy that is needed to open and close the jaws. As the couples are configured in a symmetric manner, only one 4- bar linkage is considered in the optimisation procedure. The 4- bar linkage

is parameterised as a 3rd- order Bézier curve and optimised by using a genetic algorithm. The design variables are the x and y locations of the control points of the Bézier curves and the widths of both flexures in the linkage. The objective of the optimisation was to minimise the stiffness of the total mechanism, including the jaws. The force- displacement characteristic, obtained by a numerical model, showed a zero force part and a nonzero linear part for larger displacements. A constant force mechanism, as developed in an earlier work, was thus implemented to realise a certain displacement of the jaws.

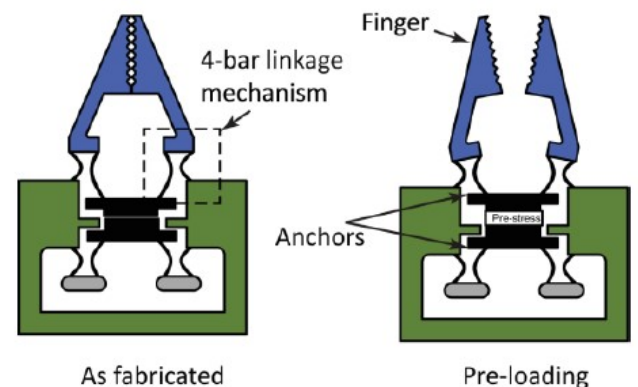


Fig. 7: Statically balanced gripper by Nguyen et al. [32]

Kuppens et al. presented a novel method to introduce prestress in a MEMS device: a flexure was elongated by a siliciumdioxide film [33] to induce buckling of the other flexures as well. The main idea behind this method is that thin films often contain residual stresses. The method was applied in an example where a linear motion stage as depicted in figure 8 is statically balanced. At the left of figure 8, the motion stage is shown. The shuttle of the motion stage is suspended by buckling flexures. The cross section of the lower flexure, which is covered with the siliciumdioxide film, is given in the subfigure (right) of figure 8. Although this specific example only illustrates the working principle in a relatively simple translational stage, it is claimed by the authors that the same method would also be applicable to balance more complicated systems. A stiffness reduction by a factor 9 to 46 is achieved with the presented setup.

Kuppens et al. [34] achieved 90.5% stiffness reduction over a 0.35 rad domain by static balancing of a rotary compliant mechanism by means of a toggle, similar to the mechanism of Pluimers et al. [35]. The mechanism is statically balanced by using the constant opposing torque approach. This approach is based on the constant opposing force approach, where two constant force mechanisms are balancing each other to obtain a zero force mechanism. Apart from the previously mentioned work from Pluimers, the use of the constant opposing force principle is rarely found in literature. Moreover, the application of the opposing constant torque approach to realise zero moment actuation is said to be completely novel. Figure 9 provides a CAD drawing of the mechanism. It is observed that the constant force mechanism

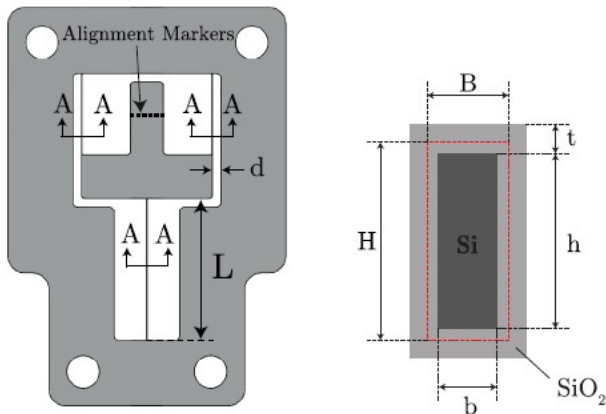


Fig. 8: The statically balanced linear motion stage (left) and the cross-section of the lower flexure (right) by Kuppens et al. [33]

from the left part of figure 8 is implemented twice, connected via a monolithically integrated bistable switch. One constant force mechanism is encircled in red in figure 9. The other one is installed in a symmetrical manner. The four plate springs intersect in the middle of the statically balanced mechanism, realising an instantaneous center of rotation. The bistable switch is located in between the constant force mechanisms. Pressing the curved beams together results in an alignment of the force-deflection characteristics of both CFM's, thus enabling statically balanced rotation. Pulling the curved beams apart, on the other hand, retrieves the non-zero stiffness of the mechanism.

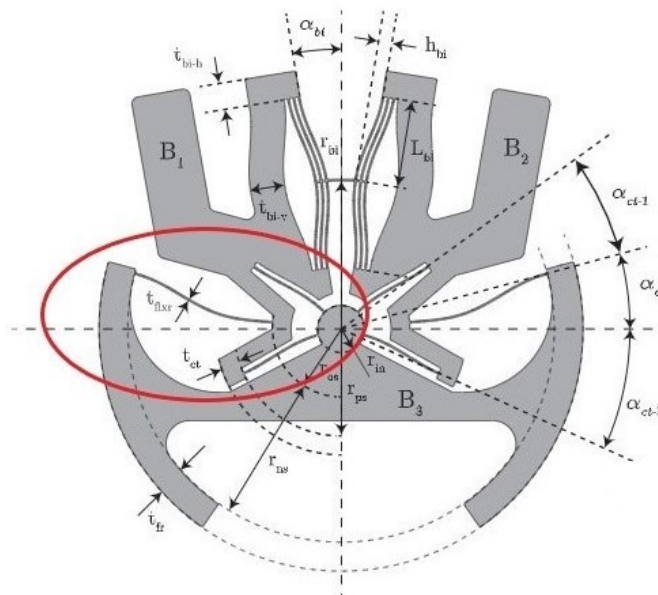


Fig. 9: Statically balanced compliant mechanism by Kuppens et al. [34]

Leishman et al. discussed the use of a modified version of the spring butterfly mechanism as a haptic interface device [36]. A haptic interface device enables touch feedback of

manual operation in situations in which the to be palpated object is distant or virtual. Because of the earlier mentioned advantages, statically balanced compliant mechanisms could be well suited to be used as haptic interface devices. Leishman used a pseudo-rigid body model to do an analytical approach. Accordingly, a prototype was produced and experimental validations were done. The CAD model of the prototype is shown in figure 10. The yellow rod is used as the actuation port, whereas the blue handle opposite to the yellow rod is the interface with the user. Although the mechanism is not perfectly balanced, the performance is said to be satisfactory for the purpose of a haptic interface device. The maximum moment at the handle was 0.0326 Nm for a handle angle of 130° , an input angle of 0° and a handle length of 0.1 m. The force transmission capability decreases with larger input and output angles and these effects are claimed to be more pronounced for rotations larger than $\pm 30^\circ$.

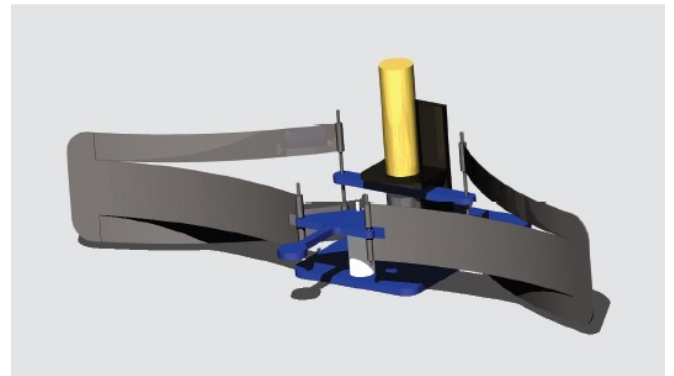


Fig. 10: CAD model of spring butterfly-based haptic interface device by Leishman et al. [36]

Jensen and Jenkins designed a statically balanced joint made from piano wire [37]. A pseudo rigid body model was developed that was subsequently optimised by an optimisation algorithm. The objective of the optimisation procedure was to find a configuration of the mechanism in which the potential energy was constant. A FEM and prototype were made to validate the optimised design variables. Figure 11a visualises the piano wire frame that was statically balanced. The figure adopted from the work of Jensen and Jenkins is slightly adjusted to visualise the imposed constraints on the mechanism. The bars marked in red are torsion bars that are constrained to remain in the same plane. The same holds for the two yellow bars: both bars are able to rotate, but they remain located in the same plane. The mechanism can be statically balanced because of an initial preload on the system. The yellow bars are rotated such that the bars between the red and yellow parts cross each other. Figure 11b illustrates a neutrally stable position of a hinge with embedded statically balanced piano wire frame. The hinge was designed to have a neutrally stable region of 180° , but the experimental results indicated that the performance of the mechanism is worse than expected due to friction.

Schultz et al. reported a neutrally stable fiber-reinforced

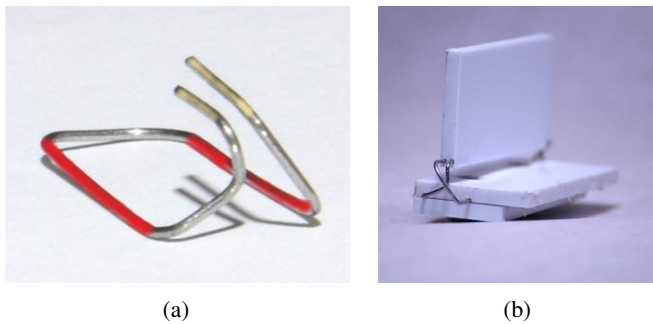


Fig. 11: Statically balanced wireform mechanism (adjusted) (a) and a prototype of a polypropylene neutrally stable hinge (b) by Jensen and Jenkins [37]

composite tape spring [38]. The fabric is supplemented by a low stiffness resin. Schultz defined an own, less strict, definition of neutral stability: if the tape spring would be left partially rolled up, the spring would stay in this exact position and thus would not roll up or deploy. The motivation for the expected neutral stability was given by means of the constitutive equations of the spring. The D- matrix of the spring, which provides the relation between imposed moments and realised curvatures, was manipulated such that the moments that are needed to achieve a certain curvature change became zero. Figure 12 depicts the tape spring in a partially deployed state. Unfortunately, no detailed specifications were given about the neutral stability of the tape spring. The focus of the design of the tape spring was on practical applicability. In practice, the mechanism needed a small actuation force in order to deploy or roll- up. To that end, a SMA wire was used to provide an actuation force and deploy the mechanism. Although the presented design was not capable of rolling up, it is believed that it would be straightforward to implement that in the current mechanism as well.



Fig. 12: Neutrally stable composite tape spring by Schultz et al. [38]

Rommers et al. presented a spherical Pseudo Rigid Body Modeling (PRBM) approach to design a “single vertex compliant facet origami mechanism” [39]. Such a compliant facet origami mechanism is rather unique as most origami mechanisms deform locally at hinges that function as the creases in traditional origami designs. A compliant facet

origami mechanism, on the other hand, allows deformation of the normally rigid connection pieces as well. Figure 13 provides the PRB model of the mechanism. The black lines that are directed to the vertex are physical hinge lines. The torsional stiffness of these hinge lines is assumed to be zero. The dotted lines are virtual hinge lines that represent the stiffness of the compliant facets. It was discovered that the compliant facets cause bistable behaviour and therefore negative stiffness. The authors recognised the potential use of this mechanism as a building block in the design of statically balanced mechanisms and designed three different joints: a constant moment joint, a gravity balanced joint and a zero moment joint [40]. In the latter work the torsional stiffnesses of the physical hinges were not assumed to be zero anymore and the analytical model was further developed. An optimisation procedure was applied to obtain the design variables that result in the desired moment- angle characteristics. Figure 14 illustrates the constant moment joint with small rods as torsion springs at the hinges to create positive stiffness. The hinges are made by an alternating pattern of Mylar type. The range of motion of the zero moment joint was 66 degrees and the range of the constant moment joint was 77 degrees. The allowed bandwidth was 3 percent of the maximum amplitude of the virtual stiffness τ_B .

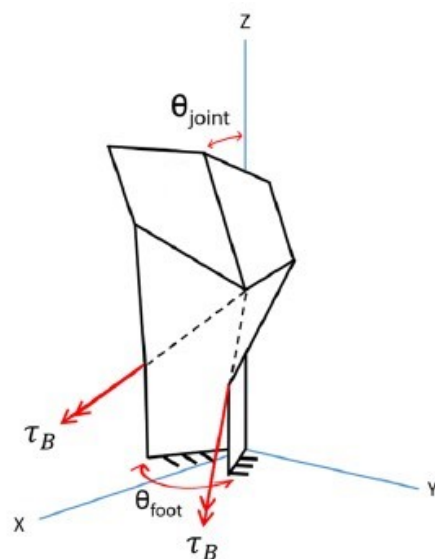


Fig. 13: PRBM of single vertex compliant facet origami mechanism presented by Rommers et al. [39]

Kok et al. described the concept and a corresponding modelling method of a neutrally stable curved crease shell structure [41]. The mechanism, depicted in figure 15, consists of two flat compliant facets that are connected to a curved crease. Figure 15 visualises the intended motion of the mechanism, from top to bottom. The top and bottom configurations denote the standard equilibria of the mechanism: no potential energy is stored. During the transition from the first equilibrium to the other, the inflection point travels along the crease from the beginning till the end. The inflection point is a point among the set of points called the “inflection

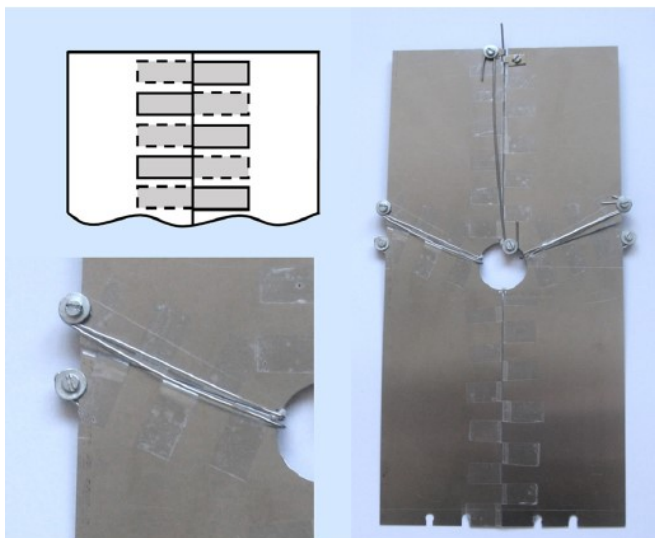


Fig. 14: Design of constant moment joint by Rommers et al. [40]

axis". At the inflection point, the positive curvature changes into a negative curvature. In order to ensure a neutrally stable path between the two extreme zero energy states, two design variables are varied: the variation of the width of the compliant facets and the variation of the curvature of the crease. It is found that the mechanism has a neutrally stable region between both endpoint equilibria when the curvature and width in the middle are slightly smaller than the curvature and width at the ends. It should be noted that the constant energy plateau shown in the paper has a finite range as the potential energy characteristic still has severe inclined parts at the begin and the end of the trajectory. Although the force and energy functions are given for some of the configurations, no performance specifications are presented.

Murphey and Pellegrino attempted to design a neutrally stable tape spring by binding two curved lamina with perpendicular curvature axes and opposing curvature senses [42]. Each individual plate was in an energy free state in the flat configuration. During the curving procedure, prestress was added to the lamina. Graphite fibre reinforced plastic (GFRP) lamina were used. The material orthotropy of the lamina, combined with a certain amount of prestress, resulted in the zero stiffness of the laminate. The tape spring is visualised in figure 16. According to an analytical analysis, the maximal actuation force needed to roll and unroll the tape spring would be 0.5N. In practice, however, difficulties were encountered during the bonding process and the actuation process. Furthermore, the production method proved to be difficult and sensitive to errors. Two different actuators were installed to inspect their effectiveness: a NiTi shape memory actuator and a PVDF piezoelectric film. Only the piezoelectric film proved to be able to actuate the tape spring, although the actuation was said to be jerky and of limited strength.

Although the mechanism is not neutrally stable yet, the

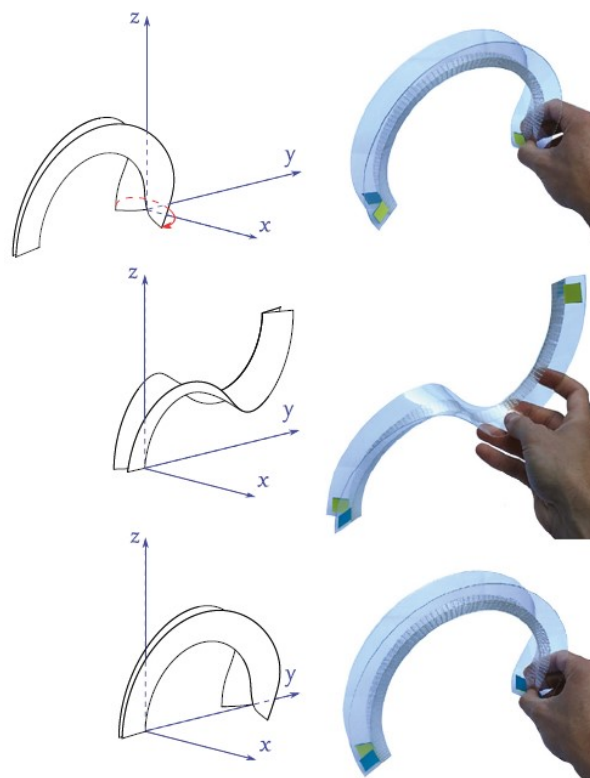


Fig. 15: Motion of neutrally stable shell structure by Kok et al. [41]



Fig. 16: Neutrally stable tape spring by Murphey and Pellegrino [42]

elaboration on the design of the Flectofin by Lienhard et al. [43] is still worthwhile to discuss. The Flectofin is a commercial compliant bending mechanism in development. Its working principle is based on the Bird-Of-Paradise flower; the *Strelitzia Reginae*. Figure 17 visualises the tropical flower (left) and the Flectofin mechanism (right). The Flectofin is actuated by bending the "backbone". This bending will enforce lateral-torsional buckling of the thin shell element that is attached to the backbone. The buckling of the shell element then causes the shell to bend to a side. The reversible and repetitive motion is enabled by elastic deformation of the entire structure. The range of motion is as large as -90° till 90° rotation of the thin plate. The plate is a composite consisting of glass fibre reinforced polymers (GFRP). This

material is selected because of its high tensile strength and low bending stiffness.

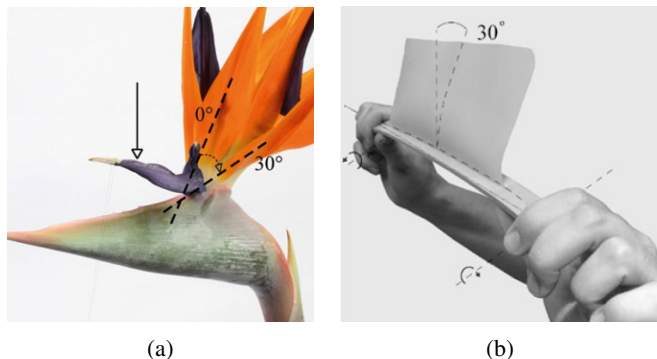


Fig. 17: The *Strelitzia Reginae* or the Bird-Of-Paradise flower (a) and the Flectofin by Lienhard et al. loaded in bending (b) [43]

A2-1-a. Planar, external conventional compensation energy, lumped compliance

Deepak et al. presented an analytical method to statically balance flexure-based compliant mechanisms [44]. The authors focused on the balancing of compliant mechanisms with lumped compliance. After defining the “effort function” that could be used as a measure of the static balancing, the flexure is replaced by a torsion spring. Subsequently, the torsion spring is substituted by a zero-free-length spring. When the torsion spring is replaced by a zero-free-length spring, a balancing spring is added to ensure neutral stability. One should then add the balancing spring to the original flexure-based compliant mechanism in order to achieve elastic balancing of this mechanism. The method was applied to several mechanisms, including a compliant probe. A schematic of the unbalanced and the balanced probe is shown in figure 18. At the left side of the image the unbalanced mechanism can be seen, whereas the statically balanced version is depicted on the right. The point P could be interpreted as the point of interest, which is able to move in the plane as described by the vector \mathbf{u} . The same point is subjected to a force vector \mathbf{f} . The positive stiffness of the flexures F_1 and F_2 , which are the red parts in the figure, is compensated by the negative stiffness of the springs Z_2^1 and Z_2^2 . Experimental validation indicated that the required actuation force was reduced by more than 70% due to the static balancing procedure. The range of motion was reported to be approximately 20% of the characteristic length scale of the mechanism itself.

A2-1-b. Planar, external conventional compensation energy, distributed compliance

Herder and Van den Berg statically balanced an approximately linear gripper with the rolling-link mechanism illustrated in figure 19. The same working principle is reported by Aguirre et al. as well [45]. The pull-pushrod is used to close and open the gripper (not shown in the figure), respectively. Initially, without any perturbation by

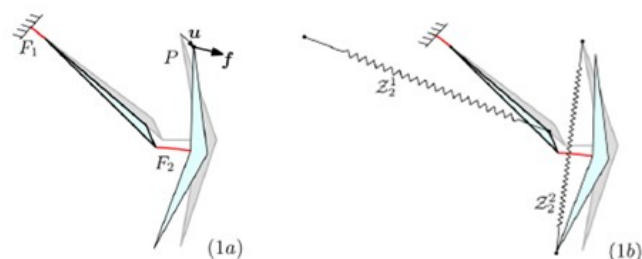


Fig. 18: The 2-DOF probe presented by Deepak et al. [44]

the pull-pushrod, the gripper is half open and the lever is positioned vertically upright. The actuation of this rod causes the rolling link to roll. As a result, the spring is relaxed when the rod is translated. During the relaxation of the spring, potential spring energy is released and used for the elastic deformation of the compliant members of the gripper. According to the performed experiments, the energy dissipation of one opening-closing cycle is approximately 0.2mJ. The maximum force perceived by the operator is 0.05N, whereas the unbalanced gripper had a maximum operating force of 12.9N.

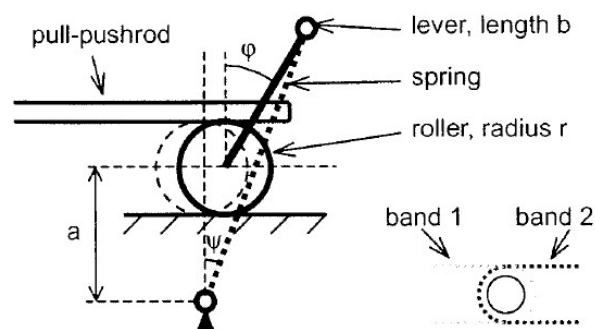


Fig. 19: The rolling-link balancing mechanism by Herder and Van den Berg [21]

Powell and Frecker modelled a static balancing mechanism to balance already fabricated ophthalmic surgical forceps [46]. The forceps are closed by the axial displacement of a tube that touches the forceps. Figure 20 visualises the forceps and the balancing mechanism. The balancing part consists of a slider-crank mechanism with a pre-tensioned spring. The spring is relaxed when the slider-crank is positioned horizontally. The total system is brought to a state of continuous equilibrium by optimisation of the slider-crank mechanism. First, a FEM model of the forceps is made. This FEM model is used to obtain the force-displacement characteristic and accordingly the potential energy as a function of the imposed displacement. Secondly, the kinematic equations and the boundary conditions of the balancing mechanism are set up. The potential energy of the total system is eventually defined by the sum of the potential energy of the constituent parts: the forceps and the balancing mechanism. The total amount of potential energy is kept constant by choosing

an objective function that minimises any deviations of the potential energy from the average amount of potential energy. The optimisation is done for several sets of precision points and for different orders of spring behaviour. The average deviation of the potential energy ranged from 0.6% till 4.2% for fourth till first order spring behaviour, respectively.

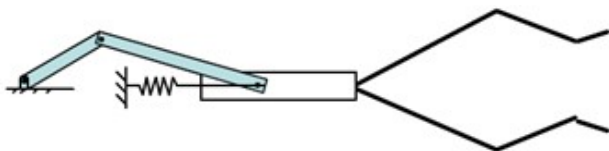


Fig. 20: The ophthalmic forceps including a static balancing mechanism designed by Powell and Frecker [46]

In the same work by Deepak et al. as discussed earlier [44] a statically balanced gripper is presented. The gripper is not balanced by the earlier discussed balancing procedure, but the unbalanced gripper is interpreted as a positive spring instead. By designing a rigid body linkage with opposite stiffness, the complete mechanism was expected to have zero stiffness. Figure 21 visualises the prototype where the rigid body compensation mechanism can be seen at the bottom. In the bottom right of the figure the suspension point F of the spring can be observed. This spring is connected to the rest of the compensation mechanism with an inextensible nylon thread. The actuation effort was reduced by 75%, which was lower than expected. According to the authors, this could be caused by frictional effects, misalignment of the vertical force or errors in the realisation of a zero-free-length spring.

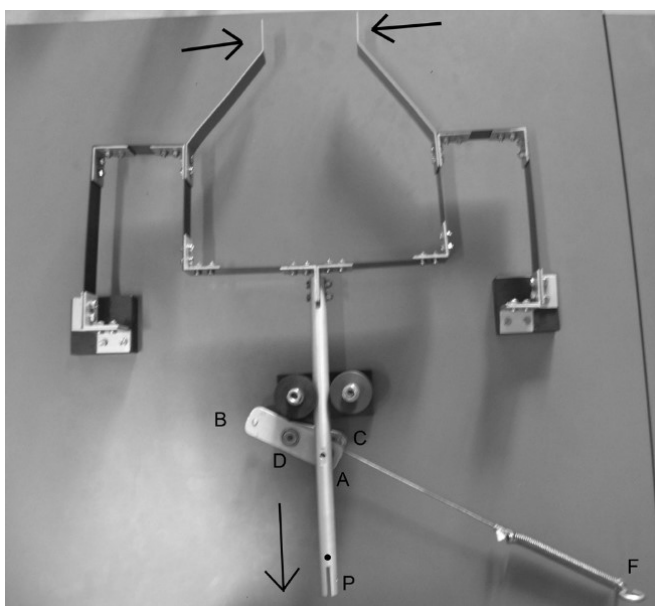


Fig. 21: The compliant gripper by Deepak et al. [44]

A2-2-a. Planar, external partially compliant compensation energy, lumped compliance

Berntsen et al. discussed the design of an internally balanced four-bar mechanism as a building block for more advanced statically balanced compliant mechanisms, like a gripper consisting of several neutrally stable four-bar mechanisms [47]. Initially curved flexures were used as a substitute for joints, such that the mechanism deforms by lumped compliance. Inspired by the use of pre-tensioned leaf springs by Dijkstra [48], it was decided to use compressed cantilever leaf springs as compensation mechanism. In contrast to Dijkstra's leaf springs, the springs in the work of Berntsen et al. describe a circular path instead of a linear trajectory. First, an analytical model was set up for both the four-bar and the compensation mechanism. The individual parts were dimensionalised using these models. Thereafter, the design parameters were optimised by using a genetic algorithm in Matlab. This algorithm was coupled to a finite element package to evaluate the performance of the interim results. The objective function was to minimise the standard deviation of the potential energy. Afterwards, the performance of the model was evaluated by both a finite element model and a prototype. The prototype is visualised in figure 22. The prestressed mechanism with no angular deflection is depicted in the top of the figure, subfigure 1, and in subfigure 3. Image-parts 2 and 4 represent the system at a rotation of -20° and 20° , respectively. According to the kinematic analysis and the finite element model, the average reduction in actuation moment was 95%. The experimentally determined moment reduction varied from 85% - 96%, depending on the range of motion. It was observed that a limited range resulted in an increase of moment reduction. The 85% was obtained for a 36° trajectory, whereas a range of motion of 20° resulted in a 96% moment reduction.

A2-2-b. Planar, external partially compliant compensation energy, distributed compliance

Morsch and Herder designed a zero stiffness compliant joint that can be used as a construction element in the design of general statically balanced compliant mechanisms [49]. The design was based on a similar zero stiffness joint that was not compliant. The conventional revolute joint was replaced by a cross-axis flexural pivot and leaf springs were used instead of helical zero-free-length springs. By replacing the conventional parts with compliant constituents, the force-deflection characteristic of the mechanism was altered. To retrieve the neutral stability an optimisation routine was applied to maximise the reduction of the actuation moment. The joint was required to move 70° from the vertical to both sides. The optimisation program used a grid search based on an analytical model that is evaluated at 50 discrete points. The configuration with the highest average moment reduction was chosen. The average moment reduction was 93% according to a finite element model and 70% reduction was measured with the experimental prototype shown in figure 23.

Tolou and Herder designed the statically balanced laparoscopic grasper visualised in figure 24 [50]. The grasper is

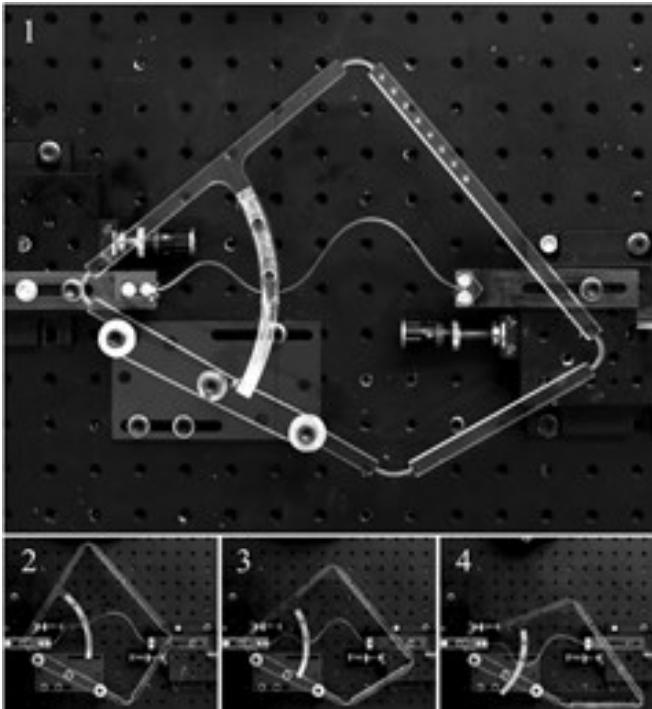


Fig. 22: Prototype of the statically balanced four-bar mechanism by Berntsen et al. [47]



Fig. 23: Zero stiffness compliant joint by Morsch and Herder [49]

initially closed and can be opened by a push on the middle beam, which is indicated by VII in the figure. The balancing mechanism consists of several prestressed beams (segments VIII and XI in figure 24). These beams are installed in a pin-pin configuration, such that the suspension points are not able to provide a reaction-moment. Both a mathematical formulation and a FEM model are made to investigate the influence of the number of balancing segments and the length of these segments on the balancing error and the maximal Von Mises stress in the balancing segments. The Von Mises stress decreases with both an increasing amount of segments and with an increasing segment length. The balancing error appeared to be independent of the amount of segments, whereas the FEM model and mathematical formulation do not agree about the effect of the segment length on the balancing error. According to the FEM model the error decreases with the length, whereas the analytical approach shows a minimum at a certain segment length. The residual force is shown to be significantly reduced compared to the required force of the unbalanced gripper, but the forces

are made dimensionless and no exact values are given. The concept is said to be an extension of the work of Herder and Van den Berg [21].

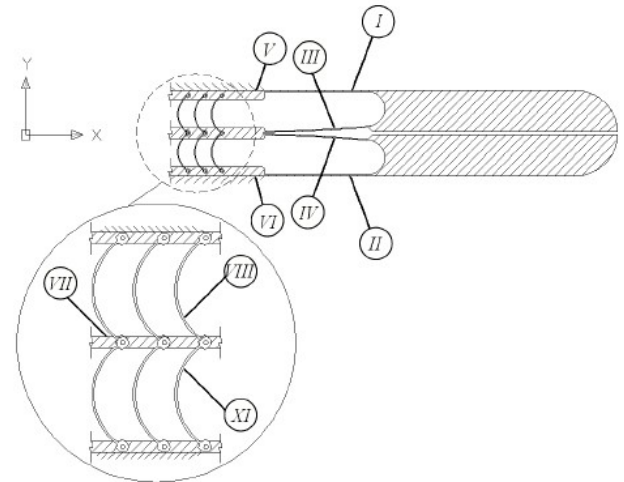


Fig. 24: The statically balanced grasper presented by Tolou and Herder [50]

A2-3-a. Planar, external compliant compensation energy, lumped compliance

No planar zero stiffness compliant mechanisms with lumped compliance and a compliant external energy container were found.

A2-3-b. Planar, external compliant compensation energy, distributed compliance

De Lange et al. designed a compliant laparoscopic grasper that is statically balanced by topology optimisation [51]. The authors claimed that their work was the first SBCM that is developed using topology optimisation. The design of the mechanism was divided into two parts: the design of a grasper with optimised deflection at the tip and the design of a compensation part. Given the deflection at the tip of the grasper, the compensation part is optimised by minimising the sum of the force-displacement characteristics of the grasper and the compensation part. The basic concepts of the grasper part and the compensation part were adopted from the work of Herder and Van den Berg [21] and the work of Stapel and Herder [52], respectively. The model of the assembly is presented in figure 25. One half of the symmetrically positioned compensation mechanism is encircled in green. The mechanism is fixed to the ground at the locations of the black, horizontal bars. The red crosses indicate an actuation location where manual operation is possible. Although it is claimed that topology optimisation is a promising method to statically balance compliant mechanisms, the Ansys model showed plastic deformation in the compensation mechanism and a compensation error as large as 14N for small displacements.

Lamers et al. presented a fully compliant statically balanced grasper that is designed by analytical formulations [53]. The building block approach was used to compensate

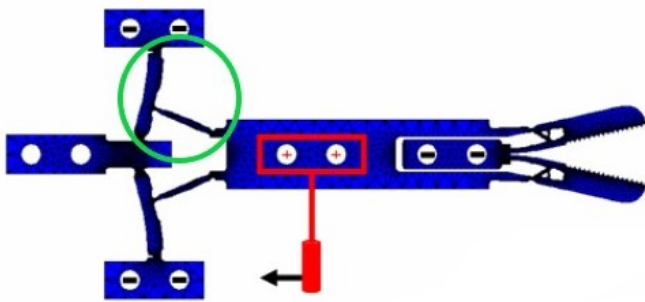


Fig. 25: Statically balanced compliant grasper designed by De Lange et al. using topology optimisation [51]

the positive stiffness of a titanium version of the compliant gripper of Herder and Van den Berg [21] with the negative stiffness of a balancer. The balancer was based on a slider-rocker linkage; a relatively well known negative stiffness mechanism. Figure 26 visualises the titanium prototype. At the left of the image the gripper by Herder and Van den Berg is recognised, whereas the larger part at the right is the compensation mechanism. The compensation mechanism consists of four slider-rocker linkages as the original linkage is duplicated in two symmetry axes. By preloading the compensation mechanism an actuation energy reduction of approximately 83.64% was achieved. The mean stiffness of the total mechanism was evaluated to be -3.14 N/mm on average.

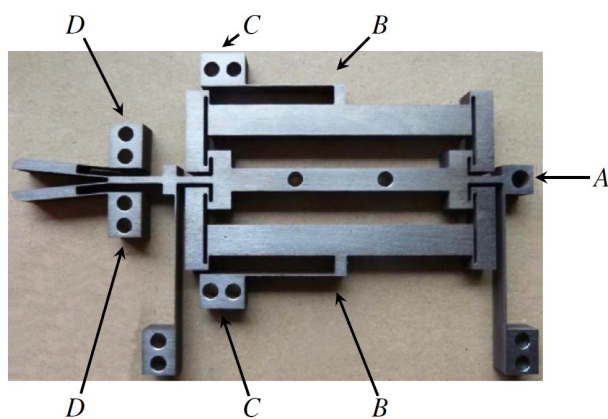


Fig. 26: Statically balanced compliant grasper by Lamers et al. [53]

Pluimers et al. introduced a compliant mechanism with a toggle to switch between a constant force and a zero force state [35]. The mechanism is depicted in figure 27. The left side of the figure illustrates the scenario in which the mechanism is in its stiff configuration, whereas the gripper is statically balanced in the right part of the figure. The gripper is statically balanced by the compensation mechanism, at the left of the main shuttle, by pressing the 3 bistable beams at the right of the main shuttle. Pressing these beams results in a phase shift of the sinusoidal force-deflection characteristic of the compensation mechanism. As a result,

the approximately linear part with negative slope of the sinus cancels the positive linear force-deflection characteristic of the gripper. The total mechanism is, in that case, a zero force mechanism and is thus statically balanced. The actuation force was reduced by 91%.

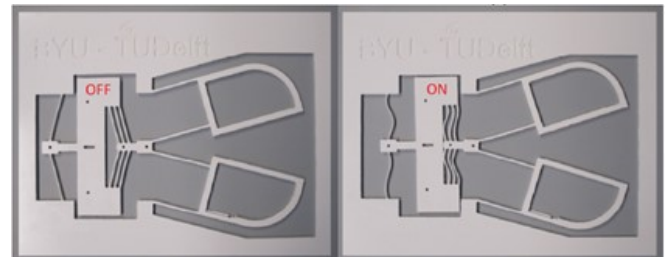


Fig. 27: Compliant mechanism with “static balancing-switch” by Pluimers et al. [35]

Stapel and Herder [52] also designed a statically balanced grasper based on the work of Herder and Van den Berg. This laparoscopic grasper was created by developing a compliant balancing mechanism that is to be installed in parallel with the “jaws” of the gripper. A slider-rocker mechanism and a double-slider were theoretically compared for this purpose. As the slider-rocker mechanism offered more possibilities for adjustment, this concept was selected as the balancing mechanism. A pseudo rigid body model was created and the theoretical performance was evaluated. The balancing error, defined as the residual force, was less than 0.03 N in the range $[-0.3, 0.3]$ mm. A schematic of the grasper is provided in figure 28.

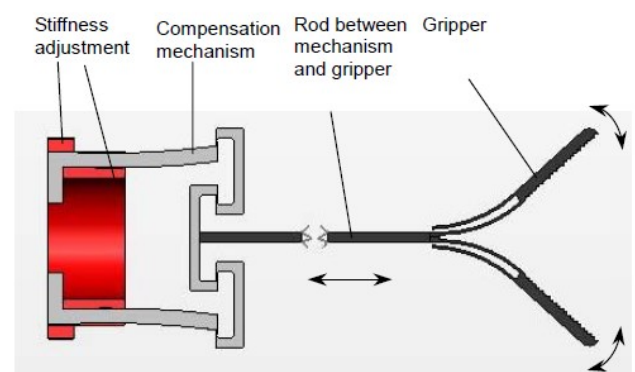


Fig. 28: Schematic of the laparoscopic grasper by Stapel and Herder [52]

Wang and Lan also designed a statically balanced compliant mechanism that was to be used in a constant force compliant gripper [54]. The statically balanced part consisted of two four-bar mechanisms that are preloaded against each other. The neutral stability is thus achieved by compensating forces with opposing forces of the same magnitude. An optimisation algorithm was used in order to find a geometry and preload that would result in a statically balanced mechanism. The objective of the optimisation was to minimise deviations in the force exerted by the mouth of the gripper.

Additionally, a constant force mechanism was designed to actuate the statically balanced part. The total mechanism is thus capable of delivering a constant force as output. A model of the gripper is depicted in figure 29. The jaws of the gripper are fully opened in the initial configuration. The closing of the jaws is caused by pulling the main body of the gripper to the right, which is realised by the constant force mechanism at the right of the figure. One couple of opposing force-four bar mechanisms is encircled with red and yellow in the sketch of the fully opened configuration. The other couple of four bars is positioned in a symmetric manner. By numerical and experimental validation it was found that the gripping force was nearly constant for an output displacement of 1.2 till 10.8 mm.

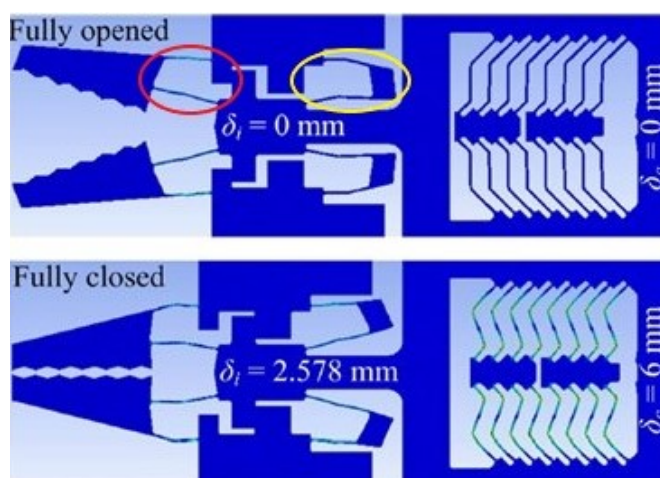


Fig. 29: Constant force compliant gripper by Wang and Lan [54]

Hoetmer et al. introduced an extension to the building block approach with negative stiffness elements [55]. After an elaboration on the proposed method a statically balanced gripper was presented as well. The building block approach originally only contained positive stiffness building blocks, but the authors claim that the extension of the approach with negative stiffness elements could provide a novel tool to statically balance any linear compliant mechanism in a systematic manner. The design method, which could also be used apart from the building block approach, is a two step approach. First, the functional element itself should be designed. This functional element should have a linear force- displacement characteristic and thus a constant positive stiffness. Secondly, the balancing segment is designed. A slightly overbalanced compliant gripper is designed using this building block approach. The gripper itself is a linear compliant mechanism designed by Kim [56]. The negative stiffness element was a compressed plate spring. The total assembly, of which the maximum operating force was reduced from 3.5N to -1N, is shown in figure 30.

Chandrasekaran et al. introduced a statically balanced remote center of rotation surgical tool [57]. A sideview of the mechanism is shown in figure 31. Although the mechanism is considered as a planar example with only one

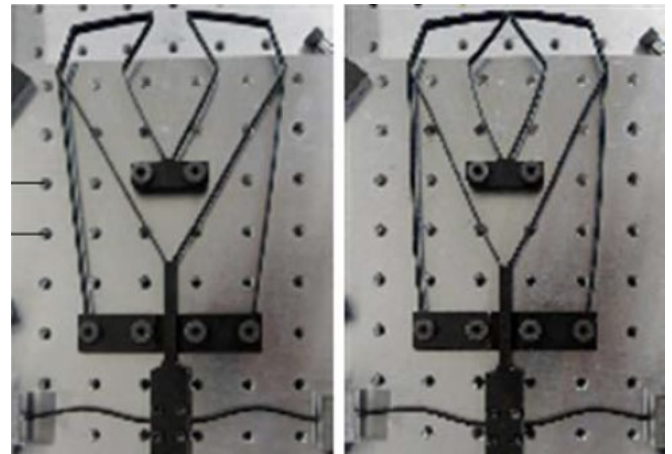


Fig. 30: Statically balanced compliant gripper designed by Hoetmer et al. using the extended building block approach [55]

rotational degree of freedom, the out of plane dimensions are considerably large as the tool uses cruciform flexures as a substitute for conventional joints. To compensate for the positive stiffness of these cruciform flexures, the stiffnesses were simply added up as the angular deflection of each flexure was expected to be the same. Subsequently, a serpentine flexure was used to provide the same, but opposite, stiffness. The serpentine was prestressed by fixing it in a deformed state at the base. A numerical optimisation routine was used to determine the length, stiffness and preload of the serpentine flexure. The objective of the optimisation was to minimise the required torque at the input link, visible halfway the serpentine flexure in figure 31. Both a FEA and an experimental validation were done. According to the experimental analysis the maximum torque was reduced by 83%.

B. Spatial zero stiffness compliant mechanisms

B1-a. Spatial, internal compensation energy, lumped compliance

No spatial zero stiffness mechanisms with lumped compliance and internal compensation energy were found.

B1-b. Spatial, internal compensation energy, distributed compliance

Dekens proposed and validated a three step method to investigate and, if possible, realise zero stiffness behaviour in shells [58]. First, the unloaded mechanism is analysed by inspecting the eigenvalues and the eigenvectors of the stiffness matrix. Then, a load is applied in the stiffest translational and rotational directions. Lastly, if a zero stiffness direction is found in the previous step, the zero stiffness direction is again analysed to inspect if the desired behaviour is still present at larger deformations. Figure 32 shows one of the analysed mechanisms: a shell mechanism being a half of a curved PVC tube. The origin in the coordinate system of the figure is constrained in all directions, whereas the eigenmodes are depicted with arrows in the point of interest. The vectors denoted as TS1

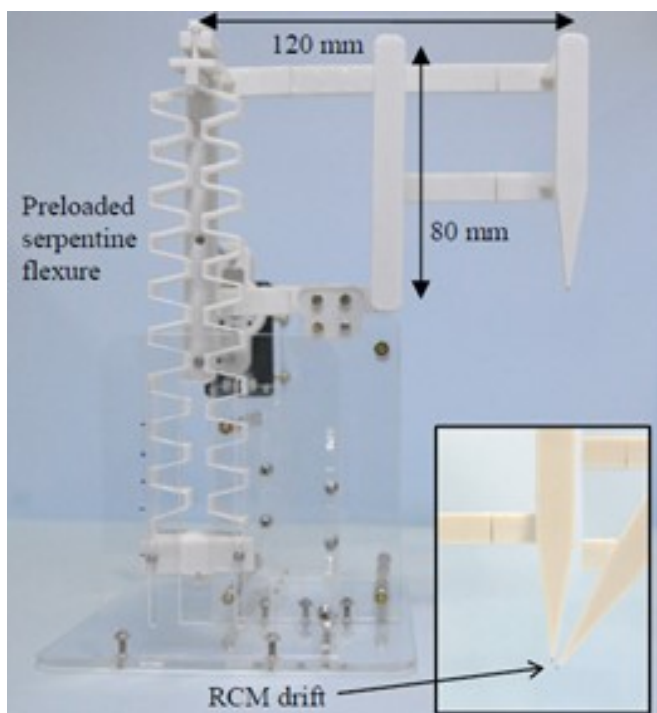


Fig. 31: Remote center of rotation statically balanced surgical tool by Chandrasekaran et al. [57]

and RS1 are the stiffest translation and rotation directions, respectively. The TS2 and RS2 directions are the second stiffest translation and rotation directions and TS3 and RS3 are the third stiffest modes. It appeared that preloading in the negative TS1 direction resulted in the most efficient zero stiffness behaviour, as this preloading direction required the least amount of preloading force. The corresponding zero stiffness direction was TS3. More generally, it was concluded that the stiffest eigendirection required the least amount of preloading in order to enable zero stiffness behaviour in the soft direction.

B2-1-x. Spatial, external conventional compensation energy

No literature examples of spatial zero stiffness mechanisms with a conventional mechanism for energy storage were found. As the category of conventional energy containers in spatial mechanisms does not contain any examples, the distinction between lumped and distributed compliance will be omitted here as well.

B2-2-a. Spatial, external partially compliant compensation energy, lumped compliance

No spatial examples of lumped compliance zero stiffness mechanisms with a partially compliant external compensation energy storage container were found.

B2-2-b. Spatial, external partially compliant compensation energy, distributed compliance

Lassooij et al. statically balanced an end effector for use in laparoscopic applications [59]. The end effector was coupled to a robotic arm with pitch and roll capabilities. The focus

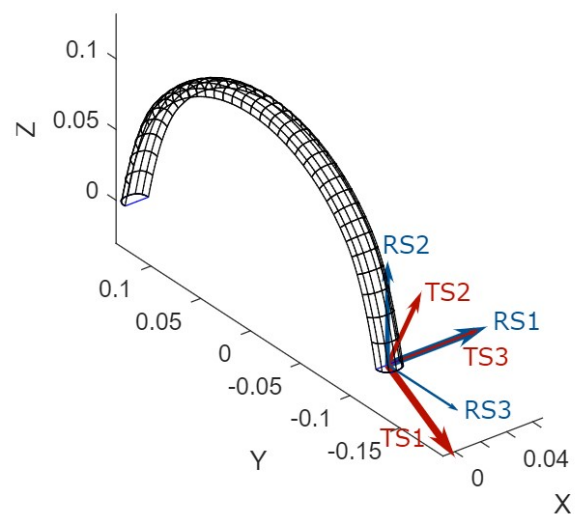


Fig. 32: Shell analysed for zero stiffness behaviour by Dekens [58]

of the work was on the end effector, which was statically balanced by employing pre-curved straight guided beams. A 3D sketch of the mechanism is given in figure 33. The force-displacement characteristic of the compensation mechanism is easily adjusted by tightening or loosening the nuts. By adjusting these nuts, the sinusoidal force-displacement characteristic is given a phase shift. As a result, the negative stiffness part is shifted to match the positive stiffness of the grasper. As the pre-curved beams are aligned collinear to the actuation direction, the mechanism is claimed to be more compact than statically balanced graspers with perpendicularly situated compensation mechanisms. The reduction in maximum actuation force and stiffness were measured to be 94% and 97%, respectively.

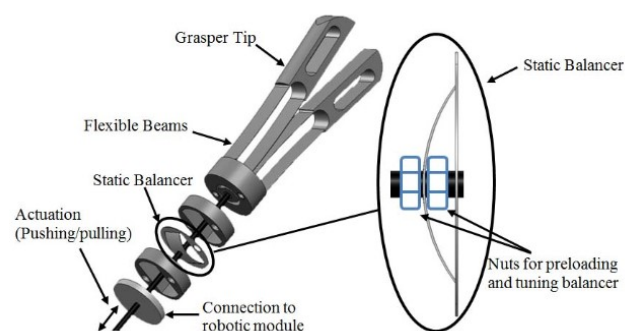


Fig. 33: Statically balanced end effector by Lassooij et al. [59]

Dunning et al. introduced a statically balanced compliant precision stage with six degrees of freedom [60]. Besides compensation for stored strain energy in the compliant members, the precision stage was also designed to remain statically balanced after applying a deadweight load. The precision stage, shown in figure 34, was conceptualised by

dividing its main task into separate functions. Subsystems were thus designed to accomplish out-of-plane motions and in-plane motions. Afterwards, the main parameters were tuned in order to enhance the neutral stability. The flexible rods at the bottom are positioned such that their contour forms an equilateral triangle. These rods are loaded near their buckling load and enable the in-plane motions of the stage. The out-of-plane motions are accomplished by three pairs of negative stiffness bi-stable buckling beams in combination with positive stiffness V-shaped beams. Measurement results showed that the translation stiffness in vertical direction was 0.4 N/mm for a 2 mm balanced domain, the out-of-plane rotational stiffnesses were 12 Nm/rad and 18.5 Nm/rad over a 10 mrad balanced domain, the in-plane translation stiffnesses in a 2 mm balanced domain were reduced from 1.1 N/mm to 0.4 N/mm after applying the load and the in-plane rotation experienced a stiffness reduction from 4.6 Nm/rad to 2.0 Nm/rad after applying the load over a 15 mrad balanced domain.

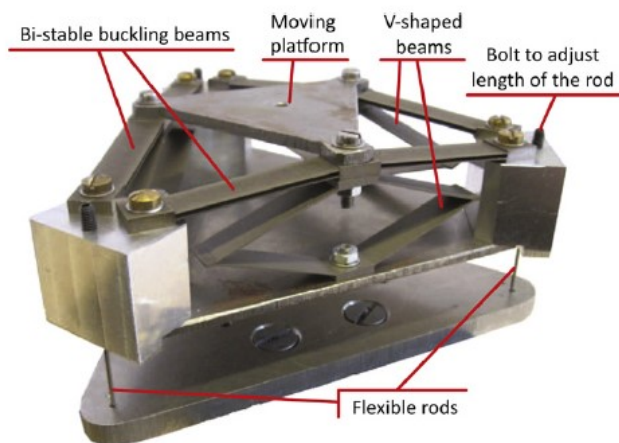


Fig. 34: Six DOF precision stage by Dunning et al. [60]

B2-3-x. Spatial, external compliant compensation energy

No examples of compliant zero stiffness mechanisms with a fully compliant external compensation energy storage container were found. A further categorisation into lumped and distributed compliance is therefore omitted.

C. Comparison of specifications

The specifications of the discussed planar and spatial zero stiffness compliant mechanisms are summarised in table II and table III, respectively. Each work is categorised based on the location of the storage of compensation energy and the type of compliance, as illustrated in figure 1. Furthermore, the synthesising method, type of force-deflection characteristic, range of motion and stiffness reduction are mentioned. The red cells that contain an asterisk refer to situations in which no reference configuration is reported. This reference configuration represents the configuration that is to be statically balanced. Thirty percent of the examples did not have any reference configuration. It can be seen that the stiffness reduction could not be evaluated

in approximately 40% of the planar cases and in one of the three spatial cases. The stiffness reduction was more often derived than mentioned in the mechanism's work. The stiffness reduction is approximately 80% - 100% in most cases. Some examples are slightly overbalanced and thus illustrate negative stiffness, recognised by the stiffness reduction greater than 100%. Apart from the comparable reduction in stiffness, the mechanisms tabulated in table II and table III also show a similar range of motion. Lastly, it is observed that almost 70% of the evaluated examples had a linear stiffness in the unbalanced configuration. This observation is in agreement with the statement of Hoetmer et al. that compliant mechanisms very often have a linear force-displacement characteristic [55].

IV. DISCUSSION

In the following, the results of chapter III will be discussed based on figure 2, table II and table III.

Chapter III already briefly summarised the most important observations on figure 2: more planar mechanisms than spatial mechanisms were found and most planar mechanisms are classified as distributed compliance mechanisms with internal compensation energy. The second largest planar group contains distributed compliance mechanisms with a compliant external energy storage mechanism. Furthermore, it was already stated that a relatively small amount of lumped compliance planar mechanisms was found and that the group of lumped compliance mechanisms with a compliant external energy storage mechanism was even empty. No spatial examples with lumped compliance were presented. The category of spatial mechanisms with internal compensation energy contained only one example, whereas two examples of spatial mechanisms with a partially compliant external storage mechanism were reported.

The fact that only a small amount of the presented mechanisms are spatial could be explained in different ways. First, presumably planar mechanisms are, in general, less complex to design and to fabricate than spatial mechanisms. Another reason for the unequal distribution of planar and spatial mechanisms could be the relatively large demand for planar statically balanced compliant mechanisms. As discussed in section III-A, a relatively large amount of planar examples is to be used as a gripper in agricultural or medical applications. The largest group and the second largest group of planar mechanisms together contain two-thirds of the total amount of zero stiffness mechanisms. These groups could be considered as the most convenient planar groups as well. The use of distributed compliance instead of lumped compliance could ensure that the stresses in the material remain relatively low, while the fully compliant nature results in the typical advantages of compliant mechanisms and a possibly monolithic assembly. It is expected that the advantages of the two largest planar categories could serve as a possible explanation for the relatively large amount of examples in these classes. The intensive use of distributed compliance and compliant designs is possibly correlated with the less frequent occurrence of mechanisms from the other

TABLE II: Summary of presented planar zero stiffness compliant mechanisms

Work	Location energy storage	Compliance	Synthesising method	Force- deflection characteristic	Range of motion	Stiffness reduction
Soroushian et al. [26]	Internal	Lumped	Structural optimisation - optimisation manufacturing parameters	*	20mm	*
Merriam et al. [30]	Internal	Lumped	Structural optimisation	Nonlinear	100°	79.6%
Lan and Wang [31]	Internal	Distributed	Structural optimisation	*	26°	*
Nguyen et al. [32]	Internal	Distributed	Structural optimisation	*	20mm	*
Kuppens et al. [33]	Internal	Distributed	Building blocks approach - buckling beam	Linear	0.38mm	88.9% - 97.8%
Kuppens et al. [34]	Internal	Distributed	-	Nonlinear	20°	90.5%
Leishman et al. [36]	Internal	Distributed	Kinematic approach (Pseudo- Rigid- Body Model)	*	60°	*
Jensen and Jenkins [37]	Internal	Distributed	Kinematic approach (Pseudo- Rigid- Body Model) i.c.w. structural optimisation	*	180°	*
Schultz et al. [38]	Internal	Distributed	Analytical formulations	*	254mm	*
Rommers et al. [39] [40]	Internal	Distributed	Kinematic approach (Pseudo- Rigid- Body Model) and structural optimisation	Nonlinear	66°	100%
Kok et al. [41]	Internal	Distributed	Iterative tuning using numerical model	*	180°	*
Murphey and Pallegirino [42]	Internal	Distributed	Analytical formulations	*	-	*
Lienhard et al. [43]	Internal	Distributed	Abstraction of working principle flower	*	180°	*
Deepak et al. [44] probe	Conventional external component	Lumped	Kinematic approach	Nonlinear	30mm	83.3% in x- direction
Herder and Van den Berg [21]	Conventional external component	Distributed	Building blocks approach	Linear	80°	99.9%
Powell and Frecker [46]	Conventional external component	Distributed	Structural optimisation	Linear	33.2mm	-
Deepak et al. [44], gripper	Conventional external component	Distributed	Kinematic approach and building blocks approach	Linear	60mm	81.9%
Berntsen et al. [47]	Partially compliant external component	Lumped	Structural optimisation	Linear	40°	100%
Morsch and Herder [49]	Partially compliant external component	Distributed	Structural optimisation	Linear	140°	79.7%
Tolou and Herder [50]	Partially compliant external component	Distributed	Building blocks approach	Linear	80°	100%
De Lange et al. [51]	Compliant external component	Distributed	Structural optimisation	Linear	10mm	93.2%
Lamers et al. [53]	Compliant external component	Distributed	Building blocks approach	Linear	0.6mm	104.7%
Pluimers et al. [35]	Compliant external component	Distributed	Parameter study using numerical model	Linear	20mm	-
Stapel and Herder [52]	Compliant external component	Distributed	Kinematic approach (Pseudo- Rigid- Body Model)	Linear	0.6mm	100.3%
Wang and Lan [54]	Compliant external component	Distributed	Structural optimisation	Nonlinear	9.6mm	39.5%
Hoetmer et al. [55]	Compliant external component	Distributed	Building blocks approach	Linear	16.9mm	120%
Chandrasekaran et al. [57]	Compliant external component	Distributed	Structural optimisation	Nonlinear	80°	83.5%

*no reference mechanism presented

TABLE III: Summary of presented spatial zero stiffness compliant mechanisms

Work	Location energy storage	Compliance	Synthesising method	Force- deflection characteristic	Range of motion	Stiffness reduction
Dekens [58]	Internal	Distributed	Own proposed 3- step method	Linear	100mm	112.5%
Lassooij et al. [59]	Partially compliant external component	Distributed	-	Linear	80°	97%
Dunning et al. [60]	Partially compliant external component	Distributed	Parameter variation using numerical model	Positive element (V- shaped beams) linear	$T_x = T_y = T_z = 2\text{mm}$, $R_x = R_y = 10\text{mrad}$, $R_z = 15\text{mrad}$	-

planar categories. The relatively small amount of lumped planar mechanisms with zero stiffness illustrate that lumped compliance is a feasible, but not often used, option for the design of planar zero stiffness compliant mechanisms. It can be seen in table II that these mechanisms generally have a stiffness reduction that is comparable to those of distributed compliance designs. The fact that only a minority of the planar designs has lumped compliance could be caused by the design goal to minimise stresses in the compliant members and the good availability of synthesis methods for distributed compliant designs. The absence of any spatial lumped compliance mechanisms does not necessarily indicate that it would be infeasible or impossible to design such a mechanism. Although the use of lumped compliance might induce high stresses in the material, it would be possible to store compensation energy in these flexures [35]. Three spatial zero stiffness mechanisms were found, of which one example utilised internal compensation energy and two examples used a partially compliant external energy storage. All three mechanisms were categorised as distributed compliance mechanisms. It should be noted that the mechanism with internal compensation energy, presented by Dekens [58], arguably does not belong to this category. As a matter of fact, prestress is applied in the stiffest direction in order to facilitate zero stiffness displacement in another direction. Without this prestress, the mechanism does not have a zero stiffness direction any more. The compensation energy is thus provided every time the mechanism is operated, so the prestress is not stored in the system. The other presented spatial zero stiffness mechanisms were the statically balanced end effector by Lassooij et al. [59] and the precision stage by Dunning et al. [60]. Both Lassooij and Dunning used a partially compliant external energy storage. Although Lassooij et al. described a gripper to be used with a laparoscopic arm to obtain pitch and roll degrees of freedom as well, only the gripper was statically balanced. The gripper is thus capable of describing spatial displacements, but energy is still required to realise pitch and roll of the end effector. In fact, one could apply this same principle to any of the described planar examples as well: it is merely a serial connection with a multiple DOF nonzero stiffness mechanism. The degree of freedom of the mouth of the gripper is the only DOF that is statically balanced. Consequently, the mouth of the gripper can be opened and

closed with approximately zero stiffness. The precision stage by Dunning et al. is designed by decomposing its motions into in- plane and out- of- plane DOF's. The mechanism of Dunning et al. could be considered as the only spatial zero stiffness mechanism with energy storage. It should be noted, however, that the concept is only applicable within a limited group of applications. In general, it is believed that the category of spatial mechanisms with distributed compliance is not investigated enough yet and represents a gap in design solutions.

Table II and table III summarise the specifications of the discussed works on zero stiffness compliant mechanisms. In 30% of the cases, no reference configuration was available. The existence of a reference configuration depends on the "point of departure": most work on zero stiffness compliant mechanisms is devoted to the static balancing of an already designed mechanism with positive stiffness, while others do not start with a preexisting design. The absence of a reference configuration induces difficulties in the comparison of the various zero stiffness mechanisms, but it is believed to be the result of the relatively wide scope of this literature review that includes mechanisms designed from different points of departure. The stiffness could not be evaluated, or was not given, for 40% of the planar mechanisms and one of the three spatial mechanisms. This indicates that for 10% of the mechanisms the reference configuration was known but no stiffness reduction could be evaluated. By inspection of table II and III it is seen that these examples have nonzero stiffness configurations with a linear force-deflection characteristic. Moreover, the range of motion is in the same order of magnitude as the range of motion of the fully evaluated examples. The stiffness reduction could not be evaluated, however, due to the absence of data to do a reliable derivation. As mentioned before, the stiffness reduction is more often derived than given by the authors of the work. This is expected to be caused by the objective of this review paper. The focus of this review was on zero stiffness compliant mechanisms, but the presented literature focused on the other interpretations of continuous equilibrium and neutral stability as well. In these cases, mainly the deviations from zero force and constant potential energy were reported, respectively.

V. CONCLUSION

The objective of this work is to present the state of the art on zero stiffness compliant mechanisms that could be used for path generation and to evaluate and compare their stiffness reduction. A categorisation is made, discriminating the mechanisms on their range of motion, location of energy storage, the nature of the possible external storage mechanism and compliance. Regarding the range of motion, the found examples were classified as being planar or spatial. Sequentially, the distinction between internal energy storage and external energy storage was made. In the case of external energy storage, the examples were categorised as having a compliant, a partially compliant or a conventional storage mechanism. The last level of categorisation concerned the division into classes with lumped compliance or distributed compliance.

Most planar mechanisms were categorised as distributed compliance mechanisms with internal energy storage. Distributed compliance and energy storage in a partially compliant external mechanism was the most occurring combination in the field of spatial zero stiffness mechanisms. No examples of spatial mechanisms with lumped compliance were found, whereas the combination of distributed compliance with internal compensation energy was demonstrated in only one spatial example. The stiffness reduction was reported to be in the 80% - 100% range in most cases. Almost 70% of the discussed examples had a linear stiffness in the unbalanced configuration.

A possible point of improvement for this literature survey would be the evaluation and tabulation of the residual forces of the systems. It should be noted, however, that some mechanisms will remain unevaluated as in these cases no performance parameters are presented at all. Moreover, one should be cautious when comparing the residual forces of the mechanisms as differences in volume could impede a fair comparison. In that case, it would be more appropriate to compare a dimensionless form of the force.

REFERENCES

- [1] L. L. Howell, *Compliant Mechanisms*, 2001.
- [2] L. L. Howell, S. P. Magleby, and B. M. B. M. Olsen, *Handbook of compliant mechanisms*.
- [3] J. Andres, G. Sanchez, and J. L. Herder, "Statically-balanced compliant micromechanisms Stable and Adjustable Mechanisms for Optical Instruments and Implants (SAMOI) View project Ferrofluid bearings View project," Tech. Rep., 2010. [Online]. Available: <https://www.researchgate.net/publication/260786326>
- [4] M. Schenk and S. D. Guest, "On zero stiffness," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 10, pp. 1701–1714, 2014.
- [5] A. G. Dunning, N. Tolou, and J. L. Herder, "Review Article: Inventory of platforms towards the design of a statically balanced six degrees of freedom compliant precision stage," *Mechanical Sciences*, vol. 2, no. 2, pp. 157–168, 2011.
- [6] D. Hogervorst, "Classification and comparison of zero stiffness compliant rotary joints," Tech. Rep., 2021.
- [7] J. Linssen, "Literature review: Deformation Induced Kinematics in Neutrally Stable Compliant Mechanisms," Tech. Rep., 2021.
- [8] S. Kok, "Literature review of occurrences and working principles of elastic neutral stability," Tech. Rep., 2020.
- [9] J. Dekens, "Literature review—Two and three dimensional zero-stiffness mechanisms," Tech. Rep., 2020.
- [10] B. Doornbal, "Review of prestressing techniques to obtain compliant shell mechanisms with zero and negative stiffness," Delft, Tech. Rep., 2018. [Online]. Available: <http://repository.tudelft.nl/>
- [11] S. Daynes and P. M. Weaver, "Stiffness tailoring using prestress in adaptive composite structures," *Composite Structures*, vol. 106, pp. 282–287, 12 2013.
- [12] L. Staats, "Review on methods of controllable stiffness for structures," Tech. Rep., 2021.
- [13] J. A. Franco, J. A. Gallego, and J. L. Herder, "Static balancing of four-bar compliant mechanisms with torsion springs by exerting negative stiffness using linear spring at the instant center of rotation," *Journal of Mechanisms and Robotics*, vol. 13, no. 3, 6 2021.
- [14] Y. Liu and Q. Xu, "Design of a 3D-printed polymeric compliant constant-force buffering gripping mechanism," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 7 2017, pp. 6706–6711.
- [15] M. Y. Barel, D. F. Machekposhti, J. L. Herder, M. Sitti, and N. Tolou, "Permanent Preloading by Acceleration for Statically Balancing MEMS Devices," in *2018 International Conference on Reconfigurable Mechanisms and Robots, ReMAR 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 8 2018.
- [16] K. A. Tolman, E. G. Merriam, and L. L. Howell, "Compliant constant-force linear-motion mechanism," *Mechanism and Machine Theory*, vol. 106, pp. 68–79, 12 2016.
- [17] C.-C. Lan, J.-H. Wang, and Y.-H. Chen, "A Compliant Constant- Force Mechanism for Adaptive Robot End- Effector Operations," in *2010 IEEE International Conference on Robotics and Automation*. I E E E, 2010.
- [18] G. Hao, "A framework of designing compliant mechanisms with nonlinear stiffness characteristics," *Microsystem Technologies*, vol. 24, no. 4, pp. 1795–1802, 4 2018.
- [19] N. Tolou, V. A. Henneken, and J. L. Herder, "Statically balanced compliant micro mechanisms (sb-mems): Concepts and simulation," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2, no. PARTS A AND B, 2010, pp. 447–454.
- [20] G. Chen and S. Zhang, "Fully-compliant statically-balanced mechanisms without prestressing assembly: Concepts and case studies," *Mechanical Sciences*, vol. 2, no. 2, pp. 169–174, 2011.
- [21] J. L. Herder and F. P. A. Van Den Berg, "STATICALLY BALANCED COMPLIANT MECHANISMS (SBCM'S), AN EXAMPLE AND PROSPECTS," in *ASME Design Engineering Technical Conferences*, 2000.
- [22] J. A. Gallego and J. Herder, "Synthesis methods in compliant mechanisms: an overview," in *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, 2009.
- [23] —, "Criteria for the static balancing of compliant mechanisms," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2, no. PARTS A AND B, 2010, pp. 465–473.
- [24] J. Herder, "Energy-free Systems," Ph.D. dissertation, TU Delft, Delft.
- [25] J. A. Gallego, "STATICALLY BALANCED COMPLIANT MECHANISMS THEORY AND SYNTHESIS," Ph.D. dissertation, TU Delft, Delft, 2013.
- [26] P. Soroushian, H. Chowdhury, and A. Nossoni, "Design and experimental verification of pseudoelastic-based constant-force springs," *Journal of Intelligent Material Systems and Structures*, vol. 14, no. 8, pp. 475–481, 8 2003.
- [27] M. C. Tanzi, S. Farè, and G. Candiani, "Biomaterials and Applications," in *Foundations of Biomaterials Engineering*. Elsevier, 2019, p. 64. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/B9780081010341000049>
- [28] CHEMEUROPE, "Pseudoelasticity."
- [29] M. Gurka, "Active hybrid structures made of shape memory alloys and fiber-reinforced composites," in *Multifunctionality of Polymer Composites: Challenges and New Solutions*. Elsevier Inc., 5 2015, pp. 25–26.
- [30] E. G. Merriam, M. Colton, S. Magleby, and L. L. Howell, "The Design of a Fully Compliant Statically Balanced Mechanism," in *Proceedings of ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2013.
- [31] C. C. Lan and J. Y. Wang, "Design of adjustable constant-force forceps for robot-assisted surgical manipulation," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 386–391.

- [32] D.-C. Nguyen, T.-V. Phan, and H.-T. Pham, "Design and Analysis of a Compliant Gripper Integrated with Constant- Force and Static Balanced Mechanism for Micro Manipulation," in *International Conference on Green Technology and Sustainable Development*. IEEE, 2018.
- [33] P. R. Kuppens, J. L. Herder, and N. Tolou, "Permanent Stiffness Reduction by Thermal Oxidation of Silicon," *Journal of Microelectromechanical Systems*, vol. 28, no. 5, pp. 900–909, 10 2019.
- [34] P. R. Kuppens, M. A. Bessa, J. L. Herder, and J. B. Hopkins, "Compliant Mechanisms That Use Static Balancing to Achieve Dramatically Different States of Stiffness," *Journal of Mechanisms and Robotics*, vol. 13, no. 2, 4 2021.
- [35] P. J. Pluimers, N. Tolou, B. D. Jensen, L. L. Howell, and J. L. Herder, "A compliant on/off connection mechanism for preloading statically balanced compliant mechanisms," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 4, no. PARTS A AND B, 2012, pp. 373–377.
- [36] L. C. Leishman, D. J. Ricks, and M. B. Colton, "Design and evaluation of statically balanced compliant mechanisms for haptic interfaces," in *ASME 2010 Dynamic Systems and Control Conference, DSCC2010*, vol. 1, 2010, pp. 859–866.
- [37] B. D. Jensen and C. H. Jenkins, "DESIGN OF SMALL-SCALE STATICALLY BALANCED COMPLIANT JOINTS," in *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2011. [Online]. Available: <http://proceedings.asmedigitalcollection.asme.org/>
- [38] M. R. Schultz, M. J. Hulse, P. N. Keller, and D. Turse, "Neutrally stable behavior in fiber-reinforced composite tape springs," *Composites Part A: Applied Science and Manufacturing*, vol. 39, no. 6, pp. 1012–1017, 6 2008.
- [39] J. Rommers, G. Radaelli, and J. L. Herder, "Pseudo-rigid-body modeling of a single vertex compliant-facet origami mechanism," *Journal of Mechanisms and Robotics*, vol. 9, no. 3, 6 2017.
- [40] —, "A design tool for a single vertex compliant-facet origami mechanism including torsional hinge lines," *Journal of Mechanisms and Robotics*, vol. 9, no. 6, 12 2017.
- [41] S. Kok, G. Radaelli, A. Amoozandeh Nobaveh, and J. Herder, "Neutrally stable transition of a curved-crease planar shell structure," *Extreme Mechanics Letters*, vol. 49, p. 101469, 11 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2352431621001759>
- [42] T. W. Murphey and S. Pellegrino, "A novel actuated composite tape-spring for deployable structures," in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 1, 2004, pp. 260–270.
- [43] J. Lienhard, S. Schleicher, S. Poppinga, T. Masselter, M. Milwich, T. Speck, and J. Knippers, "Flectofin: A hingeless flapping mechanism inspired by nature," *Bioinspiration and Biomimetics*, vol. 6, no. 4, 12 2011.
- [44] S. R. Deepak, A. N. Hansoge, and G. K. Ananthasuresh, "Application of rigid-body-linkage static balancing techniques to reduce actuation effort in compliant mechanisms," *Journal of Mechanisms and Robotics*, vol. 8, no. 2, 5 2016.
- [45] M. Aguirre, A. T. Steinorsson, T. Horeman, and J. Herder, "Technology Demonstrator for Compliant Statically Balanced Surgical Graspers," *Journal of Medical Devices, Transactions of the ASME*, vol. 9, no. 2, 6 2015.
- [46] K. M. Powell and M. I. Frecker, "METHOD FOR OPTIMIZATION OF A NONLINEAR STATIC BALANCE MECHANISM, WITH APPLICATION TO OPHTHALMIC SURGICAL FORCEPS," in *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2005. [Online]. Available: http://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2005/47446/441/2644413/441_1.pdf
- [47] L. Berntsen, D. H. Gosenshuis, and J. L. Herder, "DESIGN OF A COMPLIANT MONOLITHIC INTERNALLY STATICALLY BALANCED FOUR-BAR MECHANISM," in *Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2014. [Online]. Available: <http://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2014/46360/V05AT08A040/4216717/v05at08a040-detc2014-35054.pdf>
- [48] J. Dijkstra, "A STUDY OF SOME ASPECTS OF THE MECHANICAL BEHAVIOUR OF CROSS-SPRING PIVOTS AND PLATE SPRING MECHANISMS WITH NEGATIVE STIFFNESS," Ph.D. dissertation, Delft, 5 1979.
- [49] F. M. Morsch and J. L. Herder, "Design of a generic zero stiffness compliant joint," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2, no. PARTS A AND B, 2010, pp. 427–435.
- [50] N. Tolou and J. L. Herder, "Concept and modeling of a statically balanced compliant laparoscopic grasper," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 7, no. PARTS A AND B, 2009, pp. 163–170.
- [51] D. J. De Lange, M. Langelaar, and J. L. Herder, "Towards the design of a statically balanced compliant laparoscopic grasper using topology optimization," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2, no. PARTS A AND B, 2008, pp. 293–305.
- [52] A. Stapel and J. L. Herder, "Feasibility study of a fully compliant statically balanced laparoscopic grasper," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2 A. American Society of Mechanical Engineers, 2004, pp. 635–643.
- [53] A. J. Lamers, J. A. Gallego Sánchez, and J. L. Herder, "Design of a statically balanced fully compliant grasper," *Mechanism and Machine Theory*, vol. 92, pp. 230–239, 6 2015.
- [54] J. Y. Wang and C. C. Lan, "A constant-force compliant gripper for handling objects of various sizes," *Journal of Mechanical Design, Transactions of the ASME*, vol. 136, no. 7, 2014.
- [55] K. Hoetmer, J. L. Herder, and C. J. Kim, "A Building Block Approach for the design of statically balanced compliant mechanisms," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 7, no. PARTS A AND B, 2009, pp. 313–323.
- [56] C. Kim, "Design strategies for the topology synthesis of dual input-single output compliant mechanisms," *Journal of Mechanisms and Robotics*, vol. 1, no. 4, pp. 1–9, 11 2009.
- [57] K. Chandrasekaran, A. Somayaji, and A. Thondiyath, "REALIZATION OF A STATICALLY BALANCED COMPLIANT PLANAR REMOTE CENTER OF MOTION MECHANISM FOR ROBOTIC SURGERY," in *Proceedings of the 2018 Design of Medical Devices Conference*, 2018. [Online]. Available: <http://asmedigitalcollection.asme.org/BIO MED/proceedings-pdf/DMD2018/40789/V001T07A011/2787769/v001t07a011-dmd2018-6911.pdf>
- [58] J. Dekens, "Three- step method to design the preload of geometries to create spatial large range of motion zero- stiffness compliant mechanisms," Tech. Rep., 2021.
- [59] J. Lassoij, N. Tolou, G. Tortora, S. Caccavaro, A. Menciassi, and J. L. Herder, "A statically balanced and bi-stable compliant end effector combined with a laparoscopic 2DoF robotic arm," *Mechanical Sciences*, vol. 3, no. 2, pp. 85–93, 2012.
- [60] A. G. Dunning, N. Tolou, and J. L. Herder, "A compact low-stiffness six degrees of freedom compliant precision stage," *Precision Engineering*, vol. 37, no. 2, pp. 380–388, 4 2013.

3

Research paper

The use of a rigid linkage balancer with torsion springs to realize nonlinear moment-angle characteristics

Sjors van Nes
Dep. Precision
and Microsystems
Engineering
TU Delft
Delft, The Netherlands

Ali Amoozandeh Nobaveh
Dep. Precision
and Microsystems
Engineering
TU Delft
Delft, The Netherlands

Giuseppe Radaelli
Dep. Precision
and Microsystems
Engineering
TU Delft
Delft, The Netherlands

Just Herder
Dep. Precision
and Microsystems
Engineering
TU Delft
Delft, The Netherlands

Abstract—In this paper, the possibilities to approach various nonlinear moment-angle characteristics with a kinematically indeterminate rigid body balancer with torsion springs are examined. These torsion springs are mounted on the axes that intersect the rigid bodies. The rigid body balancer is coupled to an inverted pendulum. Although the kinematic indeterminate nature of the system enables the balancer to rotate non-proportionally along with the pendulum, the kinematics should correspond with the equilibrium configurations of the system. The required system parameters as spring stiffnesses and element lengths are obtained by optimization with a genetic algorithm. In addition to the standard optimization case, the effects of prestressed springs with contact release, nonlinear springs, optimizable initial configuration and an extra segment on the approximations are studied as well. Moreover, an extra objective function that concerns the distribution of energy among the springs is introduced. Eventually, the results that are obtained by the proposed method are verified with an experimental setup that contains a prototype of the system. The experimental results show agreement with the model with 93.47% work reduction. The corresponding model reduces the required work with more than 99%, which is higher than found in the state of the art.

Index Terms—Static balancing, preload, torsional stiffness, softening behaviour, inverted pendulum, gravity balancing, release of contact

I. INTRODUCTION

Statically balanced mechanisms are mechanisms that are in static balance for all possible configurations in their range of motion [1]. Correspondingly, the potential energy is constant throughout the range of motion of the mechanism. As a matter of fact, the potential energy function of a conservative system is obtained by integrating the load-displacement characteristic, which is constant and equal to zero. Therefore, no operating effort is required if a quasi-static translation or rotation is applied [2] [3]. As no operating effort is required, statically balanced mechanisms do not require heavy actuators or brakes and are therefore inherently relatively safe [4]. In literature, some effort is done to statically balance variations of the inverted pendulum. The inverted pendulum is a versatile model as it is used in the modeling of, for example, container walls [5], biped locomotion systems like the human body [6] [7] [8]

and wind turbines [9] [10] [11]. In the following, the common inverted pendulum with a point mass on its outer end will be referred to as just an inverted pendulum, unless mentioned otherwise. The moment induced by the mass is equal to a sine, which is a degressive characteristic. To statically balance an inverted pendulum, a balancing moment of equal magnitude but reverse direction should be generated. This balancing moment can be realized in multiple ways.

Statically balanced mechanisms typically include a balancing mass or a spring, which is often a zero free length spring [12] [13]. Masses are more frequently used, but major disadvantages are increased mass, inertia and volume of the overall system [14]. Alternatively, in order to approximate a nonlinear curve, one attempts to realize a nonlinear relationship between the rotation of the pendulum and the rotation of the energy storage unit in the balancer. This nonlinear transmission enables the balancer to provide a nonlinear load-deflection characteristic with linear springs.

Endo et al. accomplished this by the implementation of a linear spring in combination with a pulley with a nonlinear radius [15]. The pulley was thus used as a nonlinear transmission, enabling the balancer to provide a nonlinear moment-angle characteristic with a linear spring. The maximum torque at the suspension point of the pendulum was reduced by more than 90%. The statically balanced part of the range of motion was between 18° and 90° with respect to the vertical.

Bijlsma, Herder and Radaelli reported a 86.8% work reduction in the actuation of an inverted pendulum in a range of motion of two complete rotations [16]. The nonlinear counteracting moment was obtained by interconnecting the inverted pendulum and a cluster of torsion bars by a gear train. The gear train consisted out of regular gears and gears with optimized shape, which were designed to realize a nonlinear relation of the input shaft rotation and the output shaft rotation. The inverted pendulum was connected to the input shaft, whereas the output shaft was attached to the cluster of torsion bars.

Shieh and Chou evaluated the balancing qualities of a

Scotch yoke mechanism in combination with a compression spring and a gear pair to statically balance an inverted pendulum [17]. The Scotch yoke mechanism was utilized to realize a nonlinear relation between the rotation of the pendulum and the amount of energy stored in the compression spring. System parameters were adjusted such that the sum of the energy stored in the spring and the height energy of the mass was equal to a constant. A design modification with an extra pin was applied in order to statically balance the pendulum in a 180 degree range of motion.

Dede and Trease connected an optimized four-bar mechanism to an inverted pendulum to achieve a reduction in actuation energy of more than 97% in a 90 degree range of motion [18]. The nonlinear potential energy characteristic needed to balance the pendulum was obtained by an optimization of the geometry of the system, such that the relation between the rotation of the inverted pendulum and the rotation of the torsion springs was nonlinear.

Another approach for the design of a nonlinear force-deflection or moment-angle curve is the use of prestress in combination with release of contact. Whereas the previously mentioned work focused on a nonlinear rotation of the balancer during linear rotation of the pendulum, release of contact could cause the resultant stiffness to be non-constant.

Claus investigated the use of prestressed parallel and series connected torsion bars to balance an inverted pendulum [5]. With release of contact of these pretensioned bars a bilinear approximation of a quarter period of a sine was obtained. Radaelli et al. stated that the work of Claus was the only example of a system with torsion springs and a positive degressive moment-angle characteristic [19]. Radaelli subsequently built a prototype with three prestressed torsion bars to realize a trilinear approximation of the same nonlinear characteristic. A 99% work reduction was achieved by a parallel connection of these torsion bars.

Radaelli and Herder used isogeometric shape optimization to statically balance an inverted pendulum with a prestressed compliant beam [20]. Eventually, a prototype with a carbon fibre composite beam was developed, which illustrated a work reduction of 96.98% for a 180 degree rotation of the inverted pendulum.

The implementation of optimized cams as in the work of Endo et al. could be an appropriate alternative to the use of a balancer with a zero free length spring. However, the statically balanced section of the range of motion is still relatively small. Moreover, the balancer suffers from errors originated by discrepancies between the theoretical model and the prototype, like the nonzero thickness and finite stiffness of the wire that is making contact with the pulley. Although Bijlsma, Herder and Radaelli reported a relatively low work reduction, the gear train based balancer operated in a relatively large domain of two complete rotations. The gear train, however, increases the complexity of the system, whereas the cluster of torsion bars still requires available space perpendicular to the pendulum. The balancer of Shieh and Chou is less complex as the gear pair consists of two regular gears with tooth ratio 2:1. Despite

the fact that the inverted pendulum would be balanced in a 180 degree range of motion, no physical prototype is made to evaluate the performance. As a result, the friction in the Scotch yoke mechanism and the gears is not quantified yet. Furthermore, a practical implementation is also in need of a transmission that would connect the compression spring with the hinge that is experiencing the moment induced by the point mass. The four-bar linkage designed by Dede and Trease was optimized with a constraint on the stresses in the joints and the distribution of spring energy is expected to be easier controlled. A disadvantage of the presented prototype is again the occupied volume in the direction orthogonal to the degree of freedom of the pendulum, as is the case for the statically balanced systems presented by Claus and Radaelli et al. as well. The compliant carbon fibre balancer by Radaelli and Herder does not have this limitation, but energy storage and stresses could concentrate at locations like the suspension point of the balancer [21].

It is expected that it would be of great interest to present a balancer that is relatively simple, occupies minimal space and has relatively high balancing performance. In addition, it could be beneficial to distribute the energy in the system. This energy distribution could result in an inherently safer and less costly balancer. The simplicity could be manifested by eliminating the need for a transmission like the presented gear trains and cams. Omitting such a transmission would reduce friction and eventually decrease wear and maintenance costs. The volume requirement could be fulfilled by designing the balancer to be conform to the pendulum. Both the balancing performance and the energy distribution, on the other hand, depend on the method that provides the system parameters. These parameters could be selected such that the balancer realizes an as high as possible balancing performance. This selection could be done with an optimization algorithm.

The objective of this work is to examine the possibilities to statically balance various nonlinear moment-angle characteristics with a kinematically indeterminate rigid body balancer with torsion springs and to verify the results that are obtained by the proposed method with an experimental setup that contains a prototype of the system. The research effort focuses on the effect of nonlinear springs, prestressed springs with contact release and the initial configuration on the balancing performance of a three segment balancer, while the effect of nonlinear and prestressed springs (with contact release) on the energy distribution is studied as well. Moreover, the balancing potential of a four segment balancer with nonlinear and prestressed springs (with contact release) is analyzed.

After the theoretical evaluation in section III-A, section III-B describes the steps taken during the prototyping and experimental phase. Subsequently, chapter IV provides both the modeling results in section IV-A and the experimental results in section IV-B. The results are then discussed and conclusions are drawn in chapter V and chapter VI, respectively. First, chapter II will elaborate on the principle of release of contact to obtain softening behaviour in load-displacement characteristics.

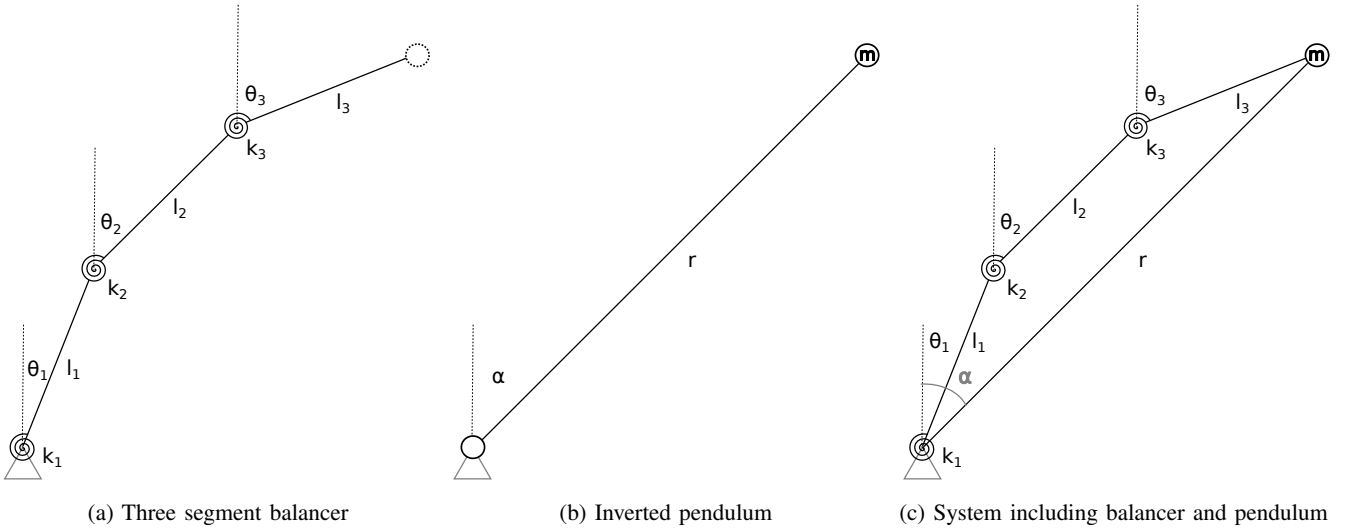


Figure 1: Schematic overview of the proposed system

II. FUNDAMENTALS

Release of contact can be used to obtain a degressive force-displacement or moment-angle characteristic. Figure 2, a cropped version of a figure from Radaelli et al., illustrates the working principle. The resultant stiffness of parallel connected springs equals the sum of the stiffnesses of the separate springs, whereas the reciprocals add up in a series connection. The stiffnesses of a parallel and series connection of N springs are provided in equation 1 and equation 2, respectively. The resultant stiffness is denoted by k_T , whereas k_i represents the stiffnesses of the individual springs.

$$k_T = \sum_{i=1}^N k_i \quad (1)$$

$$\frac{1}{k_T} = \sum_{i=1}^N \frac{1}{k_i} \quad (2)$$

A schematic of a system with a degressive characteristic and springs in series is shown in figure 2a. The leftmost spring is prestressed and hold fixed by a contact with the environment, as a result of which only the right spring is deforming when a displacement is applied. The black dot illustrates the contact with the environment, which is maintained until the applied force transcends the preload in the system. The spring is engaged and starts deforming when this preload is exceeded. The stiffness $k = \frac{F}{x}$ accordingly decreases as the resultant stiffness of a series connection of springs is lower than the stiffness of the individual springs. The same result could be obtained with the schematic of figure 2b as well. The latter case concerns a parallel connection where the prestressed right spring is loaded until the connection with the environment, again indicated with the black dot, is lost. Only the left spring is then contributing to the stiffness at the actuation point, which results in a decrease in resultant stiffness.

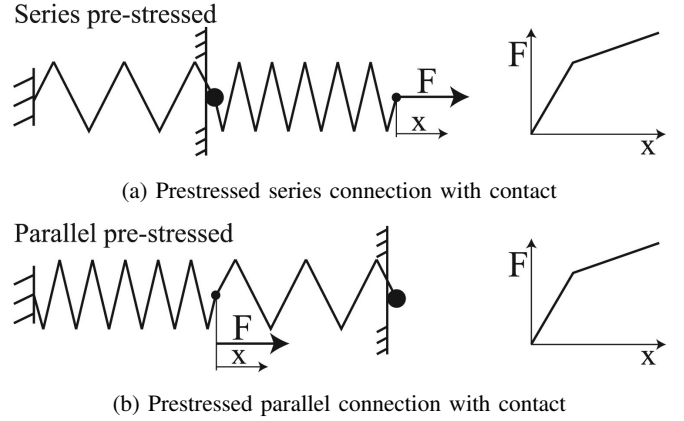


Figure 2: Series and parallel spring systems with a degressive force-displacement characteristic [19]

III. METHODS

The balancing mechanisms are first studied in MATLAB, whereafter a prototype is made and experiments are done. Section III-A describes the approach taken with respect to the MATLAB modeling part. Sequentially, section III-B elaborates on the prototyping and experiment aspects.

A. Model

Balancer mechanism

The three segment balancer is shown in its simplest form in figure 1a. By connecting the balancer to an inverted pendulum, illustrated in figure 1b, the basic four bar mechanism that is depicted in figure 1c is created. The mobility is determined by applying the Chebychev-Grübler-Kutzbach criterion formulated in equation 3 [22]. Here δ denotes the number of degrees of freedom (DOF) of the mechanism, f_α the DOF of the separate joints, g the loop connectivity, j the number of

joints and b the number of bodies. It is seen that the system consists of four joints and four bodies, where the pendulum as a body is considered to be the ground of the system.

$$\delta = \sum_{\alpha} f_{\alpha} - g(j - b + 1) \quad (3)$$

$$\delta = 4 - 3(4 - 4 + 1) = 1$$

The total system consisting of the balancer and the inverted pendulum thus has one internal degree of freedom. The same formula is applied for the four segment balancer as well. The system with the four segment balancer consists of five joints and five bodies. It is seen that the total system has two internal degrees of freedom.

$$\delta = 5 - 3(5 - 5 + 1) = 2$$

Kinematics

Because of the internal degree of freedom, the configuration of the system with the three segment balancer is known when the positions of at least two bodies are prescribed. The loop closure equations are evaluated in order to obtain analytical expressions for the angles of the other two bodies [23]. The loop closure equations for the three segment balancer system depicted in figure 1c are provided in equation 4 for the x-coordinate and equation 5 for the y-coordinate.

$$l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_3 \sin(\theta_3) = r \sin(\alpha) \quad (4)$$

$$l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3) = r \cos(\alpha) \quad (5)$$

Solving this system of equations for θ_2 and θ_3 would result in relatively long nonlinear expressions that are inconvenient to solve by hand. Therefore, MATLAB *Symbolic Math Toolbox* is used to obtain symbolic expressions for the angles of segment 2 and 3 that depend on θ_1 and α . Solving equation 4 and equation 5 thus results in equation 6 and equation 7.

$$\theta_2 = f(\theta_1, \alpha, l_1, l_2, l_3, r) \quad (6)$$

$$\theta_3 = f(\theta_1, \alpha, l_1, l_2, l_3, r) \quad (7)$$

In the case of the three segment balancer, the angle of the first segment is determined for each angle of the pendulum. This angle can not be chosen arbitrarily as it should correspond with the equilibrium configuration of the system. In order to ensure that the found configuration is an equilibrium configuration, the potential energy of the system is evaluated for a finite amount of positions of the first segment. For a given angle of the inverted pendulum and the first segment, the angles of the second and third segment are determined by analytical expressions 6 and 7. The solutions are further divided into a group with ‘‘elbow-down’’ solutions and a group with ‘‘elbow-up’’ solutions. In case of an elbow-down solution the angle of the second segment with respect to the vertical

is larger than the angle of the third segment with respect to the vertical, whereas the angle of segment three is larger than that of the second segment for the elbow-up solution. In the following, the elbow-up solutions are adopted for reasons of convenience. By evaluating the potential energy for a finite amount of values of θ_1 for each angle of the pendulum, a matrix with dimensions $n \times m$ is created. The expression for the potential energy is provided in equation 8. The angles of deformation of the first, second and third spring are denoted by α_{1ji} , α_{2ji} and α_{3ji} , respectively. Analogously, k_1 , k_2 and k_3 refer to the stiffness of the first, second and third spring. The angles of deformation are expressed in the angles of the segments with respect to the vertical in equations 9, 10 and 11.

$$V_{ji} = \frac{1}{2}k_1\alpha_{1ji}^2 + \frac{1}{2}k_2\alpha_{2ji}^2 + \frac{1}{2}k_3\alpha_{3ji}^2 \quad (8)$$

$$\alpha_{1ji} = \theta_{1ji} - \theta_{10} \quad (9)$$

$$\alpha_{2ji} = \theta_{2ji} - \theta_{1ji} - (\theta_{20} - \theta_{10}) \quad (10)$$

$$\alpha_{3ji} = \theta_{3ji} - \theta_{2ji} - (\theta_{30} - \theta_{20}) \quad (11)$$

Extra terms should be added to equation 8 if one or more springs are prestressed. The additional potential energy term that is to be included when a spring is prestressed is shown in equation 12.

$$V_{ji_p} = M_0\alpha_{ji} + \frac{M_0^2}{2k} \quad (12)$$

As illustrated in figure 1, the orientation of the segments is expressed with their angle with respect to the vertical θ . The subscript j is used to denote that the scalar value corresponds to the j^{th} configuration of the pendulum, whereas i refers to the position of the first segment. The initial angles of the first, second and third segment are indicated by θ_{10} , θ_{20} and θ_{30} , respectively. The variable M_0 lastly represents the prestress in the torsion spring. For the modeling of the system with nonlinear springs, nonlinear springs with both a first order and second order coefficient in their moment-angle characteristic are used. The analytical formulation of the corresponding potential energy, using α_k as the variable representing the deformation of the k^{th} spring, is presented in equation 13. Variables A and B are the third order and second order component of the potential energy, respectively. The three nonlinear springs are identical in this analysis.

$$V_{ji} = \sum_{k=1}^3 \left(\frac{A}{3}\alpha_{kji}^3 + \frac{B}{2}\alpha_{kji}^2 \right) \quad (13)$$

The potential energy of a prestressed nonlinear spring is provided in equation 14. Here α^* denotes the applied rotation corresponding to the preload. The relation between the preload and its corresponding rotation is found by taking the derivative of equation 14 with respect to the variable α_{ji} , resulting in

equation 15. This equation is used to calculate the internal moment of a prestressed nonlinear spring. Substituting $\alpha_{ji} = 0$ then yields an expression for the preload, given in equation 16. Rewriting for α^* provides the set of solutions given in equation 17. As M_0 is chosen to be a minimizer, the smallest non-negative α^* of the set is stored and used for calculation of the potential energy.

$$V_{ji} = \frac{A}{3} (\alpha_{ji} + \alpha^*)^3 + \frac{B}{2} (\alpha_{ji} + \alpha^*)^2 \quad (14)$$

$$M_{ji} = A(\alpha_{ji} + \alpha^*)^2 + B(\alpha_{ji} + \alpha^*) \quad (15)$$

$$M_0 = M_{ji}|_{\alpha_{ji}=0} = A\alpha^{*2} + B\alpha^* \quad (16)$$

$$\alpha^* = \frac{-B \pm \sqrt{B^2 + 4AM_0}}{2A} \quad (17)$$

The analysis of the four segment balancer is analogous to that of the three segment balancer, albeit that an extra degree of freedom is introduced. As a result, the position of an extra segment should be known in order to fully define the configuration of the system. Therefore, for each precision point and position of the first segment, segment two is swept through its range of motion as well. As with the analysis of the three segment balancer the lowest potential energy configuration is selected.

Performance evaluation

The balancing performance of the balancers is evaluated by means of the normalized root mean square error, as shown in equation 18.

$$f_1 = \frac{1}{mgr} \sqrt{\frac{\sum_{j=1}^N (M_{1_j} - M_{obj_j})^2}{N}} \quad (18)$$

The objective moment at a certain angle of the pendulum is denoted by M_{obj_j} , whereas M_{1_j} is the actual balancing-moment at this configuration. The sum of the squared differences of these values is then divided by the amount of evaluated angles of the pendulum N . The root mean square error is divided by the amplitude of the objective moment-angle curve in order to facilitate a convenient comparison of systems with different masses and pendulum lengths. In this case, the amplitude is equal to the magnitude of the point mass times the gravitational constant and the length of the pendulum, respectively.

Equation 19 is used as a measure of the energy distribution between the springs. The squared difference in potential energy between spring 1 and spring 2 is denoted by ΔV_{12_j} , as formulated in equation 20. Similarly, equation 21 and equation 22 quantify these squared differences for the first and the third spring, and the second and the third spring, respectively.

$$f_2 = \frac{1}{mgrN} \sum_{j=1}^N \sqrt{(\Delta V_{12_j})^2 + (\Delta V_{13_j})^2 + (\Delta V_{23_j})^2} \quad (19)$$

$$(\Delta V_{12_j})^2 = (V_{1m_j} - V_{2m_j})^2 \quad (20)$$

$$(\Delta V_{13_j})^2 = (V_{1m_j} - V_{3m_j})^2 \quad (21)$$

$$(\Delta V_{23_j})^2 = (V_{2m_j} - V_{3m_j})^2 \quad (22)$$

Objective functions

The moment-angle characteristic corresponding to an inverted pendulum with a point mass connected to its end is given in equation 23. To statically balance the inverted pendulum, a balancing moment with equal magnitude but opposite sign is needed. The objective function for this basic inverted pendulum is provided in equation 24.

$$M_p = -mgr \sin(\alpha) \quad (23)$$

$$M_{obj_p} = mgr \sin(\alpha) \quad (24)$$

To examine the versatility of the method, five other moment-angle objective functions are adopted as well. Equation 18 and its corresponding optimization formulation are used for all of these functions. The balancer will be coupled to an inverted pendulum again. As a result, the distance of the connection-point with the environment to the outer end will be restricted to be equal to the length of the pendulum. Equation 25 represents a progressive objective curve, while equation 26 and equation 27 denote objective curves with a transition from progressive to degressive behaviour and vice versa, respectively. Equation 28 is a normalized fit of the moment-angle characteristic used at Laevo. The last objective function will be a scaled half period of a sine, as formulated in equation 29. In this work, the length $r = 1$ and the gravitational force $mg = 1$.

$$M_{obj_h} = -mgr \cos(\alpha) + mgr \quad (25)$$

$$M_{obj_{ns}} = 0.5 + \frac{4}{3\pi} \arctan \left(\tan \left(\frac{3\pi}{8} \right) \left(\frac{4}{\pi} \alpha - 1 \right) \right) \quad (26)$$

$$M_{obj_{sn}} = 0.5 \frac{\tan \left(1.5 \left(\alpha - \frac{\pi}{4} \right) \right)}{\tan \left(1.5 \left(\frac{\pi}{4} \right) \right)} + 0.5 \quad (27)$$

$$M_{obj_L} = -0.25\alpha^4 + 1.34\alpha^3 - 2.91\alpha^2 + 2.82\alpha - 0.01 \quad (28)$$

$$M_{obj_s} = \sin(2\alpha) \quad (29)$$

Optimization

The genetic algorithm solver from the MATLAB *Optimization Toolbox* is used to find the system parameters that result in the lowest possible normalized root mean square error of the system. The simplest optimization study concerns the optimization of the spring stiffnesses and the element lengths, as illustrated below.

$$\min_{k_1, k_2, k_3, l_1, l_2, l_3} \sqrt{\frac{\sum_{j=1}^N (M_{1j} - M_{\text{obj}j})^2}{N}}$$

$$s.t. 0 \leq k_i \leq \frac{3}{2}$$

$$\frac{1}{3} \leq l_i \leq \frac{1}{2}$$

As the energy distribution among the springs is considered as well, an extra objective function is formulated.

$$\min_{k_1, k_2, k_3, l_1, l_2, l_3} \frac{1}{mgrN} \sum_{j=1}^N \sqrt{(\Delta V_{12j})^2 + (\Delta V_{13j})^2 + (\Delta V_{23j})^2}$$

$$(\Delta V_{12j})^2 = (V_{1m_j} - V_{2m_j})^2$$

$$(\Delta V_{13j})^2 = (V_{1m_j} - V_{3m_j})^2$$

$$(\Delta V_{23j})^2 = (V_{2m_j} - V_{3m_j})^2$$

$$s.t. 0 \leq k_i \leq \frac{3}{2}$$

$$\frac{1}{3} \leq l_i \leq \frac{1}{2}$$

The optimization is slightly more involved for the optimization of prestressed springs, nonlinear springs, the angle of the first segment and the four segment balancer. The typical lower- and upperbounds are provided below.

$$0 \leq M_{3_0} \leq 1$$

$$0 \leq M_{2_0} \leq 1$$

$$-2 \leq A \leq 2$$

$$-2 \leq B \leq 2$$

$$-\pi \leq \theta_{1_0} \leq \pi$$

For the Laevo and 180 degree sine objective functions, $-3 \leq A \leq 3$ and $-3 \leq B \leq 3$. In the case of the four segment balancer, the element lengths are constrained to be $\frac{1}{4} \leq l_i \leq \frac{3}{8}$. Iteratively, it was found that the upperbound of the stiffness should be relaxed for the hardening-softening, softening-hardening, Laevo and 180 degree sine objective functions. For these objectives, the stiffness is restricted to be $0 \leq k_i \leq 2.5$. Lastly, the linear and nonlinear spring parameters are relaxed as well in the case of an optimization run with optimizable angle of the first segment. The lower- and upperbounds are defined below.

$$0 \leq k_i \leq 4.5$$

$$-3.5 \leq A \leq 3.5$$

$$-3.5 \leq B \leq 3.5$$

All optimization runs are executed with a pendulum length $r = 1$ and an initial configuration with zero potential energy or a potential energy that equals the prestress in the springs that are enabled via contact release.

Because the amount of readily available clock springs is limited, an optimization routine is implemented that selects off-the-shelf springs from the Lesjöfors catalogue such that the stiffness ratios approximately correspond with the stiffness ratios found by optimization. The genetic algorithm from MATLAB *Optimization Toolbox* is used to execute the optimization. The minimizers x_1, x_2, x_3, x_4, x_5 and x_6 are the entries of the vector with available stiffnesses v_k . A parallel connection of springs is allowed as well. The springs with stiffnesses k_1 and k_2 thus correspond with the first axis, those with k_3 and k_4 with the second axis and k_5 and k_6 correspond with the third axis. The ratio k_1/k_2 is represented by the variable r_1 , whereas r_2 is equal to k_3/k_4 .

$$\min_{x_1, x_2, x_3, x_4, x_5, x_6} \sqrt{\left(\frac{k_1 + k_2}{k_3 + k_4} - r_1\right)^2 + \left(\frac{k_5 + k_6}{k_3 + k_4} - r_2\right)^2}$$

$$k_1 = v_k(x_1)$$

$$k_2 = v_k(x_2)$$

$$k_3 = v_k(x_3)$$

$$k_4 = v_k(x_4)$$

$$k_5 = v_k(x_5)$$

$$k_6 = v_k(x_6)$$

$$s.t. 1 \leq x_i \leq 13$$

The design variables are constrained to be integer values, such that their number corresponds with a spring from the catalogue of available springs. In total, 12 different springs are included. The upperbound of x_i is equal to 13 as $x_i = 13$ corresponds with a stiffness $k = 0$, to allow for no parallel connection of springs as well.

Modeling scheme

The modeling approach for the standard three segment balancer is summarized in the schematic in figure 3. First, the MATLAB solver selects values for the minimizers. Thereafter, the pendulum is given a small perturbation. For the new position of the pendulum, the potential energy is calculated for all possible configurations of the balancer. This results in an array with a length equal to the amount of evaluated configurations. Sequentially, this loop of evaluation is repeated for the exact same selection of system parameters until the pendulum angle is equal to its upperbound. The fifth block in figure 3 corresponds with this situation. The potential energy matrix, created by concatenating the separate arrays for each angle of the pendulum, is converted into an array again. This is done by storing the lowest potential energy value for each row. Finally, the objective function, in this case the normalized root mean square error, is calculated and the large loop of figure 3 starts over.

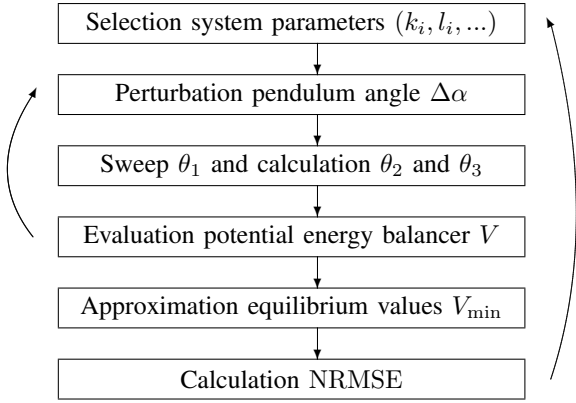


Figure 3: Modeling scheme standard three segment balancer

B. Prototype and experiment

Prototype setup

The prototype represents the system consisting of the inverted pendulum and the three segment balancer, as depicted in figure 1c. The system is mounted horizontally, as a result of which the moment induced by the gravitational force is perpendicular to the balancing moment. The mass that is connected to the outer end of the model of the inverted pendulum is omitted in the prototype. The segments of the balancer are 3D-printed PLA, whereas the pendulums are laser-cut PMMA parts. The system consists of two pendulums to realize a more symmetric design. One pendulum is mounted below the balancer, whereas the other one is located above the balancer. The PLA segments are interconnected by stepped steel axes with a slit to facilitate connection with the torsion springs. The arbors of the clock springs are mounted on the heads of these axes, whereas their outer connection points are fixated to the segments by means of M3 threaded rod. The system is shown in the initial configuration in figure 4a.

Ball bearings are used to enable a rotation of the second axis with respect to the first segment, to rotate the third axis relative to the third segment and to rotate the pendulums with regard to the axes they are mounted on. Moreover, two ball bearings are installed on the L-shaped PLA part, which is called the “pushing bracket” in figure 4, to attach it to the first and main axis. A FUTEK LSB200 Miniature S-Beam Jr. Load Cell is connected to the other end of this part. The ends of the first axis contain a ball bearing as well, to connect this axis with a Thorlabs MB3060/M breadboard and two Thorlabs XE25L225/M construction rails via 3D-printed PLA connection parts. Set crews are applied to constrain the degrees of freedom between the axes and segments that should not rotate relative to each other. A Cherry AN8 angular position sensor is attached to the upper PLA plateau. This Hall effect sensor consists of a rotating part and a stationary part. The rotating part is connected to the upper pendulum by another PLA part and does not touch the fixed member.

System parameters

Both the system parameters obtained from optimization and the properties of the prototype are provided in table I. The segment lengths of the prototype are halved with respect to the lengths obtained from the optimization, which is done to facilitate 3D-printing. The stiffness ratio of the first and second optimized spring is provided in equation 30, whereas the ratio of the third and second spring is given in equation 31. These are the ratios corresponding to the spring stiffnesses found by optimization.

$$r_1 = \frac{k_1}{k_2} = \frac{1.27}{0.85} = 1.50 \quad (30)$$

$$r_2 = \frac{k_3}{k_2} = \frac{4.11}{0.85} = 4.85 \quad (31)$$

The stiffness ratios corresponding to the prototype are provided in equation 32 and equation 33.

$$r_{1p} = \frac{k_{1p}}{k_{2p}} = \frac{0.037}{0.025} = 1.51 \quad (32)$$

$$r_{2p} = \frac{k_{3p}}{k_{2p}} = \frac{0.12}{0.025} = 4.84 \quad (33)$$

Parameter	Optimization	Prototype	Unit
k_1	1.27	0.037	Nm/rad
k_2	0.85	0.025	Nm/rad
k_3	4.11	0.12	Nm/rad
l_1	0.34	0.17	m
l_2	0.46	0.23	m
l_3	0.48	0.24	m
θ_{10}	0.66	0.66	rad
r	1.00	0.50	m

Table I: System parameters corresponding to prototype

IV. RESULTS

A. Model

The optimization results for the various configurations of the three segment balancer are depicted in figure 5. The distinct objective functions are shown along the x-axis, whereas the y-axis illustrates the best obtained work reduction for each system. The work in the balanced and reference configurations is determined by calculating the area below the moment-angle curve for both the balanced and the unbalanced system, respectively. The MATLAB function *trapz* is used to estimate this area. The work reduction percentage is then calculated by dividing the difference in work by the work in the reference configuration, as shown in equation 34. The required work in the reference configuration is indicated by W_{ref} , whereas W_{bal} represents the work corresponding to the balanced system.

$$W_{\text{red}} = 100 \frac{W_{\text{ref}} - W_{\text{bal}}}{W_{\text{ref}}} \quad (34)$$

The circular markers indicate linear balancers, whereas the square markers represent systems with nonlinear springs. The balancers with prestress with contact release are referred to as

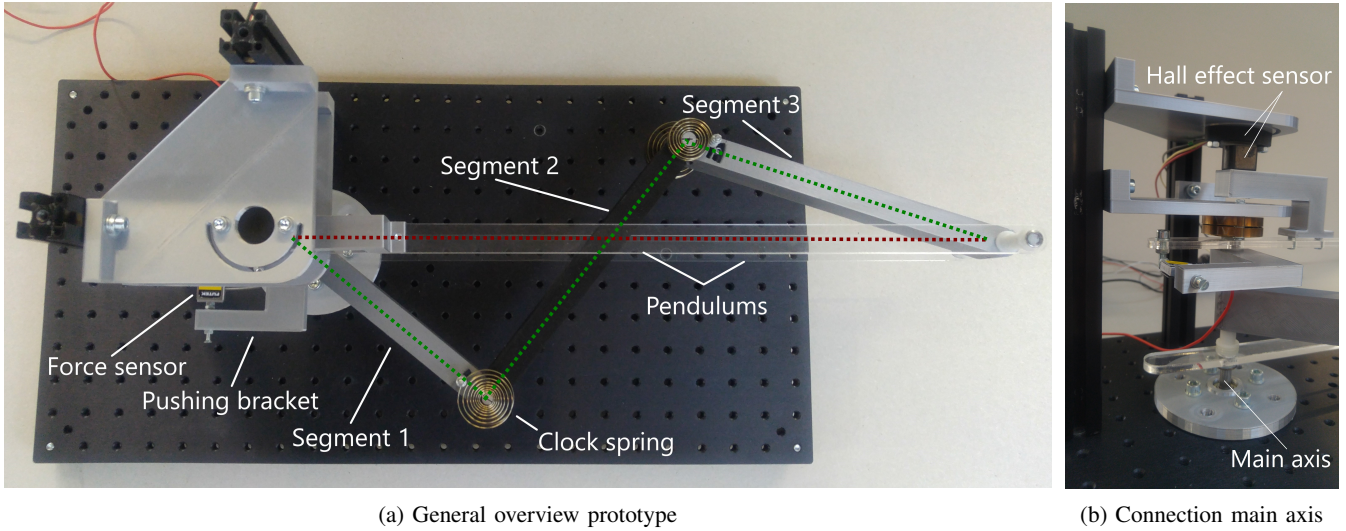


Figure 4: Prototype three segment balancer with pendulums

having “Prestress” in the legend. The work reduction of the regular three segment balancer is 38.47% for the 180 degree sine objective function and is omitted in the figure in order to preserve the overview.

Analogously, the performance of the four segment balancer is presented in figure 6. As seen in the legend of the figure, only systems without optimizable initial configuration are evaluated in case of the 4 segment balancer. The 43.96% work reduction of the regular balancer is again omitted for the 180 degree sine objective moment-angle characteristic.

The results of the multi-objective optimization are shown in figure 7. The normalized root mean square error is represented by the x-axis, whereas the y-axis quantifies the magnitude of the objective function that is related to the energy distribution between the springs.

B. Experiment

The measured moment-angle characteristic corresponding to the prototype of the three segment balancer is shown in figure 8. The blue dotted curve indicates the original objective function, which is a quarter period of a sine. The red curve denotes the result obtained by the MATLAB model, where the discrepancies in spring stiffness ratios are taken into account. The grey plot lastly represents the measured hysteresis loop. The shown hysteresis loop is obtained by executing ten measurement runs, concatenating the obtained data arrays and applying a moving average filter that averages 0.5% of the total amount of data points to create a new datapoint. The work reduction is found to be 93.47%.

V. DISCUSSION

In general, figure 5 and figure 6 show relatively high balancing performance for all objective functions. Six three segment balancers and three four segment balancers reduce the required work for the sine objective function by more than

99%. These balancers thus realize a higher work reduction than the systems mentioned in section I. Although the other objective functions are balanced with a relatively high work reduction as well, only the progressive and Laevo objective characteristics have a maximum work reduction that is similar to that of the sine objective function. The measurement results, shown in figure 8, illustrate softening behaviour of the balancer and moment-angle points that are comparable to that of the expected balancing curve. In the following, the results corresponding to the optimization of the three segment balancer and four segment balancer will be analyzed further. Sequentially, figure 7 will be discussed. This figure illustrates the found approximations of the Pareto set for three distinct balancers. Lastly, an interpretation of the measurement results is given. In the following discussion, the balancers with prestressed springs with contact release will be referred to as the balancers with prestress for reasons of convenience.

A couple of observations can be made by inspection of figure 5. The first of which is regarding the performance of the regular three segment balancer. The regular three segment balancer has the lowest work reduction for the 90 degree sine, the degressive-progressive, the Laevo and the 180 degree sine objective functions. For the progressive objective curve, on the other hand, the regular balancer has the highest work reduction of all balancers. The second observation is the fact that the three segment balancers with linear springs and an optimizable initial angle of the first segment have the lowest work reduction of all balancers for the progressive objective function, whereas their work reduction for the other objective functions is relatively high. For the objective functions other than the progressive objective function, the work reduction of these balancers with optimizable initial angle of segment 1 is higher than the work reduction of the other linear balancers. The third observation concerns the relatively low work

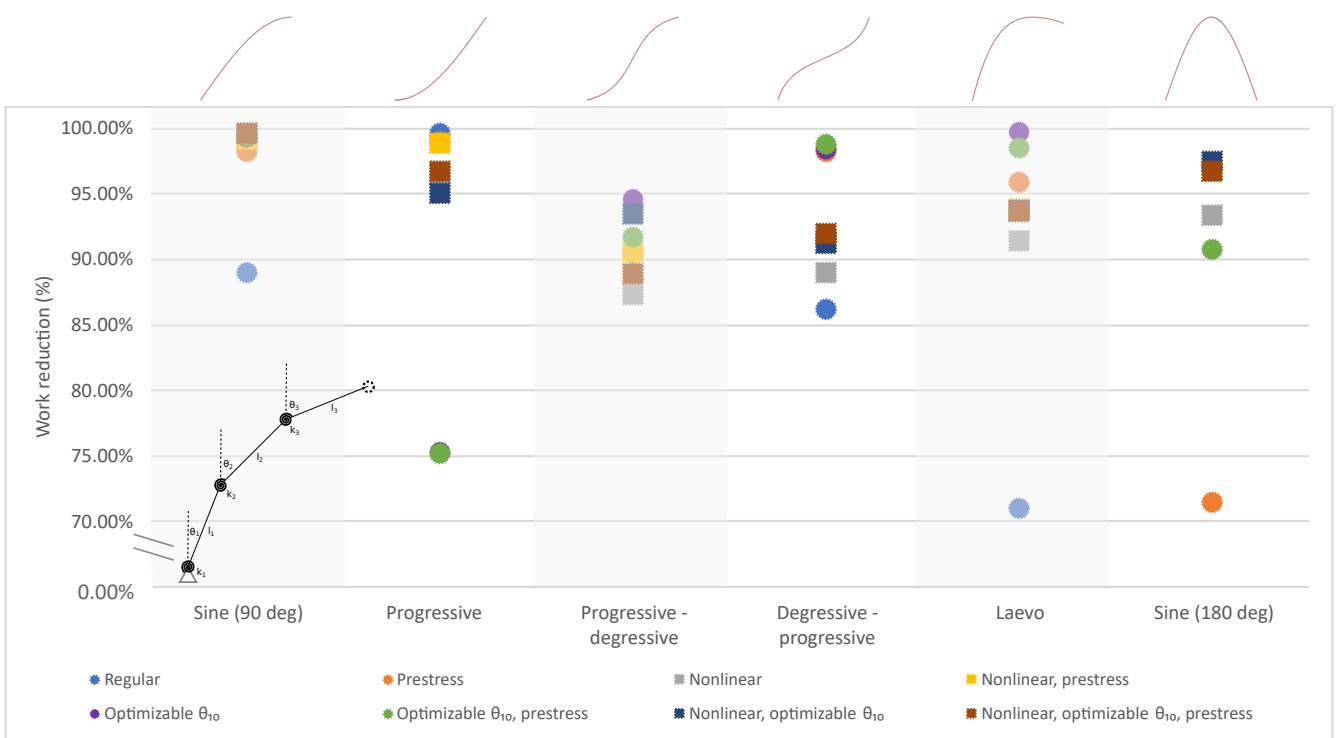


Figure 5: Modeling results three segment balancer

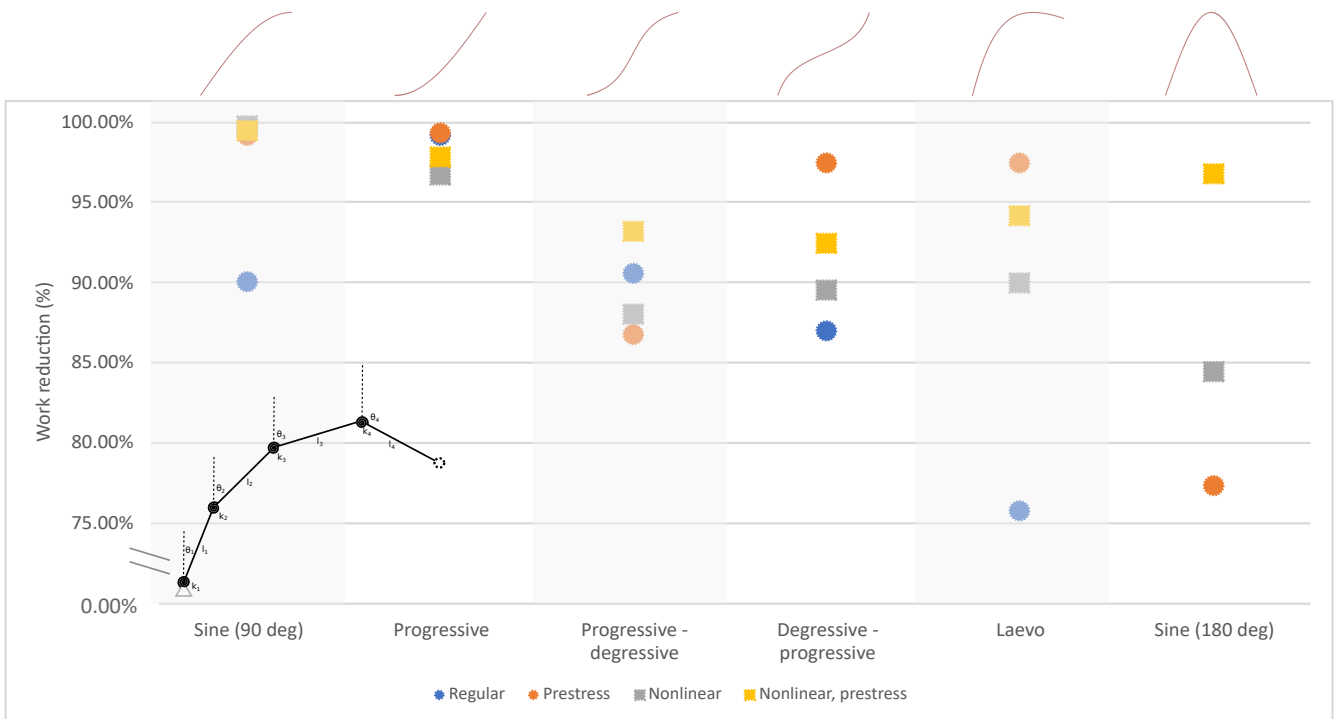


Figure 6: Modeling results four segment balancer

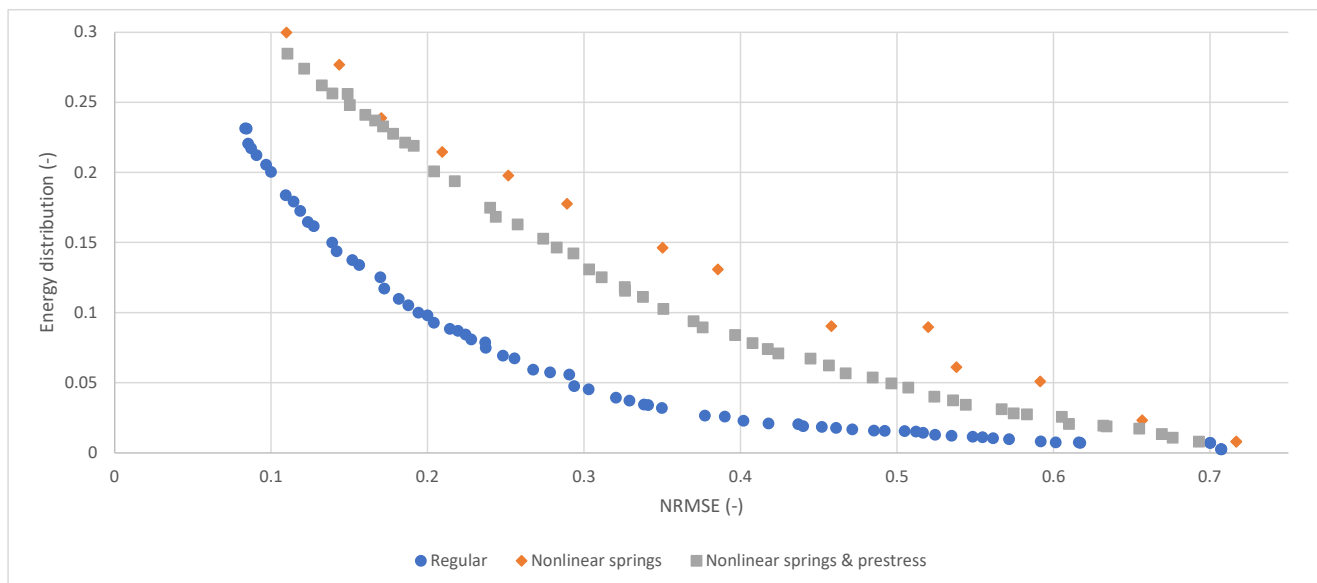


Figure 7: Pareto sets three segment balancer

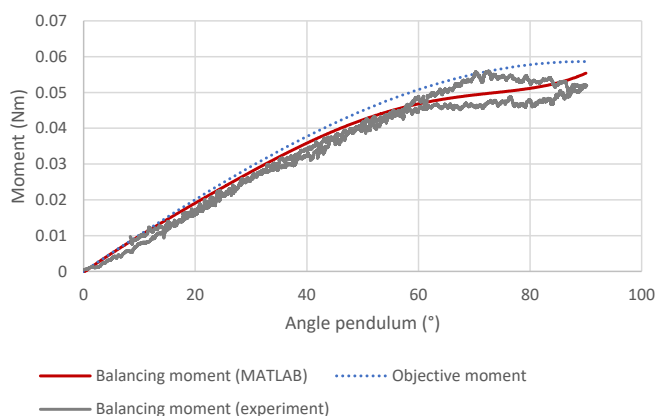


Figure 8: Measurement results experimental setup

reduction of all three segment balancers for the progressive-degressive objective function, as no balancer is able to reduce the on the pendulum exerted work by more than 95%. It should be noted as well that all balancers have a performance that is relatively close to the average for this objective function. Lastly, greater differences in performance are seen for the degressive-progressive, Laevo and 180 degree sine objective functions when comparing balancers with linear springs to the nonlinear variants. If the regular balancer is not taken into consideration, it can be stated that the linear configurations realize higher work reduction than the nonlinear balancers for the degressive-progressive and the Laevo objective functions. On the other hand, most balancers with nonlinear springs achieve a higher work reduction than the systems equipped with linear springs for the 180 degree sine objective function.

The regular three segment balancer, the linear balancer

without prestress and without optimizable initial angle, is the balancer with the lowest work reduction for all objective functions except the progressive and progressive-degressive characteristics. It is understood that the balancers with prestress and nonlinear springs have an advantage compared to the standard balancer. As a matter of fact, prestress and the corresponding release of contact enable the forced engagement of springs. As the springs in the balancers are connected in series, this engagement allows for lower stiffness from the angle of activation onward. The latter facilitates softening behaviour of the balancer itself, which is of great use in approximating objective functions with softening behaviour as the 90 degree sine, the degressive-progressive, the Laevo and the 180 degree sine objective functions. The balancers with nonlinear springs are also able to approximate these functions relatively well, as the optimizer is able to select springs that already have a degressive load-displacement characteristic. One should be careful interpreting the relatively high work reduction of the regular balancer for the progressive and the progressive-degressive objective functions, however. Theoretically, the prestress, optimizable initial angle of segment 1 and the nonlinear springs are only additions to the standard balancer. In other words, the optimization algorithm is allowed to select zero prestress, an initial angle of segment 1 equal to 0° and a second order coefficient of the nonlinear springs equal to zero. The only complication is that the nonlinear springs are confined to have the same characteristic, which degrades this freedom for the nonlinear balancers. It is expected that this extra design freedom is not well utilized as the optimizer converged to a relatively high local minimum for each of these non-regular balancers. The results, on the other hand, are obtained with use of a genetic algorithm, which is an algorithm with a random nature. Consequently, no firm conclusions can

be drawn from the collected data.

The work reduction of the linear balancers with optimizable initial angle of segment 1 is relatively low compared to the performance of the other linear balancers for the progressive objective curve. Again, one should be aware of the fact that the former mentioned balancers are comparable to the regular balancer, which has the highest work reduction of all configurations for this objective function. The fact that no theoretical restriction for a better approximation of the goal function exists is highlighted by the moment-angle characteristics obtained by the optimization routine. The progressive objective function is, for both linear balancers with optimizable angle of segment 1, approximated by a linear curve. The same balancers, on the other hand, show both softening and hardening behaviour in their approximation of the progressive-degressive and degressive-progressive objective functions. As the genetic algorithm is ran several times for all of these objective curves, it is expected that the solutions for the progressive objective curve under discussion are local minima.

The observation regarding the relatively low work reduction for the progressive-degressive objective function is not directly explained by inspection of the optimization results. It is found that only the linear and nonlinear balancers with optimizable initial angle of segment 1 realize a moment-angle characteristic where both hardening and softening behaviour can be observed. The other balancers either have a linear approximation or a progressive balancing characteristic. In the case of the balancers with both hardening and softening behaviour, the curve has less curvature than the objective function and thus is closer to a linear approximation. Again, no theoretical restrictions are met and the cause of the relatively low work reduction is expected to originate from the solver.

The last remark about figure 5 concerned the relatively high work reduction of the linear balancers for the degressive-progressive and Laevo objective functions, compared to the performance of the nonlinear balancers. The nonlinear balancers, on the other hand, generally reduce the work more than the linear versions for the 180 degree sine function. It should be stressed that, although the nonlinear springs have a second order moment-angle characteristic, all springs are confined to have the same characteristic. This restriction could impede the selection of spring ratios that allow the balancer to describe a higher order objective curve as the degressive-progressive and the Laevo characteristic. The relatively high work reduction of the nonlinear balancers for the 180 degree sine objective function is expected to be caused by the quality of a second order fit of the sine. The nonlinear balancers, even the balancers with prestress and optimizable initial angle of segment 1, obtain their non-linearity from the nonlinear springs only. This is in contrast with the linear balancers that realize their nonlinear behaviour by a nonlinear rotation of the first segment with respect to the pendulum.

The four observations made by inspection of figure 5 are also applicable to figure 6, which illustrates the work reduction realized by the four segment balancers. The balancing quality of the nonlinear balancer for the 180 degree sine objective

function is significantly lower than that of the three segment counterpart, however. By further inspection of the results, it was found that the work reduction would be significantly higher if the first segment would rotate proportionally with the pendulum. The current relation between both rotations, however, is a slightly progressive one. As mentioned before, all nonlinear springs are constrained to have the exact same moment-angle characteristic. A second order moment-angle characteristic corresponds to a third order potential energy curve, which is recognized by its progressive shape. This potential energy characteristic, combined with the extra internal degree of freedom of the four segment balancer, is expected to cause the lower work reduction of the four segment balancer. As a matter of fact, equilibrium should be satisfied, which is restricting the spring on the main axis to store energy by rotation. Apart from this nonlinear case, it would be expected that the four segment balancer is able to realize a higher work reduction for a certain objective curve than the three segment balancer. Theoretically, the introduction of an extra segment would only enlarge the optimization freedom. The optimization freedom is enlarged as an extra degree of freedom is enabled, which is only an addition to the possibilities of the three segment balancer. More generally, the three segment balancer can be interpreted as a subset of the four segment balancer as all configurations of the three segment balancer could be realized with the four segment balancer as well. The latter only holds when the lower- and upperbounds of the segment lengths and spring stiffnesses would be fully relaxed. As this is not the case in the current work, it could be the cause of the fact that only 75% of the four segment balancers achieve a higher work reduction than their three segment counterpart.

Figure 7 illustrates the approximations of three Pareto sets found by the genetic algorithm. Only the regular, nonlinear and prestressed nonlinear variants of the three segment balancer are included. It should be emphasized that the found points are not guaranteed to be located on the actual Pareto set, as they are merely approximations. It is seen that the plot for the regular balancer generally is the set with the lowest objective function values. The system with nonlinear springs is represented by the orange plot, which contains points with relatively high objective function values. The approximation of the Pareto front corresponding to the system with nonlinear and prestressed springs, plotted in grey, is located between the other two plots. Whereas the normalized root mean square error directly depends on the angle of rotation of the first segment and the stiffness of the corresponding spring, the energy distribution depends on both the stiffness of all springs and the rotation of all segments. As the nonlinear springs are restricted to have the same moment-angle characteristic, the energy distribution only depends on the rotation of the segments for those balancers. It is expected that this dependency is the main cause of the differences in magnitude of the objective function values between the regular and the nonlinear systems.

The measurement results are compared with the expected moment-angle characteristic in figure 8. The red curve is the balancing moment obtained by MATLAB, which accounts for

the deviations in spring stiffness ratios. The grey plot is the measured hysteresis loop after averaging. Again, a couple of observations are made by inspection of the characteristics. It is seen that the measured balancing moment obtained by the experiments is smaller than the balancing moment provided by MATLAB. This deviation originates from a relatively large deviation in two of the ten measurement runs. No clear cause of these deviations was found, but it is observed that small imperfections in the initial configuration could have a relatively large effect on the measured characteristic. A second remarkable fact is the relatively large friction band at larger angles of the pendulum. This section of the characteristic, ranging from approximately 60° to 90° , should have the smallest slope as well. This decrease in slope of the balancing moment is only achieved when the plot of the angle of rotation of the first segment against the rotation of the pendulum is a degressive plot. The internal degree of freedom of the balancer is thus utilized more at larger angles. Relatively large friction in the bearings that facilitate the internal DOF could be a cause of the larger friction band at larger angles. The last observation concerns the upper and the lower part of the hysteresis loop. In theory, the upper part would correspond with the rotation of the pendulum from 0° to 90° , whereas the lower part describes the moment-angle relation for the returning rotation from 90° to 0° . This holds for figure 8 at angles of the pendulum larger than 50° , but the orientation is the other way around at smaller angles. It is expected that the part of the hysteresis loop that corresponds with the returning rotation should be located lower than shown in the figure. The measurement results illustrate friction in the two bearings that facilitate rotation of the pushing bracket. An argument for this claim is the open end of the returning part of the hysteresis loop at approximately 8° . At this angle, the pendulum is no longer pushed and is no longer rotating while the force sensor still registers a force. Although this friction is known to exist, it is not expected to cause the intersection of the lower and upper parts of the hysteresis loop. Instead, it is expected that friction in the bearings of the balancer impairs the kinematics and causes the reverse motion to be deviating from the 0° to 90° rotation.

A recommendation for future research would be to investigate the effect of relaxed lower- and upperbounds of the segment lengths on the optimization results of the three and four segment balancers. Currently, the three segment balancers with prestress only allow for prestress and the corresponding release of contact on the spring on the second axis. The third spring could be preloaded and fixed instead, to analyse the possibilities of the balancer more exhaustively.

Moreover, it would be of great value to study the performance of extra balancers with respect to the multi-objective optimization. Due to limited resources, the Pareto sets are approximated for only three different three segment balancers. The linear balancers other than the regular version could possibly have Pareto optimal points with relatively low objective function values.

Besides this further evaluation of the multi-objective op-

timization, it is expected that significant potential exists to gather measurement results that are closer to the expected measurement curve. If one would be able to realize a better alignment of the segments and pendulums, height differences would be limited. A minimum height difference prevents the kinematics from being impaired. Although steel segments increase the mass and therefore the difference in height energy for a given angle of misalignment, the segments are likely to have less deformation when the bearing is inserted. This decrease in deformation could eventually improve the alignment and reduce height differences.

VI. CONCLUSION

To conclude, in this work the possibilities to statically balance various nonlinear moment-angle characteristics by a rigid body balancer with torsion springs are examined. It appeared to be possible to select system parameters that result in an approximation of the given objective function.

The performance of the balancers that approximate a quarter period of a sine is relatively high, as 75% of the balancers realize a work reduction higher than 99%. Although some of the objective functions are approximated with a lower work reduction, the performance for the other objective characteristics is comparable to that of the sine balancer. Softening behaviour was obtained by applying prestress with contact release, but the same degressive behaviour was also realized by the regular balancer with a relatively large initial angle of segment 1 and the balancers with nonlinear springs. Negative stiffness was achieved with the nonlinear balancers and linear balancers with optimizable angle of the first segment.

Lastly, the results that are obtained by the proposed method were verified with an experimental setup that contains a prototype of the system. In spite of the friction in the system, the measurement results provide a proof of concept as the measured characteristic is degressive and results in a work reduction of 93.47%.

REFERENCES

- [1] J. Herder, "Energy- free Systems: Theory, conception and design of statically balanced spring mechanisms," Ph.D. dissertation, TU Delft, Delft, 11 2001.
- [2] G. Radaelli, "An energy approach to static balancing of systems with torsion stiffness," in *Proceedings of the ASME Design Engineering Technical Conference*. Delft: TU Delft, 9 2009.
- [3] J. L. Herder and F. P. A. Van Den Berg, "Statically balanced compliant mechanisms (SBCM's), an example and prospects," in *ASME Design Engineering Technical Conferences*. Baltimore: ASME, 12 2000, pp. 853–859.
- [4] A. Gopalswamy, P. Gupta, and M. Vidyasagar, "A new parallelogram linkage configuration for gravity compensation using torsional springs," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 664–669, 4 1992.
- [5] M. R. Claus, "Gravity balancing using configurations of torsion bars with application to the HCI foldable container," TU Delft, Delft, Tech. Rep., 12 2008.
- [6] O. Boubaker, "The inverted Pendulum: A fundamental Benchmark in Control Theory and Robotics," in *2012 International Conference on Education and e-Learning Innovations*, National Institute of Applied Sciences and Technology. Tunis: IEEE, 11 2012.

- [7] H. Hemami, F. C. Weimer, and S. H. Koozekanani, "Some Aspects of the Inverted Pendulum Problem for Modeling of Locomotion Systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 6, pp. 658–661, 12 1973.
- [8] T. Kwon and J. K. Hodgins, "Momentum-Mapped Inverted Pendulum Models for Controlling Dynamic Human Motions," *ACM Transactions on Graphics*, vol. 36, no. 1, 1 2017.
- [9] S. Bhattacharya, R. De Risi, D. Lombardi, A. Ali, H. E. Demirci, and S. Haldar, "On the seismic analysis and design of offshore wind turbines," *Soil Dynamics and Earthquake Engineering*, vol. 145, 6 2021.
- [10] P. Varella Barca Guimarães and S. Moreira Ávila, "Control of an offshore wind turbine modeled as discrete system," in *2nd ECCOMAS Young Investigators Conference (YIC 2013)*, Brasilia, 8 2013. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00855865>
- [11] P. V. B. Guimaraes, S. M. Avila, M. A. M. Shzu, Z. J. G. Del Prado, and M. V. G. Morais, "Vibration control of an offshore wind turbine modeled as an inverted pendulum," in *11th International Conference on Vibration Problems*, Brasilia, 9 2013. [Online]. Available: <http://www.dailymail.co.uk/news/article-2013233/The-wind-turbine-backlash-Growing-public->
- [12] G. J. Walsh, D. A. Streit, and B. J. Gilmore, "Spatial spring equilibrators theory," *Mechanism and Machine Theory*, vol. 26, no. 2, pp. 155–170, 3 1991.
- [13] F. L. S. Te Riele and J. L. Herder, "Perfect static balance with normal springs," in *ASME 2001 Design Engineering Technical Conferences*. New York: ASME, 11 2001, pp. 571–578.
- [14] Y. Chheta R. and R. Joshi M., "A Review on Passive Gravity Compensation," in *International Conference on Electronics, Communication and Aerospace Technology*. Coimbatore: IEEE, 4 2017, pp. 184–189.
- [15] G. Endo, H. Yamada, A. Yajima, M. Ogata, and S. Hirose, "A passive weight compensation mechanism with a non-circular pulley and a spring," in *Proceedings - IEEE International Conference on Robotics and Automation*. Anchorage: IEEE, 5 2010, pp. 3843–3848.
- [16] B. G. Bijlsma, "Design of a compact gravity equilibrators with an unlimited range of motion," *Journal of Mechanisms and Robotics*, vol. 9, no. 6, 3 2012.
- [17] W. B. Shieh and B. S. Chou, "A novel spring balancing device on the basis of a Scotch yoke mechanism," in *2015 IFToMM World Congress Proceedings, IFToMM 2015*. New Taipei: National Taiwan University, 2015.
- [18] E. Dede and B. Trease, "Statically-balanced Compliant Four-bar Mechanism for Gravity Compensation," The University of Michigan, Michigan, Tech. Rep., 2004.
- [19] G. Radaelli, R. Buskermolen, R. Barents, and J. L. Herder, "Static balancing of an inverted pendulum with prestressed torsion bars," in *Mechanism and Machine Theory*, vol. 108. Delft: Elsevier Ltd, 2 2017, pp. 14–26.
- [20] G. Radaelli and J. L. Herder, "Isogeometric shape optimization for compliant mechanisms with prescribed load paths," in *Proceedings of the ASME Design Engineering Technical Conference*, vol. 5A. Delft: American Society of Mechanical Engineers (ASME), 8 2014.
- [21] —, "Gravity balanced compliant shell mechanisms," *International Journal of Solids and Structures*, vol. 118–119, pp. 1339–1351, 7 2017.
- [22] A. Müller, "Generic mobility of rigid body mechanisms," *Mechanism and Machine Theory*, vol. 44, no. 6, pp. 1240–1255, 6 2009.
- [23] L. Howell and A. Midha, "A Loop-Closure Theory for the Analysis and Synthesis of Compliant Mechanisms," *Journal of Mechanical Design*, vol. 118, no. 125, 1994.

4

Discussion

4.1. Discussion research paper

The research paper, discussed in chapter 3, illustrated relatively high performance of the balancers. As a matter of fact, six three segment balancers and three four segment balancers realized a work reduction higher than 99%. As mentioned in the paper, this work reduction is higher than that of the mechanisms presented in the state of the art. The work reduction of the prototype was 93.47%. The MATLAB model achieved a higher reduction in actuation effort of 99.33%. This deviation in work reduction is partly caused by the fact that the measured load values are generally lower than expected. It is seen that two out of ten measurement runs illustrated a deviating moment-angle characteristic when compared to the other measurement runs. These moment-angle curves appeared to be lower than the other plots and are expected to cause the lower average moment values at angles of the pendulum until 60°. The experiments, however, still provide a proof of concept as the measured moment-angle curve is a degressive characteristic.

4.2. Discussion application as exoskeleton

It is expected that the proposed balancer has significant potential for use in a lower back supporting exoskeleton. The balancing quality in terms of work reduction is much larger than that of the state of the art presented in section 1, both for the MATLAB model and the prototype. Although the contribution of friction was found to be relatively large in the research paper, the spring stiffnesses are easily scaled in case of an application as exoskeleton. As long as the friction in the bearings is not increased, enlarged internal spring moments will reduce the contribution of friction. Despite the fact that it will be hard to acquire torsion springs with the right stiffnesses, it is possible to design and produce own springs. Incorporation of the spring design and production in the design process could increase the development costs, but this approach will eventually facilitate full in-house development of the mechanism. Moreover, the other constituent parts of the balancer can be obtained or made relatively easily and are less costly.

An important requirement of an exoskeleton is that it should conform the human body. One of the main advantages of the presented balancer is the flexibility to be applied in different use cases. As a matter of fact, the spring stiffnesses and segment lengths, among others, can be selected such that the kinematics suit the intended application. For a given angle of the pendulum and the first segment, the balancer could attain two different postures: an elbow up configuration and elbow down configuration. Extreme cases such as the case where segment 1 is at its lower- or upperbound form an exception. In this work, the elbow up configuration is analyzed because of the potentially better fit with the human body. A restriction that is not taken into account yet is the allowed range of motion of the first segment when the balancer is used in an actual exoskeleton. As a matter of fact, interesting behaviour is seen for relatively large initial angles of the first segment. The balancer appeared to be able to show softening behaviour without prestress and contact release with these large initial angles. If the presented design is to be used as an actual exoskeleton, one should ensure that the first segment always has a smaller angle with respect to the vertical than the pendulum. This would degrade the possibilities for the latter softening behaviour. As nonlinear springs are not easily obtained, the exoskeleton would be dependent on the principle of prestress and release of contact. This principle is expected to be relatively easy to apply. The contacts only restrict the segments to rotate with respect to each other, which could be achieved by means of a simple bracket.

In addition to the conformability to the human body, the balancer should allow for the other degrees of freedom of the human body as much as possible. Apart from cases wherein these DOFs are to be constrained, like in the situation of a patient with insufficient muscular capacity to hold his or her body in a certain configuration, translations and rotations other than forward bending should be unaffected in their kinematics. Because of the stiffness of the balancer in the rotation directions other than forward bending, a connection of the balancer with the human body that allows for these rotations is needed.

The springs on the first axis of the prototype, on the other hand, are connected with the environment by means of a bolt, nut and connection plateau. If the balancer is to be used as an exoskeleton, this plateau is no longer available. Instead, the outer ring of the clock springs should be connected with a rigid part of the human body that does not experience any displacements during forward bending, like the hip.

4.3. Recommendations for future research

As a recommendation for future research, it would be of great relevance to develop an optimization procedure that accounts for variations of the length of the pendulum. This would contribute significantly to the practical application of the balancer, as the human back is known to extend during flexion. The performance of the balancer can be maximized by measuring the effective length of the back of the wearer during flexion and its reverse motion at the beginning of the design phase of the balancer. Accordingly, a model can be made with a known radius-angle characteristic $r(\alpha)$. Alternatively, the connections of the balancer with the back can be altered. As a result, the mechanism is allowed to extend and contract by a small amount. This extra degree of freedom would prevent the user of the system to be forced to describe a perfect quarter of a sine. It should be noted, however, that the first approach is recommended as the latter method will degrade the balancing performance.

If the exact moment-angle objective function is known, one could focus on this particular function during the optimization phase. The current work examined various versions of the three and four segment balancers for different moment-angle objective functions. As computational power and time are to be invested in the optimization of one scenario only, more research can be done on the type of algorithm and the algorithm settings that provide the lowest objective function values for that scenario. Until now, all optimization runs are done with the genetic algorithm from MATLAB on standard settings. Other population and crossover settings could improve the optimization performance as the optimization time could be reduced or lower optima could be found. One could, for example, compare different settings of the genetic algorithm in terms of their influence on the optimization time and eventual objective function value. It should be mentioned, however, that the latter investigation could be very costly as multiple runs are needed in order to make general conclusions or predictions. As a matter of fact, the random nature of the algorithm causes one optimization run to be not representative for the performance of the algorithm as a whole.

Dependent on the field of application, it might be worth the effort to design a housing for the axes with bearings and clock springs. Especially in agricultural or hospital applications, it is required that there is no accumulation of dust or other substances at these locations. Alternatively, one could decide to convert the rigid body mechanism into a compliant version. This could, however, come at the cost of concentrated deflections and high local stresses in the balancer.

Lastly, it is recommended to do an attempt to balance the human back in lateral bending too. Other exoskeletons, like the PLAD soft exoskeleton [17], support the human body in this degree of freedom as well. Although the PLAD illustrates relatively low balancing quality in this direction, users with reduced muscle activity might find this multi-directional support advantageous.

5

Conclusion

This work analyzed a rigid body balancer with torsion springs that could be incorporated in an exoskeleton that supports the human back by balancing the moment that is induced around the hip by forward bending. More specifically, the research paper examined the possibilities to statically balance various nonlinear moment-angle characteristics. The ability to approximate these nonlinear characteristics originates from the internal degree of freedom of the total system, consisting of the balancer and the inverted pendulum as a model of the human back. This internal DOE, on the other hand, also posed extra design difficulties as the optimization routine required the equilibrium configurations.

The research paper demonstrated that the proposed rigid body balancer could be used to approximate a given moment-angle characteristic. Most of the balancers with linear, un-prestressed springs realized higher work reduction than the state of the art on sine balancers presented in the paper. Moreover, balancers were found that are able to approximate progressive, progressive-degressive and degressive-progressive objective functions. In addition, system parameters were obtained that resulted in approximations of negative stiffness objective functions. The performances of the latter are comparable to that of the sine balancers. The implementation of nonlinear springs and linear, prestressed springs with release of contact is shown to be useful for some of the presented objective characteristics. The experiments with the prototype provided a proof of concept as the measured characteristic illustrated both degressive behaviour and correspondence to the objective function.

Generally, it can be concluded that the presented balancers are able to approximate the quarter of a sine objective function relatively well. As a matter of fact, 75% of the three segment balancers and 75% of the four segment balancers realized a work reduction higher than 99%. This is higher than reported in the studied state of the art. The work reduction of the prototype was 93.47%. Although the availability of clock springs with the correct stiffnesses was and will be limited, the proposed system could be conform to the human body as the kinematics can be controlled early in the design phase.

Appendices

In the following sections, extra information is included in the form of appendices. It concerns information that is not provided in the research paper or in the regular chapters of this thesis.

In order to find the correct equilibrium values of the system with the MATLAB script, one should do an energy analysis of the possible postures of the balancer for each angle of the pendulum. As the angle of the first segment is swept through its range of motion, a lowerbound and upperbound of this sweep are to be established. Generally, these bounds are interpreted as situations in which the second and third segment are perfectly aligned with respect to each other. Equation A.1 and equation A.3 are used to evaluate the corresponding internal angles of the four bar. These equations are provided, together with formulations for other angles that are expected to be of use, in appendix A.

To check the behaviour of the balancers with prestress and release of contact, one would need Free Body Diagrams of the balancers. The FBD of the three segment balancer is included in appendix B, whereas appendix C provides a less elaborated FBD of the four segment balancer. The expressions for the reaction moments can be used to check the written MATLAB scripts and its calculations. As a matter of fact, the external moments at the nodes of the springs should correspond with the internal moments that are created by the springs.

An extra appendix, appendix D, is related to optimization results and figures of a four segment balancer with prestress and release of contact. The above described correspondence of the internal and external moments and more figures are included, as they illustrate the working principle of this balancer well.

The moment-angle characteristic of a given, conservative system can be derived by differentiating the potential energy formulation with respect to its degree of freedom. This fact is used to spot potential errors in the MATLAB code early in the design phase. As the moment-angle and potential energy characteristics did not agree for a couple of iterations at the start of the research, it was decided to elaborate on the potential energy and to do an attempt to enforce the Lagrange equations. Unfortunately, the latter did not succeed because of the relatively involved potential energy formulation. Nevertheless, the formulation of the potential energy and the requirements to fulfill in order to apply Lagrange are given in appendix E.

It is expected that it is useful to verify the results in another way than with the moment-angle and potential energy curves as well. The latter is done with help of Artas SAM software [21]. The software is developed to analyze the loads and kinematics of mechanisms. Although the to be evaluated mechanism should be exactly constrained, which is not the case for the balancer that is described in this work, the program is still useful to check the free body diagrams that are included in appendix B and C. An example of a check with a four segment balancer is included in appendix F.

Apart from the more theoretical parts described above, it might be of interest to obtain more information about the design of the prototype. Appendix G will depict some renders of the Solidworks [22] model. In order to construct an exact same version of this, all constituent parts are listed as well. References to the manufacturer sites are included, if applicable. Some parts, especially PLA parts, are made at the university. In those cases, the Solidworks drawing is included to make the overview complete. Appendix H discusses points of attention for the assembly phase that will minimize friction and optimize the alignment of the balancer. Despite the recommendation to use off-the-shelf clock springs, as described in appendix G, the reader is encouraged to implement springs of own design if that would result in actual stiffnesses that are closer to the stiffnesses that are provided by the optimization procedure. Appendix I includes a qualitative evaluation of watercutted and lassercutted springs of various thicknesses. The springs that are used in the prototype, the off-the-shelf clock springs from Lesjöfors, are quantitatively evaluated in appendix J. Lastly, when the prototype and the experimental setup are fully constructed, LabVIEW [23] code is needed to read the output of the sensors. The LabVIEW block-scheme is presented in appendix K.

The last three appendices are more abstract as these include the information that is needed to run the same MATLAB scripts in the same way as done in this research. The MATLAB code is given in appendix N, whereas appendix L informs the reader about the use of a Linux cluster to increase the available computational power. As an intermezzo, appendix M tabulates the optimization results that were omitted from the paper for reasons of readability and overview.

A

Geometric analysis

In the following, the geometry of a four bar consisting of an inverted pendulum and a 3 segment balancer will be analyzed. A sketch of the system is shown in figure A.1. The black line indicates the inverted pendulum, whereas the blue lines represent the rigid segments of the balancer. The green line with length "q" serves as an imaginary connection, such that the four bar is divided into two triangles. Typically, the lengths of the blue segments and the black segment are known. The length of the imaginary, green segment is subject to change as the pendulum rotates.

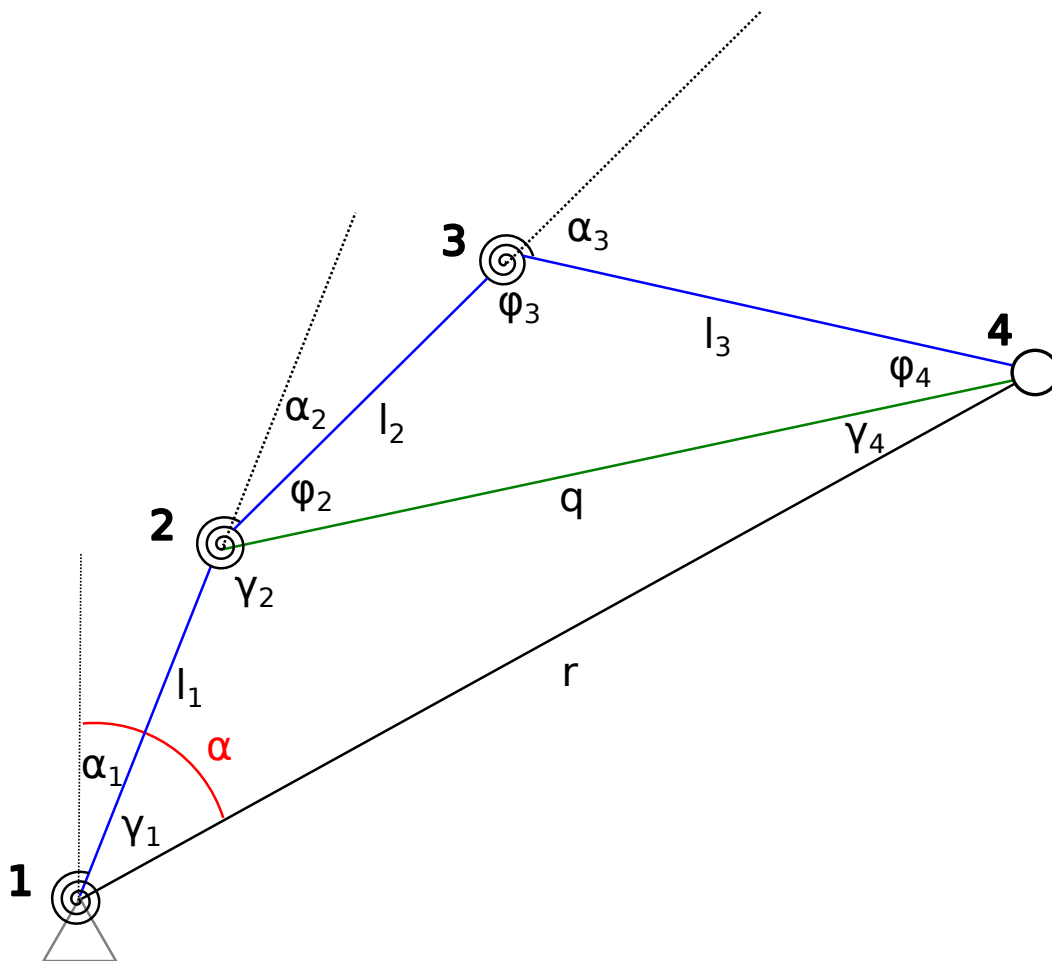


Figure A.1: Definition of angles

$$\gamma_2 = \arccos\left(\frac{q^2 + l_1^2 - r^2}{2ql_1}\right) \quad (\text{A.1})$$

$$\gamma_1 = \arccos\left(\frac{r^2 + l_1^2 - q^2}{2rl_1}\right) \quad (\text{A.2})$$

$$\gamma_4 = \arccos\left(\frac{r^2 + q^2 - l_1^2}{2rq}\right) \quad (\text{A.3})$$

$$\phi_3 = \arccos\left(\frac{l_2^2 + l_3^2 - q^2}{2l_2l_3}\right) \quad (\text{A.4})$$

$$\phi_2 = \arccos\left(\frac{q^2 + l_2^2 - l_3^2}{2ql_2}\right) \quad (\text{A.5})$$

$$\phi_4 = \arccos\left(\frac{q^2 + l_3^2 - l_2^2}{2ql_3}\right) \quad (\text{A.6})$$

$$q = \sqrt{l_2^2 + l_3^2 - 2l_2l_3 \cos(\phi_3)} = \sqrt{r^2 + l_1^2 - 2rl_1 \cos(\gamma_1)} \quad (\text{A.7})$$

B

FBD three segment balancer

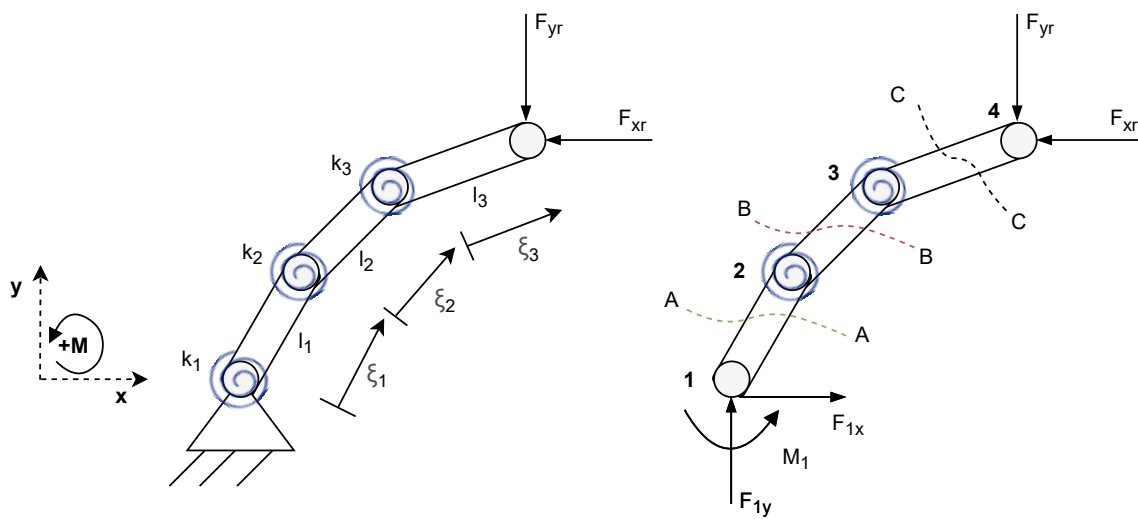


Figure B.1: Schematic of compensation mechanism (left) and setup for Free Body Diagrams (right)

B.1. Segment 1

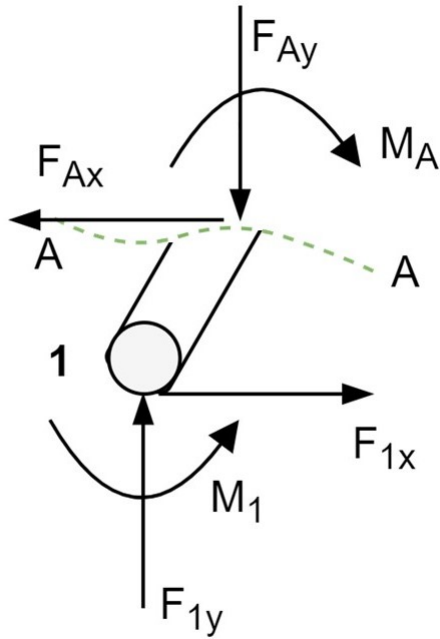


Figure B.2: FBD first segment

Equilibrium in x-direction:

$$\sum F_x = F_{1x} - F_{Ax} = 0$$

$$F_{Ax} = F_{1x}$$

Equilibrium in y-direction:

$$\sum F_y = F_{1y} - F_{Ay} = 0$$

$$F_{Ay} = F_{1y}$$

Moment equilibrium:

In the following, θ_1 , θ_2 and θ_3 will represent the angles of the first, second and third segment with respect to the vertical, respectively.

$$\sum M_{AA} = M_1 - M_A - F_{1y}\xi_1 \sin(\theta_1) + F_{1x}\xi_1 \cos(\theta_1) = 0$$

$$M_A = M_1 - F_{1y}\xi_1 \sin(\theta_1) + F_{1x}\xi_1 \cos(\theta_1)$$

$$M_A|_{\xi_1=0} = M_1$$

B.2. Segment 2

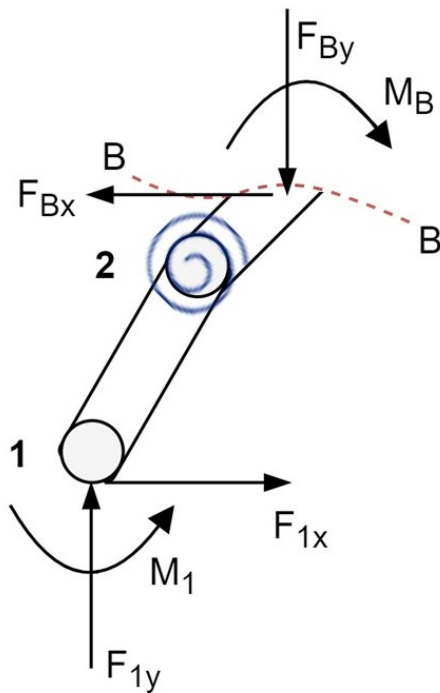


Figure B.3: FBD second segment

Equilibrium in x-direction:

$$\sum F_x = F_{1x} - F_{Bx} = 0$$

$$F_{Bx} = F_{1x}$$

Equilibrium in y-direction:

$$\sum F_y = F_{1y} - F_{By} = 0$$

$$F_{By} = F_{1y}$$

Moment equilibrium:

$$\sum M_{BB} = M_1 - M_B - F_{1y}(l_1 \sin(\theta_1) + \xi_2 \sin(\theta_2)) + F_{1x}(l_1 \cos(\theta_1) + \xi_2 \cos(\theta_2)) = 0$$

$$M_B = M_1 - F_{1y}(l_1 \sin(\theta_1) + \xi_2 \sin(\theta_2)) + F_{1x}(l_1 \cos(\theta_1) + \xi_2 \cos(\theta_2))$$

$$M_B|_{\xi_2=0} = M_1 - F_{1y}l_1 \sin(\theta_1) + F_{1x}l_1 \cos(\theta_1)$$

B.3. Segment 3

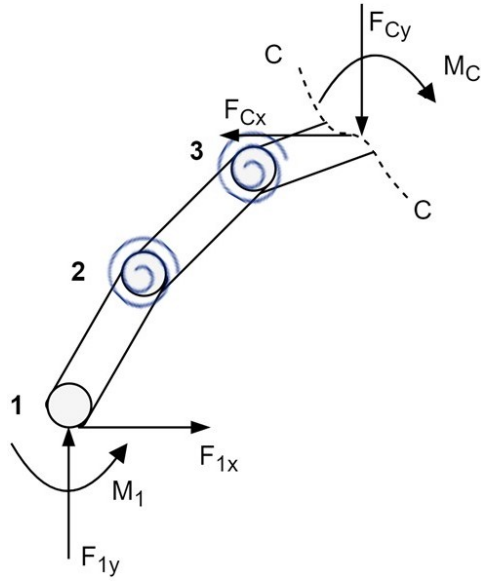


Figure B.4: FBD third segment

Equilibrium in x-direction:

$$\sum F_x = F_{1x} - F_{Cx} = 0$$

$$F_{Cx} = F_{1x}$$

Equilibrium in y-direction:

$$\sum F_y = F_{1y} - F_{Cy} = 0$$

$$F_{Cy} = F_{1y}$$

Moment equilibrium:

$$\sum M_{CC} = M_1 - M_C - F_{1y}(l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + \xi_3 \sin(\theta_3)) + F_{1x}(l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + \xi_3 \cos(\theta_3)) = 0$$

$$M_C = M_1 - F_{1y}(l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + \xi_3 \sin(\theta_3)) + F_{1x}(l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + \xi_3 \cos(\theta_3))$$

$$M_C|_{\xi_3=0} = M_1 - F_{1y}(l_1 \sin(\theta_1) + l_2 \sin(\theta_2)) + F_{1x}(l_1 \cos(\theta_1) + l_2 \cos(\theta_2))$$

B.4. Reaction forces

Some scenarios require analytical expressions for the magnitude of the reaction forces expressed in, among others, the internal spring moments. These expressions can be derived as shown below.

$$M_2 = M_B|_{\xi_2=0} = M_1 - F_{1y}l_1 \sin(\theta_1) + F_{1x}l_1 \cos(\theta_1) \quad (\text{B.1})$$

$$M_3 = M_C|_{\xi_3=0} = M_1 - F_{1y}(l_1 \sin(\theta_1) + l_2 \sin(\theta_2)) + F_{1x}(l_1 \cos(\theta_1) + l_2 \cos(\theta_2)) \quad (\text{B.2})$$

Equation B.1 is accordingly written into an expression for the reaction force in horizontal direction, as provided in equation B.3.

$$F_{1x} = \frac{M_2 - M_1 + F_{1y} l_1 \sin(\theta_1)}{l_1 \cos(\theta_1)} \quad (\text{B.3})$$

Equation B.2 is rewritten into an equation for the vertical reaction force, F_{1y} .

$$F_{1y} = \frac{M_1 - M_3 + F_{1x} (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)}$$

$$F_{1y} = \frac{M_1 - M_3 + \frac{M_2 - M_1 + F_{1y} l_1 \sin(\theta_1)}{l_1 \cos(\theta_1)} (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)}$$

$$F_{1y} = \frac{M_1 - M_3}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)} + \frac{(M_2 - M_1) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \cos(\theta_1) (l_1 \sin(\theta_1) + l_2 \sin(\theta_2))} + F_{1y} \tan(\theta_1) \frac{l_1 \cos(\theta_1) + l_2 \cos(\theta_2)}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)}$$

$$F_{1y} \left(1 - \tan(\theta_1) \frac{l_1 \cos(\theta_1) + l_2 \cos(\theta_2)}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)} \right) = \frac{M_1 - M_3}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)} + \frac{(M_2 - M_1) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \cos(\theta_1) (l_1 \sin(\theta_1) + l_2 \sin(\theta_2))}$$

$$F_{1y} \left(\frac{l_2 \sin(\theta_2) - l_2 \tan(\theta_1) \cos(\theta_2)}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)} \right) = \frac{M_1 - M_3}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2)} + \frac{(M_2 - M_1) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \cos(\theta_1) (l_1 \sin(\theta_1) + l_2 \sin(\theta_2))}$$

All parts of the equation contain the same term $l_1 \sin(\theta_1) + l_2 \sin(\theta_2)$, which can be eliminated.

$$F_{1y} (l_2 \sin(\theta_2) - l_2 \tan(\theta_1) \cos(\theta_2)) = (M_1 - M_3) + \frac{(M_2 - M_1) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \cos(\theta_1)}$$

Equation B.4 finally provides an expression for the vertical reaction force.

$$F_{1y} = \frac{M_1 - M_3}{l_2 \sin(\theta_2) - l_2 \tan(\theta_1) \cos(\theta_2)} + \frac{(M_2 - M_1) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2))}{l_1 \cos(\theta_1) (l_2 \sin(\theta_2) - l_2 \tan(\theta_1) \cos(\theta_2))} \quad (\text{B.4})$$

C

FBD four segment balancer

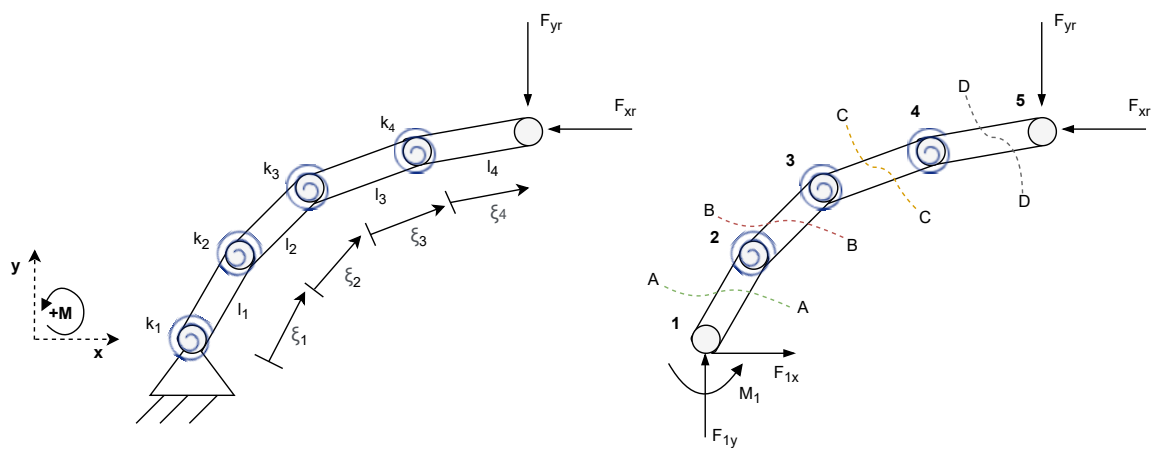


Figure C.1: Schematic of compensation mechanism (left) and setup for Free Body Diagrams (right)

C.1. Segment 4

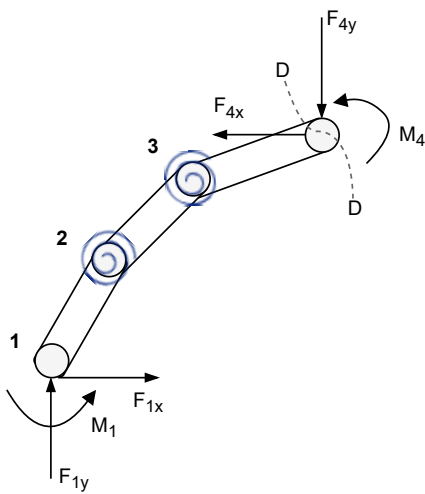


Figure C.2: FBD segment 4

Equilibrium in x-direction:

$$\sum F_x = F_{1x} - F_{4x} = 0$$

$$F_{4x} = F_{1x}$$

Equilibrium in y-direction:

$$\sum F_y = F_{1y} - F_{4y} = 0$$

$$F_{4y} = F_{1y}$$

Moment equilibrium:

$$\sum M_{DD} = 0 = M_1 + M_4 + F_{1x}(l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)) - F_{1y}(l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_3 \sin(\theta_3)) \quad (C.1)$$

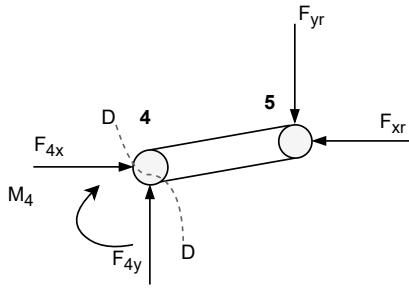


Figure C.3: FBD segment 4

Equilibrium in x-direction:

$$\sum F_x = F_{4x} - F_{xr} = 0$$

$$F_{4x} = F_{xr}$$

Equilibrium in y-direction:

$$\sum F_y = F_{4y} - F_{yr} = 0$$

$$F_{4y} = F_{yr}$$

Moment equilibrium:

$$\sum M_5 = 0 = -M_4 + F_{4x}l_4 \cos(\theta_4) - F_{4y}l_4 \sin(\theta_4)$$

C.2. Reaction forces

Rewriting the equation of the sum of the moments in node 5:

$$F_{4x} = \frac{M_4 + F_{4y}l_4 \sin(\theta_4)}{l_4 \cos(\theta_4)}$$

Using $F_{1x} = F_{4x}$ and $F_{1y} = F_{4y}$:

$$F_{1x} = \frac{M_4 + F_{1y}l_4 \sin(\theta_4)}{l_4 \cos(\theta_4)} \quad (C.2)$$

Substituting equation C.2 for F_{1x} in equation C.1:

$$M_1 + M_4 + \frac{M_4 + F_{1y} l_4 \sin(\theta_4)}{l_4 \cos(\theta_4)} (l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)) - F_{1y} (l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_3 \sin(\theta_3)) = 0$$

After expanding this equation:

$$M_1 + M_4 + \frac{M_4}{l_4 \cos(\theta_4)} (l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)) + F_{1y} (\tan(\theta_4) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)) - (l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_3 \sin(\theta_3))) = 0$$

After rewriting this equation:

$$F_{1y} = \frac{M_1 + M_4 + \frac{M_4}{l_4 \cos(\theta_4)} (l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3))}{l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_3 \sin(\theta_3) - \tan(\theta_4) (l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3))}$$

$$F_{1x} = \frac{M_4 + F_{1y} l_4 \sin(\theta_4)}{l_4 \cos(\theta_4)}$$

D

Release of contact

The figures corresponding to the MATLAB optimization results of a four segment balancer with release of contact of its springs are included in this section. Although it appeared to be one of the more costly balancers to optimize, both in terms of optimization time and programming effort, the figures are insightful and therefore discussed here. The working principle of release of contact to obtain softening behaviour is elaborated in the paper, included in chapter 3, and therefore not discussed here. The system parameters that are obtained from the optimization routine are shown in table D.1. The length of the pendulum is chosen to be $r = 1\text{m}$.

Parameter	Value	Unit
k_1	0.95	Nm/rad
k_2	0.21	Nm/rad
k_3	0.14	Nm/rad
k_4	0.06	Nm/rad
M_{3_0}	0.25	Nm
M_{2_0}	0.61	Nm
l_1	0.33	m
l_2	0.29	m
l_3	0.36	m
l_4	0.36	m

Table D.1: Optimization minimizers

Figure D.1 depicts the four plots that are made with the obtained optimization results of the balancer. The kinematics of the balancer are visualized in figure D.1a. The inverted pendulum itself is omitted from this figure. The red posture is the relaxed initial configuration, whereas the blue circles indicate the locations of the mass for 30 different angles of the pendulum. The black lines correspondingly represent the configurations of the balancer for those angles of the pendulum. The softening behaviour is recognized by inspection of the rotation of the first segment. Initially, the distance between two lines is relatively large, whereas it decreases from a certain angle of rotation onward. This distance is even smaller for the last few configurations of the first segment. A decreasing distance between two succeeding lines corresponds with softening behaviour, as the relation of the rotation of the first segment and that of the pendulum is a degressive one. The objective characteristic, the achieved curve and the residual moment are plotted in figure D.1b. The achieved balancing moment of the balancer, plotted in blue, is a degressive and non-smooth characteristic. The latter is caused by the instantaneous activation of springs. The potential energy of the total system is plotted against the angle of rotation of the pendulum in figure D.1c. It is seen that the potential energy is approximately constant. The last figure, figure D.1d, presents the internal and external moments of the second and third spring. The blue and red curves correspond to the internal spring moments, whereas the yellow and purple characteristics indicate the external loads on these points. It is observed that M_{3m} , the internal load of the third spring, is initially constant and intersects M_{3l} , which is the external load on the third

spring. The constant value attained by M3m is equal to the prestress on that spring, whereas the discussed intersection of both curves indicates that the external moment is equal to the preload. For larger angles of the pendulum, both characteristics coincidence as force and moment equilibrium should be satisfied. Moreover, the preloaded spring is enabled and will thus decrease the resultant stiffness of the balancer. The latter is observed by the decreased slope of the moment-angle curve. The same phenomenon is seen for the second spring, represented by the yellow and blue curves of the figure.

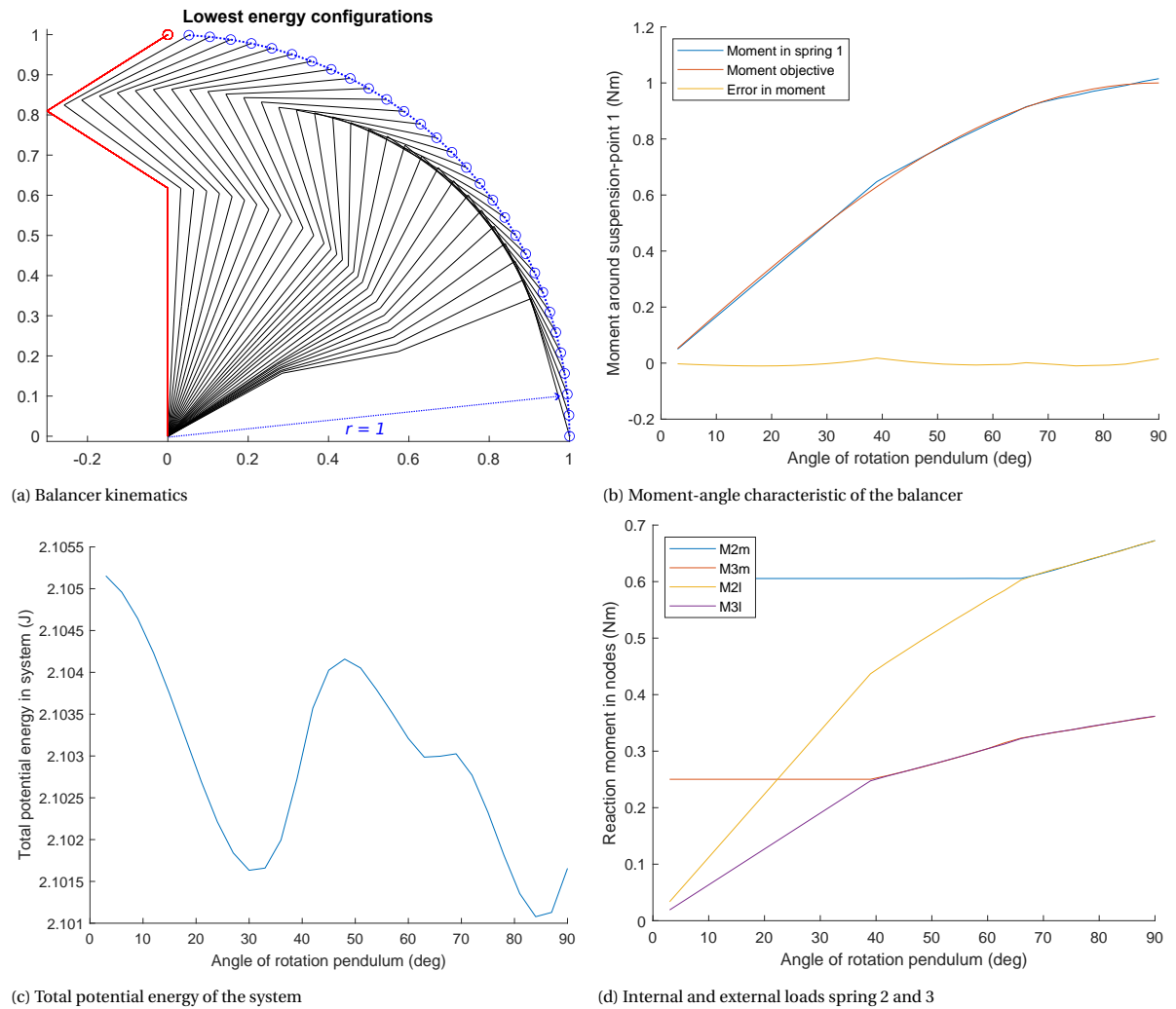


Figure D.1: Kinematics, moment-angle characteristic, potential energy curve and loads on spring 2 and 3 for the four segment balancer with release of contact of the springs

E

Lagrange

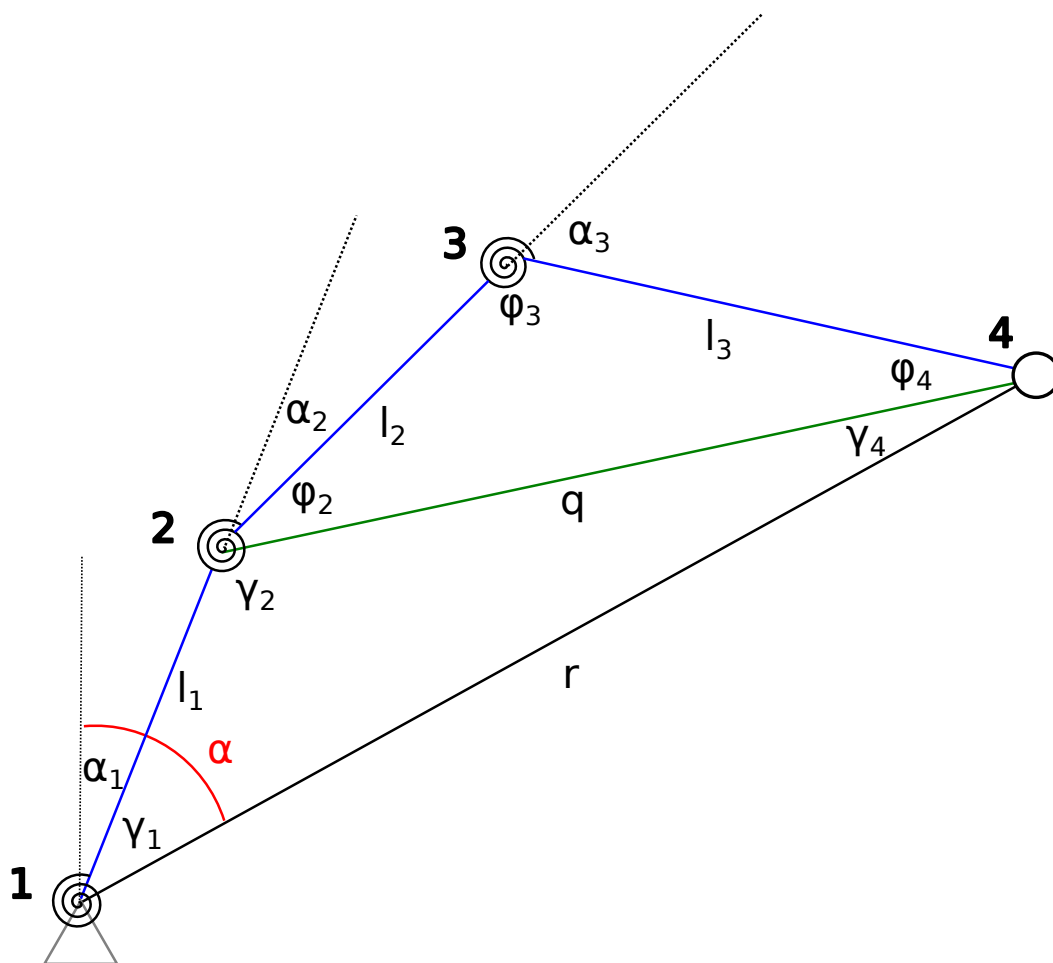


Figure E.1: Definition of angles

Requirements Lagrange: generalized coordinates should be [24]:

- Independent
- Holonomic: as many generalized coordinates as DOF
- Complete: location of bodies always fully defined

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_j} - \frac{\partial T}{\partial q_j} + \frac{\partial V}{\partial q_j} = Q_j \quad (\text{E.1})$$

As the analysis is quasi-static and no non-conservative forces are involved $T = Q_j = 0$ and equation E.1 reduces to equation E.2.

$$\frac{\partial V}{\partial q_j} = 0 \quad (\text{E.2})$$

By selecting $q_1 = \alpha$ and $q_2 = \alpha_1$ equation E.3 should hold.

$$\frac{\partial V}{\partial q_j} = 0 \begin{cases} \frac{\partial V}{\partial \alpha} = 0 \\ \frac{\partial V}{\partial \alpha_1} = 0 \end{cases} \quad (\text{E.3})$$

The total potential energy consists of the energy stored in the torsion springs and the height energy of the mass.

$$V = mgr \cos(\alpha) + \frac{1}{2} k_1 \alpha_1^2 + \frac{1}{2} k_2 \alpha_2^2 + \frac{1}{2} k_3 \alpha_3^2 \quad (\text{E.4})$$

$$\alpha_1 = \theta_1 - \theta_{1_0} \quad (\text{E.5})$$

$$\alpha_2 = \pi - \gamma_2 - \phi_2 - (\theta_{2_0} - \theta_{1_0}) \quad (\text{E.6})$$

$$\alpha_3 = \pi - \phi_3 - (\theta_{3_0} - \theta_{2_0}) \quad (\text{E.7})$$

$$q = \sqrt{r^2 + l_1^2 - 2rl_1 \cos(\alpha - \alpha_1 + \theta_{1_0})} \quad (\text{E.8})$$

$$\gamma_2 = \arccos\left(\frac{q^2 + l_1^2 - r^2}{2ql_1}\right) \quad (\text{E.9})$$

$$\phi_2 = \arccos\left(\frac{q^2 + l_2^2 - l_3^2}{2ql_2}\right) \quad (\text{E.10})$$

$$\phi_3 = \arccos\left(\frac{l_2^2 + l_3^2 - q^2}{2l_2l_3}\right) \quad (\text{E.11})$$

Substituting equation E.9, equation E.10 and equation E.11 into equations E.6 and E.7 yields equations E.12 and E.13.

$$\alpha_2 = \pi - \arccos\left(\frac{q^2 + l_1^2 - r^2}{2ql_1}\right) - \arccos\left(\frac{q^2 + l_2^2 - l_3^2}{2ql_2}\right) - (\theta_{2_0} - \theta_{1_0}) \quad (\text{E.12})$$

$$\alpha_3 = \pi - \arccos\left(\frac{l_2^2 + l_3^2 - q^2}{2l_2l_3}\right) - (\theta_{3_0} - \theta_{2_0}) \quad (\text{E.13})$$

$$\begin{aligned} V = mgr \cos(\alpha) + \frac{1}{2} k_1 \alpha_1^2 \\ + \frac{1}{2} k_2 \left(\pi - \arccos\left(\frac{q^2 + l_1^2 - r^2}{2ql_1}\right) - \arccos\left(\frac{q^2 + l_2^2 - l_3^2}{2ql_2}\right) - (\theta_{2_0} - \theta_{1_0}) \right)^2 \\ + \frac{1}{2} k_3 \left(\pi - \arccos\left(\frac{l_2^2 + l_3^2 - q^2}{2l_2l_3}\right) - (\theta_{3_0} - \theta_{2_0}) \right)^2 \end{aligned}$$

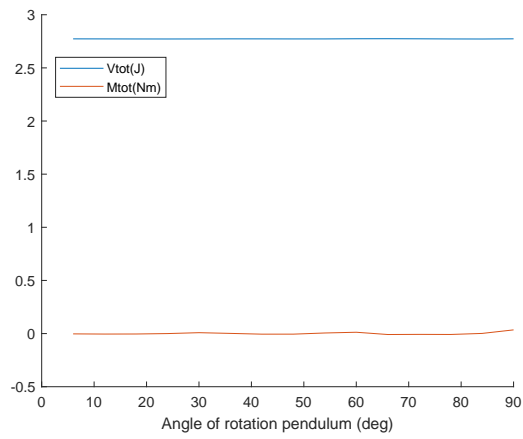
F

SAM

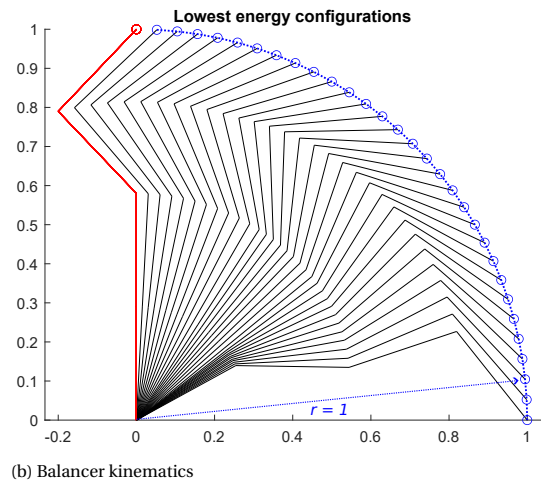
Early in the MATLAB modeling phase, the results obtained by the written MATLAB scripts were verified with use of Artas SAM software [21]. The results correspond to a four segment balancer with segment lengths equal to 29% of the length of the pendulum. The fourth spring is omitted and therefore has a stiffness value of 0.00 Nm/rad. The second and third spring, on the other hand, are prestressed. The system properties are summarized in table F.1. Figure F.1 illustrates both the moment-angle and the potential energy characteristics in subfigure F.1a and the kinematics of the balancer in subfigure F.1b. As the to be analyzed mechanism should be exactly constrained in SAM, three data arrays are inserted. These arrays originate from MATLAB and contain the discrete angles of the pendulum, the angles of the first segment and the found angles of the second segment. The SAM software is thus only used to check whether the calculations regarding the loads and potential energy are executed correctly. A potential mistake in the analysis of the equilibrium angles would not be detected via this method. The moment-angle and potential energy characteristics found by SAM are shown in figure F.2.

Parameter	Magnitude	Unit
k_1	0.97	Nm/rad
k_2	0.11	Nm/rad
k_3	0.31	Nm/rad
k_4	0.00	Nm/rad
M_{01}	0.00	Nm
M_{02}	0.60	Nm
M_{03}	0.23	Nm
l_1	0.29	m
l_2	0.29	m
l_3	0.29	m
l_4	0.29	m
r	1.00	m

Table F.1: System parameters



(a) Potential energy and moment-angle plot



(b) Balancer kinematics

Figure E1: MATLAB analysis four segment balancer with prestress on springs 2 and 3

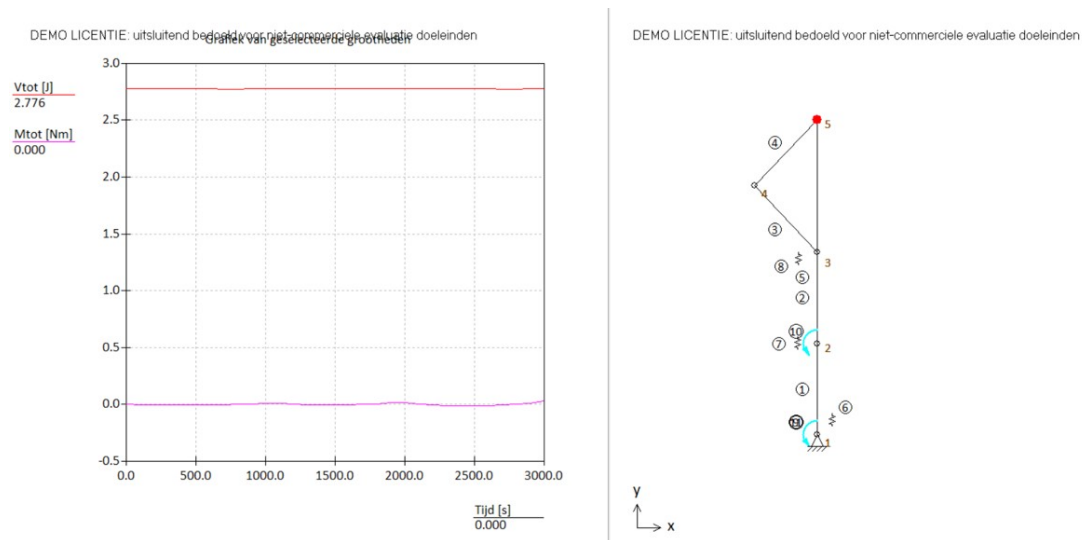


Figure E2: SAM analysis four segment balancer with prestress on springs 2 and 3

G

Solidworks

G.1. Solidworks model assembly

Renders of the Solidworks model of the prototype are provided in figure G.1, G.2 and G.3 for an overview, side view and top view, respectively.

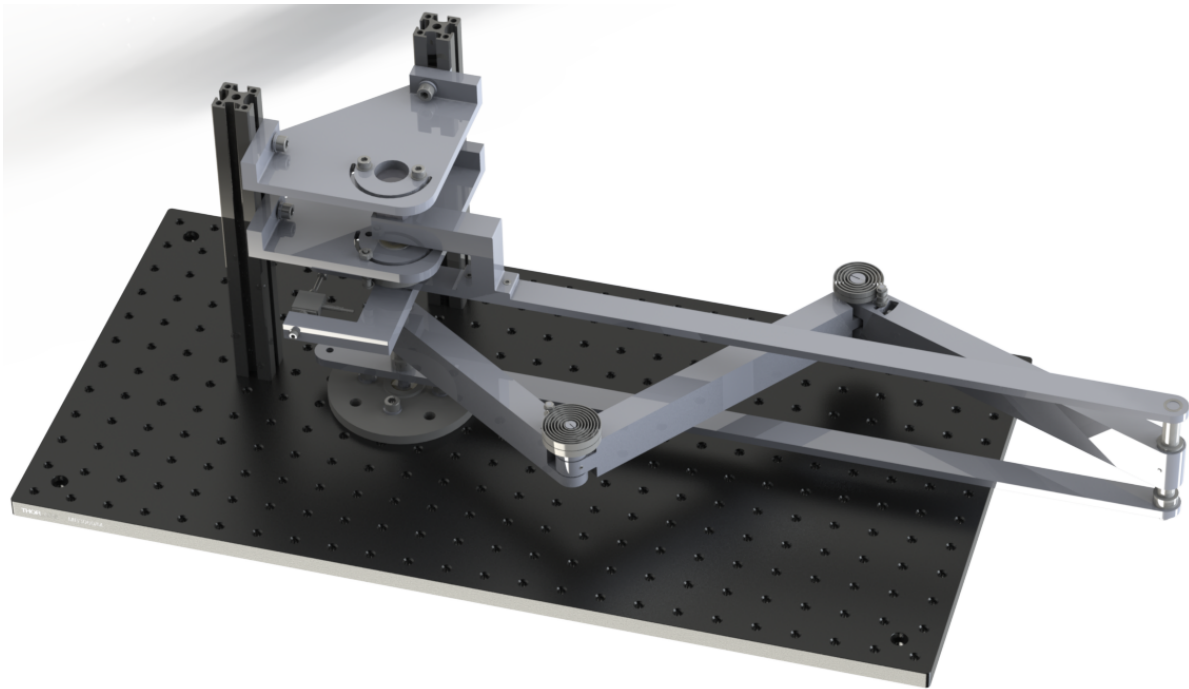


Figure G.1: Overview SW model

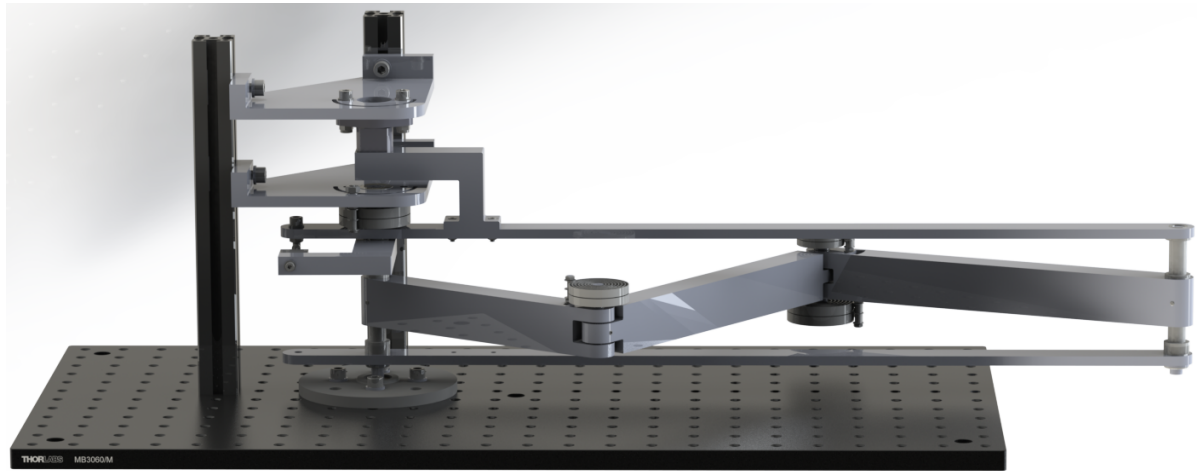


Figure G.2: Side view SW model

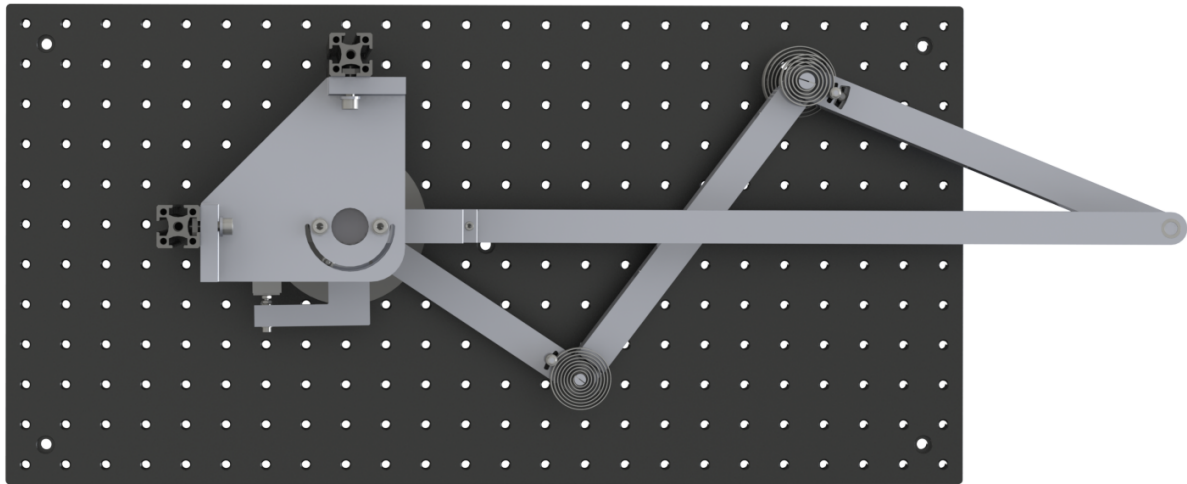


Figure G.3: Top view SW model

G.2. Solidworks model parts

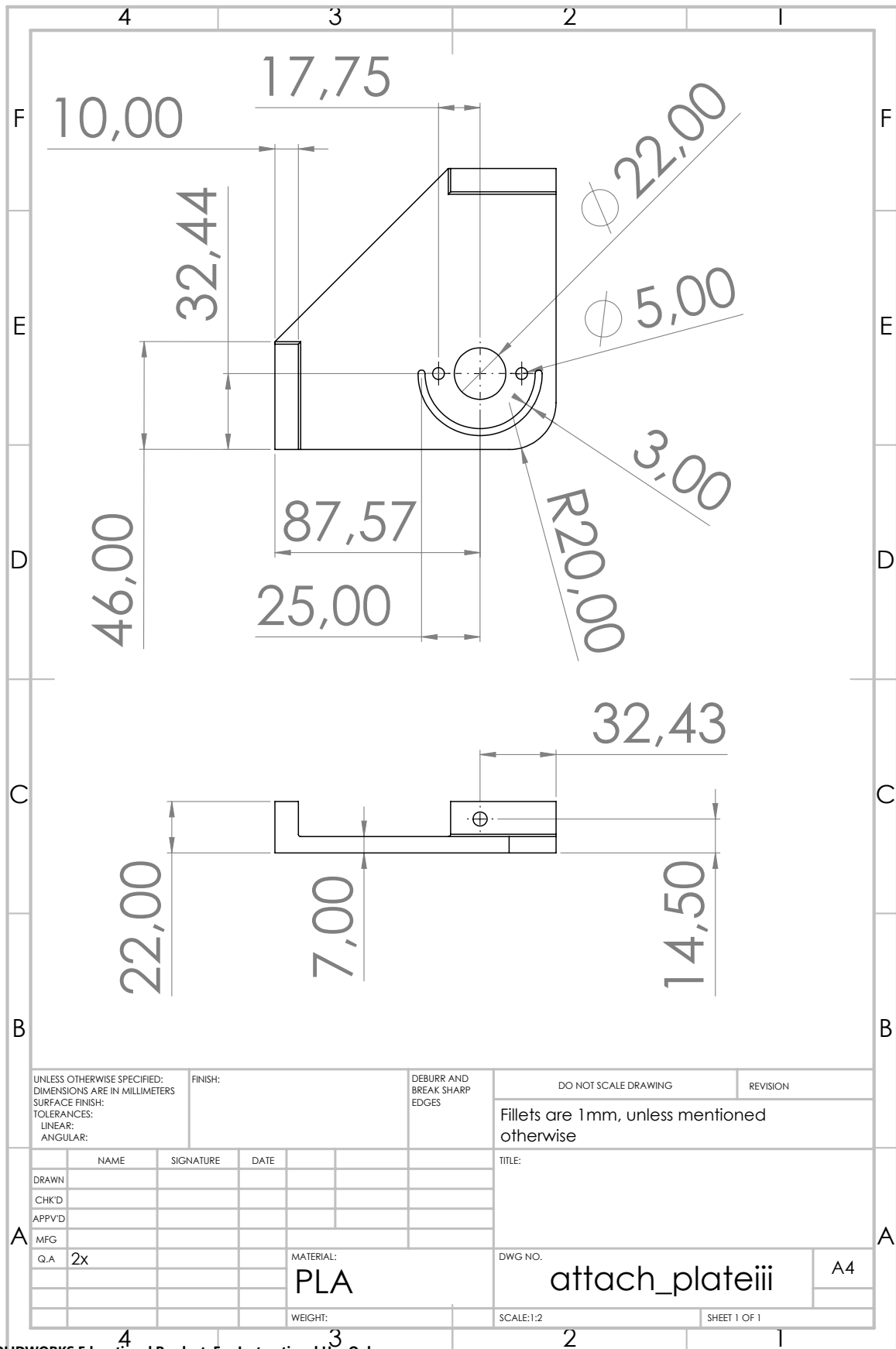
Off-the-shelf parts:

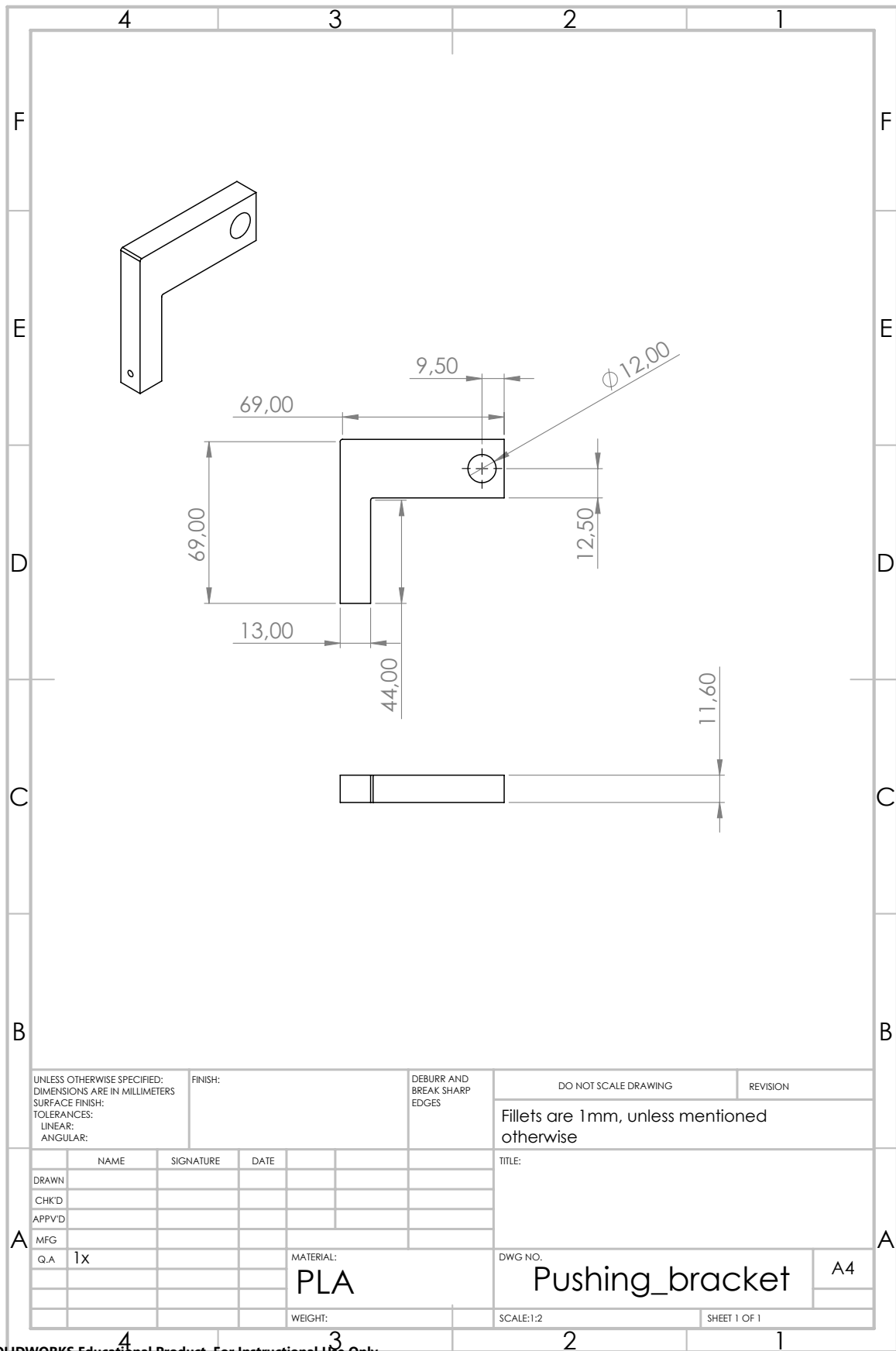
- 1x Thorlabs MB3060/M [25]
- 2x Thorlabs XE25L225/M construction rail [26]
- 1x Cherry AN8 angle position sensor [27]
- 1x FUTEK LSB200 FSH00102 load cell [28]
- 4x 907 Lesjöfors clock spring [29]
- 2x 903 Lesjöfors clock spring [30]
- 2x 908 Lesjöfors clock spring [31]
- 2x RS PRO 8mm-22mm miniature ball bearing [32]
- 8x NMB 8mm-12mm radial ball bearing [33]
- 2x NMB 6mm-10mm radial ball bearing [34]

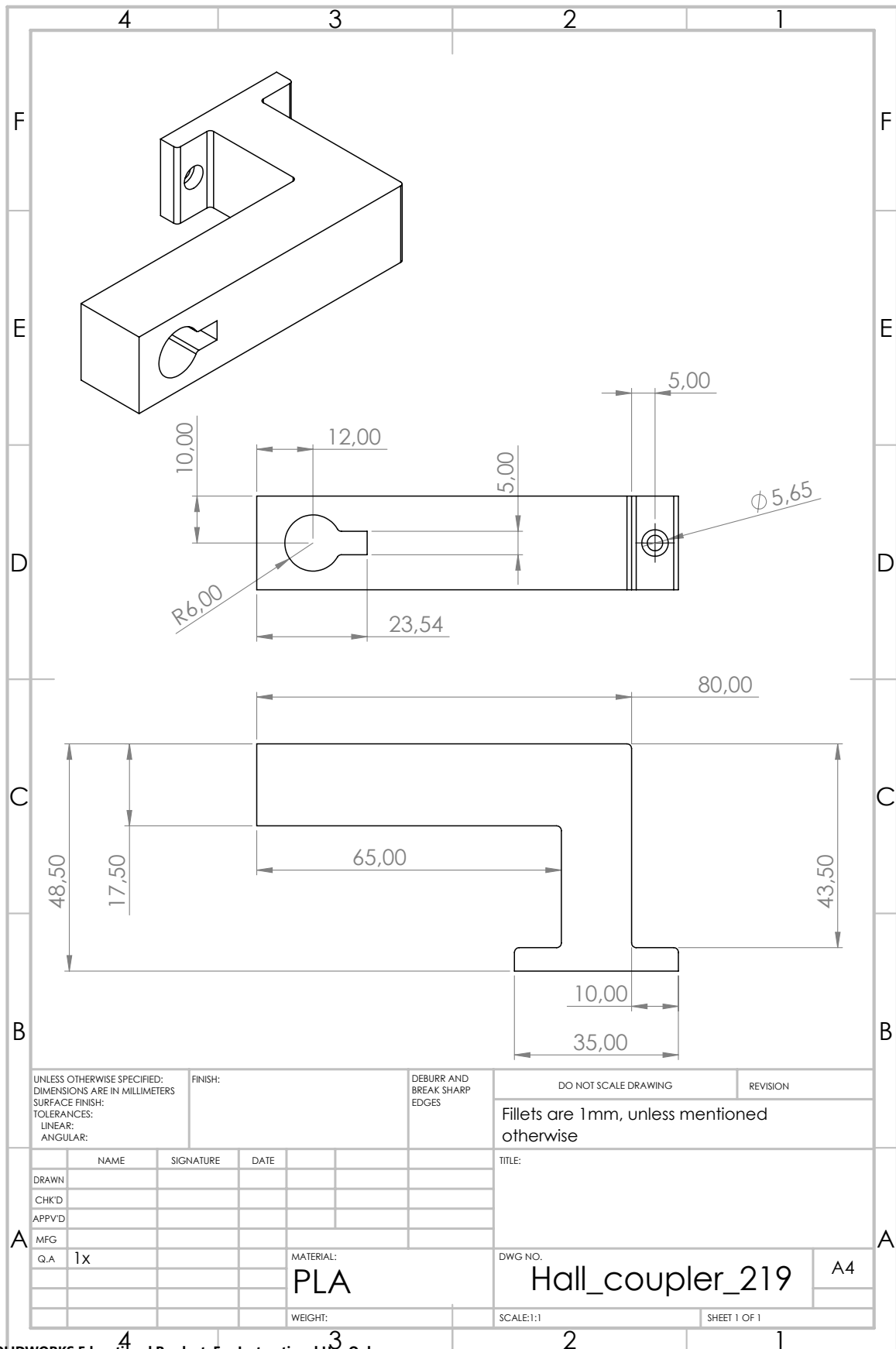
- 5x nylon 8mm bearing [35]
- 9x nylon M8 washer [36]
- 2x metal M8 washer [37]
- 8x metal M6 washer [38]
- 3x metal M5 washer [39]
- 20x metal M3 washer [40]
- 2x M8 Starlock [41]
- 17x M3 hexagon nut [42]
- 3x M5 hexagon nut [43]
- 8x M6 20mm cylinder head screw [44]
- 3x M5 20mm cylinder head screw [45]
- 2x M3 20mm cylinder head screw [46]
- 2x M3 12mm cylinder head screw [47]
- 1x M3 30mm flathead screw [48]
- 5x M3 12mm set screw [49]
- 3x M3 thread [50]

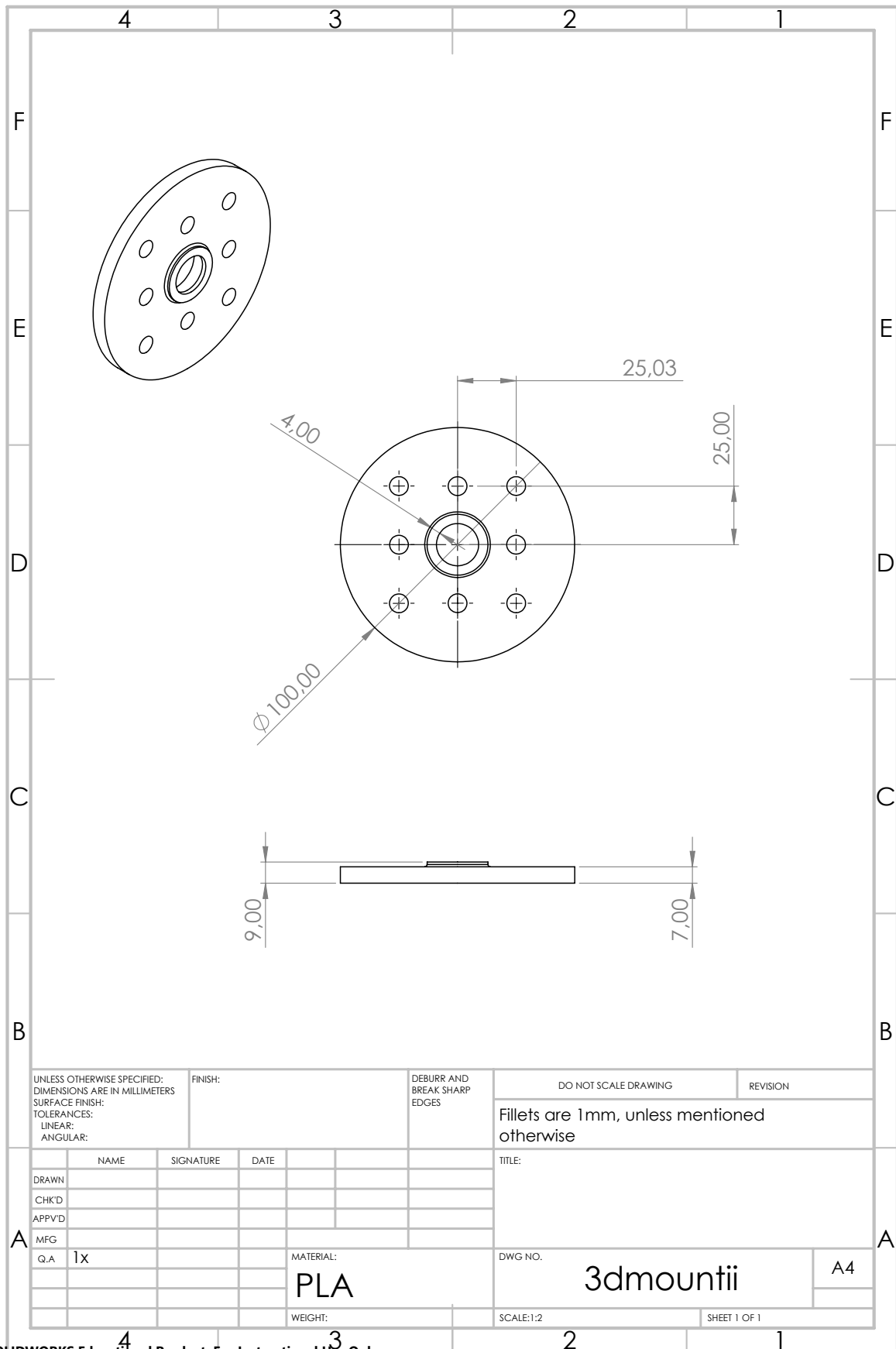
Other parts:

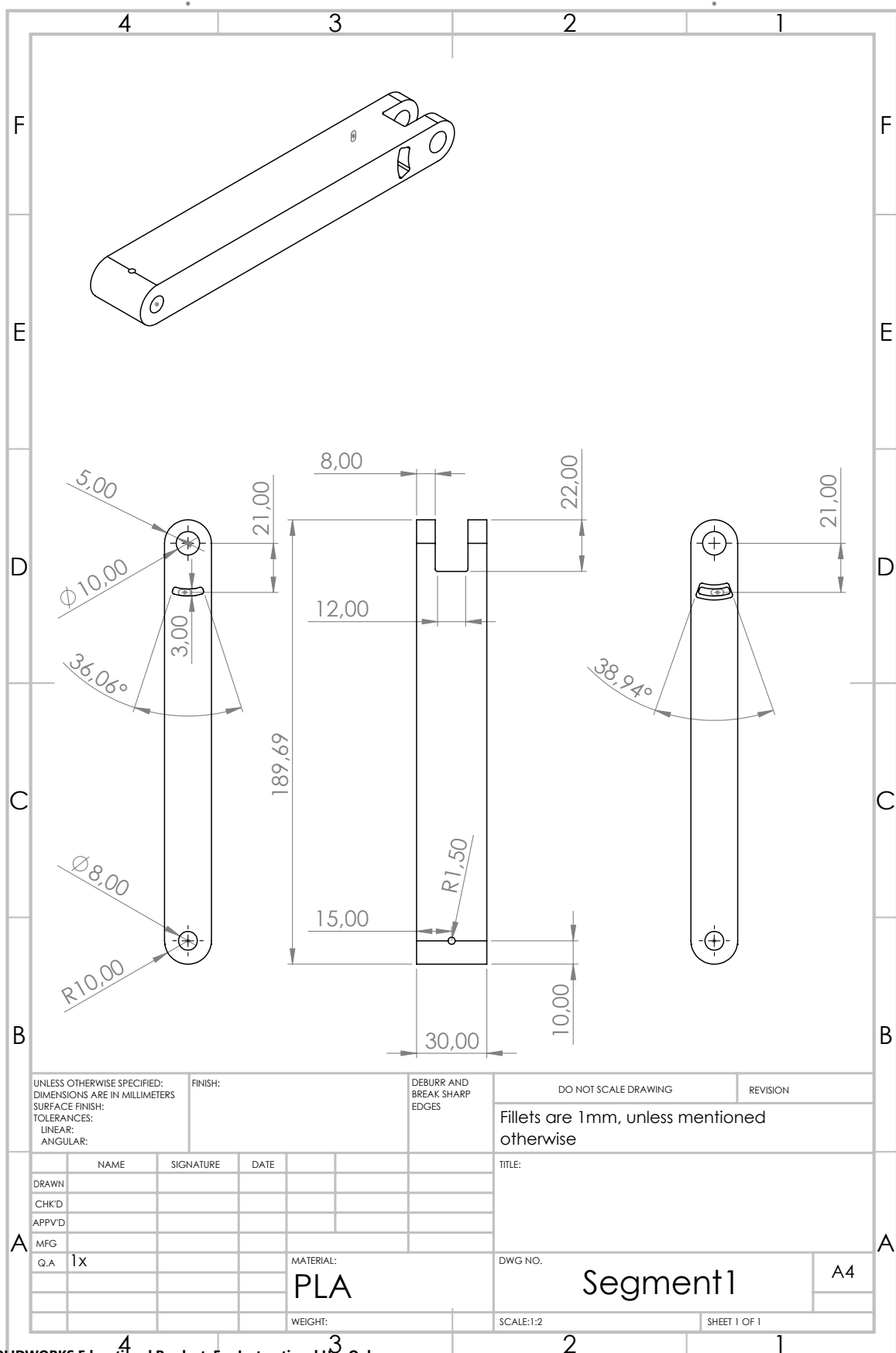
- 2x PLA "Attach plate"
- 1x PLA "Pushing bracket"
- 1x PLA "Hall coupler"
- 1x PLA "3dmount"
- 1x PLA "Segment1"
- 1x PLA "Segment2"
- 1x PLA "Segment3"
- 2x PMMA "Pendulum"
- 1x steel "Armaturerod1"
- 1x steel "Armaturerod2"
- 1x steel "Armaturerod3"
- 1x steel "Pendulumrod"

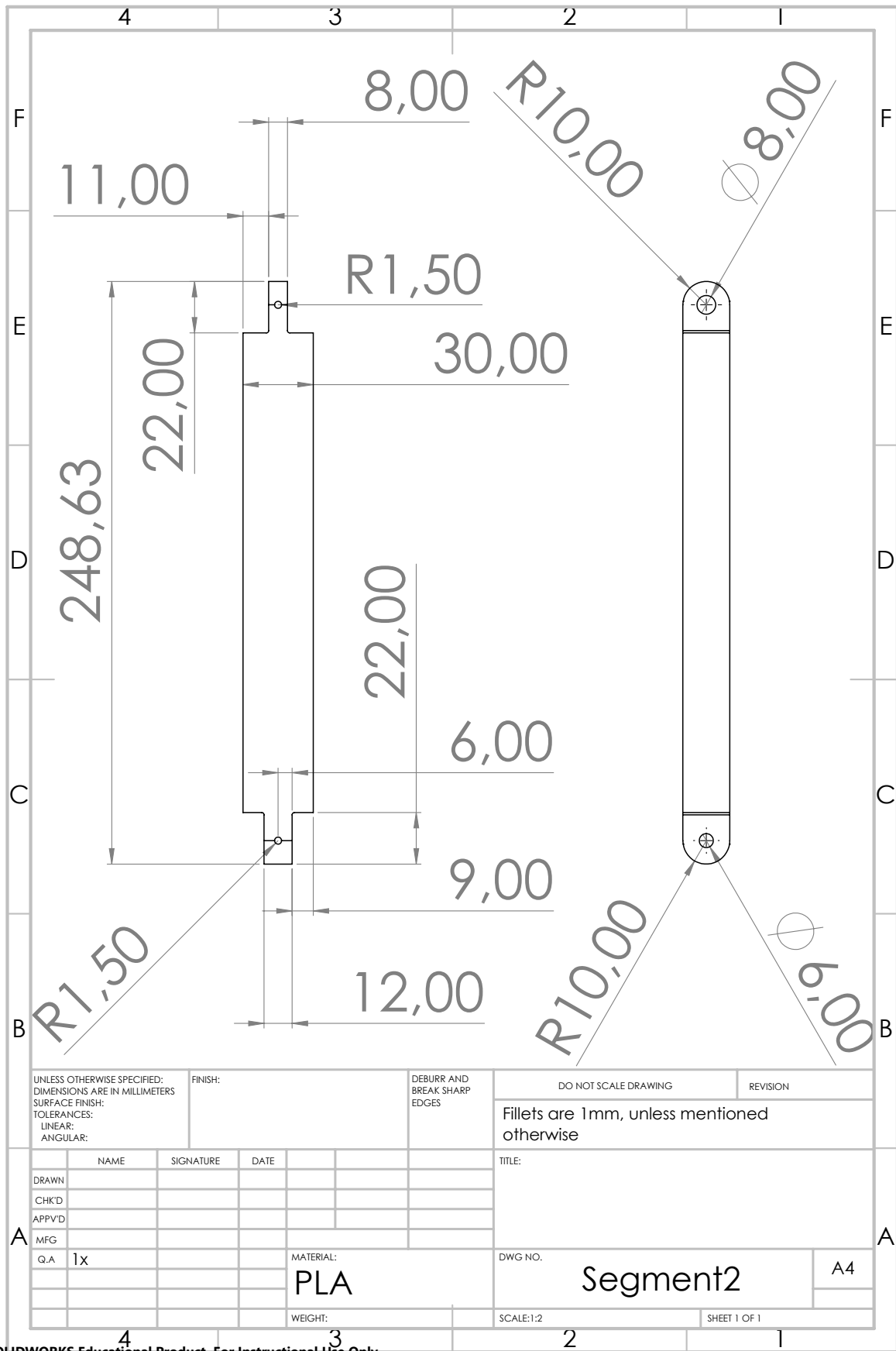


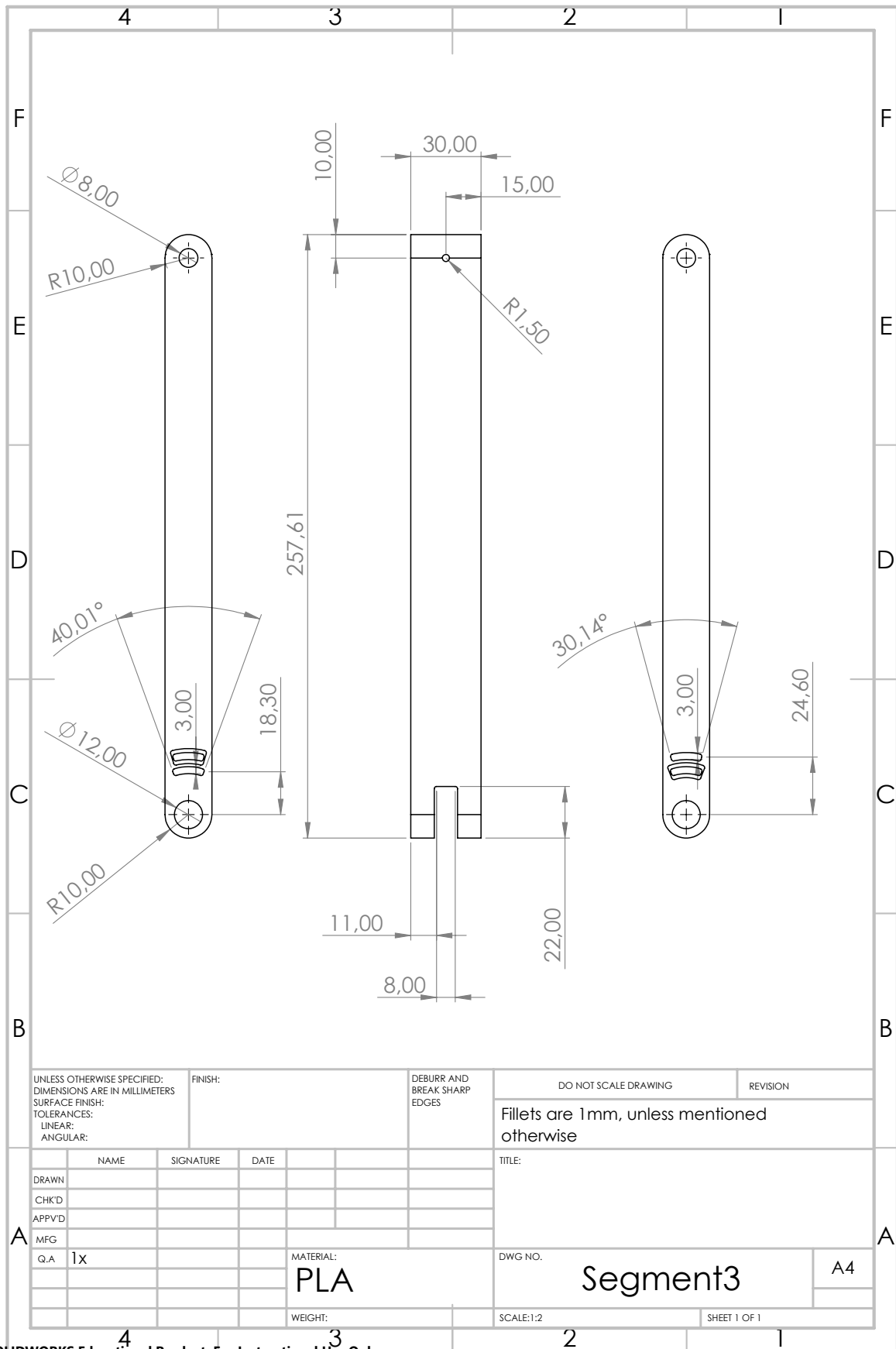




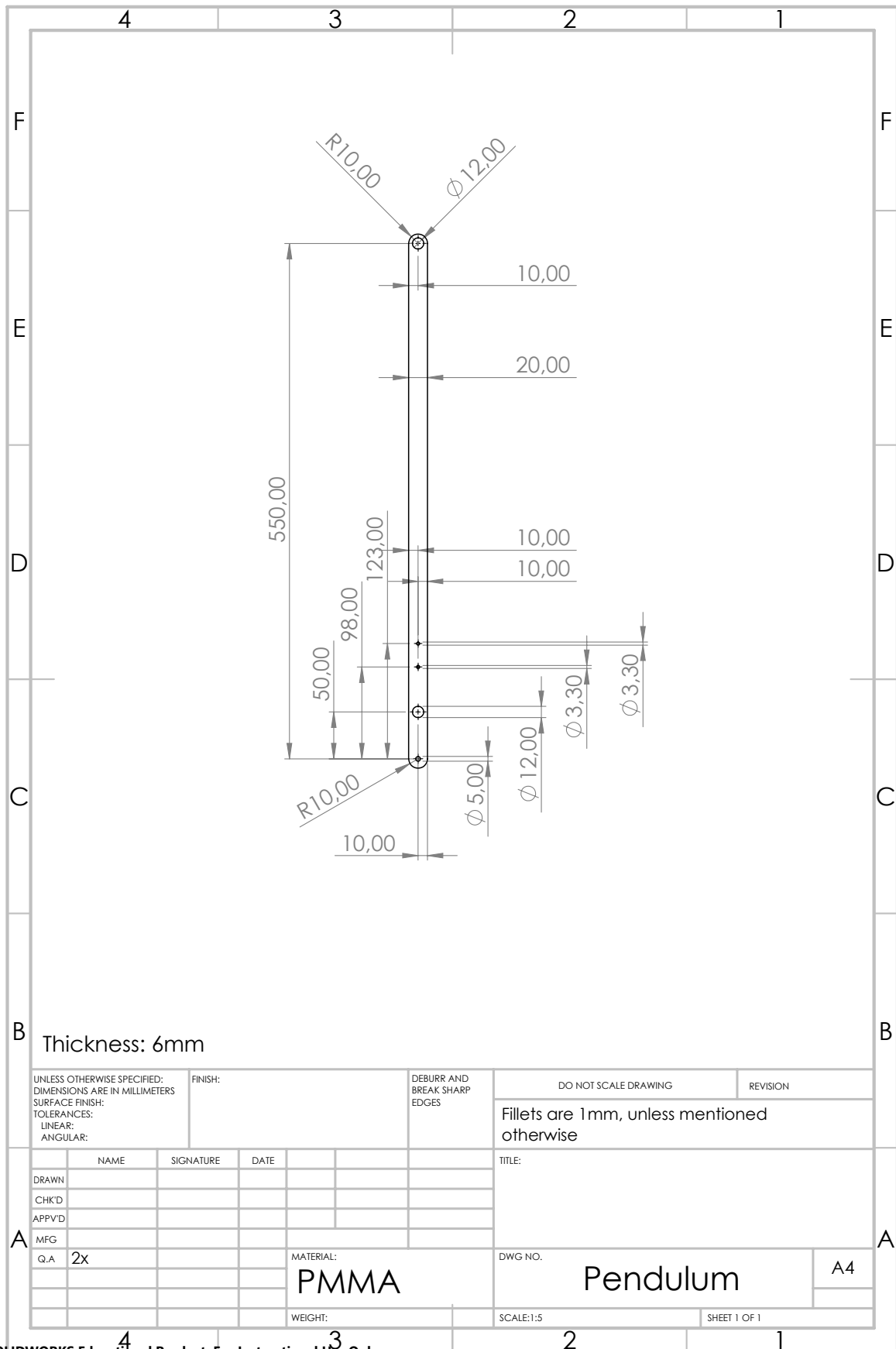


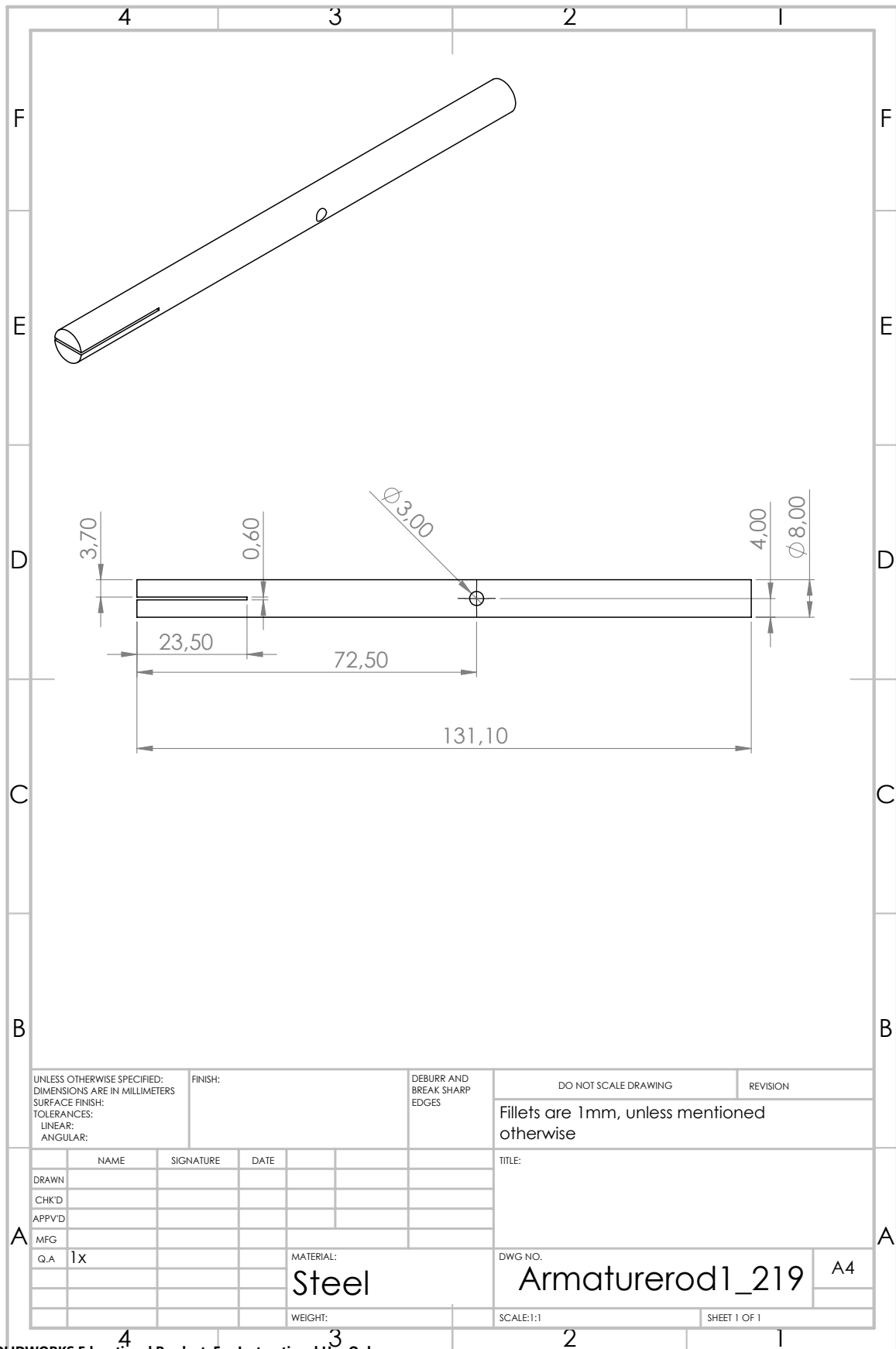


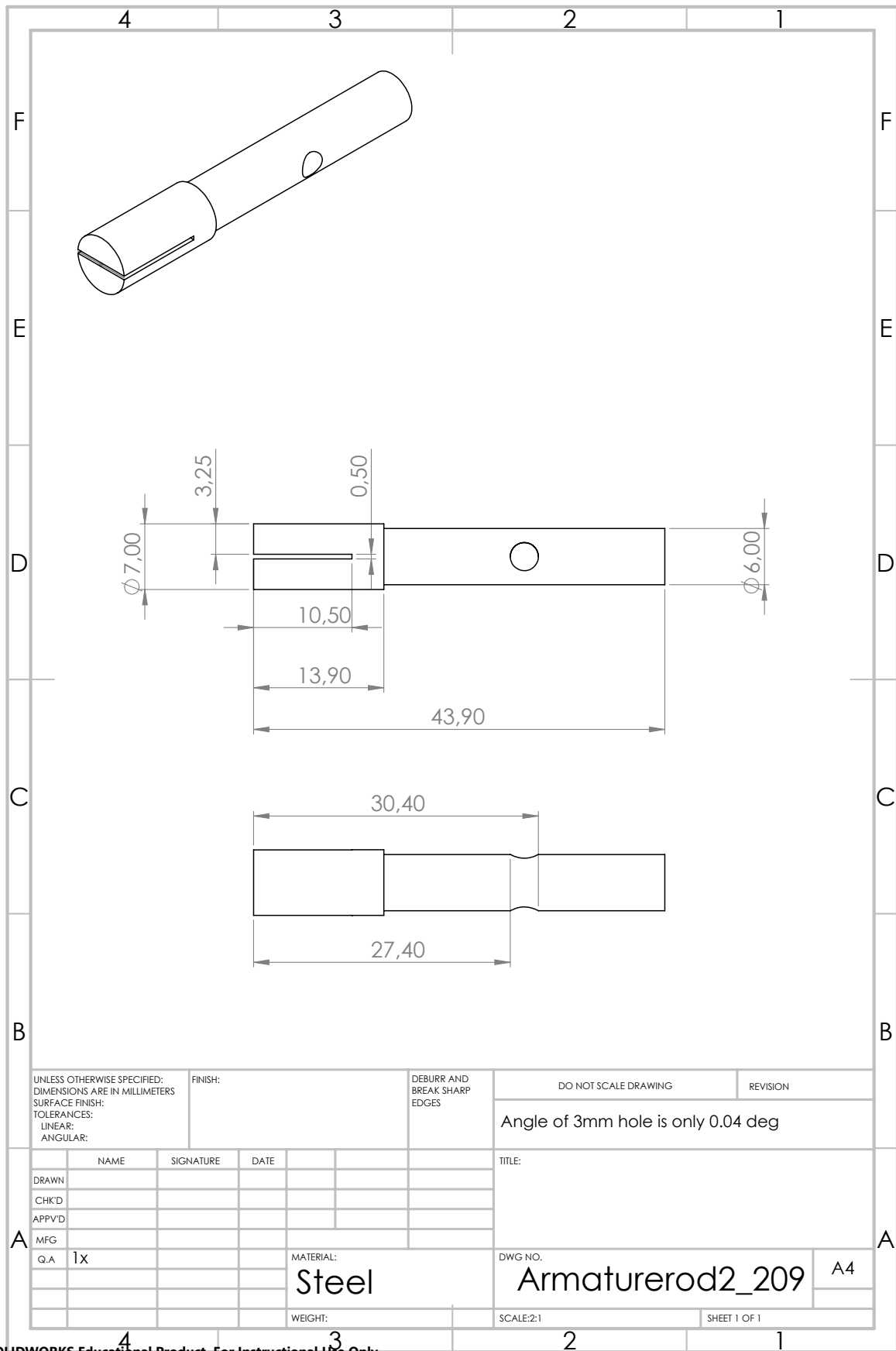


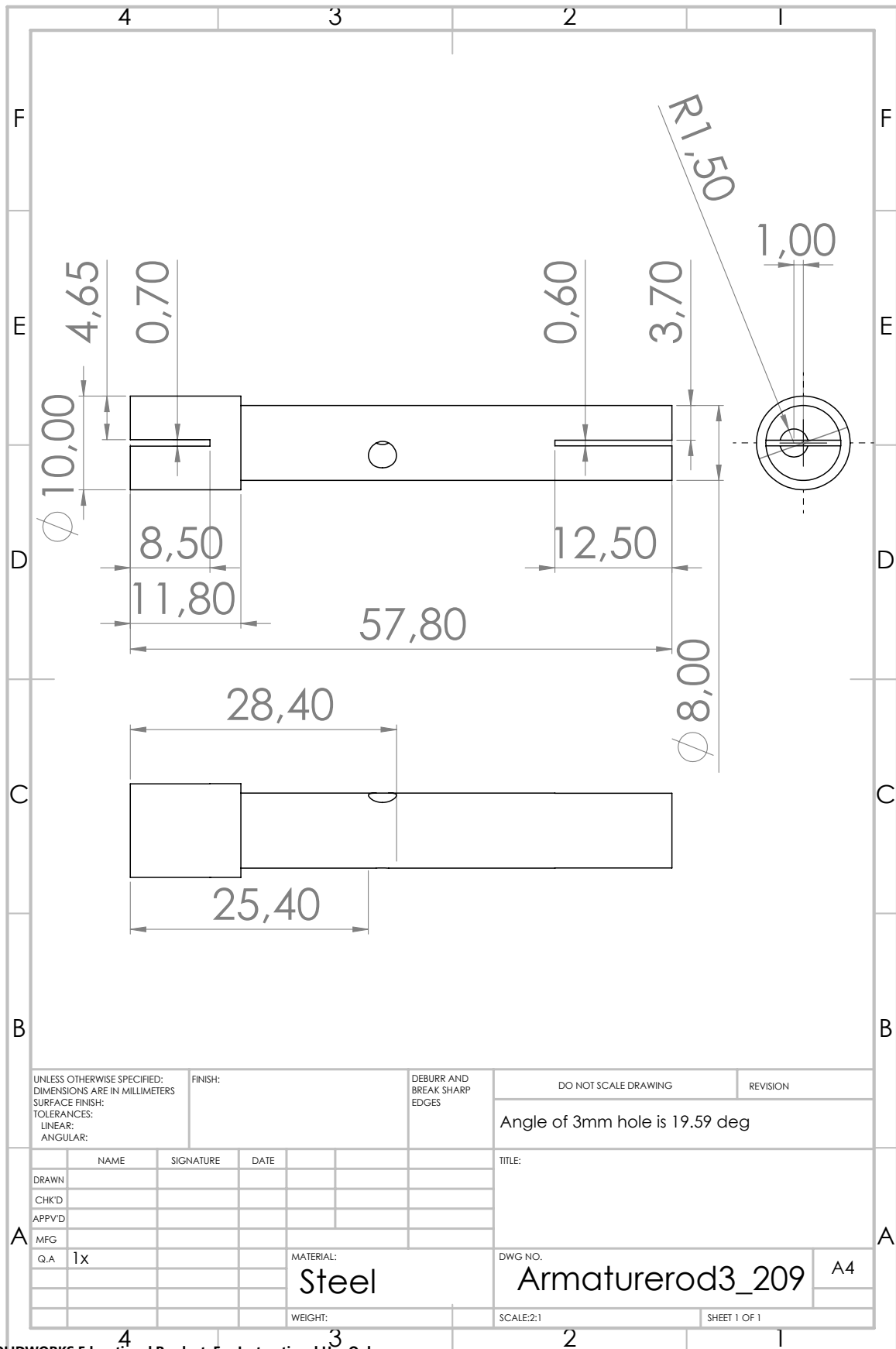


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:	DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
				Fillets are 1mm, unless mentioned otherwise		
NAME	SIGNATURE	DATE	TITLE:			
DRAWN						
CHK'D						
APP'VD						
MFG						
Q.A	1X		MATERIAL: PLA	DWG NO.		Segment3
			WEIGHT:	SCALE: 1:2		SHEET 1 OF 1

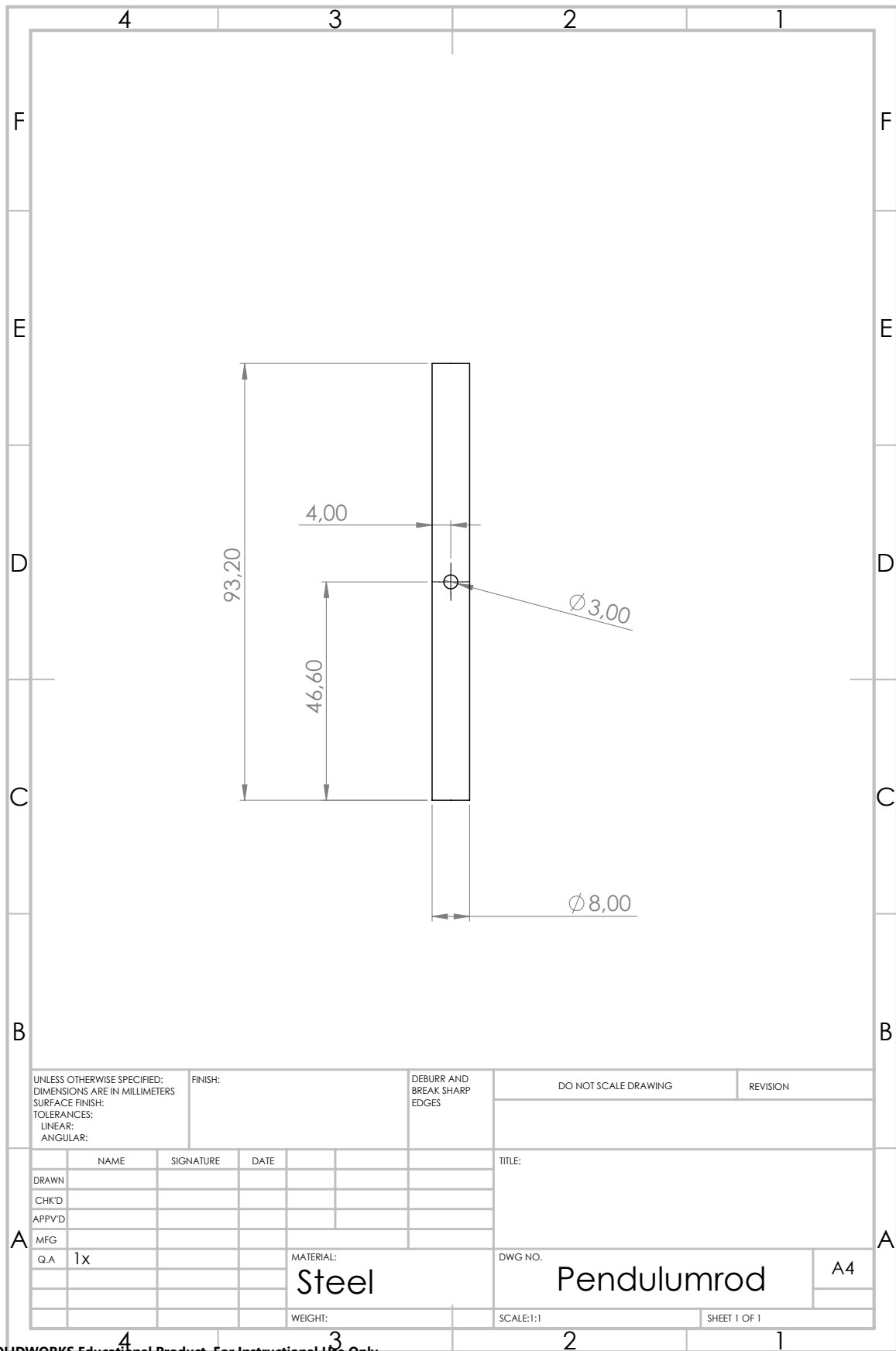








UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:			DEBURR AND BREAK SHARP EDGES			DO NOT SCALE DRAWING			REVISION		
									Angle of 3mm hole is 19.59 deg					
NAME			SIGNATURE			DATE			TITLE:					
DRAWN														
CHK'D														
APP'VD														
MFG														
Q.A			1X			MATERIAL:			DWG NO.			A4		
						Steel			Armaturerod3_209					
						WEIGHT:			SCALE:2:1			SHEET 1 OF 1		



H

Assembly

This chapter will elaborate on special remarks regarding the assembling process. The goal of the following sections is not to give an exhaustive overview, but merely to highlight potential difficulties of the assembling phase of the project. Section H.1 will discuss the installation of the steel axes into the PLA segments, whereafter a possible approach for the connection of the first axis with the environment is proposed in section H.2. Lastly, section H.3 and section H.4 will elaborate on typical problems regarding the installation of double springs and the upside down fixation of clock springs, respectively.

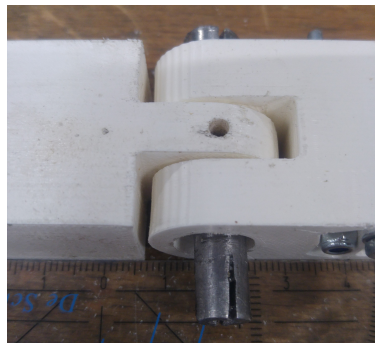
H.1. Installation axes

Typically, a significant load is required to insert the steel axes into the PLA parts. Although this could result in an appropriate clamping connection, immediately eliminating the degrees of freedom of the shaft with respect to the segment, a too high load could result in failure of the segments. Possible approaches to reduce the required load are listed below.

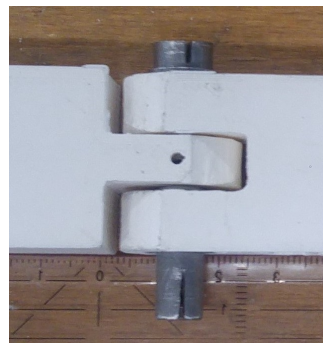
- Heat the PLA locally with use of a heat source
- Design for larger holes in the segments
- File the holes before inserting the axes
- Drill segment holes with correct diameter

The use of a hairdryer to heat the PLA locally was found to work relatively well in some cases. One should be aware of possible significant deformation of the PLA, which is undesired in some instances. It is therefore recommended to only heat segments that do not have any volume restrictions, like the hole in the first segment corresponding to the main axis. The “fingers” of the second segment, however, should not deform as this could easily result in contact with the fingers of the other segments. This contact will result in friction, which could have dramatic consequences for the performance of the balancer. For these holes, of the segments that should have minimum deformation, it might be useful to iteratively design for larger holes of the segments. Too large holes will result in play, whereas too small holes could result in the before mentioned failure of the segments. The applied force by installation of the axes can be reduced by filing or drilling the segments as well. During the assembly phase, it was discovered that a proper mounting of the segment is needed in order to drill without significant displacement and/or deformation. This mounting is not always available or possible, depending on the drill that is used. Furthermore, it was found that filing is a more delicate approach as the effect on the fit of the axis can be inspected immediately.

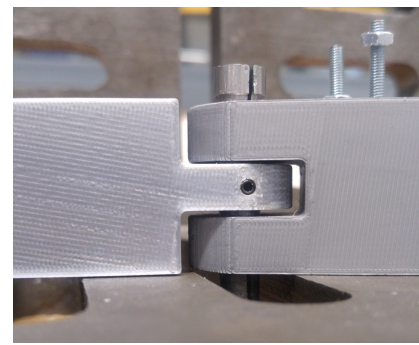
Two additional remarks should be made regarding the alignment of the segments and the axes. By inserting an axis between segment 1 and 2 or between segment 2 and 3, the fingers might deform. This is illustrated in figure H.1, where figure H.1a depicts the resulting misaligned axis. This deformation was mitigated by inserting strips of grinding paper between the gaps before the installation of the axes. As a result, the fingers remained in their horizontal orientation as depicted in figure H.1b. As the segments should be aligned with respect to each other as well, a perforated steel block is used to ensure that the bottom faces of the segments remain parallel. An image of the latter is shown in figure H.1c.



(a) Poorly aligned axis due to deformation of the upper fingers



(b) Better aligned axis by use of strips of paper



(c) Use of perforated steel block to align segments with respect to each other

Figure H.1: Alignment segments and axes

H.2. Connection main axis with environment

Section H.1 already discussed points of attention regarding the installation of axes into the segments. It is expected that the alignment of these axes is important as the potential energy of the balancer can be affected by any misalignments. As a matter of fact, if one of the axes of the balancer would be installed under an angle, the next segments and axes would be oriented under an angle as well. This angle causes the balancer to experience a difference in height energy during its rotation. This difference in potential energy will result in other equilibrium positions of the system and thus impaired kinematics. It is expected that any misalignment of the first, main, axis has a relatively large effect on these equilibrium positions as a large arm will result in a large height difference for a given angle.

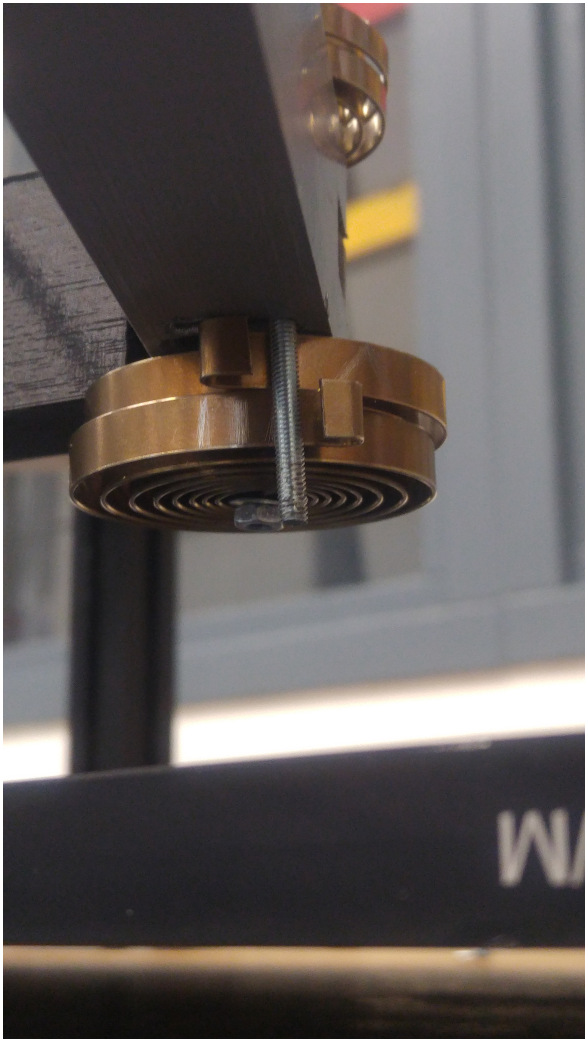
To realize a correct orientation of the main axis, it is recommended to not immediately tighten the bolts in the bottom connection part. This cylindrically shaped PLA part is seen in the left bottom corners of figure G.1 and figure G.2. Instead, it is advisable to first assemble the system without mounting the springs. Then, one is able to detect any preference positions that could be caused by a non-vertical main axis. The orientation of this axis might then be adjusted before fixating the bolts.

H.3. Mounting double springs

Sets of springs can be installed analogously to single springs, but the geometry of the to be mounted springs might differ. In most cases, the angle between the inner and outer connection parts deviates significantly. For these cases, one of the springs should be reversed. As a result, one spring is initially under compression, while the other is tensioned. The resulting initial moment of the set will be zero. An exemplifying figure is shown in figure H.2a.

H.4. Constraining upside down double springs

As the third axis is equipped with four springs in total, one set is located below the segments. Depending on the thickness of the groove in the axis, the clock springs might shift along the axis as a result of the gravitational force. An approach to constrain the clock springs in vertical direction is the use of a small ring in combination with a setscrew and a nut, as depicted in figure H.2b. In this case, a 8mm deep 2.5mm hole is drilled into the axis. Sequentially, the hole is provided with a M3 thread.



(a) One spring will be loaded in compression and the other in tension



(b) Holding the springs by means of a setscrew, nut and a ring

Figure H.2: Mounting double springs



Spring design

Because of the relatively low stiffness of the off-the-shelf clock springs, it was attempted to design stiffer springs by adjusting the geometry. Increasing the stiffness would increase the internal moment, as a result of which the contribution of friction to the measurements results would be reduced.

The geometry of the Lesjöfors 907 spring was altered to increase the stiffness. An expression for the stiffness of a clock spring is given in equation I.1 [51] [52], where E denotes the Young's modulus of the material, b the out of plane thickness, t the in plane thickness and L the effective length of the spring. It is seen that the in plane thickness has a third power relation with the stiffness. It is therefore decided to alter this spring parameter.

$$k_c = \frac{Ebt^3}{12L} \quad (I.1)$$

Eventually, the in plane thickness was multiplied by $5^{1/3} = 1.71$ to realize an increase in stiffness by a factor five. The spring was both watercutted out of a 3mm thick AISI 301 plate and lasercutted out of 2mm, 3mm and 4mm thick DC01 quality steel plates. Figure I.1 depicts top view images of these springs. Eventually, only the stiffness of the lasercutted 2mm thick spring was evaluated in the universal test bench. This spring appeared, by visual inspection, the most useful of the four. The watercutted spring is shown in figure I.1a. The non-constant in plane thickness is caused by displacement of the material. It appeared that the spring could not be appropriately constrained, as a result of which the water jet displaced the thread during the cutting process. The lasercutted 3mm and 4mm thick springs are visualized in figure I.1c and figure I.1d, respectively. Again, significant deviations in in plane thickness are observed. Moreover, the 4mm version does not have an arbor as the spring fell apart during the cutting process. This is expected to be caused by a relatively high temperature of the material. As a matter of fact, the power of the laser is constant, but the cutting speed decreases for thicker plates. As a result, the material is exposed to the heat of the laser for a longer period of time. Although the 2mm lasercutted spring, shown in figure I.1b, appears to have less deviation in its in plane thickness than the other lasercutted springs, the arbor and the outer connection ring are not as well aligned as those of the thicker springs.



(a) Watercutted 3mm thick spring (b) Lasercutted 2mm thick spring (c) Lasercutted 3mm thick spring (d) Lasercutted 4mm thick spring

Figure I.1: Watercutted and lasercutted, modified versions of the 907 clock spring

J

Spring evaluation

As described in section G.2, clock springs are used to store the required potential energy in the balancer. The presented, most current, balancer is equipped with a total of eight clock springs. To obtain the required spring stiffness ratios, three different types of clock springs are used. These springs will be distinguished based on their Lesjöfors part number. These part numbers and the properties of the corresponding springs are presented in table J.1.

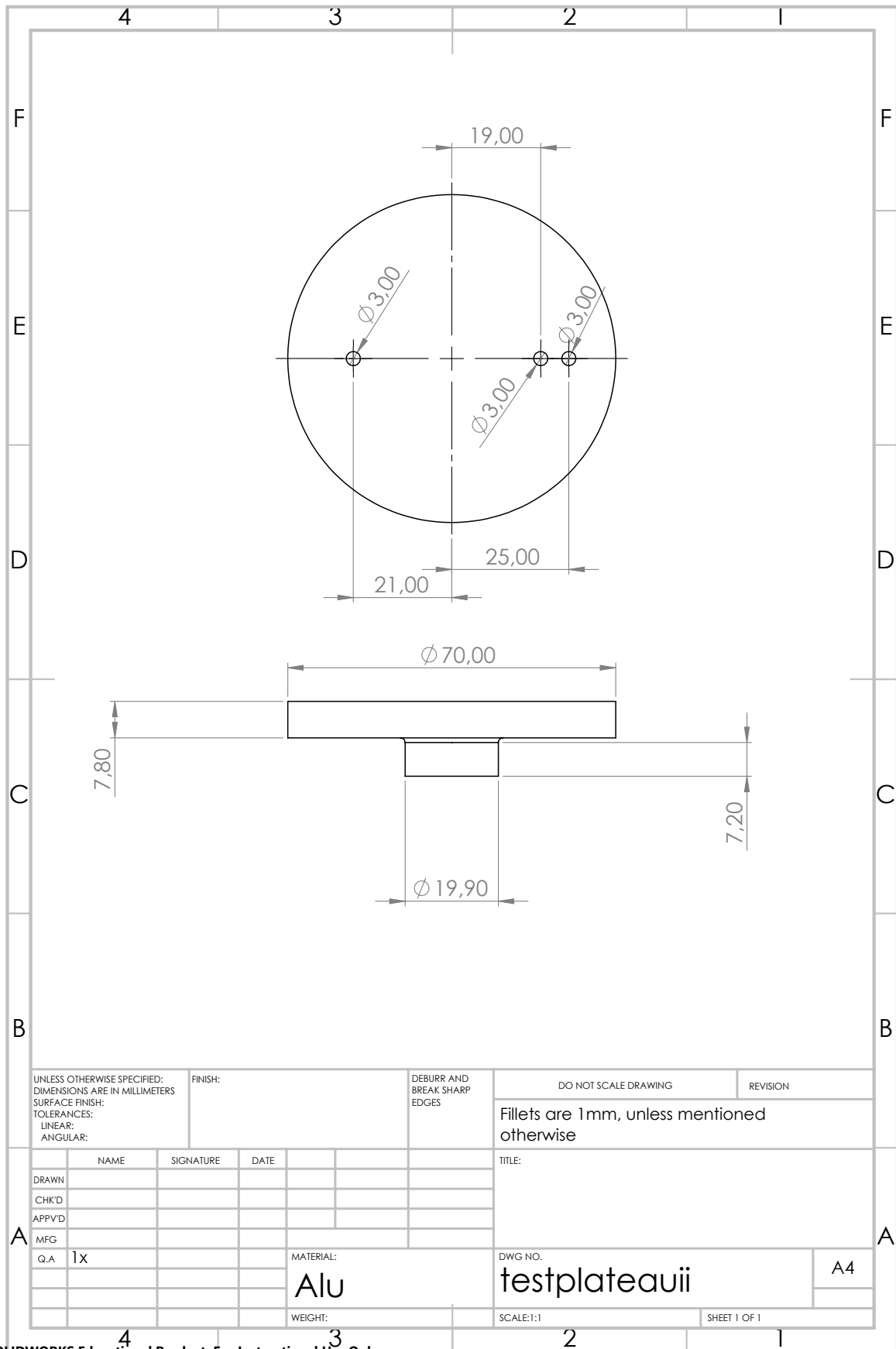
Spring	Stiffness (Nm/rad)	Amount of windings (-)	Thickness (mm)	Width (mm)	Inner radius (mm)	Outer radius (mm)
907	0.037	8	0.6	6.0	4.0	25
903	0.025	8	0.5	5.0	3.5	21
908	0.082	5	0.7	4.0	5.0	19

Table J.1: Properties distinct clock springs

Two extra 907 springs, one extra 903 and one additional 908 spring are evaluated next to the eight mentioned springs. Moreover a lasercutted, modified version of the 907 spring is tested as well. The results of these measurements are tabulated and discussed in section J.2 and section J.3, respectively. First, the required preparation and background information on the measurement procedure are elaborated in section J.1.

J.1. Spring evaluation preparation

The stiffnesses of the clock springs are evaluated by measuring a part of their moment-angle characteristic on a Zwickroell Z005 AllroundLine universal test bench [53]. The test bench is shown in figure J.1. Figure J.1a depicts the mounting plateau, a clock spring, the input shaft and the output shaft. An overview of the test bench is presented in figure J.1b and figure J.1c. During the assembly phase at the turning lathe and milling table, spare axes are made as well. These copies are used to realize the connection between the head of the test bench and the arbor of the clock springs. As the head of the measurement device has a 8mm hole, the spare axes corresponding to the 907 and 908 springs are already suited to use as input axes for the torsion tests. The 903 spring, however, is typically mounted on a stepped shaft with diameters of 6mm and 7mm. Another 8mm shaft is therefore provided with a 0.5mm thick groove to facilitate connection with the head of the test bench. The mounting plateau is a stepped aluminium shaft with three 3mm diameter holes to attach the clock springs. The part of the plateau with a diameter of approximately 20mm fits into the bottom part of the test bench and the remaining degrees of freedom are constrained by means of a set screw. A Solidworks drawing of the mounting plateau is included below.



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:			FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION		
							Fillets are 1mm, unless mentioned otherwise				
DRAWN			SIGNATURE			DATE			TITLE:		
CHK'D											
APPV'D											
MFG											
Q.A			1X			MATERIAL: Alu			DWG NO. testplateauii		
						WEIGHT:			SCALE:1:1		
									SHEET 1 OF 1		

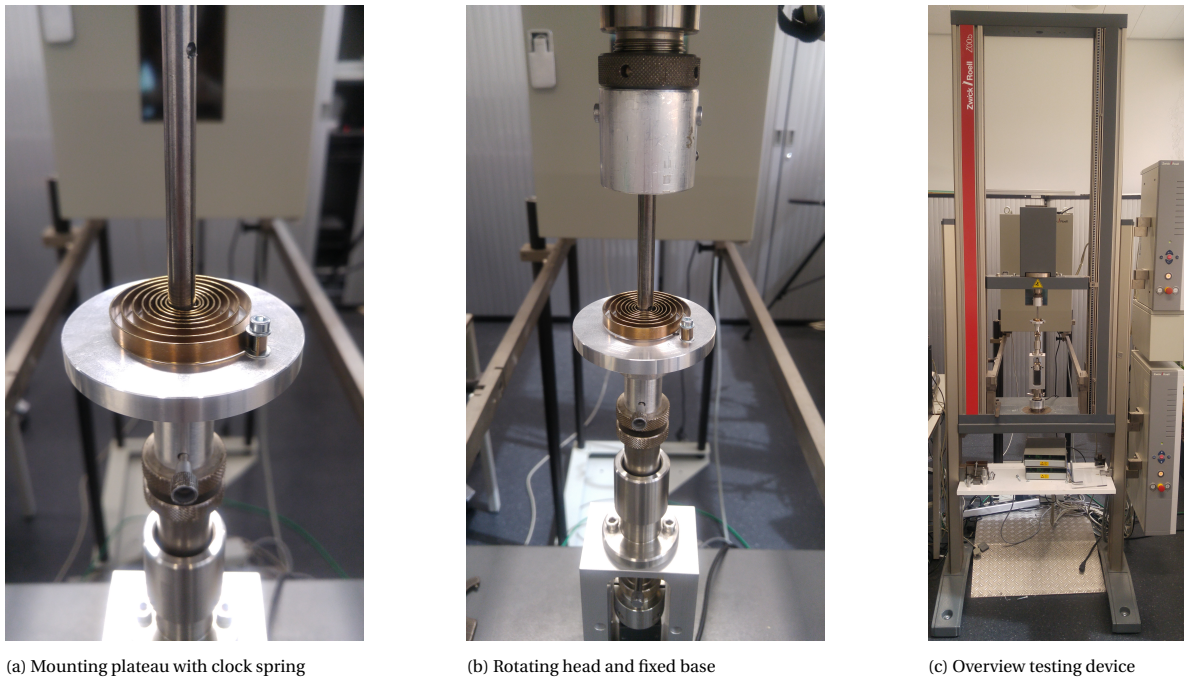


Figure J.1: Measurement setup ZwickRoell universal test bench

J.2. Spring evaluation results

The measurement results are provided in table J.2. The first column represents the tested springs, where the Lesjöfors part number is used as a shorthand notation. As multiple springs of the same type are tested, the number in between brackets is scribed into the corresponding springs as well. The 907 (mod) spring is the only lasercutted spring. The second, third and fourth columns represent the stiffnesses that are derived from a measurement run. The fifth column contains the averages of those runs. Lastly, the stiffnesses provided by the spring manufacturer and the deviation between expected and measured stiffnesses are shown in column six and seven, respectively.

Spring	Stiffness measurement 1 (Nm/rad)	Stiffness measurement 2 (Nm/rad)	Stiffness measurement 3 (Nm/rad)	Average measurement value (Nm/rad)	Expected stiffness (Nm/rad)	Deviation (%)
907	0.034	0.034	0.036	0.035	0.037	-7.03
907 (ii)	0.037	0.037	0.037	0.037	0.037	-1.32
907 (iii)	0.036	0.036	0.036	0.036	0.037	-3.89
907 (iv)	0.037	0.037	0.037	0.037	0.037	0.23
907 (v)	0.038	0.038	0.038	0.038	0.037	0.84
907 (vi)	0.037	0.037	0.037	0.037	0.037	-1.23
907 (mod)	0.033	0.033	0.033	0.033	0.051	-34.92
903	0.021	0.021	0.021	0.021	0.025	-16.03
903 (ii)	0.022	0.022	0.022	0.022	0.025	-11.90
903 (iii)	0.022	0.022	0.022	0.022	0.025	-12.29
908	0.069	0.071	0.071	0.070	0.082	-14.34
908 (ii)	0.071	0.071	0.072	0.071	0.082	-12.93
908 (iii)	0.070	0.069	0.068	0.069	0.082	-15.99

Table J.2: Measured stiffnesses compared to expected stiffnesses

J.3. Spring evaluation discussion

The last column of table J.2, the column presenting the deviation of the average measured value from the expected stiffness, illustrates that the 907 springs generally have the lowest stiffness deviation. An exception is the lasercutted variant, having a 34.92% deviation from the expected stiffness. The 903 and 908 springs, on the other hand, have relatively large stiffness deviations. As the deviations of the latter are negative, the springs are softer than they would be in theory. Discrepancies in stiffness are expected to originate from deviating spring geometries. An expression for the stiffness of a clock spring is shown in equation I.1. As can be seen from the latter equation, the in plane thickness has a third power relation with the stiffness. Furthermore, the spring stiffness changes proportionally with the (out of plane) thickness and inversely with the effective length of the spring.

No deviations in thickness and width were observed with a caliper. The effective length, however, was less trivial to measure. It should be noted that the inner and outer connection parts of the springs appeared to be not perfectly aligned. This alignment problem on its own is not expected to cause the larger discrepancies in spring stiffnesses, as the inner connection part could be rotated relatively easy without a significant change in effective length of the spring. A possible side effect, on the other hand, could be the forced contact of the arbor of the spring with the circumference of the axes the spring is mounted on. This contact could have a significant effect on the effective length of the spring and thus the measured stiffness. Extra contact, however, would result in a decreased effective length of the spring and thus an increase in stiffness. This contact could only attribute to the deviation in stiffness if the spring supplier accounts for this contact, whereas no contact is present in practice. Although the assumptions and expectations of the spring supplier are not known, apart from the information on the corresponding website, it should be stressed that the implemented axes have the same diameter as the arbor of the clock spring and should therefore be well suited for the application.

A further observation is the relatively small deviation between the measured stiffnesses. It is therefore expected that the repeatability of the measurement setup is relatively high. The repeatability of the fabrication technique of the clock springs is expected to be less high, as the mutual deviations in stiffness between the distinct springs of the same type are significant.



TU cluster

The cluster of the Precision and Microsystems Engineering department is used to allow for relatively time efficient optimization runs. The cluster consists of multiple computers connected to one main computer, also called the main node. These computers are faster than the laptop that should have done the calculations otherwise, but the main advantage is that multiple optimization tasks can be uploaded at the same time. As these tasks are actually performed on different calculation units, the amount of tasks does not influence the needed optimization time. The latter would not hold when MATLAB Parallel Optimization is used on one machine: each task is devoted to one or multiple cores and the computational power per task therefore decreases.

To work on the aforementioned cluster from a Windows machine, one will need a version of notepad, a (VPN) connection with the TU Delft network, PuTTY [54] and WinSCP [55]. PuTTY is used to prepare and monitor jobs, while WinSCP is needed to exchange files with the local personal folder on the cluster. An example notepad file that executes a job in the cluster, which should be given the extension “.pbs” is shown in figure L.1. This file should be copied to the personal folder on the cluster with WinSCP, while the following command should be typed in PuTTY: “qsub notepadfilename.pbs”.

```
1  #!/bin/sh
2
3  #request 20 processors on 1 node
4  #PBS -l nodes=1:ppn=20
5
6  #define the name of the job
7  #PBS -N Optimization_Sjors
8
9  #provide mail adress
10 #PBS -M a.c.vannes@student.tudelft.nl
11
12 #give mail preferences
13 #PBS -m abe
14
15 # Make sure I'm the only one that can read my output
16 umask 0077
17
18 #change to the working directory
19 cd $PBS_O_WORKDIR
20
21 #load MATLAB
22 module load matlab/2020b
23
24 #define the name of the MATLAB file
25 matlab -r Spring_selector
```

Figure L.1: TU cluster example PBS file

M

Optimization results

In the following, all obtained optimization results will be provided. Section M.1 will include the results of optimization without spring-environment contact, whereas section M.2 elaborates on separate optimization runs with the contact angle as minimizer. Section M.3, on the other hand, contains the results for extra optimization attempts with relaxed lower- and upperbounds for the segment lengths. More elaborate tables that also include the values of the minimizers and the root mean square error are provided in section M.4.

M.1. Optimization without contact

Table M.1 presents the work reduction of the three segment balancers and table M.2 concerns those of the four segment balancers without optimized initial angle of segment 1.

Objective curve	Regular	Prestress	Nonlinear	Nonlinear, prestress	Opt. θ_{10}	Opt. θ_{10} , prestress	Nonlinear, opt. θ_{10}	Nonlinear, opt. θ_{10} , prestress
Sine (90 deg)	88.95%	98.20%	99.28%	99.10%	99.33%	99.45%	99.56%	99.62%
Progressive	99.61%	98.97%	96.52%	98.90%	75.25%	75.13%	95.06%	96.68%
Progressive-degressive	90.48%	90.50%	87.25%	90.60%	94.54%	91.71%	93.41%	88.92%
Degressive-progressive	86.18%	98.17%	89.00%	91.35%	98.42%	98.76%	91.24%	92.00%
Laevo	70.94%	95.85%	91.38%	93.79%	99.66%	98.47%	93.69%	93.66%
Sine (180 deg)	38.47%	71.42%	93.34%	96.73%	97.10%	90.72%	97.49%	96.70%

Table M.1: Optimization results three segment balancer

Objective curve	Regular	Prestress	Nonlinear	Nonlinear, prestress
Sine (90 deg)	90.00%	99.12%	99.72%	99.43%
Progressive	99.16%	99.32%	96.67%	97.81%
Progressive-degressive	90.56%	86.72%	88.02%	93.17%
Degressive-progressive	86.95%	97.40%	89.46%	92.39%
Laevo	75.74%	97.44%	89.92%	94.15%
Sine (180 deg)	43.96%	77.26%	84.44%	96.72%

Table M.2: Optimization results four segment balancer

M.2. Optimization with contact

Contact is only enabled for some of the three segment balancers, as seen in table M.3. Only the objective curves with progressive parts are included. A balancer with nonlinear springs is indicated with “NL”.

Objective curve	Regular	Prestress	NL	NL, prestress	Opt. θ_{10}	Opt. θ_{10} , prestress	NL, opt. θ_{10}	NL, opt. θ_{10} , prestress
Progressive	96.51%	96.87%	97.96%	99.05%	75.23%	75.13%	94.83%	97.37%
Progressive-degressive	91.93%	91.56%	88.74%	90.80%	85.68%	91.30%	94.82%	91.32%
Degressive-progressive	86.19%	98.43%	89.04%	91.34%	86.32%	90.89%	90.75%	92.42%

Table M.3: Optimization results three segment balancer

M.3. Optimization with relaxed lower- and upperbounds for segment lengths

The lower- and upperbound of the segment lengths are relaxed. These bounds are defined to be the following:

$$0.1 \leq l_i \leq 0.9$$

Table M.4 represents the three segment balancer, whereas the results of the four segment balancers are shown in table M.5. The optimization procedure was done without enabling contact of the springs with the environment.

Objective curve	Regular	Prestress	Nonlinear	Nonlinear, prestress
Sine (90 deg)	98.19%	98.15%	98.80%	99.02%
Laevo	87.33%	95.13%	93.60%	93.81%

Table M.4: Optimization results three segment balancer with relaxed bounds for segment lengths

Objective curve	Regular	Prestress	Nonlinear	Nonlinear, prestress
Sine (90 deg)	95.54%	98.25%	96.30%	99.59%
Laevo	80.81%	92.41%	93.51%	93.49%

Table M.5: Optimization results four segment balancer with relaxed bounds for segment lengths

M.4. Optimization results elaborated

M.4.1. Sine

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														RMSE	Best theor. Approx.	Work red.
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4						
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	1.5	1.5	1.5	1.5	1.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	1.3808	0.4359	0.087	-	-	-	0.4868	0.4951	0.4507	-		0.0835	0.0861	88.95%		
4 Seg.	1.3125	0.9037	1.197	0.1568	-	-	0.3609	0.3739	0.3745	0.3083		0.0797	0.0913	90.00%		
3 Seg.: prestress spring 2	0.9341	0.1211	0.0927	-	-	0.3983	0.4764	0.4975	0.4986	-		0.0135	0.01664	98.20%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	0.01664	-		
4 Seg.: prestress spring 2 & 3	0.9512	0.2082	0.1392	0.0614	0.2502	0.6055	0.3252	0.2926	0.357	0.3558		0.00668	0.007022	99.12%		

Nonlinear springs											RMSE	Work red.	
	A	B	M03	M02	I1	I2	I3	I4					
LB	-2	-2	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25				
UB	2	2	1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375				
3 Seg.	-0.6047	1.5601	-	-	0.3334	0.4987	0.482	-			0.0054		99.28%
4 Seg.	-0.6123	1.564	-	-	0.3641	0.2529	0.2547	0.2545			0.002335		99.72%
3 Seg.: prestress spring 2	-0.4559	1.3625	-	0.0001	0.4853	0.3339	0.3366	-			0.006549		99.10%
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-			-		-
4 Seg.: prestress spring 2 & 3	-0.8031	1.7909	0.0123	0.0168	0.267	0.3107	0.3706	0.3557			0.0047		99.43%

Figure M.1

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg															RMSE	Best theor. Approx.	Work red.
Linear springs																	
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10						
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25						
UB	4.5	4.5	4.5	4.5	4.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375						
3 Seg.	1.2701	0.8473	4.1078	-	-	-	0.3394	0.4573	0.4752	-		0.6641	0.0059	99.33%			
4 Seg.	-	-	-	-	-	-	-	-	-	-		-	-	-			
3 Seg.: prestress spring 2	0.9668	0.1762	2.4054	-	-	0.5237	0.3727	0.4438	0.3792	-		0.3891	0.0043	99.45%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	-	-			
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-		-	-	-			

Nonlinear springs												RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4	theta10				
LB	-3.5	-3.5	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25				
UB	3.5	3.5	1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375				
3 Seg.	-0.8125	1.801	-	-	0.3482	0.4915	0.4993	-			0.5869	0.0033	99.56%
4 Seg.	-	-	-	-	-	-	-	-			-	-	-
3 Seg.: prestress spring 2	-0.6346	1.5943	-	0	0.4797	0.4934	0.4998	-			0.4729	0.0028	99.62%
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-			-	-	-
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-			-	-	-

Figure M.2

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg															RMSE	Best theor. Approx.	Work red.
Linear springs																	
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4							
LB	0	0	0	0	0	0	0	0.1	0.1	0.1	0.1						
UB	1.5	1.5	1.5	1.5	1.5	1	1	0.9	0.9	0.9	0.9						
3 Seg.	1.194	0.4247	0.1294	-	-	-	0.7298	0.6937	0.573	-		0.0193	0.0861	98.19%			
4 Seg.	1.2841	1.1961	1.0113	0.2982	-	-	0.4036	0.5063	0.5752	0.5128		0.04	0.0913	95.54%			
3 Seg.: prestress spring 2	0.9385	0.2559	0.1553	-	-	0.4737	0.3792	0.7545	0.5973	-		0.0141	0.01664	98.15%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	0.01664	-			
4 Seg.: prestress spring 2 & 3	0.9706	0.4726	0.5166	0.3371	0.3985	0.586	0.1628	0.2636	0.339	0.5038		0.016	0.007022	98.25%			

Nonlinear springs												RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4					
LB	-2	-2	0	0	0	0.1	0.1	0.1	0.1	0.1			
UB	2	2	1	1	1	0.9	0.9	0.9	0.9	0.9			
3 Seg.	-0.3613	1.2221	-	-	0.6372	0.2142	0.5479	-			0.0087		98.80%
4 Seg.	0.8623	0.3156	-	-	0.2428	0.3528	0.1356	0.3536			0.0311		96.30%
3 Seg.: prestress spring 2	-0.2693	1.1032	-	0.578	0.3887	0.8611	0.263	-			0.0075		99.02%
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-			-		-
4 Seg.: prestress spring 2 & 3	-0.7191	1.6941	0.1912	0.0092	0.1957	0.1862	0.1641	0.5304			0.0033		99.59%

Figure M.3

M.4.2. Progressive

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4		RMSE	Best theor. Approx.	Work red.		
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	1.5	1.5	1.5	1.5	1.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	1.3582	0.0165	0.7429	-	-	-	0.3923	0.4893	0.3976	-		0.0017	0.1016	99.61%		
4 Seg.	1.2788	0.0233	0.4838	0.9178	-	-	0.2544	0.3355	0.3087	0.257		0.0038	0.1016	99.16%		
3 Seg.: prestress spring 2	0.9811	0.0393	0.0965	-	-	0	0.4007	0.342	0.3555	-		0.0043	0.0176	98.97%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-			0.0176			
4 Seg.: prestress spring 2 & 3	1.2206	0.0304	1.2034	0.7933	0.9325	0.0035	0.2788	0.2832	0.3291	0.2808		0.0031	0.007	99.32%		

Nonlinear springs																
	A	B	M03	M02	I1	I2	I3	I4		RMSE			Work red.			
LB	-2	-2		0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25							
UB	2	2		1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375							
3 Seg.	0.5868	0.1931	-	-	0.3478	0.3925	0.335	-		0.0144			96.52%			
4 Seg.	0.3467	-0.4282	-	-	0.3104	0.2622	0.2996	0.3656		0.0183			96.67%			
3 Seg.: prestress spring 2	0.4207	0.0361	-	0.3642	0.4658	0.3362	0.3528	-		0.0049			98.90%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-								
4 Seg.: prestress spring 2 & 3	0.8027	-0.0119		0.9086	0.0313	0.3465	0.2609	0.3307	0.2745	0.0104			97.81%			

Figure M.4

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	RMSE	Best theor. Approx.	Work red.		
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	4.5	4.5	4.5	4.5	4.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	0.5163	4.3788	1.5123	-	-	-	0.3545	0.4992	0.5	-		1.3896	0.1017	75.25%		
4 Seg.	-	-	-	-	-	-	-	-	-	-						
3 Seg.: prestress spring 2	0.5152	1.5643	2.5515	-	-	0.944	0.3382	0.5	0.5	-		0.0152	0.1021	75.13%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-						
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-						

Nonlinear springs																
	A	B	M03	M02	I1	I2	I3	I4	theta10	RMSE			Work red.			
LB	-2	-2		0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25							
UB	2	2		1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375							
3 Seg.	0.6154	0.3584	-	-	0.3355	0.3342	0.4961	-		0.3727	0.0217		95.06%			
4 Seg.	-	-	-	-	-	-	-	-								
3 Seg.: prestress spring 2	0.3978	0.0612	-	0.3279	0.4719	0.4519	0.409	-		1.0353	0.0215		96.68%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-								
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-								

Figure M.5

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	contactan	theta10	RMSE	Best theor. Approx.	Work red.	
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	1.5	1.5	1.5	1.5	1.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	0.8971	0.089	0.3411	-	-	-	0.4029	0.4568	0.4343	-		0.8693	-	0.015	96.51%	
4 Seg.	-	-	-	-	-	-	-	-	-	-						
3 Seg.: prestress spring 2	0.895	0.0908	0.559	-	-	0.0003	0.3383	0.4369	0.4255	-		0.7989	-	0.0132	96.87%	
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-						
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-						

Nonlinear springs																
	A	B	M03	M02	I1	I2	I3	I4	contactan	theta10	RMSE		Work red.			
LB	-2	-2		0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25							
UB	2	2		1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375							
3 Seg.	0.3146	-0.2665	-	-	0.4594	0.3341	0.3787	-		0.8058	-	0.0094	97.96%			
4 Seg.	-	-	-	-	-	-	-	-								
3 Seg.: prestress spring 2	0.4342	0.0252	-	0.3134	0.4999	0.334	0.4844	-		0.4053	-	0.0044	99.05%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-								
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-								

Figure M.6

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																	
Linear springs																	
LB	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	contactan	theta10	RMSE	Best theor.	Approx.	Work red.	
UB	4.5	4.5	4.5	4.5	1	1	0.33-0.25	0.33-0.25	0.33-0.25	0.33-0.25	0.25	0.25	0.375				
3 Seg.	0.5165	2.8885	3.1711	-	-	-	0.4123	0.4909	0.4872	-	0.7515	1.3085	0.1017			75.23%	
4 Seg.	-	-	-	-	-	-	-	-	-	-	-	-	-			-	
3 Seg.: prestress spring 2	0.5152	1.3208	3.605	-	-	0.4759	0.3374	0.4978	0.4958	-	0.9532	1.3677	0.1021			75.13%	
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-	-	-	-			-	
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-	-	-	-			-	

Nonlinear springs																	
LB	A	B	M03	M02	I1	I2	I3	I4	contactan	theta10	RMSE	Best theor.	Approx.	Work red.			
UB	2	2	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.5-0.375	0.375	0.25	0.375						
3 Seg.	0.6425	0.3745	-	-	0.3347	0.3357	0.4997	-	1.277	0.4309	0.0226			94.83%			
4 Seg.	-	-	-	-	-	-	-	-	-	-	-			-			
3 Seg.: prestress spring 2	0.3576	0.0916	-	0.8926	0.4265	0.3596	0.4994	-	0.2809	0.8602	0.0109			97.37%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-	-			-			
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-	-			-			

Figure M.7

M.4.3. Progressive-degressive

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																	
Linear springs																	
LB	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	contactan	theta10	RMSE	Best theor.	Approx.	Work red.	
UB	2.5	2.5	2.5	2.5	1	1	0.33-0.25	0.33-0.25	0.33-0.25	0.33-0.25	0.25	0.25	0.375				
3 Seg.	0.9601	0.0001	2.4904	-	-	-	0.4986	0.3661	0.337	-	-	-	0.0585			90.48%	
4 Seg.	1.0179	1.3803	0.0007	1.9876	-	-	0.357	0.288	0.373	0.3366	-	-	0.0606			90.56%	
3 Seg.: prestress spring 2	0.9668	0.0014	2.4967	-	-	0	0.4983	0.3357	0.3758	-	-	-	0.0586			90.50%	
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-	-	-	-			-	
4 Seg.: prestress spring 2 & 3	0.6792	0.0264	0.878	2.4662	0.5026	0.6944	0.2847	0.3388	0.3168	0.2999	-	-	0.0829			86.72%	

Nonlinear springs																	
LB	A	B	M03	M02	I1	I2	I3	I4	contactan	theta10	RMSE	Best theor.	Approx.	Work red.			
UB	2	2	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.5-0.375	0.375	0.25	0.375						
3 Seg.	0.2203	0.6432	-	-	0.4883	0.4976	0.4774	-	-	-	0.072			87.25%			
4 Seg.	0.1947	0.8758	-	-	0.2795	0.3739	0.2674	0.339	-	-	0.071			88.02%			
3 Seg.: prestress spring 2	0.3146	0.3632	-	0.4713	0.3341	0.5	0.4735	-	-	-	0.0542			90.60%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-	-			-			
4 Seg.: prestress spring 2 & 3	0.4588	0.2601	0.2306	0.4467	0.2846	0.3635	0.375	0.3241	-	-	0.0438			93.17%			

Figure M.8

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																	
Linear springs																	
LB	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	RMSE	Best theor.	Approx.	Work red.		
UB	4.5	4.5	4.5	4.5	1	1	0.33-0.25	0.33-0.25	0.33-0.25	0.33-0.25	0.25	0.375					
3 Seg.	4.3662	0.2491	0.0005	-	-	-	0.4996	0.5	0.4996	-	0.639	0.0348			94.54%		
4 Seg.	-	-	-	-	-	-	-	-	-	-	-	-			-		
3 Seg.: prestress spring 2	0.8311	0.0193	0.0009	-	-	0.0067	0.4052	0.499	0.4972	-	-1.1515	0.0531			91.71%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-	-	-			-		
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-	-	-			-		

Nonlinear springs																	
LB	A	B	M03	M02	I1	I2	I3	I4	theta10	RMSE	Best theor.	Approx.	Work red.				
UB	2	2	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.5-0.375	0.375	0.25	0.375						
3 Seg.	0.849	0.2714	-	-	0.4449	0.3821	0.4669	-	0.9152	0.0391			93.41%				
4 Seg.	-	-	-	-	-	-	-	-	-	-			-				
3 Seg.: prestress spring 2	0.7453	0.4793	-	0.0001	0.4759	0.4867	0.4752	-	1.0114	0.0632			88.92%				
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-			-				
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-			-				

Figure M.9

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														
Linear springs														
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx. Work red.
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25			
UB	4.5	4.5	4.5	4.5	4.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375			
3 Seg.	0.8322	0.0973	1.4333	-	-	-	0.3962	0.4665	0.426	-	-	0.3549	0.053	91.93%
4 Seg.														
3 Seg.: prestress spring 2	0.8315	0.092	0.1653	-	-	0.0145	0.3709	0.4732	0.4209	-	-	0.3408	0.0533	91.56%
3 Seg.: prestress spring 3														
4 Seg.: prestress spring 2 & 3														

Nonlinear springs														
	A	B	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx. Work red.		
LB	-2	-2	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	2	2	1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	0.0275	0.7354	-	-	0.334	0.4984	0.4452	-	-	0.1993	0.0631		88.74%	
4 Seg.														
3 Seg.: prestress spring 2	0.3476	0.3319	-	0.4554	0.3341	0.5	0.4764	-	-	0.8874	0.0541		90.80%	
3 Seg.: prestress spring 3														
4 Seg.: prestress spring 2 & 3														

Figure M.10

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														
Linear springs														
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx. Work red.
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25			
UB	4.5	4.5	4.5	4.5	4.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375			
3 Seg.	0.6743	2.5389	0.7194	-	-	-	0.3426	0.4976	0.4868	-	1.352	1.4539	0.0831	85.68%
4 Seg.														
3 Seg.: prestress spring 2	0.8287	1.0232	0.1058	-	-	0.108	0.3506	0.4988	0.4972	-	1.2021	0.0459	0.0535	91.30%
3 Seg.: prestress spring 3														
4 Seg.: prestress spring 2 & 3														

Nonlinear springs														
	A	B	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx. Work red.		
LB	-4.5	-4.5	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	4.5	4.5	1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	0.9012	0.1365	-	-	0.3853	0.4221	0.4155	-	0.9261	1.8591	0.0305		94.82%	
4 Seg.														
3 Seg.: prestress spring 2	0.3858	0.3051	-	0.8764	0.334	0.4998	0.4999	-	0.7313	1.6328	0.0515		91.32%	
3 Seg.: prestress spring 3														
4 Seg.: prestress spring 2 & 3														

Figure M.11

M.4.4. Degressive-progressive

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														
Linear springs														
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx. Work red.
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25			
UB	2.5	2.5	2.5	2.5	2.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375			
3 Seg.	0.6324	2.3421	0.8924	-	-	-	0.4998	0.4998	0.3938	-			0.0806	86.18%
4 Seg.	0.7247	2.2405	0.9827	0.0411	-	-	0.361	0.3133	0.375	0.2667			0.0792	86.95%
3 Seg.: prestress spring 2	1.6965	0.0952	0.0009	-	-	0.1367	0.4689	0.4213	0.4776	-			0.0132	98.17%
3 Seg.: prestress spring 3														
4 Seg.: prestress spring 2 & 3	1.3166	0.1279	2.005	0.6868	0.7725	0.2416	0.2643	0.3621	0.3538	0.3744			0.0173	97.40%

Nonlinear springs														
	A	B	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx. Work red.		
LB	-2	-2	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	2	2	1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	-0.2911	1.0077	-	-	0.4502	0.4988	0.4505	-	0.0649				89.00%	
4 Seg.	-0.3014	1.0991	-	-	0.3502	0.3747	0.3625	0.3563	0.0682				89.46%	
3 Seg.: prestress spring 2	-0.2036	0.9112	-	0.1818	0.334	0.4957	0.4249	-	0.0516				91.35%	
3 Seg.: prestress spring 3														
4 Seg.: prestress spring 2 & 3	-0.2507	1.0577	0.0916	0.1399	0.2511	0.3541	0.375	0.287	0.0519				92.39%	

Figure M.12

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg																		
Linear springs																		
LB	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	RMSE	Best theor. Approx.	Work red.				
UB	4.5	4.5	4.5	4.5	4.5	1	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25							
3 Seg.	3.3836	0.3294	0.4068	-	-	-	0.4868	0.4369	0.3878	-	0.8163	0.0113		98.42%				
4 Seg.																		
3 Seg.: prestress spring 2	2.1364	0.0985	0.4534	-	-	0.2666	0.3453	0.3487	0.4627	-	0.553	0.0093		98.76%				
3 Seg.: prestress spring 3																		
4 Seg.: prestress spring 2 & 3																		

Nonlinear springs																		
LB	A	B	M03	M02	I1	I2	I3	I4	theta10	RMSE	Work red.							
UB	2	2	1	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25									
3 Seg.	-0.2247	1.1196	-	-	0.3359	0.411	0.4897	-	0.9213	0.0526	91.24%							
4 Seg.																		
3 Seg.: prestress spring 2	-0.224	0.9791	-	0.3021	0.3342	0.5	0.5	-	0.6883	0.048	92.00%							
3 Seg.: prestress spring 3																		
4 Seg.: prestress spring 2 & 3																		

Figure M.13

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg															
Linear springs															
LB	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx.	Work red.
UB	4.5	4.5	4.5	4.5	4.5	1	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25				
3 Seg.	0.6283	3.2139	1.5473	-	-	-	0.4351	0.4649	0.3927	-	-	0.9429	0.0807		86.19%
4 Seg.															
3 Seg.: prestress spring 2	1.2614	0.1343	0.039	-	-	0.1847	0.3959	0.5	0.4102	-	-	1.4119	0.0109		98.43%
3 Seg.: prestress spring 3															
4 Seg.: prestress spring 2 & 3															

Nonlinear springs															
LB	A	B	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Work red.			
UB	2	2	1	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25						
3 Seg.	-0.3131	1.0406	-	-	0.4	0.4997	0.4905	-	-	0.4297	0.0645	89.04%			
4 Seg.															
3 Seg.: prestress spring 2	-0.2059	0.9154	-	0.1775	0.3341	0.498	0.4256	-	-	1.0801	0.0516	91.34%			
3 Seg.: prestress spring 3															
4 Seg.: prestress spring 2 & 3															

Figure M.14

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg															
Linear springs															
LB	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Best theor. Approx.	Work red.
UB	4.5	4.5	4.5	4.5	4.5	1	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25				
3 Seg.	0.5938	4.3689	4.3492	-	-	-	0.3563	0.4953	0.4919	-	1.352	1.8324	0.0801		86.32%
4 Seg.															
3 Seg.: prestress spring 2	0.7789	0.4116	0.4538	-	-	0.552	0.3695	0.4866	0.4953	-	0.8885	0.818	0.0537		90.89%
3 Seg.: prestress spring 3															
4 Seg.: prestress spring 2 & 3															

Nonlinear springs															
LB	A	B	M03	M02	I1	I2	I3	I4	theta10	contactan	RMSE	Work red.			
UB	4.5	4.5	1	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25						
3 Seg.	-0.2277	1.071	-	-	0.3341	0.4949	0.4453	-	1.0494	1.7399	0.0558	90.75%			
4 Seg.															
3 Seg.: prestress spring 2	-0.2406	0.9838	-	0.3708	0.334	0.4994	0.5	-	0.8642	0.2396	0.0448	92.42%			
3 Seg.: prestress spring 3															
4 Seg.: prestress spring 2 & 3															

Figure M.15

M.4.5. Laevo

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														RMSE	Best theor. Approx.	Work red.
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10					
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	2.5	2.5	2.5	2.5	2.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	1.5	0.5073	0.0003	-	-	-	0.5	0.4999	0.4927	-		0.2671	0.2755	70.94%		
4 Seg.	1.9225	1.1771	0.467	0.0001	-	-	0.36	0.375	0.375	0.3658		0.2365	0.279	75.74%		
3 Seg.: prestress spring 2	1.9362	0.0231	0.0002	-	-	0.4968	0.4278	0.5	0.5	-		0.0406	0.0437	95.85%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	0.0437	-		
4 Seg.: prestress spring 2 & 3	2.0313	0.0545	0.2988	0.0049	0.2108	0.6278	0.3307	0.3457	0.375	0.3359		0.0271	0.0207	97.44%		

Nonlinear springs														RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4	theta10						
LB	-3	-3		0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	3	3		1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	-1.1842	2.2747	-	-	-	0.4995	0.3334	0.4993	-		0.08003	-	91.38%		
4 Seg.	-1.3289	2.4358	-	-	-	0.375	0.3596	0.2503	0.2638		0.098	-	89.92%		
3 Seg.: prestress spring 2	-0.9992	2.066	-	-	0.4623	0.3549	0.3882	0.4477	-		0.0572	-	93.79%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-		-	-	-		
4 Seg.: prestress spring 2 & 3	-1.0165	2.087	-	0.2561	0.4699	0.3213	0.3177	0.323	0.335		0.0587	-	94.15%		

Figure M.16

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														RMSE	Best theor. Approx.	Work red.
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10					
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	3.5	3.5	3.5	3.5	3.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	3.3664	0.5915	0.9847	-	-	-	0.3518	0.4999	0.4995	-		1.3365	0.0037	99.66%		
4 Seg.	-	-	-	-	-	-	-	-	-	-		-	-	-		
3 Seg.: prestress spring 2	2.1753	0.003	0.0009	-	-	0.497	0.4756	0.4998	0.4926	-		0.3068	0.0164	98.47%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	-	-		
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-		-	-	-		

Nonlinear springs														RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4	theta10						
LB	-3.5	-3.5		0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	3.5	3.5		1	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
3 Seg.	-0.9527	2.0145	-	-	-	0.3545	0.4256	0.4282	-		0.9766	0.0587	93.69%		
4 Seg.	-	-	-	-	-	-	-	-	-		-	-	-		
3 Seg.: prestress spring 2	-0.958	2.0217	-	-	0.5217	0.4056	0.3925	0.4953	-		0.8973	0.0589	93.66%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-		-	-	-		
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-		-	-	-		

Figure M.17

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														RMSE	Best theor. Approx.	Work red.
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10					
LB	0	0	0	0	0	0	0	0.1	0.1	0.1	0.1					
UB	2.5	2.5	2.5	2.5	2.5	1	1	0.9	0.9	0.9	0.9					
3 Seg.	1.6398	0.1799	0.1631	-	-	-	0.8641	0.5486	0.5882	-		0.1288	0.2755	87.33%		
4 Seg.	1.5073	1.5799	0.9472	0.1644	-	-	0.6431	0.176	0.6611	0.5207		0.1863	0.279	80.81%		
3 Seg.: prestress spring 2	2.04	0.2091	0.2125	-	-	0.4658	0.4368	0.8666	0.57	-		0.0471	0.0437	95.13%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	0.0437	-		
4 Seg.: prestress spring 2 & 3	2.1317	0.3356	0.8089	0.6702	0.3449	0.5405	0.1664	0.2338	0.4922	0.6389		0.0836	0.0207	92.41%		

Nonlinear springs														RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4	theta10						
LB	-3	-3		0	0	0	0.1	0.1	0.1	0.1					
UB	3	3		1	1	1	0.9	0.9	0.9	0.9					
3 Seg.	-0.977	2.0431	-	-	-	0.8995	0.4927	0.5839	-		0.0596	-	93.60%		
4 Seg.	-1.0863	2.1614	-	-	-	0.6688	0.317	0.3643	0.3525		0.0622	-	93.51%		
3 Seg.: prestress spring 2	-1.0144	2.0813	-	-	0.5013	0.2978	0.5405	0.6288	-		0.057	-	93.81%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-		-	-	-		
4 Seg.: prestress spring 2 & 3	-1.1493	2.2374	-	0.4394	0.4558	0.1822	0.1885	0.3093	0.6705		0.0619	-	93.49%		

Figure M.18

M.4.6. Sine, 180 degree

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg														RMSE	Best theor. Approx.	Work red.
Linear springs																
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4						
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	2.5	2.5	2.5	2.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375						
3 Seg.	0.6362	2.3752	0.0636	-	-	-	0.4998	0.4816	0.3841	-		0.4486		38.47%		
4 Seg.	1.8588	0.5803	0.2827	0.0013	-	-	0.3713	0.375	0.375	0.375		0.4308		43.96%		
3 Seg.: prestress spring 2	1.5268	0	0.0001	-	-	0.369	0.5	0.5	0.5	-		0.2445		71.42%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-		-		
4 Seg.: prestress spring 2 & 3	1.8369	0.0011	0.0025	0.0002	0.2358	0.4781	0.3683	0.375	0.375	0.375		0.2321		77.26%		

Nonlinear springs														RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4							
LB	-3	-3		0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25						
UB	3	3		1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375						
3 Seg.	-1.6673	2.5479	-	-	-	0.5	0.3342	0.4915	-			0.0515		93.34%	
4 Seg.	-1.5797	2.4223	-	-	-	0.375	0.375	0.2527	0.3554			0.1125		84.44%	
3 Seg.: prestress spring 2	-1.6017	2.4953	-	-	0.661	0.3344	0.334	0.3341	-			0.0242		96.73%	
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-			-		-	
4 Seg.: prestress spring 2 & 3	-1.5903	2.4835		0.4107	0.6911	0.25	0.2501	0.25	0.2501			0.025		96.72%	

Figure M.19

Resolution: M = 90, N1 = 1000 for 3 - seg M = 15, N1 = 150, N2 = 150 for 4 - seg															RMSE	Best theor. Approx.	Work red.
Linear springs																	
	k1	k2	k3	k4	M03	M02	I1	I2	I3	I4	theta10						
LB	0	0	0	0	0	0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25						
UB	4.5	4.5	4.5	4.5	4.5	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375						
3 Seg.	3.8637	0.2544	1.3649	-	-	-	0.3502	0.4991	0.4971	-		1.2367	0.0217	97.10%			
4 Seg.	-	-	-	-	-	-	-	-	-	-		-	-	-			
3 Seg.: prestress spring 2	2.5	0.2743	1.909	-	-	0.1276	0.3346	0.4651	0.4761	-		0.9992	0.0713	90.72%			
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-	-		-	-	-			
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-	-		-	-	-			

Nonlinear springs															RMSE	Work red.
	A	B	M03	M02	I1	I2	I3	I4								
LB	-3.5	-3.5		0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25							
UB	3.5	3.5		1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375							
3 Seg.	-1.639	-2.5266	-	-	-	0.3852	0.4002	0.4306	-			0.6247	0.0183	97.49%		
4 Seg.	-	-	-	-	-	-	-	-	-			-	-	-		
3 Seg.: prestress spring 2	-1.5965	2.4891	-	-	0.9482	0.465	0.3905	0.4728	-			0.9176	0.0243	96.70%		
3 Seg.: prestress spring 3	-	-	-	-	-	-	-	-	-			-	-	-		
4 Seg.: prestress spring 2 & 3	-	-	-	-	-	-	-	-	-			-	-	-		

Figure M.20

N

MATLAB code

In the following, the required MATLAB scripts are provided. Section N.1 will include the code for the clock spring selection from the catalogue of the spring supplier. Sequentially, the MATLAB script that is used to calculate the work reduction corresponding to the prototype is given in section N.2. Section N.3 and section N.4 contain the main code for the three segment balancer and four segment balancer, respectively.

N.1. Spring selection

```
1  clc                                %clear command window
2  clear variables                    %empty workspace
3  close all                          %close all windows
4
5  %the lowest value the minimizer can attain is equal to 1:
6  %the first entry of the vector with springs
7  lb = [1,1,1,1,1,1];
8  %the highest value the minimizer can attain is equal to 13:
9  %the last entry of the vector with springs
10 ub = [13,13,13,13,13,13];
11 %six optimizers, i.e. six vacancies for clock springs
12 nvars = 6;
13 %all design variables should be integer-valued
14 intcon = [1 2 3 4 5 6];
15
16 %do the optimization "n" times
17 n = 100;
18 %preallocate a matrix to store the minimizer values
19 testvectorX = zeros(nvars,n);
20 %preallocate a matrix to store the objective function value for each run
21 testvectorY = zeros(1,n);
22
23 %do a genetic algorithm optimization "n" times
24 for i = 1:1:n
25     %[output] = ga(input)
26     %objective function "f135", see bottom of script
27     %nvars is the dimension of the objective function
28     %refer to lower (lb) and upper (ub) bounds
29     %the variables listed in intcon should be integer values
30     [xvec,fval,exitflag,output,population,scores] = ga(@f135,nvars...
31         ,[],[],[],[],lb,ub,[],intcon);
32
33     %store the minimizer solution values in the i-th column of the
34     %"testvectorX" matrix...
35     testvectorX(:,i) = transpose(xvec);
36     %...and write the objective function value to the corresponding column
37     %of the "testvectorY" vector
38     testvectorY(1,i) = fval;
39 end
40
41 %load the vector with spring stiffnesses, obtained from spring supplier
42 load Rvec
43 %store the lowest possible objective function value and call this value
44 %"testvectorYopt"
45 %save the corresponding index as well
46 [testvectorYopt,I] = min(testvectorY);
47 %the "testvectorXopt" vector contains the values of the minimizers for
48 %the lowest possible objective function value
49 testvectorXopt = testvectorX(:,I);
```

```

50 %translate the integer-valued minimizers into a corresponding spring
51 %stiffnesses by reading the entries from the spring stiffness vector
52 stiffnesses = [R(testvectorXopt(1)) R(testvectorXopt(2))...
53               R(testvectorXopt(3)) R(testvectorXopt(4))...
54               R(testvectorXopt(5)) R(testvectorXopt(6))];
55
56 %the objective function
57 function e = f315(x)
58 %the to be approximated ratio between the nett stiffness on axis 1 w.r.t.
59 %the nett stiffness on axis 2
60 A = 1.498937760758821;
61 %the to be approximated ratio between the nett stiffness on axis 3 w.r.t.
62 %the nett stiffness on axis 2
63 B = 4.847839773931474;
64 %load the vector with spring stiffnesses again
65 load Rvec %#ok<LOAD>
66
67 %calculate the objective function as a measure of a resultant
68 %deviation from the required spring stiffness ratios
69 e = sqrt((((R(x(1))+R(x(2)))/(R(x(3))+R(x(4))))-A)^2 +...
70          (((R(x(5))+R(x(6)))/(R(x(3))+R(x(4))))-B)^2);
71 end

```

N.2. Work reduction

```

1  close all                                %close all windows
2  clear variables                          %empty workspace
3
4  load Angle_lower                         %load angle data lower part of hysteresis loop
5  load Angle_upper                         %load angle data upper part of hysteresis loop
6  load Moment_lower                       %load moment data lower part of hysteresis loop
7  load Moment_upper                       %load moment data upper part of hysteresis loop
8
9  %M_obj is a fit of the expected moment-angle curve
10 %Below, the arrays "Angle_upper" and "Angle_lower" are inserted as its
11 %Argument to calculate the corresponding fit value
12 M_obj_u = ((0.00000000044283)*Angle_upper.^5) -...
13           ((0.0000000082536)*Angle_upper.^4) +...
14           ((0.00000048168)*Angle_upper.^3) - ((0.000014378)*Angle_upper.^2) +...
15           0.0011225*Angle_upper - 0.00029428;
16
17 M_obj_l = ((0.00000000044283)*Angle_lower.^5) -...
18           ((0.0000000082536)*Angle_lower.^4) +...
19           ((0.00000048168)*Angle_lower.^3) - ((0.000014378)*Angle_lower.^2) +...
20           0.0011225*Angle_lower - 0.00029428;
21
22 figure
23 hold on
24 %plot(Angle_lower,M_obj_l)
25 plot(Angle_upper,M_obj_u)
26 plot(Angle_lower,Moment_lower)
27 plot(Angle_upper,Moment_upper)
28 xlabel('Angle pendulum (deg)')
29 ylabel('Moment (Nm)')
30
31
32 %Obtain the work corresponding to the lower part of the hysteresis loop
33 Work_lower = trapz(Angle_lower(117:23479),...
34                  abs(Moment_lower(117:23479)-M_obj_l(117:23479)));
35
36 %Obtain the work corresponding to the upper part of the hysteresis loop
37 Work_upper = trapz(Angle_upper(167:33411),...
38                  abs(Moment_upper(167:33411)-M_obj_u(167:33411)));
39
40 %Obtain the work corresponding to the reference configuration
41 Work_ref = trapz(Angle_upper(167:33411),M_obj_u(167:33411))+...
42           trapz(Angle_lower(117:23479),M_obj_l(117:23479));
43
44 %Calculate the work reduction
45 Work_red = (((Work_lower+Work_upper)-Work_ref)/Work_ref)*100;

```

N.3. Three segment balancer

```

1  clc                                      %clear command window
2  clear variables                          %empty workspace
3  close all                                %close all windows
4
5  %prompt that asks for desired configurations and saves...
6  %...preference by means of variable X
7  prompt = 'Knee up (1) or knee down (0)?';
8  X = input(prompt);

```



```

9
10 %with (1) or without (0) prestress on springs
11 prestress = 0;
12 %with (1) or without (0) nonlinear springs
13 nonlinearity = 0;
14 %type objective function between quotation marks...
15 objective = "sinus";
16
17 %select manually: elbow up solution (X = 1;) or elbow down (X = 0;)
18 %X = 1;
19 M = 90; %amount of precision points
20 %amount of configurations per precision point
21 N = 1000;
22
23 %stiffness spring 1 (Nm/rad)
24 k1 = 0.03732627 + 0.03755394;
25 %stiffness spring 2 (Nm/rad)
26 k2 = 0.0217042 + 0.021610197;
27 %stiffness spring 3 (Nm/rad)
28 k3 = 0.035792513 + 0.03678242 + 0.071335373 + 0.068828657;
29
30 l1 = 0.16968851561841500; %length first segment (m)
31 l2 = 0.22863398856099401; %length second segment (m)
32 l3 = 0.23760765730766202; %length third segment (m)
33
34 %angle at which contact is enabled at spring 2 (rad)
35 contactan = 100;
36 %initial angle of segment 1 (rad)
37 theta10 = 0.664071011410003;
38 %length pendulum (m)
39 r = 0.5;
40
41 %constant mass times grav. constant (N)
42 mg = 0.029321787763455*2*2;
43
44 %angle between segment 3 and pendulum when segment 2 and 3 are aligned
45 D3 = acos((r^2 + (l2+l3)^2 - l1^2)/(2*r*(l2+l3)));
46 %angle between segment 1 and segment 2 " "
47 D2 = acos((l1^2 + (l2+l3)^2 - r^2)/(2*l1*(l2+l3)));
48
49 %angle of pendulum and first segment in case of deviating...
50 %... definitions of angles
51 phir0 = (pi/2);
52 phi10 = (pi/2) - theta10;
53 Mtheta10 = -theta10;
54
55 %if initial angle of first segment is greater than zero or equal to zero
56 if theta10 >= 0
57 %initial angle segment 2
58 theta20ku = pi/2 - real(pi - acos((l1*cos(phi10) - r*cos(phir0) + ...
59 l3*cos(log(-((l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i))-...
60 l1^2*exp(phir0*i)*exp(phi10*i) + l2^2*exp(phir0*i)*exp(phi10*i) +...
61 l3^2*exp(phir0*i)*exp(phi10*i) -...
62 r^2*exp(phir0*i)*exp(phi10*i) - 2*l2*l3*exp(phir0*i)*exp(phi10*i))*...
63 (l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) -...
64 l1^2*exp(phir0*i)*exp(phi10*i) + l2^2*exp(phir0*i)*exp(phi10*i) +...
65 l3^2*exp(phir0*i)*exp(phi10*i) -...
66 r^2*exp(phir0*i)*exp(phi10*i) + 2*l2*l3*exp(phir0*i)*...
67 exp(phi10*i)))^(1/2) - l1*r*exp(phir0*2i) - l1*r*exp(phi10*2i) +...
68 l1^2*exp(phir0*i)*exp(phi10*i) - l2^2*exp(phir0*i)*exp(phi10*i) +...
69 l3^2*exp(phir0*i)*exp(phi10*i) +...
70 r^2*exp(phir0*i)*exp(phi10*i))/(2*(l1*l3*exp(phir0*i) -...
71 l3*r*exp(phi10*i))))*i)/l2));
72 %initial angle segment 3
73 theta30ku = pi/2 - real(-log(-((l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) -...
74 l1^2*exp(phir0*i)*exp(phi10*i) +...
75 l2^2*exp(phir0*i)*exp(phi10*i) + l3^2*exp(phir0*i)*exp(phi10*i) -...
76 r^2*exp(phir0*i)*exp(phi10*i) -...
77 2*l2*l3*exp(phir0*i)*exp(phi10*i))*(l1*r*exp(phir0*2i) +...
78 l1*r*exp(phi10*2i) - l1^2*exp(phir0*i)*exp(phi10*i) +...
79 l2^2*exp(phir0*i)*exp(phi10*i) + l3^2*exp(phir0*i)*exp(phi10*i) -...
80 r^2*exp(phir0*i)*exp(phi10*i) +...
81 2*l2*l3*exp(phir0*i)*exp(phi10*i)))^(1/2) - l1*r*exp(phir0*2i) -...
82 l1*r*exp(phi10*2i) + l1^2*exp(phir0*i)*exp(phi10*i) -...
83 l2^2*exp(phir0*i)*exp(phi10*i) + l3^2*exp(phir0*i)*exp(phi10*i) +...
84 r^2*exp(phir0*i)*exp(phi10*i))/(2*(l1*l3*exp(phir0*i) -...
85 l3*r*exp(phi10*i))))*i);
86
87 %calculate the deviations in x and y of the coordinates of the compensator, respectively
88 DEV10 = l1*sin(theta10) + l2*sin(theta20ku) + l3*sin(theta30ku) - r*sin(0);
89 DEV20 = l1*cos(theta10) + l2*cos(theta20ku) + l3*cos(theta30ku) - r*cos(0);
90
91 %if any of the deviations is greater than its threshold
92 if abs(DEV10) > 10^-12 || abs(DEV20) > 10^-8

```

```

93     %other formulation initial angle segment 2
94     theta20ku = pi/2 - real(pi + acos((l1*cos(phi10) - r*cos(phir0) + ...
95         l3*cos(log(-((l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) - ...
96         l1^2*exp(phir0*1i)*exp(phi10*1i) + l2^2*exp(phir0*1i)*exp(phi10*1i) + ...
97         l3^2*exp(phir0*1i)*exp(phi10*1i) - ...
98         r^2*exp(phir0*1i)*exp(phi10*1i) - 2*l2*l3*exp(phir0*1i)*exp(phi10*1i))*...
99         (l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) - ...
100        l1^2*exp(phir0*1i)*exp(phi10*1i) + l2^2*exp(phir0*1i)*exp(phi10*1i) + ...
101        l3^2*exp(phir0*1i)*exp(phi10*1i) - ...
102        r^2*exp(phir0*1i)*exp(phi10*1i) + 2*l2*l3*exp(phir0*1i)*...
103        exp(phi10*1i)))^(1/2) - l1*r*exp(phir0*2i) - l1*r*exp(phi10*2i) + ...
104        l1^2*exp(phir0*1i)*exp(phi10*1i) - l2^2*exp(phir0*1i)*exp(phi10*1i) + ...
105        l3^2*exp(phir0*1i)*exp(phi10*1i) + ...
106        r^2*exp(phir0*1i)*exp(phi10*1i))/(2*(l1*l3*exp(phir0*1i) - ...
107        l3*r*exp(phi10*1i))))*1i)/12));
108     end
109 end
110
111 %if initial angle of first segment is smaller than zero
112 if theta10 < 0
113     %initial angle segment 2
114     theta20ku = real(asin((l3*sin(log(-(l1*r + ((l1*r - l1^2*exp(Mtheta10*1i))*...
115         exp(0*1i) + l2^2*exp(Mtheta10*1i)*exp(0*1i) + ...
116         l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) - ...
117         2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) + ...
118         l1*r*exp(Mtheta10*2i)*exp(0*2i))*(l1*r - l1^2*exp(Mtheta10*1i)*exp(0*1i) + ...
119         l2^2*exp(Mtheta10*1i)*exp(0*1i) + ...
120         l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) + ...
121         2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) + ...
122         l1*r*exp(Mtheta10*2i)*exp(0*2i))))^(1/2) - l1^2*exp(Mtheta10*1i)*exp(0*1i) + ...
123         l2^2*exp(Mtheta10*1i)*exp(0*1i) - ...
124         l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) + ...
125         l1*r*exp(Mtheta10*2i)*exp(0*2i))/(2*(l3*r*exp(Mtheta10*1i) - ...
126         l1*l3*exp(Mtheta10*2i)*exp(0*1i))))*1i + l1*sin(Mtheta10) + r*sin(0))/12));
127     %initial angle segment 3
128     theta30ku = real(-log(-(l1*r + ((l1*r - l1^2*exp(Mtheta10*1i)*exp(0*1i) + ...
129         l2^2*exp(Mtheta10*1i)*exp(0*1i) + ...
130         l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) - ...
131         2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) + ...
132         l1*r*exp(Mtheta10*2i)*exp(0*2i))*(l1*r - l1^2*exp(Mtheta10*1i)*exp(0*1i) + ...
133         l2^2*exp(Mtheta10*1i)*exp(0*1i) + ...
134         l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) + ...
135         2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) + ...
136         l1*r*exp(Mtheta10*2i)*exp(0*2i))))^(1/2) - l1^2*exp(Mtheta10*1i)*exp(0*1i) + ...
137         l2^2*exp(Mtheta10*1i)*exp(0*1i) - ...
138         l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) + ...
139         l1*r*exp(Mtheta10*2i)*exp(0*2i))/(2*(l3*r*exp(Mtheta10*1i) - ...
140         l1*l3*exp(Mtheta10*2i)*exp(0*1i))))*1i);
141     end
142
143 %confine with elbow up solutions for the initial angles for now
144 theta20 = theta20ku;
145 theta30 = theta30ku;
146
147 %preallocate all variables for better performance...
148 %...vectors
149 alpha = zeros(1,M);
150 alpha1m = zeros(1,M);
151 alpha2m = zeros(1,M);
152 alpha3m = zeros(1,M);
153 theta100 = zeros(1,M);
154 theta1ff = zeros(1,M);
155 theta1m = zeros(1,M);
156 theta2m = zeros(1,M);
157 theta3m = zeros(1,M);
158 BEGIN = zeros(1,M);
159 END = zeros(1,M);
160 STEP = zeros(1,M);
161 M1m = zeros(1,M);
162 M2m = zeros(1,M);
163 M3m = zeros(1,M);
164 Mobj = zeros(1,M);
165 phir = zeros(1,M);
166 fit = zeros(1,M);
167 Vm = zeros(1,M);
168 V1m = zeros(1,M);
169 V2m = zeros(1,M);
170 V3m = zeros(1,M);
171 Vtm = zeros(1,M);
172 F1y = zeros(1,M);
173 F1x = zeros(1,M);
174 F2y = zeros(1,M);
175 F2x = zeros(1,M);
176 F1r = zeros(1,M);

```

```

177 M1l = zeros(1,M);
178 M2l = zeros(1,M);
179 M3l = zeros(1,M);
180 phi = zeros(1,M);
181 T1b = zeros(1,M);
182 Tub = zeros(1,M);
183 Start = zeros(1,M);
184 Stop = zeros(1,M);
185 %...matrices
186 DEV1 = zeros(M,N);
187 DEV2 = zeros(M,N);
188 theta1 = zeros(M,N);
189 theta2 = zeros(M,N);
190 theta3 = zeros(M,N);
191 theta2kd = zeros(M,N);
192 theta3kd = zeros(M,N);
193 theta2ku = zeros(M,N);
194 theta3ku = zeros(M,N);
195 phi1 = zeros(M,N);
196 Mtheta1 = zeros(M,N);
197 theta1P = zeros(M,N);
198 d = zeros(M,N);
199 alpha1 = zeros(M,N);
200 alpha2 = zeros(M,N);
201 alpha3 = zeros(M,N);
202 V = zeros(M,N);
203 V1 = zeros(M,N);
204 V2 = zeros(M,N);
205 V3 = zeros(M,N);
206 M1 = zeros(M,N);
207 M2 = zeros(M,N);
208 M3 = zeros(M,N);
209 x1 = zeros(M,N);
210 x2 = zeros(M,N);
211 x3 = zeros(M,N);
212 y1 = zeros(M,N);
213 y2 = zeros(M,N);
214 y3 = zeros(M,N);
215 F1xt = zeros(M,N);
216 F1yt = zeros(M,N);
217 F2xt = zeros(M,N);
218 F2yt = zeros(M,N);
219 M2lt = zeros(M,N);
220 M3lt = zeros(M,N);
221
222 Count = 0; %error counter
223 Count2 = 0; %second error counter
224
225 %activation = 0 (only spring 1 active) or activation = 1 (all springs active)
226 activation = 0;
227 theta1ln = pi/2;
228 alphaln = pi/2;
229
230 %start a loop throughout all precision points
231 for j = 1:1:M
232 %divide the 90 deg range of motion into equally sized segments
233 alpha(j) = (pi/2)*(j/M);
234
235 %lowerbound of theta1 such that precision point (j) is still reached
236 theta100(j) = alpha(j) - pi + D2 + D3;
237 %upperbound of theta1 such that precision point (j) is still reached
238 theta1ff(j) = alpha(j) - (-pi + D2 + D3);
239
240 %if arm,consisting of segment 2 and segment 3,can not be fully stretched...
241 %...segment 1 should be given a full rotation for sweep as no physical
242 %lower and upperbound exist
243
244 %check whether segment 2 and 3 can be aligned (stretched arm)
245 if (l1+r) <= (l2+l3)
246 %alternative formulation lowerbound of theta1 if arm cannot be
247 %stretched
248 theta100(j) = alpha(j) - pi;
249 %alternative formulation upperbound of theta1 if arm cannot be
250 %stretched
251 theta1ff(j) = alpha(j) + pi;
252 end
253
254 BEGIN(j) = theta100(j); %begin interval
255 END(j) = theta1ff(j); %end interval
256
257 STEP(j) = (END(j)-BEGIN(j))/N; %stepsize
258
259 %loop for segment 1 angle sweep
260 for i = 1:1:N

```

```

261
262 %increase angle with steps equal to the stepsize STEP(j)
263 if prestress == 0
264     theta1(j,i) = BEGIN(j) + STEP(j)*i;
265 end
266
267
268 if prestress == 1
269     %springs 2 and 3 not involved yet
270     if activation == 0
271         %the angle of the first segment when only spring 1 is active
272         theta1(j,i) = alpha(j)+theta10;
273     end
274
275     %springs 2 and 3 engaged
276     if activation == 1
277         %increase angle with steps equal to the stepsize STEP(j)
278         theta1(j,i) = BEGIN(j) + STEP(j)*i;
279     end
280 end
281
282 %the following holds when contact is not engaged
283 if alpha2(j,i) < contactan
284
285 %the expressions within this loop are valid for theta1 < 0
286 if theta1(j,i) < 0
287     %Mtheta1(j,i) is used instead of theta1(j,i) for practical reasons
288     Mtheta1(j,i) = - theta1(j,i);
289
290     %formulation for angle segment 3: elbow up
291     theta3ku(j,i) = real(-log(-(11*r + ((11*r - 11^2*exp(Mtheta1(j,i)*1i)*...
292         exp(alpha(j)*1i) + 12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
293         13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
294         exp(alpha(j)*1i) - 2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
295         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
296         (11*r - 11^2*exp(Mtheta1(j,i)*1i)*...
297         exp(alpha(j)*1i) + 12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
298         13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
299         exp(alpha(j)*1i) + 2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
300         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
301         11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
302         12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - 13^2*exp(Mtheta1(j,i)*1i)*...
303         exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
304         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(13*r*exp(Mtheta1(j,i)*1i) - ...
305         11*13*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i)))*1i);
306
307     %formulation for angle segment 2: elbow up
308     theta2ku(j,i) = real(asin((13*sin(log(-(11*r + ...
309         ((11*r - 11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
310         12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
311         13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - ...
312         r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - ...
313         2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
314         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
315         (11*r - 11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
316         12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
317         13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
318         exp(alpha(j)*1i) + 2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
319         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
320         11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
321         12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - 13^2*exp(Mtheta1(j,i)*1i)*...
322         exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
323         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(13*r*exp(Mtheta1(j,i)*1i) - ...
324         11*13*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i)))*1i) + ...
325         11*sin(Mtheta1(j,i) + r*sin(alpha(j))/12));
326
327     %formulation for angle segment 3: elbow down
328     theta3kd(j,i) = real(-log((- 11*r + ((11*r - 11^2*exp(Mtheta1(j,i)*1i)*...
329         exp(alpha(j)*1i) + 12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
330         13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - ...
331         r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - 2*12*13*exp(Mtheta1(j,i)*1i)*...
332         exp(alpha(j)*1i) + ...
333         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
334         (11*r - 11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
335         12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 13^2*exp(Mtheta1(j,i)*1i)*...
336         exp(alpha(j)*1i) - ...
337         r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
338         2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + ...
339         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) + ...
340         11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - ...
341         12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 13^2*exp(Mtheta1(j,i)*1i)*...
342         exp(alpha(j)*1i) + r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - ...
343         11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
344         (2*(13*r*exp(Mtheta1(j,i)*1i) - ...
345         11*13*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i)))*1i);

```

```

345 %formulation for angle segment 2: elbow down
346 theta2kd(j,i) = pi + real(- asin((l3*sin(log((- l1*r +...
347 ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
348 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
349 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
350 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
351 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
352 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
353 (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
354 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
355 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
356 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
357 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
358 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) +...
359 l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
360 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
361 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
362 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
363 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
364 (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
365 l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))) *1i) +...
366 l1*sin(Mtheta1(j,i) + r*sin(alpha(j)))/l2));
367
368 %select the angles for segment 2 and 3 corresponding to elbow up if X = 1 is selected
369 if X == 1
370     theta2(j,i) = theta2ku(j,i);
371     theta3(j,i) = theta3ku(j,i);
372 end
373
374 %select the angles for segment 2 and 3 corresponding to elbow down if X = 0 is selected
375 if X == 0
376     theta2(j,i) = theta2kd(j,i);
377     theta3(j,i) = theta3kd(j,i);
378 end
379
380 %calculate the deviations in x and y of the coordinates of the compensator, respectively
381 DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
382     r*sin(alpha(j));
383 DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
384     r*cos(alpha(j));
385
386 %if the absolute value of any of these deviations transcends a certain threshold,
387 %then use alternative formulations for theta2
388 if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
389     if X == 1 %elbow up
390         %other formulation for angle segment 2
391         theta2(j,i) = pi + real( - asin((l3*sin(log(-(l1*r +...
392 ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
393 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
394 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
395 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
396 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
397 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
398 (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
399 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
400 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
401 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
402 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
403 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
404 l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
405 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
406 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
407 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
408 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
409 (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
410 exp(alpha(j)*1i))) *1i) +...
411 l1*sin(Mtheta1(j,i) + r*sin(alpha(j)))/l2));
412
413 %even other formulation for angle segment 2 if angle is larger than 180 deg
414 if theta2(j,i) > pi
415     theta2(j,i) = -pi + real( - asin((l3*sin(log(-(l1*r +...
416 ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
417 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
418 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
419 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
420 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
421 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
422 (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
423 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
424 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
425 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
426 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
427 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
428 l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...

```

```

429             l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
430             l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
431             r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
432             l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
433             (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
434             l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i)))*1i) +...
435             l1*sin(Mtheta1(j,i) + r*sin(alpha(j)))/l2));
436         end
437     end
438
439     if X == 0 %elbow down
440     %other formulation for angle segment 2
441     theta2(j,i) = real(asin((l3*sin(log((- l1*r +...
442     ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
443     l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
444     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
445     r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
446     2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
447     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
448     (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
449     l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
450     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
451     r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
452     2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
453     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) +...
454     l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
455     l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
456     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
457     r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
458     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
459     (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
460     exp(alpha(j)*1i)))*1i) + l1*sin(Mtheta1(j,i) + r*sin(alpha(j)))/l2));
461     end
462 end
463 end
464
465 %the expressions within this loop are valid for theta1 >= 0
466 if theta1(j,i) >= 0
467     %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
468     phir(j) = (pi/2) - alpha(j);
469     %angle of segment 1 with respect to positive x-axis (anti-clockwise positive)
470     phil(j,i) = (pi/2) - theta1(j,i);
471
472     %formulation for angle segment 3: elbow up
473     theta3ku(j,i) = pi/2 - real(-log(-((l1*r*exp(phir(j)*2i) +...
474     l1*r*exp(phil(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
475     l2^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
476     l3^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
477     r^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
478     2*l2*l3*exp(phir(j)*1i)*exp(phil(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
479     l1*r*exp(phil(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
480     l2^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
481     l3^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
482     r^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
483     2*l2*l3*exp(phir(j)*1i)*exp(phil(j,i)*1i)))^(1/2) -...
484     l1*r*exp(phir(j)*2i) - l1*r*exp(phil(j,i)*2i) +...
485     l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
486     l2^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
487     l3^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
488     r^2*exp(phir(j)*1i)*exp(phil(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
489     l3*r*exp(phil(j,i)*1i)))*1i);
490
491     %formulation for angle segment 2: elbow up
492     theta2ku(j,i) = pi/2 - real(pi - acos((l1*cos(phil(j,i)) - r*cos(phir(j)) +...
493     l3*cos(log(-((l1*r*exp(phir(j)*2i) + l1*r*exp(phil(j,i)*2i) -...
494     l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
495     exp(phil(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
496     r^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
497     2*l2*l3*exp(phir(j)*1i)*exp(phil(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
498     l1*r*exp(phil(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
499     l2^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
500     l3^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
501     r^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
502     2*l2*l3*exp(phir(j)*1i)*exp(phil(j,i)*1i)))^(1/2) - l1*r*exp(phir(j)*2i) -...
503     l1*r*exp(phil(j,i)*2i) + l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) -...
504     l2^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
505     l3^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
506     r^2*exp(phir(j)*1i)*exp(phil(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
507     l3*r*exp(phil(j,i)*1i)))*1i)/l2));
508
509     %formulation for angle segment 3: elbow down
510     theta3kd(j,i) = pi/2 - real(-log(-((l1*r*exp(phir(j)*2i) +...
511     l1*r*exp(phil(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...
512     l2^2*exp(phir(j)*1i)*exp(phil(j,i)*1i) +...

```

```

513     13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
514     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
515     2*12*13*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(11*r*exp(phir(j)*2i) +...
516     11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
517     12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
518     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
519     2*12*13*exp(phir(j)*1i)*exp(phi1(j,i)*1i))^(1/2) +...
520     11*r*exp(phir(j)*2i) + 11*r*exp(phi1(j,i)*2i) -...
521     11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
522     12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
523     13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
524     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(11*13*exp(phir(j)*1i) -...
525     13*r*exp(phi1(j,i)*1i))))*1i);
526
527 %formulation for angle segment 2: elbow down
528 theta2kd(j,i) = 2*pi + pi/2 - real(pi + acos((11*cos(phi1(j,i)) -...
529     r*cos(phir(j)) + 13*cos(log(((11*r*exp(phir(j)*2i) +...
530     11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
531     12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
532     13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
533     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*...
534     exp(phi1(j,i)*1i))*(11*r*exp(phir(j)*2i) +...
535     11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
536     12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
537     13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
538     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
539     2*12*13*exp(phir(j)*1i)*exp(phi1(j,i)*1i))^(1/2) + 11*r*exp(phir(j)*2i) +...
540     11*r*exp(phi1(j,i)*2i) -...
541     11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
542     12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
543     13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
544     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(11*13*exp(phir(j)*1i) -...
545     13*r*exp(phi1(j,i)*1i))))*1i)/12);
546
547 %select the angles of segment 2 and 3 corresponding to elbow up if X = 1 is selected
548 if X == 1
549     theta2(j,i) = theta2ku(j,i);
550     theta3(j,i) = theta3ku(j,i);
551 end
552
553 %select the angles of segment 2 and 3 corresponding to elbow down if X = 0 is selected
554 if X == 0
555     theta2(j,i) = theta2kd(j,i);
556     theta3(j,i) = theta3kd(j,i);
557 end
558
559 %calculate the deviations in x and y of the coordinates of the compensator, respectively
560 DEV1(j,i) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i)) + 13*sin(theta3(j,i)) -...
561     r*sin(alpha(j));
562 DEV2(j,i) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i)) + 13*cos(theta3(j,i)) -...
563     r*cos(alpha(j));
564
565 %if the absolute value of any of these deviations transcends a certain threshold,
566 %then use alternative formulations for theta2
567 if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-8
568     if X == 1 %elbow up
569         %other formulation for angle segment 2
570         theta2(j,i) = 2*pi + pi/2 - real(pi + acos((11*cos(phi1(j,i)) -...
571             r*cos(phir(j)) + 13*cos(log(-(((11*r*exp(phir(j)*2i) +...
572             11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
573             12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
574             13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*...
575             exp(phi1(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
576             (11*r*exp(phir(j)*2i) + 11*r*exp(phi1(j,i)*2i) -...
577             11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 12^2*exp(phir(j)*1i)*...
578             exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
579             r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*...
580             exp(phi1(j,i)*1i))^(1/2) - 11*r*exp(phir(j)*2i) -...
581             11*r*exp(phi1(j,i)*2i) + 11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
582             12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
583             13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + r^2*exp(phir(j)*1i)*...
584             exp(phi1(j,i)*1i))/(2*(11*13*exp(phir(j)*1i) -...
585             13*r*exp(phi1(j,i)*1i))))*1i)/12);
586
587 %even other formulation for angle segment 2 if angle is larger
588 %than 180 deg
589 if theta2(j,i) > pi
590     theta2(j,i) = pi/2 - real(pi + acos((11*cos(phi1(j,i)) -...
591         r*cos(phir(j)) + 13*cos(log(-(((11*r*exp(phir(j)*2i) +...
592         11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*...
593         exp(phi1(j,i)*1i) + 12^2*exp(phir(j)*1i)*...
594         exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
595         exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*...
596         exp(phi1(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*...

```

```

597         exp(phi1(j,i)*1i))*(11*r*exp(phir(j)*2i) + ...
598         11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*...
599         exp(phi1(j,i)*1i) + 12^2*exp(phir(j)*1i)*...
600         exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
601         exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*...
602         exp(phi1(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*...
603         exp(phi1(j,i)*1i))^(1/2) - 11*r*exp(phir(j)*2i) - ...
604         11*r*exp(phi1(j,i)*2i) + 11^2*exp(phir(j)*1i)*...
605         exp(phi1(j,i)*1i) - 12^2*exp(phir(j)*1i)*...
606         exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
607         exp(phi1(j,i)*1i) + r^2*exp(phir(j)*1i)*...
608         exp(phi1(j,i)*1i))/(2*(11*13*exp(phir(j)*1i) - ...
609         13*r*exp(phi1(j,i)*1i))) * 1i)/12);
610     end
611 end
612
613 if X == 0 %elbow down
614 %other formulation for angle segment 2
615     theta2(j,i) = pi/2 - real(pi - acos((11*cos(phi1(j,i)) - ...
616     r*cos(phir(j)) + 13*cos(log(((11*r*exp(phir(j)*2i) + ...
617     11*r*exp(phi1(j,i)*2i) - 11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + ...
618     12^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
619     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - ...
620     2*12*13*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
621     (11*r*exp(phir(j)*2i) + 11*r*exp(phi1(j,i)*2i) - ...
622     11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 12^2*exp(phir(j)*1i)*...
623     exp(phi1(j,i)*1i) + 13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - ...
624     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + ...
625     2*12*13*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) + ...
626     11*r*exp(phir(j)*2i) + 11*r*exp(phi1(j,i)*2i) - ...
627     11^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 12^2*exp(phir(j)*1i)*...
628     exp(phi1(j,i)*1i) - 13^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - ...
629     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(11*13*exp(phir(j)*1i) - ...
630     13*r*exp(phi1(j,i)*1i))) * 1i)/12);
631 end
632 end
633 end
634
635 %in the case of a horizontally positioned segment 1, MATLAB solve() has
636 %troubles finding a solution... Therefore, perturb by small amount to solve
637 if theta1(j,i) == pi/2
638     theta1(j,i) = pi/2 + STEP(j);
639 end
640
641 %the expressions within this loop are valid for theta1 > pi/2
642 if theta1(j,i) > pi/2
643     %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
644     phir(j) = (pi/2) - alpha(j);
645     %angle of segment 1 with respect to positive x-axis (clockwise positive)
646     theta1P(j,i) = theta1(j,i) - pi/2;
647
648 %formulation for angle segment 3: elbow up
649     theta3ku(j,i) = real(-log(-(11*r + ((11*r - 11^2*exp(phir(j)*1i)*...
650     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
651     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
652     exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
653     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i)*...
654     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
655     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
656     exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
657     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - 11^2*exp(phir(j)*1i)*...
658     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - ...
659     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
660     exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
661     (2*(11*13*exp(phir(j)*1i)*1i - ...
662     13*r*exp(phir(j)*2i)*exp(theta1P(j,i)*1i)*1i)));
663
664 %formulation for angle segment 2: elbow up
665     theta2ku(j,i) = real(asin((13*sin(log(-(11*r + ((11*r - 11^2*exp(phir(j)*1i)*...
666     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
667     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
668     exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
669     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i)*...
670     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
671     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
672     exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + ...
673     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - 11^2*exp(phir(j)*1i)*...
674     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - ...
675     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
676     exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
677     (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
678     exp(theta1P(j,i)*1i)*1i) - 11*cos(theta1P(j,i)) + ...
679     r*cos(phir(j)))/12);
680

```



```

681 %formulation for angle segment 3: elbow down
682 theta3kd(j,i) = real(-log((- 11*r + ((11*r - 11^2*exp(phir(j)*1i))*...
683     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
684     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
685     exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
686     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i)*...
687     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
688     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
689     exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
690     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))^(1/2) + 11^2*exp(phir(j)*1i)*...
691     exp(theta1P(j,i)*1i) - 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
692     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
693     exp(theta1P(j,i)*1i) - 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
694     (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
695     exp(theta1P(j,i)*1i)*1i));
696
697 %formulation for angle segment 2: elbow down
698 theta2kd(j,i) = pi +...
699     real( - asin((13*sin(log((- 11*r + ((11*r - 11^2*exp(phir(j)*1i))*...
700     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
701     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
702     exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
703     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i)*...
704     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
705     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
706     exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
707     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))^(1/2) +...
708     11^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 12^2*exp(phir(j)*1i)*...
709     exp(theta1P(j,i)*1i) + 13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
710     r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 11*r*exp(phir(j)*2i)*...
711     exp(theta1P(j,i)*2i))/(2*(11*13*exp(phir(j)*1i)*1i -...
712     13*r*exp(phir(j)*2i)*exp(theta1P(j,i)*1i)*1i)))*1i) -...
713     11*cos(theta1P(j,i)) + r*cos(phir(j))/12);
714
715 %select the angles of the second and third segment corresponding to elbow/ up...
716 %...if X = 1 is selected
717 if X == 1
718     theta2(j,i) = theta2ku(j,i);
719     theta3(j,i) = theta3ku(j,i);
720 end
721
722 %select the angles of the second and third segment corresponding to elbow/ down...
723 %...if X = 0 is selected
724 if X == 0
725     theta2(j,i) = theta2kd(j,i);
726     theta3(j,i) = theta3kd(j,i);
727 end
728
729 %calculate the deviations in x and y of the coordinates of the compensator, respectively
730 DEV1(j,i) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i)) + 13*sin(theta3(j,i)) -...
731     r*sin(alpha(j));
732 DEV2(j,i) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i)) + 13*cos(theta3(j,i)) -...
733     r*cos(alpha(j));
734
735 %if the absolute value of any of these deviations transcends a certain threshold,
736 %then use alternative formulations for theta2
737 if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
738     if X == 1 %elbow up
739         %other formulation for angle segment 2
740         theta2(j,i) = pi + real( - asin((13*sin(log((- (11*r +...
741         ((11*r - 11^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
742         12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
743         exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
744         2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*...
745         exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i)*...
746         exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
747         13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
748         exp(theta1P(j,i)*1i) +...
749         2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
750         11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))^(1/2) -...
751         11^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
752         12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 13^2*exp(phir(j)*1i)*...
753         exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
754         11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
755         (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
756         exp(theta1P(j,i)*1i)*1i))*1i) -...
757         11*cos(theta1P(j,i)) + r*cos(phir(j))/12);
758     end
759
760     if X == 0 %elbow down
761         %other formulation for angle segment 2
762         theta2(j,i) = real(asin((13*sin(log((- 11*r +...
763         ((11*r - 11^2*exp(phir(j)*1i))*...
764         exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...

```

```

765         13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
766         r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*...
767         exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*...
768         (11*r - 11^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
769         12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
770         exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
771         2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
772         11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))^(1/2) +...
773         11^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 12^2*exp(phir(j)*1i)*...
774         exp(theta1P(j,i)*1i) + 13^2*exp(phir(j)*1i)*...
775         exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
776         exp(theta1P(j,i)*1i) - 11*r*exp(phir(j)*2i)*...
777         exp(theta1P(j,i)*2i))/(2*(11*13*exp(phir(j)*1i)*1i -...
778         13*r*exp(phir(j)*2i)*exp(theta1P(j,i)*1i)*1i)))*1i) -...
779         11*cos(theta1P(j,i)) + r*cos(phir(j))/12);
780     end
781 end
782 end
783
784 end
785
786 %if the current angle of deformation of the second spring is larger than...
787 %...the contact angle - while the angle of the previous posture is not -...
788 %...contact is just engaged
789 if i > 1 && (alpha2(j,i) > contactan) && (alpha2(j,i-1) < contactan)
790     %save angle of segment 1 corresponding to contact
791     theta1ln = theta1(j,i);
792     %save angle of pendulum corresponding to contact
793     alpha1n = alpha(j);
794 end
795
796 %initial relative angle of segment 2
797 alpha20 = theta20 - theta10;
798 %angle of rotation torsion spring 2
799 alpha2(j,i) = theta2(j,i) - theta1(j,i) - alpha20;
800
801 %if the angle of deformation of the second spring is larger than the...
802 %...contact angle
803 if alpha2(j,i) > contactan
804     %retrieve angle of segment 1
805     theta1(j,i) = theta1ln + (alpha(j)-alpha1n);
806
807 %the expressions within this loop are valid for theta1 < 0
808 if theta1(j,i) < 0
809     %Mtheta1(j,i) is used instead of theta1(j,i) for practical reasons
810     Mtheta1(j,i) = - theta1(j,i);
811
812 %formulation for angle third segment: elbow up
813 theta3ku(j,i) = real(-log(-(11*r + ((11*r - 11^2*exp(Mtheta1(j,i)*1i)*...
814     exp(alpha(j)*1i) + 12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
815     13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
816     exp(alpha(j)*1i) - 2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
817     11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
818     (11*r - 11^2*exp(Mtheta1(j,i)*1i)*...
819     exp(alpha(j)*1i) + 12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
820     13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
821     exp(alpha(j)*1i) + 2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
822     11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))^(1/2) -...
823     11^2*exp(Mtheta1(j,i)*1i)*...
824     exp(alpha(j)*1i) + 12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
825     13^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
826     r^2*exp(Mtheta1(j,i)*1i)*...
827     exp(alpha(j)*1i) + 11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
828     (2*(13*r*exp(Mtheta1(j,i)*1i) -...
829     11*13*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i);
830
831 %formulation for angle second segment: elbow up
832 theta2ku(j,i) = real(asin((13*sin(log(-(11*r +...
833     ((11*r - 11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
834     12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 13^2*exp(Mtheta1(j,i)*1i)*...
835     exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
836     2*12*13*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
837     11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
838     (11*r - 11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
839     12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 13^2*exp(Mtheta1(j,i)*1i)*...
840     exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 2*12*13*...
841     exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 11*r*exp(Mtheta1(j,i)*2i)*...
842     exp(alpha(j)*2i))^(1/2) - 11^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
843     12^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - 13^2*exp(Mtheta1(j,i)*1i)*...
844     exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
845     11*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(13*r*exp(Mtheta1(j,i)*1i) -...
846     11*13*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) + 11*sin(Mtheta1(j,i)) +...
847     r*sin(alpha(j))/12);
848

```

```

849 %formulation for angle third segment: elbow down
850 theta3kd(j,i) = real(-log((- l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
851     exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
852     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
853     exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
854     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
855     (l1*r - l1^2*exp(Mtheta1(j,i)*1i))*...
856     exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
857     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
858     exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
859     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))^(1/2) +...
860     l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
861     l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*...
862     exp(alpha(j)*1i) + r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
863     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i) -...
864     l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i)))^1i);
865
866 %formulation for angle second segment: elbow down
867 theta2kd(j,i) = pi +...
868     real(- asin((l3*sin(log((- l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
869     exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
870     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
871     exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
872     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
873     (l1*r - l1^2*exp(Mtheta1(j,i)*1i))*...
874     exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
875     l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
876     exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
877     l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))^(1/2) +...
878     l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l2^2*exp(Mtheta1(j,i)*1i)*...
879     exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
880     r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l1*r*exp(Mtheta1(j,i)*2i)*...
881     exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i) -...
882     l1*l3*exp(Mtheta1(j,i)*2i))*...
883     exp(alpha(j)*1i)))^1i + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j))/12));
884
885 %select the angles for the second and third segment corresponding to elbow up...
886 %...if X = 1 is selected
887 if X == 1
888     theta2(j,i) = theta2ku(j,i);
889     theta3(j,i) = theta3ku(j,i);
890 end
891
892 %select the angles for the second and third segment corresponding to elbow down...
893 %...if X = 0 is selected
894 if X == 0
895     theta2(j,i) = theta2kd(j,i);
896     theta3(j,i) = theta3kd(j,i);
897 end
898
899 %calculate the deviations in x and y of the coordinates of the compensator, respectively
900 DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
901     r*sin(alpha(j));
902 DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
903     r*cos(alpha(j));
904
905 %if the absolute value of any of these deviations transcends a certain threshold,
906 %then use alternative formulations for theta2
907 if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
908     if X == 1 %elbow up
909         %other formulation for angle segment 2
910         theta2(j,i) = pi + real(- asin((l3*sin(log(-(l1*r +...
911             ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
912             l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
913             l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
914             r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
915             2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
916             l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
917             (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
918             l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
919             l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
920             r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
921             2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
922             l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))^(1/2) -...
923             l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
924             l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
925             l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
926             r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
927             l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
928             (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
929             exp(alpha(j)*1i)))^1i + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j))/12));
930
931 %even other formulation for angle segment 2 if angle is
932 %larger than 180 deg

```

```

933         if theta2(j,i) > pi
934             theta2(j,i) = -pi + real( - asin((l3*sin(log(-(l1*r +...
935                 ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
936                 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
937                 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
938                 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
939                 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
940                 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))...
941                 (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
942                 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
943                 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
944                 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
945                 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
946                 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))...
947                 (1/2) - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
948                 l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
949                 l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
950                 r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
951                 l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
952                 (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
953                 l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) +...
954                 l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/12));
955         end
956     end
957
958     if X == 0 %elbow down
959     %other formulation for angle segment 2
960     theta2(j,i) = real(asin((l3*sin(log(-(l1*r +...
961         ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
962         l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
963         l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
964         r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
965         2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
966         l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))*...
967         (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
968         l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
969         l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
970         r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
971         2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
972         l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))^(1/2) +...
973         l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
974         l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
975         l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
976         r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
977         l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
978         (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
979         exp(alpha(j)*1i))))*1i) + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/12));
980     end
981 end
982 end
983
984 %the expressions within this loop are valid for theta1 >= 0
985 if theta1(j,i) >= 0
986     %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
987     phir(j) = (pi/2) - alpha(j);
988     %angle of segment 1 with respect to positive x-axis (anti-clockwise positive)
989     phil(j,i) = (pi/2) - theta1(j,i);
990
991     %formulation for angle third segment: elbow up
992     theta3ku(j,i) = pi/2 - real(-log(-((l1*r*exp(phir(j)*2i) +...
993         l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
994         l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
995         exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
996         2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
997         l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
998         l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
999         exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1000         2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))^(1/2) -...
1001         l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
1002         l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
1003         exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1004         r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
1005         l3*r*exp(phi1(j,i)*1i))))*1i);
1006
1007     %formulation for angle second segment: elbow up
1008     theta2ku(j,i) = pi/2 - real(pi - acos((l1*cos(phi1(j,i)) - r*cos(phir(j)) +...
1009         l3*cos(log(-((l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
1010         l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
1011         exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1012         r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
1013         exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
1014         l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
1015         exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1016         r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...

```

```

1017     exp(phi1(j,i)*1i))^(1/2) - l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
1018     l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
1019     exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1020     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
1021     l3*r*exp(phi1(j,i)*1i)))*/12);
1022
1023 %formulation for angle third segment: elbow down
1024 theta3kd(j,i) = pi/2 - real(-log(((l1*r*exp(phir(j)*2i) +...
1025     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1026     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1027     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1028     2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
1029     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1030     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1031     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1032     2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))^(1/2) + l1*r*exp(phir(j)*2i) +...
1033     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1034     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
1035     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
1036     (2*(l1*l3*exp(phir(j)*1i) - l3*r*exp(phi1(j,i)*1i)))*1i);
1037
1038 %formulation for angle second segment: elbow down
1039 theta2kd(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
1040     r*cos(phir(j)) + l3*cos(log(((l1*r*exp(phir(j)*2i) +...
1041     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1042     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1043     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1044     2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
1045     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1046     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1047     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1048     2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))^(1/2) + l1*r*exp(phir(j)*2i) +...
1049     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1050     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
1051     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
1052     (2*(l1*l3*exp(phir(j)*1i) - l3*r*exp(phi1(j,i)*1i)))*1i)/12);
1053
1054 %select the angles of the second and third segment corresponding to elbow up...
1055 %...if X = 1 is selected
1056 if X == 1
1057     theta2(j,i) = theta2ku(j,i);
1058     theta3(j,i) = theta3ku(j,i);
1059 end
1060
1061 %select the angles of the second and third segment corresponding to elbow down...
1062 %...if X = 0 is selected
1063 if X == 0
1064     theta2(j,i) = theta2kd(j,i);
1065     theta3(j,i) = theta3kd(j,i);
1066 end
1067
1068 %calculate the deviations in x and y of the coordinates of the compensator, respectively
1069 DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
1070     r*sin(alpha(j));
1071 DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
1072     r*cos(alpha(j));
1073
1074 %if the absolute value of any of these deviations transcends a certain threshold,
1075 %then use alternative formulations for theta2
1076 if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-8
1077     %display("Alternative formulation for pos theta1 active")
1078     if X == 1 %elbow up
1079         %other formulation for angle segment 2
1080         theta2(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
1081             r*cos(phir(j)) + l3*cos(log(-((l1*r*exp(phir(j)*2i) +...
1082             l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1083             l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1084             exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1085             2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
1086             l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1087             l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1088             exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1089             2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))^(1/2) -...
1090             l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
1091             l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
1092             exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1093             r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
1094             l3*r*exp(phi1(j,i)*1i)))*1i)/12);
1095
1096 %even other formulation for angle segment 2 if angle is
1097 %larger than 180 deg
1098 if theta2(j,i) > pi
1099     theta2(j,i) = pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
1100         r*cos(phir(j)) + l3*cos(log(-((l1*r*exp(phir(j)*2i) +...

```

```

1101         l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*...
1102         exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1103         l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1104         r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1105         2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
1106         (l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
1107         l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1108         l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1109         l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1110         r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1111         2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) -...
1112         l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
1113         l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1114         l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1115         l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1116         r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
1117         (2*(l1*l3*exp(phir(j)*1i) -...
1118         l3*r*exp(phi1(j,i)*1i)))*1i)/l2));
1119     end
1120 end
1121
1122 if X == 0 %elbow down
1123 %other formulation for angle segment 2
1124 theta2(j,i) = pi/2 - real(pi - acos((l1*cos(phi1(j,i)) -...
1125     r*cos(phir(j)) + l3*cos(log(((l1*r*exp(phir(j)*2i) +...
1126     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1127     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1128     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1129     2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
1130     (l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
1131     l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
1132     exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
1133     r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...
1134     exp(phi1(j,i)*1i)))^(1/2) + l1*r*exp(phir(j)*2i) +...
1135     l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
1136     l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
1137     exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
1138     (2*(l1*l3*exp(phir(j)*1i) - l3*r*exp(phi1(j,i)*1i)))*1i)/l2));
1139 end
1140 end
1141 end
1142
1143 %in the case of a horizontally positioned segment 1, MATLAB solve() has
1144 %troubles finding a solution... Therefore, perturb by small amount to solve
1145 if theta1(j,i) == pi/2
1146     theta1(j,i) = pi/2 + STEP(j);
1147 end
1148
1149 %the expressions within this loop are valid for theta1 > pi/2
1150 if theta1(j,i) > pi/2
1151     %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
1152     phir(j) = (pi/2) - alpha(j);
1153     %angle of segment 1 with respect to positive x-axis (clockwise positive)
1154     theta1P(j,i) = theta1(j,i) - pi/2;
1155
1156     %formulation for angle of third segment: elbow up
1157     theta3ku(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
1158     exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1159     l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
1160     r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
1161     exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*...
1162     (l1*r - l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1163     l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
1164     exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1165     2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1166     l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - l1^2*exp(phir(j)*1i)*...
1167     exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
1168     l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1169     exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
1170     (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
1171     exp(theta1P(j,i)*1i)*1i))*1i);
1172
1173     %formulation for angle of second segment: elbow up
1174     theta2ku(j,i) = real(asin((l3*sin(log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
1175     exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1176     l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1177     exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1178     l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
1179     exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1180     l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1181     exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1182     l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - l1^2*exp(phir(j)*1i)*...
1183     exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
1184     l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...

```

```

1185     exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
1186     (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
1187     exp(theta1P(j,i)*1i)*1i)))*1i) - 11*cos(theta1P(j,i)) + r*cos(phir(j)))/12));
1188
1189 %formulation for angle of third segment: elbow down
1190 theta3kd(j,i) = real(-log((- 11*r + ((11*r - 11^2*exp(phir(j)*1i))*...
1191     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1192     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1193     exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1194     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i))*...
1195     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1196     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1197     exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1198     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))^(1/2) + 11^2*exp(phir(j)*1i))*...
1199     exp(theta1P(j,i)*1i) - 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1200     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
1201     exp(theta1P(j,i)*1i) - 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
1202     (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
1203     exp(theta1P(j,i)*1i)*1i))*1i);
1204
1205 %formulation for angle of second segment: elbow down
1206 theta2kd(j,i) = pi +...
1207     real(- asin((13*sin(log((- 11*r + ((11*r - 11^2*exp(phir(j)*1i))*...
1208     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1209     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1210     exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1211     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(11*r - 11^2*exp(phir(j)*1i))*...
1212     exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1213     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1214     exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1215     11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))^(1/2) + 11^2*exp(phir(j)*1i))*...
1216     exp(theta1P(j,i)*1i) - 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1217     13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
1218     exp(theta1P(j,i)*1i) - 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
1219     (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
1220     exp(theta1P(j,i)*1i)*1i))*1i) - 11*cos(theta1P(j,i)) + r*cos(phir(j)))/12));
1221
1222 %select the angles of the second and third segment corresponding to elbow up...
1223 %...if X = 1 is selected
1224 if X == 1
1225     theta2(j,i) = theta2ku(j,i);
1226     theta3(j,i) = theta3ku(j,i);
1227 end
1228
1229 %select the angles of the second and third segment corresponding to elbow down...
1230 %...if X = 0 is selected
1231 if X == 0
1232     theta2(j,i) = theta2kd(j,i);
1233     theta3(j,i) = theta3kd(j,i);
1234 end
1235
1236 %calculate the deviations in x and y of the coordinates of the compensator, respectively
1237 DEV1(j,i) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i)) + 13*sin(theta3(j,i)) -...
1238     r*sin(alpha(j));
1239 DEV2(j,i) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i)) + 13*cos(theta3(j,i)) -...
1240     r*cos(alpha(j));
1241
1242 %if the absolute value of any of these deviations transcends a certain threshold,
1243 %then use alternative formulations for theta2
1244 if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
1245     if X == 1 %elbow up
1246         %other formulation for angle segment 2
1247         theta2(j,i) = pi +...
1248             real(- asin((13*sin(log((-11*r + ((11*r - 11^2*exp(phir(j)*1i))*...
1249             exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1250             13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1251             exp(theta1P(j,i)*1i) - 2*12*13*exp(phir(j)*1i)*...
1252             exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*...
1253             (11*r - 11^2*exp(phir(j)*1i))*...
1254             exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1255             13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1256             exp(theta1P(j,i)*1i) + 2*12*13*exp(phir(j)*1i)*...
1257             exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*...
1258             exp(theta1P(j,i)*2i))^(1/2) - 11^2*exp(phir(j)*1i))*...
1259             exp(theta1P(j,i)*1i) + 12^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
1260             13^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1261             exp(theta1P(j,i)*1i) + 11*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
1262             (2*(11*13*exp(phir(j)*1i)*1i - 13*r*exp(phir(j)*2i)*...
1263             exp(theta1P(j,i)*1i)*1i))*1i) - 11*cos(theta1P(j,i)) +...
1264             r*cos(phir(j))/12));
1265     end
1266
1267     if X == 0 %elbow down
1268         %other formulation for angle segment 2

```

```

1269         theta2(j,i) = real(asin((l3*sin(log((- l1*r +...
1270             ((l1*r - l1^2*exp(phir(j)*1i))*...
1271             exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1272             l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1273             exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i))*...
1274             exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i))*...
1275             exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i))*...
1276             exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1277             l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
1278             exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i))*...
1279             exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i))*...
1280             exp(theta1P(j,i)*2i))^(1/2) + l1^2*exp(phir(j)*1i))*...
1281             exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
1282             l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i))*...
1283             exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
1284             (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i))*...
1285             exp(theta1P(j,i)*1i)*1i))*1i) - l1*cos(theta1P(j,i)) +...
1286             r*cos(phir(j))/l2));
1287     end
1288 end
1289 end
1290
1291 end
1292
1293 %if the angle of the third segment is smaller than -90 deg...
1294 %...do a phase shift of 360 deg
1295 if theta3(j,i) < -pi/2
1296     theta3(j,i) = theta3(j,i) + 2*pi;
1297 end
1298
1299 %evaluate the deviations in x and y again, respectively
1300 DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
1301     r*sin(alpha(j));
1302 DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
1303     r*cos(alpha(j));
1304
1305 d(j,i) = sqrt((r*sin(alpha(j))-l1*cos(theta1(j,i)))^2 +...
1306     (r*cos(alpha(j))-l1*sin(theta1(j,i)))^2);
1307
1308 %check condition upper loop closure
1309 if l3-l2+d(j,i) < 0
1310     %mark error with variable "Count2"
1311     Count2 = Count2 + 1;
1312 end
1313
1314 %check condition upper loop closure
1315 if (l3-l2-d(j,i)) > 0
1316     %mark error with variable "Count2"
1317     Count2 = Count2 + 1;
1318 end
1319
1320 %if any of these deviations transcends a certain treshold then throw an error
1321 if abs(DEV1(j,i)) > 10^-10 || abs(DEV2(j,i)) > 10^-10
1322     Count = Count + 1;
1323 end
1324
1325 %initial relative angle of segment 1
1326 alpha10 = theta10;
1327 %initial relative angle of segment 2
1328 alpha20 = theta20 - theta10;
1329 %initial relative angle of segment 3
1330 alpha30 = theta30 - theta20;
1331
1332 %angle of rotation torsion spring 1
1333 alpha1(j,i) = theta1(j,i) - alpha10;
1334 %angle of rotation torsion spring 2
1335 alpha2(j,i) = theta2(j,i) - theta1(j,i) - alpha20;
1336 %angle of rotation torsion spring 3
1337 alpha3(j,i) = theta3(j,i) - theta2(j,i) - alpha30;
1338
1339 if prestress == 0 && nonlinearity == 0
1340     %potential energy spring 1
1341     V1(j,i) = ((k1/2)*alpha1(j,i)^2);
1342     %potential energy spring 2
1343     V2(j,i) = ((k2/2)*alpha2(j,i)^2);
1344     %potential energy spring 3
1345     V3(j,i) = ((k3/2)*alpha3(j,i)^2);
1346     %total potential energy
1347     V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);
1348
1349
1350     %the spring moment in spring 1
1351     M1(j,i) = k1*alpha1(j,i);
1352     %the spring moment in spring 2

```



```

1353     M2(j,i) = k2*alpha2(j,i);
1354     %the spring moment in spring 3
1355     M3(j,i) = k3*alpha3(j,i);
1356 end
1357
1358 %if springs are prestressed
1359 if prestress == 1 && nonlinearity == 0
1360     %potential energy spring 1
1361     V1(j,i) = ((k1/2)*alpha1(j,i)^2);
1362     %potential energy spring 2
1363     V2(j,i) = ((k2/2)*alpha2(j,i)^2) + M0*alpha2(j,i) + ((k2/2)*(M0/k2)^2);
1364     %potential energy spring 3
1365     V3(j,i) = ((k3/2)*alpha3(j,i)^2);
1366     %total potential energy
1367     V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);
1368
1369
1370     %the spring moment in spring 1
1371     M1(j,i) = k1*alpha1(j,i);
1372     %the spring moment in spring 2
1373     M2(j,i) = k2*alpha2(j,i) + M0;
1374     %the spring moment in spring 3
1375     M3(j,i) = k3*alpha3(j,i);
1376 end
1377
1378 %if springs are nonlinear
1379 if prestress == 0 && nonlinearity == 1
1380     %potential energy spring 1
1381     V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
1382     %potential energy spring 2
1383     V2(j,i) = (A/3)*alpha2(j,i)^3 + (B/2)*alpha2(j,i)^2;
1384     %potential energy spring 3
1385     V3(j,i) = (A/3)*alpha3(j,i)^3 + (B/2)*alpha3(j,i)^2;
1386     %total potential energy
1387     V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);
1388 end
1389
1390 %if springs are prestressed and nonlinear
1391 if prestress == 1 && nonlinearity == 1
1392     %first solution prestress angle: angle of rotation corresponding to prestress
1393     alphastar1 = (-B + sqrt(B^2 + 4*M0*A))/(2*A);
1394     %second solution prestress angle: angle of rotation corresponding to prestress
1395     alphastar2 = (-B - sqrt(B^2 + 4*M0*A))/(2*A);
1396
1397     %allow only for nonnegative solutions; set to NaN if negative
1398     if alphastar1 < 0
1399         alphastar1 = NaN;
1400     end
1401
1402     %allow only for nonnegative solutions; set to NaN if negative
1403     if alphastar2 < 0
1404         alphastar2 = NaN;
1405     end
1406
1407     %store solutions prestress angle in array called "alphastars"
1408     alphastars = [alphastar1,alphastar2];
1409
1410     %store the smallest solution for the prestress angle
1411     alphastar = min(abs(alphastars));
1412
1413     %potential energy spring 1
1414     V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
1415     %potential energy spring 2
1416     V2(j,i) = (A/3)*(alpha2(j,i)+alphastar)^3 + (B/2)*(alpha2(j,i)+alphastar)^2;
1417     %potential energy spring 3
1418     V3(j,i) = (A/3)*alpha3(j,i)^3 + (B/2)*alpha3(j,i)^2;
1419     %total potential energy springs
1420     V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);
1421
1422
1423     %the spring moment in spring 1
1424     M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
1425     %the spring moment in spring 2
1426     M2(j,i) = A*(alpha2(j,i)+alphastar)^2 + B*(alpha2(j,i)+alphastar);
1427     %the spring moment in spring 3
1428     M3(j,i) = A*alpha3(j,i)^2 + B*alpha3(j,i);
1429 end
1430
1431 %coordinates of nodes
1432 %x - coordinate origin (and first spring)
1433 x0 = 0;
1434 %y - coordinate origin (and first spring)
1435 y0 = 0;
1436 %x - coordinate 2nd spring

```

```

1437 x1(j,i) = l1*sin(theta1(j,i));
1438 %y - coordinate 2nd spring
1439 y1(j,i) = l1*cos(theta1(j,i));
1440 %x - coordinate 3rd spring
1441 x2(j,i) = x1(j,i) + l2*sin(theta2(j,i));
1442 %y - coordinate 3rd spring
1443 y2(j,i) = y1(j,i) + l2*cos(theta2(j,i));
1444 %x - coordinate end effector
1445 x3(j,i) = x2(j,i) + l3*sin(theta3(j,i));
1446 %y - coordinate end effector
1447 y3(j,i) = y2(j,i) + l3*cos(theta3(j,i));
1448
1449 %do the plotting for the initial configuration
1450 %x - coordinate origin (and first spring)
1451 x00 = 0;
1452 %y - coordinate origin (and first spring)
1453 y00 = 0;
1454 %x - coordinate 2nd spring
1455 x10 = l1*sin(theta10);
1456 %y - coordinate 2nd spring
1457 y10 = l1*cos(theta10);
1458 %x - coordinate 3rd spring
1459 x20 = x10 + l2*sin(theta20);
1460 %y - coordinate 3rd spring
1461 y20 = y10 + l2*cos(theta20);
1462 %x - coordinate end effector
1463 x30 = x20 + l3*sin(theta30);
1464 %y - coordinate end effector
1465 y30 = y20 + l3*cos(theta30);
1466
1467 %vertical reaction force at segment 1 (positive upwards)
1468 F1yt(j,i) = (M1(j,i) - M3(j,i) - (M3(j,i)/(l3*cos(theta3(j,i))))*...
1469 (l1*cos(theta1(j,i))+l2*cos(theta2(j,i))))/...
1470 ((l1*sin(theta1(j,i))+l2*sin(theta2(j,i))) - tan(theta3(j,i))*...
1471 (l1*cos(theta1(j,i))+l2*cos(theta2(j,i))));
1472
1473 %horizontal reaction force at segment 1 (positive to the right)
1474 F1xt(j,i) = (-M3(j,i) + F1yt(j,i)*l3*sin(theta3(j,i)))/(l3*cos(theta3(j,i)));
1475
1476 %the load (moment) on nodes 2 and 3 (where springs 2 and 3 are located), respectively
1477 M2lt(j,i) = M1(j,i) + F1xt(j,i)*l1*cos(theta1(j,i)) - F1yt(j,i)*l1*sin(theta1(j,i));
1478 M3lt(j,i) = M1(j,i) + F1xt(j,i)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i))) -...
1479 F1yt(j,i)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i)));
1480
1481 if prestress == 1
1482 %if the load (moment) on node 2 transcends the pretension...
1483 %...the spring will be activated
1484 if M2lt(j,i) > M0
1485 activation = 1;
1486 end
1487 end
1488
1489 end
1490 end
1491
1492 %find minimum potential energy and it's corresponding index
1493 [Vmin,I] = min(V,[],2);
1494 figure(1) %create figure
1495 %plot following plot commands in that same figure
1496 hold on
1497 axis equal
1498 title('Lowest energy configurations')
1499
1500 %start a loop throughout all precision points
1501 for j = 1:M
1502 %divide the 90 deg range of motion into equally sized segments
1503 alpha(j) = (pi/2)*(j/M);
1504
1505 %plot connection line between spring 1 and 2 in black
1506 plot([x0 x1(j,I(j))],[y0 y1(j,I(j))],'k')
1507 %plot connection line between spring 2 and 3 in black
1508 plot([x1(j,I(j)) x2(j,I(j))],[y1(j,I(j)) y2(j,I(j))],'k')
1509 %plot connection line between spring 3 and end effector in black
1510 plot([x2(j,I(j)) x3(j,I(j))],[y2(j,I(j)) y3(j,I(j))],'k')
1511 %plot the location of the end effector of the pendulum with a circle
1512 plot(r*sin(alpha(j)),r*cos(alpha(j)),'b--o')
1513
1514 %plot connection line between spring 1 and 2 in black
1515 plot([x00 x10],[y00 y10],'red')
1516 %plot connection line between spring 2 and 3 in black
1517 plot([x10 x20],[y10 y20],'red')
1518 %plot connection line between spring 3 and end effector in black
1519 plot([x20 x30],[y20 y30],'red')
1520 %plot the location of the end effector of the pendulum with a circle

```

```

1521 plot(r*sin(0),r*cos(0),'r--o')
1522
1523 %angle of spring 1 w.r.t. vertical, corresponding to equilibrium
1524 theta1m(j) = theta1(j,I(j));
1525 %angle of spring 2 w.r.t. vertical, corresponding to equilibrium
1526 theta2m(j) = theta2(j,I(j));
1527 %angle of spring 3 w.r.t. vertical, corresponding to equilibrium
1528 theta3m(j) = theta3(j,I(j));
1529
1530 %deformation angle of spring 1, corresponding to equilibrium
1531 alpha1m(j) = alpha1(j,I(j));
1532 %deformation angle of spring 2, corresponding to equilibrium
1533 alpha2m(j) = alpha2(j,I(j));
1534 %deformation angle of spring 3, corresponding to equilibrium
1535 alpha3m(j) = alpha3(j,I(j));
1536
1537
1538 if prestress == 0 && nonlinearity == 0
1539     %internal moment spring 1, corresponding to equilibrium
1540     M1m(j) = k1*alpha1(j,I(j));
1541     %internal moment spring 2, corresponding to equilibrium
1542     M2m(j) = k2*alpha2(j,I(j));
1543     %internal moment spring 3, corresponding to equilibrium
1544     M3m(j) = k3*alpha3(j,I(j));
1545
1546
1547     %potential energy spring 1, corresponding to equilibrium
1548     V1m(j) = V1(j,I(j));
1549     %potential energy spring 2, corresponding to equilibrium
1550     V2m(j) = V2(j,I(j));
1551     %potential energy spring 3, corresponding to equilibrium
1552     V3m(j) = V3(j,I(j));
1553     %total potential energy springs, corresponding to equilibrium
1554     Vtm(j) = V1m(j) + V2m(j) + V3m(j);
1555 end
1556
1557 %if springs are prestressed
1558 if prestress == 1 && nonlinearity == 0
1559     %internal moment spring 1, corresponding to equilibrium
1560     M1m(j) = k1*alpha1(j,I(j));
1561     %internal moment spring 2, corresponding to equilibrium
1562     M2m(j) = k2*alpha2(j,I(j)) + M0;
1563     %internal moment spring 3, corresponding to equilibrium
1564     M3m(j) = k3*alpha3(j,I(j));
1565 end
1566
1567 %if springs are nonlinear
1568 if prestress == 0 && nonlinearity == 1
1569     %internal moment spring 1, corresponding to equilibrium
1570     M1m(j) = A*alpha1(j,I(j))^2 + B*alpha1(j,I(j));
1571     %internal moment spring 2, corresponding to equilibrium
1572     M2m(j) = A*alpha2(j,I(j))^2 + B*alpha2(j,I(j));
1573     %internal moment spring 3, corresponding to equilibrium
1574     M3m(j) = A*alpha3(j,I(j))^2 + B*alpha3(j,I(j));
1575
1576
1577     %potential energy spring 1, corresponding to equilibrium
1578     V1m(j) = (A/3)*alpha1(j,I(j))^3 + (B/2)*alpha1(j,I(j))^2;
1579     %potential energy spring 2, corresponding to equilibrium
1580     V2m(j) = (A/3)*alpha2(j,I(j))^3 + (B/2)*alpha2(j,I(j))^2;
1581     %potential energy spring 3, corresponding to equilibrium
1582     V3m(j) = (A/3)*alpha3(j,I(j))^3 + (B/2)*alpha3(j,I(j))^2;
1583 end
1584
1585 %if springs are prestressed and nonlinear
1586 if prestress == 1 && nonlinearity == 1
1587     %internal moment spring 1, corresponding to equilibrium
1588     M1m(j) = A*alpha1(j,I(j))^2 + B*alpha1(j,I(j));
1589     %internal moment spring 2, corresponding to equilibrium
1590     M2m(j) = A*(alpha2(j,I(j))+alphastar)^2 + B*(alpha2(j,I(j))+alphastar);
1591     %internal moment spring 3, corresponding to equilibrium
1592     M3m(j) = A*alpha3(j,I(j))^2 + B*alpha3(j,I(j));
1593 end
1594
1595 %vertical reaction force at segment 1 (positive upwards)
1596 F1y(j) = (M1m(j) - M3m(j) - (M3m(j)/(13*cos(theta3(j,I(j))))))*...
1597     ((11*cos(theta1(j,I(j)))+12*cos(theta2(j,I(j)))))/...
1598     ((11*sin(theta1(j,I(j)))+12*sin(theta2(j,I(j)))) - tan(theta3(j,I(j))))*...
1599     ((11*cos(theta1(j,I(j)))+12*cos(theta2(j,I(j)))));
1600
1601 %horizontal reaction force at segment 1 (positive to the right)
1602 F1x(j) = (-M3m(j) + F1y(j)*13*sin(theta3(j,I(j))))/(13*cos(theta3(j,I(j))));
1603
1604 Tlb(j) = ((alpha(j)-pi/2)*180/pi);

```

```

1605 Tub(j) = ((alpha(j)+pi/2)*180/pi);
1606
1607 %the external load (moment) on nodes 1, 2 and 3...
1608 %...(where springs 1, 2 and 3 are located), respectively
1609 M1l(j) = F1y(j)*r*sin(alpha(j)) - F1x(j)*r*cos(alpha(j));
1610 M2l(j) = M1m(j) + F1x(j)*l1*cos(theta1(j,I(j))) - F1y(j)*l1*sin(theta1(j,I(j)));
1611 M3l(j) = M1m(j) + F1x(j)*(l1*cos(theta1(j,I(j)))+l2*cos(theta2(j,I(j)))) -...
1612     F1y(j)*(l1*sin(theta1(j,I(j)))+l2*sin(theta2(j,I(j))));
1613
1614 %objective moment-angle characteristics
1615 if objective == "sinus"
1616     Vm(j) = mg*r*cos(alpha(j));           %original value
1617     Mobj(j) = mg*r*sin(alpha(j));        %original value
1618 end
1619
1620 if objective == "Laevo"
1621     Vm(j) = (0.05022*alpha(j)^5 - 0.33575*alpha(j)^4 + 0.97*alpha(j)^3 -...
1622     1.412*alpha(j)^2 + 0.006501*alpha(j) + 1);
1623     Mobj(j) = (-0.2511*alpha(j)^4 + 1.343*alpha(j)^3 - 2.91*alpha(j)^2 +...
1624     2.824*alpha(j) - 0.006501);
1625 end
1626
1627 if objective == "stiffening"
1628     Vm(j) = sin(alpha(j)) - alpha(j);
1629     Mobj(j) = -cos(alpha(j))+1;
1630 end
1631
1632 if objective == "sqrt"
1633     Vm(j) = - (2/3)*alpha(j)^(3/2);
1634     Mobj(j) = sqrt(alpha(j));
1635 end
1636
1637 if objective == "quadratic"
1638     Vm(j) = - (1/3)*alpha(j)^(3);
1639     Mobj(j) = alpha(j)^2;
1640 end
1641
1642 if objective == "hardening-softening"
1643     Vm(j) = 0.25*cos(2*alpha(j)-pi/2) - 0.5*alpha(j);
1644     Mobj(j) = (sin(2*alpha(j)-pi/2)+1)/2;
1645 end
1646
1647 if objective == "hardening-softening2"
1648     Vm(j) = -0.5*alpha(j)+(-0.333333+0.424413*alpha(j))*atan(2.41421-3.07387...
1649     *alpha(j))+0.0690356*log(9.8696-21.4521*alpha(j)+13.6569*alpha(j)^2);
1650     Mobj(j) = 0.5 + (4/(3*pi))*atan(tan((3*pi)/8)*((4/pi)*alpha(j)-1));
1651 end
1652
1653 if objective == "softening-hardening"
1654     Vm(j) = 0.5*log(cos(alpha(j)-pi/4)) - 0.5*alpha(j);
1655     Mobj(j) = 0.5*tan(alpha(j)-pi/4) + 0.5;
1656 end
1657
1658 if objective == "softening-hardening2"
1659     Vm(j) = -0.5*alpha(j) - 0.0690356*log(1 + tan(1.1781 - 1.5*alpha(j))^2);
1660     Mobj(j) = 0.5*tan(1.5*(alpha(j)-pi/4))/tan(1.5*(pi/4))+0.5;
1661 end
1662
1663 if objective == "sinuspi"
1664     Vm(j) = 0.5*cos(2*alpha(j));
1665     Mobj(j) = sin(2*alpha(j));
1666 end
1667
1668 end
1669
1670 %print the root mean square error (objective function)
1671 e = sqrt(mean((M1m - Mobj).^2)) %ok<NOPTS>
1672 vd = mean(sqrt((V1m-V2m).^2 + (V1m-V3m).^2 + (V2m-V3m).^2));
1673
1674 IntM = trapz(alpha,abs(M1m-Mobj));
1675 IntMo = trapz(alpha,Mobj);
1676
1677 figure(2)
1678 hold on
1679 plot(alpha*180/pi,M1m)
1680 plot(alpha*180/pi,Mobj)
1681 plot(alpha*180/pi,M1m - Mobj)
1682 xlabel('Angle of rotation pendulum from vertical (deg)')
1683 ylabel('Moment around suspension-point 1 (Nm)')
1684 legend('Moment in spring 1','Objective moment','Error in moment',...
1685     'location','northwest')
1686
1687 figure(3)
1688 hold on

```

```

1689 plot(alpha*180/pi,Vmin+transpose(Vm))
1690 xlabel('Angle of rotation pendulum from vertical (deg)')
1691 ylabel('Total potential energy in system (J)')
1692
1693 figure(4)
1694 hold on
1695 plot(alpha*180/pi,V1m)
1696 plot(alpha*180/pi,V2m)
1697 plot(alpha*180/pi,V3m)
1698 xlabel('Angle of rotation pendulum from vertical (deg)')
1699 ylabel('Energy storage in springs (J)')
1700 legend('Energy spring 1','Energy spring 2','Energy spring 3','location','northwest')
1701
1702 figure(5)
1703 hold on
1704 plot(alpha*180/pi,M2m)
1705 plot(alpha*180/pi,M3m)
1706 plot(alpha*180/pi,M2l)
1707 plot(alpha*180/pi,M3l)
1708 xlabel('Angle of rotation pendulum from vertical (deg)')
1709 ylabel('Reaction moments in springs (Nm)')
1710 legend('M2m','M3m','M2l','M3l','location','northwest')
1711
1712 figure(6)
1713 hold on
1714 plot(alpha*180/pi,F1x)
1715 plot(alpha*180/pi,F1y)
1716 xlabel('Angle of rotation pendulum from vertical (deg)')
1717 ylabel('Reaction force in point 1 (N)')
1718 legend('F1x','F1y','location','northwest')
1719
1720 figure(7)
1721 hold on
1722 plot(alpha*180/pi,alpha1m*180/pi)
1723 xlabel('Angle of rotation pendulum from vertical (deg)')
1724 ylabel('Angle of rotation spring 1 (deg)')

```

N.4. Four segment balancer

```

1  clc                                %clear command window
2  clear variables                     %empty workspace
3  close all                           %close all windows
4
5  %with (1) or without (0) prestress on springs
6  prestress = 1;
7  %with (1) or without (0) nonlinear springs
8  nonlinearity = 0;
9  %type objective function between quotation marks...
10 objective = "sinus";
11
12 M = 15;                             %amount of precision points
13
14 %amount of configurations of segment 1 per precision point
15 N1 = 150;
16 %number of configurations of segment 2 per precision point
17 N2 = 150;
18
19 k1 = 0.971;                          %stiffness spring 1 (Nm/rad)
20 k2 = 0.105;                          %stiffness spring 2 (Nm/rad)
21 k3 = 0.310;                          %stiffness spring 3 (Nm/rad)
22 k4 = 0;                               %stiffness spring 4 (Nm/rad)
23 M02 = 0.596;                         %preload spring 2 (Nm)
24 M03 = 0.225;                         %preload spring 3 (Nm)
25 l1 = 0.29;                           %length first segment (m)
26 l2 = 0.29;                           %length second segment (m)
27 l3 = 0.29;                           %length third segment (m)
28 l4 = 0.29;                           %length fourth segment (m)
29
30 %length pendulum (m)
31 r = 1;
32 %constant mass times grav. constant (N)
33 mg = 1;
34
35 Count = 0;                            %error counter
36 Count2 = 0;                          %second error counter
37 Count3 = 0;                          %third error counter
38 %-----
39 %definition of angles in the initial (relaxed) configuration
40 thetai1 = 0*pi/180;                   %independent variable: initial angle segment 1
41 %dependent angles initial configuration
42 thetai1 = -thetai1;
43 Ari = (pi/2);
44 A1i = (pi/2) - thetai1;

```

```

45 theta1pi = theta1i - (pi/2);
46
47 %if initial angle of first segment is greater than zero or equal to zero
48 if theta1i >= 0
49     %initial upperbound angle segment 2
50     theta2fi = (pi/2) - real(-log((- ((- 11^2*exp(A1i*1i)*exp(Ari*1i) +...
51         12^2*exp(A1i*1i)*exp(Ari*1i) + 13^2*exp(A1i*1i)*exp(Ari*1i) +...
52         14^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
53         11*r*exp(A1i*2i) + 11*r*exp(Ari*2i) -...
54         2*12*13*exp(A1i*1i)*exp(Ari*1i) - 2*12*14*exp(A1i*1i)*exp(Ari*1i) +...
55         2*13*14*exp(A1i*1i)*exp(Ari*1i))*(- 11^2*exp(A1i*1i)*exp(Ari*1i) +...
56         12^2*exp(A1i*1i)*exp(Ari*1i) + 13^2*exp(A1i*1i)*exp(Ari*1i) +...
57         14^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
58         11*r*exp(A1i*2i) + 11*r*exp(Ari*2i) +...
59         2*12*13*exp(A1i*1i)*exp(Ari*1i) + 2*12*14*exp(A1i*1i)*exp(Ari*1i) +...
60         2*13*14*exp(A1i*1i)*exp(Ari*1i)))^(1/2) -...
61         11^2*exp(A1i*1i)*exp(Ari*1i) - 12^2*exp(A1i*1i)*exp(Ari*1i) +...
62         13^2*exp(A1i*1i)*exp(Ari*1i) + 14^2*exp(A1i*1i)*exp(Ari*1i) -...
63         r^2*exp(A1i*1i)*exp(Ari*1i) + 11*r*exp(A1i*2i) + 11*r*exp(Ari*2i) +...
64         2*13*14*exp(A1i*1i)*exp(Ari*1i))/...
65         (2*(11*12*exp(Ari*1i) - 12*r*exp(A1i*1i))))*1i);
66     %initial lowerbound angle segment 2
67     theta20i = (pi/2) - real(-log((( (- 11^2*exp(A1i*1i)*exp(Ari*1i) +...
68         12^2*exp(A1i*1i)*exp(Ari*1i) + 13^2*exp(A1i*1i)*exp(Ari*1i) +...
69         14^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
70         11*r*exp(A1i*2i) + 11*r*exp(Ari*2i) -...
71         2*12*13*exp(A1i*1i)*exp(Ari*1i) - 2*12*14*exp(A1i*1i)*exp(Ari*1i) +...
72         2*13*14*exp(A1i*1i)*exp(Ari*1i))*(- 11^2*exp(A1i*1i)*exp(Ari*1i) +...
73         12^2*exp(A1i*1i)*exp(Ari*1i) + 13^2*exp(A1i*1i)*exp(Ari*1i) +...
74         14^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
75         11*r*exp(A1i*2i) + 11*r*exp(Ari*2i) +...
76         2*12*13*exp(A1i*1i)*exp(Ari*1i) + 2*12*14*exp(A1i*1i)*exp(Ari*1i) +...
77         2*13*14*exp(A1i*1i)*exp(Ari*1i)))^(1/2) -...
78         11^2*exp(A1i*1i)*exp(Ari*1i) -...
79         12^2*exp(A1i*1i)*exp(Ari*1i) + 13^2*exp(A1i*1i)*exp(Ari*1i) +...
80         14^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
81         11*r*exp(A1i*2i) + 11*r*exp(Ari*2i) +...
82         2*13*14*exp(A1i*1i)*exp(Ari*1i))/...
83         (2*(11*12*exp(Ari*1i) - 12*r*exp(A1i*1i))))*1i);
84     end
85
86 %if initial angle of first segment is smaller than zero
87 if theta1i < 0
88     %initial upperbound angle segment 2
89     theta2fi = real(-log(-(((11*r - 11^2*exp(0*1i)*exp(theta1ni*1i) +...
90         12^2*exp(0*1i)*exp(theta1ni*1i) + 13^2*exp(0*1i)*exp(theta1ni*1i) +...
91         14^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) -...
92         2*12*13*exp(0*1i)*exp(theta1ni*1i) -...
93         2*12*14*exp(0*1i)*exp(theta1ni*1i) +...
94         2*13*14*exp(0*1i)*exp(theta1ni*1i) +...
95         11*r*exp(0*2i)*exp(theta1ni*2i))*...
96         (11*r - 11^2*exp(0*1i)*exp(theta1ni*1i) +...
97         12^2*exp(0*1i)*exp(theta1ni*1i) + 13^2*exp(0*1i)*exp(theta1ni*1i) +...
98         14^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) +...
99         2*12*13*exp(0*1i)*exp(theta1ni*1i) +...
100        2*12*14*exp(0*1i)*exp(theta1ni*1i) +...
101        2*13*14*exp(0*1i)*exp(theta1ni*1i) +...
102        11*r*exp(0*2i)*exp(theta1ni*2i)))^(1/2) +...
103        11*r - 11^2*exp(0*1i)*exp(theta1ni*1i) -...
104        12^2*exp(0*1i)*exp(theta1ni*1i) + 13^2*exp(0*1i)*exp(theta1ni*1i) +...
105        14^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) +...
106        2*13*14*exp(0*1i)*exp(theta1ni*1i) + 11*r*exp(0*2i)*exp(theta1ni*2i))/...
107        (2*(12*r*exp(theta1ni*1i) - 11*12*exp(0*1i)*exp(theta1ni*2i))))*1i);
108     %initial lowerbound angle segment 2
109     theta20i = real(-log(-(11*r - ((11*r - 11^2*exp(0*1i)*exp(theta1ni*1i) +...
110         12^2*exp(0*1i)*exp(theta1ni*1i) + 13^2*exp(0*1i)*exp(theta1ni*1i) +...
111         14^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) -...
112         2*12*13*exp(0*1i)*exp(theta1ni*1i) -...
113         2*12*14*exp(0*1i)*exp(theta1ni*1i) +...
114         2*13*14*exp(0*1i)*exp(theta1ni*1i) +...
115         11*r*exp(0*2i)*exp(theta1ni*2i))*...
116         (11*r - 11^2*exp(0*1i)*exp(theta1ni*1i) +...
117         12^2*exp(0*1i)*exp(theta1ni*1i) + 13^2*exp(0*1i)*exp(theta1ni*1i) +...
118         14^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) +...
119         2*12*13*exp(0*1i)*exp(theta1ni*1i) +...
120         2*12*14*exp(0*1i)*exp(theta1ni*1i) +...
121         2*13*14*exp(0*1i)*exp(theta1ni*1i) +...
122         11*r*exp(0*2i)*exp(theta1ni*2i)))^(1/2) -...
123         11^2*exp(0*1i)*exp(theta1ni*1i) - 12^2*exp(0*1i)*exp(theta1ni*1i) +...
124         13^2*exp(0*1i)*exp(theta1ni*1i) + 14^2*exp(0*1i)*exp(theta1ni*1i) -...
125         r^2*exp(0*1i)*exp(theta1ni*1i) + 2*13*14*exp(0*1i)*exp(theta1ni*1i) +...
126         11*r*exp(0*2i)*exp(theta1ni*2i))/...
127         (2*(12*r*exp(theta1ni*1i) - 11*12*exp(0*1i)*exp(theta1ni*2i))))*1i);
128     end

```

```

129 %initial angle of the second segment
130 theta2i = 0;
131
132
133 %initial angle of the second segment, (CCW positive) with respect to the
134 %positive x-axis
135 A2i = (pi/2) - theta2i;
136
137 %the length of the imaginary connection line between the origin and the
138 %node at the end of the second segment...
139 l12i = sqrt((l1*sin(theta1i) + l2*sin(theta2i))^2 +...
140 (l1*cos(theta1i) + l2*cos(theta2i))^2);
141 %...and its angle with respect to the vertical
142 phi12i = (pi/2) - atan((l1*sin(theta1i) +...
143 l2*sin(theta2i))/(l1*cos(theta1i) + l2*cos(theta2i)));
144
145 Mtheta12i = - atan((l1*sin(theta1i) + l2*sin(theta2i))/...
146 (l1*cos(theta1i) + l2*cos(theta2i)));
147
148 %the angle of the third segment, corresponding to the system in its initial
149 %configuration
150 theta3i = pi/2 - real(pi - acos((l12i*cos(phi12i) - r*cos(Ari) +...
151 l4*cos(log(-((l12i*r*exp(Ari*2i) + l12i*r*exp(phi12i*2i) -...
152 l12i^2*exp(Ari*1i)*exp(phi12i*1i) + l3^2*exp(Ari*1i)*exp(phi12i*1i) +...
153 l4^2*exp(Ari*1i)*exp(phi12i*1i) - r^2*exp(Ari*1i)*exp(phi12i*1i) -...
154 2*l3*l4*exp(Ari*1i)*exp(phi12i*1i))*(l12i*r*exp(Ari*2i) +...
155 l12i*r*exp(phi12i*2i) - l12i^2*exp(Ari*1i)*exp(phi12i*1i) +...
156 l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) -...
157 r^2*exp(Ari*1i)*exp(phi12i*1i) +...
158 2*l3*l4*exp(Ari*1i)*exp(phi12i*1i)))^(1/2) - l12i*r*exp(Ari*2i) -...
159 l12i*r*exp(phi12i*2i) + l12i^2*exp(Ari*1i)*exp(phi12i*1i) -...
160 l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) +...
161 r^2*exp(Ari*1i)*exp(phi12i*1i))/...
162 (2*(l12i*l4*exp(Ari*1i) - l4*r*exp(phi12i*1i))))*1i)/l3));
163
164 %the angle of the fourth segment, corresponding to the system in its initial
165 %configuration
166 theta4i = pi/2 - real(-log(-((l12i*r*exp(Ari*2i) + l12i*r*exp(phi12i*2i) -...
167 l12i^2*exp(Ari*1i)*exp(phi12i*1i) + l3^2*exp(Ari*1i)*exp(phi12i*1i) +...
168 l4^2*exp(Ari*1i)*exp(phi12i*1i) - r^2*exp(Ari*1i)*exp(phi12i*1i) -...
169 2*l3*l4*exp(Ari*1i)*exp(phi12i*1i))*(l12i*r*exp(Ari*2i) +...
170 l12i*r*exp(phi12i*2i) - l12i^2*exp(Ari*1i)*exp(phi12i*1i) +...
171 l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) -...
172 r^2*exp(Ari*1i)*exp(phi12i*1i) +...
173 2*l3*l4*exp(Ari*1i)*exp(phi12i*1i)))^(1/2) - l12i*r*exp(Ari*2i) -...
174 l12i*r*exp(phi12i*2i) + l12i^2*exp(Ari*1i)*exp(phi12i*1i) -...
175 l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) +...
176 r^2*exp(Ari*1i)*exp(phi12i*1i))/...
177 (2*(l12i*l4*exp(Ari*1i) - l4*r*exp(phi12i*1i))))*1i);
178
179 %if the node at the end of the second segment is located left to the y-axis
180 %in the initial configuration
181 if (l1*sin(theta1i) + l2*sin(theta2i)) < 0
182 %define alternative formulation initial angle segment 3
183 theta3i = real(asin((l4*sin(log(-l12i*r +...
184 ((l12i*r - l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
185 l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
186 l4^2*exp(Mtheta12i*1i)*exp(0*1i) -...
187 r^2*exp(Mtheta12i*1i)*exp(0*1i) -...
188 2*l3*l4*exp(Mtheta12i*1i)*exp(0*1i) +...
189 l12i*r*exp(Mtheta12i*2i)*exp(0*2i))*(l12i*r -...
190 l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
191 l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
192 l4^2*exp(Mtheta12i*1i)*exp(0*1i) - r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
193 2*l3*l4*exp(Mtheta12i*1i)*exp(0*1i) +...
194 l12i*r*exp(Mtheta12i*2i)*exp(0*2i)))^(1/2) -...
195 l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
196 l3^2*exp(Mtheta12i*1i)*exp(0*1i) -...
197 l4^2*exp(Mtheta12i*1i)*exp(0*1i) -...
198 r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
199 l12i*r*exp(Mtheta12i*2i)*exp(0*2i))/(2*(l4*r*exp(Mtheta12i*1i) -...
200 l12i*l4*exp(Mtheta12i*2i)*exp(0*1i))))*1i) +...
201 l12i*sin(Mtheta12i) + r*sin(0))/l3));
202
203 %define alternative formulation initial angle segment 4
204 theta4i = real(-log(-l12i*r + ((l12i*r - l12i^2*exp(Mtheta12i*1i)*...
205 exp(0*1i) + l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
206 l4^2*exp(Mtheta12i*1i)*exp(0*1i) - r^2*exp(Mtheta12i*1i)*exp(0*1i) -...
207 2*l3*l4*exp(Mtheta12i*1i)*exp(0*1i) +...
208 l12i*r*exp(Mtheta12i*2i)*exp(0*2i))*...
209 (l12i*r - l12i^2*exp(Mtheta12i*1i)*...
210 exp(0*1i) + l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
211 l4^2*exp(Mtheta12i*1i)*...
212 exp(0*1i) - r^2*exp(Mtheta12i*1i)*exp(0*1i) +...

```

```

213     2*13*14*exp(Mtheta12i*1i)*...
214     exp(0*1i) + 112i*r*exp(Mtheta12i*2i)*exp(0*2i))^ (1/2) -...
215     112i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
216     13^2*exp(Mtheta12i*1i)*exp(0*1i) -...
217     14^2*exp(Mtheta12i*1i)*exp(0*1i) -...
218     r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
219     112i*r*exp(Mtheta12i*2i)*exp(0*2i)/(2*(14*r*exp(Mtheta12i*1i) -...
220     112i*14*exp(Mtheta12i*2i)*exp(0*1i))) *1i);
221 end
222 %-----
223 %preallocate all variables for better performance...
224 %...vectors
225 alpha = zeros(1,M);
226 theta100 = zeros(1,M);
227 theta1ff = zeros(1,M);
228 theta100pa = zeros(1,M);
229 theta1ffpa = zeros(1,M);
230 theta100pa2 = zeros(1,M);
231 theta1ffpa2 = zeros(1,M);
232 BEGIN1 = zeros(1,M);
233 END1 = zeros(1,M);
234 STEP1 = zeros(1,M);
235 BEGIN1pa = zeros(1,M);
236 END1pa = zeros(1,M);
237 STEP1pa = zeros(1,M);
238 BEGIN1pa2 = zeros(1,M);
239 END1pa2 = zeros(1,M);
240 STEP1pa2 = zeros(1,M);
241 Ar = zeros(1,M);
242 alpha1m = zeros(1,M);
243 alpha2m = zeros(1,M);
244 alpha3m = zeros(1,M);
245 alpha4m = zeros(1,M);
246 theta1m = zeros(1,M);
247 theta2m = zeros(1,M);
248 theta3m = zeros(1,M);
249 theta4m = zeros(1,M);
250 M1m = zeros(1,M);
251 M2m = zeros(1,M);
252 M3m = zeros(1,M);
253 M4m = zeros(1,M);
254 Mobj = zeros(1,M);
255 Fix = zeros(1,M);
256 Fiy = zeros(1,M);
257 M2l = zeros(1,M);
258 M3l = zeros(1,M);
259 Vm = zeros(1,M);
260 fit = zeros(1,M);
261
262 %preallocation of matrices (number of rows = M, number of columns = N1)
263 theta1 = zeros(M,N1);
264 theta1n = zeros(M,N1);
265 theta2f = zeros(M,N1);
266 theta20 = zeros(M,N1);
267 BEGIN2 = zeros(M,N1);
268 END2 = zeros(M,N1);
269 STEP2 = zeros(M,N1);
270 A1 = zeros(M,N1);
271 theta1p = zeros(M,N1);
272 theta1sw = zeros(M,N1);
273 theta1sw2 = zeros(M,N1);
274 theta1fa = zeros(M,N1);
275 alpha1 = zeros(M,N1);
276 V1 = zeros(M,N1);
277 M1 = zeros(M,N1);
278 x1 = zeros(M,N1);
279 y1 = zeros(M,N1);
280 passedX = zeros(M,N1);
281 theta23 = zeros(M,N1);
282
283 %preallocation of tensors (number of rows = M, number of columns = N1,
284 %number of "pages" = N2)
285 theta2 = zeros(M,N1,N2);
286 theta3 = zeros(M,N1,N2);
287 theta4 = zeros(M,N1,N2);
288 DEV1 = zeros(M,N1,N2);
289 DEV2 = zeros(M,N1,N2);
290 DEV11 = zeros(M,N1,N2);
291 DEV22 = zeros(M,N1,N2);
292 A2 = zeros(M,N1,N2);
293 alpha2 = zeros(M,N1,N2);
294 alpha3 = zeros(M,N1,N2);
295 alpha4 = zeros(M,N1,N2);
296 M2 = zeros(M,N1,N2);

```



```

297 M3 = zeros(M,N1,N2);
298 M4 = zeros(M,N1,N2);
299 F1xt = zeros(M,N1,N2);
300 F1yt = zeros(M,N1,N2);
301 M2lt = zeros(M,N1,N2);
302 M3lt = zeros(M,N1,N2);
303 V2 = zeros(M,N1,N2);
304 V3 = zeros(M,N1,N2);
305 V4 = zeros(M,N1,N2);
306 V = zeros(M,N1,N2);
307 x2 = zeros(M,N1,N2);
308 y2 = zeros(M,N1,N2);
309 x3 = zeros(M,N1,N2);
310 y3 = zeros(M,N1,N2);
311 x4 = zeros(M,N1,N2);
312 y4 = zeros(M,N1,N2);
313 Mtheta12 = zeros(M,N1,N2);
314 theta12P = zeros(M,N1,N2);
315 phi12 = zeros(M,N1,N2);
316 phi12v = zeros(M,N1,N2);
317 l12 = zeros(M,N1,N2);
318 d = zeros(M,N1,N2);
319
320 %-----
321 % figure(1)
322 % hold on
323 % axis equal
324
325 %start a loop throughout all precision points
326 for j = 1:1:M
327 %divide the 90 deg range of motion into equally sized segments
328 alpha(j) = (pi/2)*(j/M);
329
330 %lowerbound of theta1 such that precision point (j) is still reached
331 theta100(j) = alpha(j) - acos((r^2 + l1^2 - (l2 + l3 + l4)^2)/(2*r*l1));
332 %upperbound of theta1 such that precision point (j) is still reached
333 theta1ff(j) = alpha(j) + acos((r^2 + l1^2 - (l2 + l3 + l4)^2)/(2*r*l1));
334
335 %check whether segment 2,3 and 4 can be aligned (stretched arm)
336 if (l1+r) <= (l2+l3+l4)
337 %alternative formulation lowerbound of theta1 if arm cannot be
338 %stretched
339 theta100(j) = alpha(j) - pi;
340 %alternative formulation upperbound of theta1 if arm cannot be
341 %stretched
342 theta1ff(j) = alpha(j) + pi;
343 end
344
345 BEGIN1(j) = theta100(j); %begin interval
346 END1(j) = theta1ff(j); %end interval
347 STEP1(j) = (END1(j)-BEGIN1(j))/N1; %stepsize
348
349 %inserted code needed for prestress here
350 if prestress == 1
351 %lowerbound of theta1
352 theta100pa(j) = alpha(j) - acos((r^2+(l1+l2)^2 - ...
353 (l3+l4)^2)/(2*r*(l1+l2)));
354 %upperbound of theta1
355 theta1ffpa(j) = alpha(j) + acos((r^2+(l1+l2)^2 - ...
356 (l3+l4)^2)/(2*r*(l1+l2)));
357 %define boundaries segment 1 sweep
358 BEGIN1pa(j) = theta100pa(j);
359 END1pa(j) = theta1ffpa(j);
360 %define stepsize segment 1 sweep
361 STEP1pa(j) = (END1pa(j)-BEGIN1pa(j))/N1;
362
363 %calculate angle between segment 2 and 3
364 psi23 = pi - abs(theta3i-theta2i);
365 %calculate length of imaginary connection line between begin of
366 %segment 2 and end of segment 3
367 l23 = sqrt(l2^2 + l3^2 - 2*l2*l3*cos(psi23));
368
369 %lower - and upperbound of segment 1 when spring 2 is engaged and
370 %spring 3 is still making contact with the environment
371 theta100pa2(j) = alpha(j) - acos((l1^2+r^2 - (l23+l4)^2)/(2*r*l1));
372 theta1ffpa2(j) = alpha(j) + acos((l1^2+r^2 - (l23+l4)^2)/(2*r*l1));
373
374 %check whether the imaginary connection line "l23" and segment 4
375 %can be aligned (stretched arm)
376 if (l1+r) <= (l23+l4)
377 %alternative formulation lowerbound of theta1 if arm cannot be
378 %stretched
379 theta100pa2(j) = alpha(j) - pi;
380 %alternative formulation upperbound of theta1 if arm cannot be

```

```

381         %stretched
382         theta1ffpa2(j) = alpha(j) + pi;
383     end
384
385     %define boundaries segment 1 sweep
386     BEGIN1pa2(j) = theta100pa2(j);
387     END1pa2(j) = theta1ffpa2(j);
388     %define stepsize segment 1 sweep
389     STEP1pa2(j) = (END1pa2(j)-BEGIN1pa2(j))/N1;
390 end
391 %
392
393 if prestress == 0
394
395 %loop for segment 1 angle sweep for N1 different angles of segment 1
396 for i = 1:1:N1
397
398 %loop for segment 2 angle sweep for N2 different angles of segment 2
399 for k = 1:1:N2
400
401 %increase angle with steps equal to the stepsize STEP(j)
402 theta1(j,i) = BEGIN1(j) + STEP1(j)*i;
403
404 %the expressions within this loop are valid for theta1 < 0
405 if theta1(j,i) < 0
406     %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
407     theta1n(j,i) = - theta1(j,i);
408
409 %lowerbound and upperbound of segment 2, respectively
410 %for given precision point and angle of segment 1
411 theta20(j,i) = real(-log(-(11*r - ((11*r - 11^2*exp(alpha(j)*1i)*...
412     exp(theta1n(j,i)*1i) + 12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
413     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 14^2*exp(alpha(j)*1i)*...
414     exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
415     2*12*13*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
416     2*12*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
417     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
418     11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))*...
419     (11*r - 11^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
420     12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
421     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
422     14^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
423     r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
424     2*12*13*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
425     2*12*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
426     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
427     11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i)))^(1/2) -...
428     11^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
429     12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
430     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
431     14^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
432     r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
433     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
434     11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))/...
435     (2*(12*r*exp(theta1n(j,i)*1i) -...
436     11*12*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))))*1i);
437
438 theta2f(j,i) = real(-log(-(((11*r - 11^2*exp(alpha(j)*1i)*...
439     exp(theta1n(j,i)*1i) + 12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
440     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
441     14^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
442     r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - 2*12*13*exp(alpha(j)*1i)*...
443     exp(theta1n(j,i)*1i) - 2*12*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i)+...
444     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 11*r*exp(alpha(j)*2i)*...
445     exp(theta1n(j,i)*2i))*(11*r - 11^2*exp(alpha(j)*1i)*...
446     exp(theta1n(j,i)*1i) + 12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
447     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 14^2*exp(alpha(j)*1i)*...
448     exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
449     2*12*13*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
450     2*12*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
451     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
452     11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i)))^(1/2) +...
453     11*r - 11^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
454     12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
455     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 14^2*exp(alpha(j)*1i)*...
456     exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
457     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
458     11*r*exp(alpha(j)*2i)*...
459     exp(theta1n(j,i)*2i))/(2*(12*r*exp(theta1n(j,i)*1i) -...
460     11*12*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))))*1i);
461
462 %compensate for erroneous results due to periodicity of the loop
463 %closure equations
464 if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi

```

```

465     theta2f(j,i) = theta2f(j,i) + 2*pi;
466 end
467
468 %prevent the upperbound of segment 2 from being smaller
469 %than the lowerbound
470 if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
471     theta2f(j,i) = theta2f(j,i) + 2*pi;
472 end
473
474 %define boundaries segment 2 sweep
475 BEGIN2(j,i) = theta20(j,i);
476 END2(j,i) = theta2f(j,i);
477 %define stepsize segment 2 sweep
478 STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;
479
480 %start angle of segment 2 equal to lowerbound,increase with stepsize
481 theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;
482
483 %angle connection line origin and endpoint segment 2
484 Mtheta12(j,i,k) = - atan((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))/...
485     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))));
486
487 %if endpoint of second segment is in Q3
488 if (11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0 &&...
489     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0
490
491     %angle connection line origin and endpoint segment 2
492     Mtheta12(j,i,k) = atan(abs(11*cos(theta1(j,i)) + ...
493         12*cos(theta2(j,i,k)))/...
494         abs(11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))) + pi/2;
495 end
496
497 %length of imaginary connection line between origin and end of segment 2
498 l12(j,i,k) = sqrt((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))^2 + ...
499     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))^2);
500
501 %angle of segment 3 and segment 4
502 %for given precision point & angle segment 1 & angle segment 2
503 theta3(j,i,k) = real(asin((14*sin(log(-(l12(j,i,k)*r + ...
504     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
505     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
506     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
507     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
508     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
509     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - ...
510     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
511     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
512     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
513     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
514     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
515     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
516     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
517     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
518     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
519     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
520     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
521     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - ...
522     112(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) + ...
523     112(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/13));
524
525 theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ...
526     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
527     exp(alpha(j)*1i) + ...
528     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
529     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
530     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
531     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
532     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - ...
533     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
534     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
535     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
536     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
537     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
538     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
539     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
540     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
541     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
542     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
543     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
544     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - ...
545     112(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i);
546
547 %compensate for erroneous results due to periodicity of the loop
548 %closure equations

```

```

549     if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %#ok<COMPNOT>
550         theta4(j,i,k) = 2*pi + real(-log(-(l12(j,i,k)*r +...
551             ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
552             exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
553             exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
554             exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
555             exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i))*...
556             exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
557             exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
558             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
559             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
560             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
561             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
562             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
563             exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - l12(j,i,k)^2*...
564             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
565             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
566             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
567             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
568             exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(14*r*...
569             exp(Mtheta12(j,i,k)*1i) -...
570             l12(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
571     end
572
573     %calculate the deviations in x and y of the coordinates
574     %of the compensator,respectively
575     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
576         l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
577     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
578         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
579
580     %if the absolute value of any of these deviations transcends a
581     %certain threshold, then use alternative formulation for theta3
582     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-12
583         theta3(j,i,k) = pi + real( - asin((14*sin(log(-(l12(j,i,k)*r +...
584             ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
585             exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
586             exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
587             exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
588             exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i))*...
589             exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
590             exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
591             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
592             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
593             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
594             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
595             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
596             exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - l12(j,i,k)^2*...
597             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
598             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
599             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
600             exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
601             exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
602             (2*(14*r*exp(Mtheta12(j,i,k)*1i) -...
603             l12(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
604             l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/13));
605     end
606
607     %if endpoint of second segment is in Q1
608     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
609         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) > 0
610
611         %angle pendulum w.r.t. positive x-axis, (CCW positive)
612         Ar(j) = (pi/2) - alpha(j);
613         %angle segment 1 w.r.t. positive x-axis, (CCW positive)
614         A1(j,i) = (pi/2) - theta1(j,i);
615         %angle segment 2 w.r.t. positive x-axis, (CCW positive)
616         A2(j,i,k) = (pi/2) - theta2(j,i,k);
617         %angle imaginary connection line origin and endpoint segment 2
618         phi12(j,i,k) = atan((l1*sin(A1(j,i)) +...
619             l2*sin(A2(j,i,k)))/(l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
620
621
622         %angle of segment 3 and segment 4
623         %for given precision point & angle segment 1 & angle segment 2
624         theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
625             r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) +...
626             l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i))*...
627             exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
628             l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i))*...
629             exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
630             (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
631             l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
632             l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i))*...

```

```

633     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
634     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
635     112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
636     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
637     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
638     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
639     (2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
640     14*r*exp(phi12(j,i,k)*1i))) *1i)/13));
641
642
643     theta4(j,i,k) = pi/2 - real(-log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
644     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
645     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
646     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
647     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
648     (112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
649     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
650     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
651     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
652     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
653     112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
654     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
655     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
656     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
657     (2*(112(j,i,k)*14*exp(Ar(j)*1i) - 14*r*exp(phi12(j,i,k)*1i))) *1i);
658
659     %calculate the deviations in x and y of the coordinates of the compensator,
660     respectively
661     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
662     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
663     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
664     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
665
666     %if the absolute value of any of these deviations transcends a
667     %certain threshold, then use alternative formulations for
668     %theta3
669     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
670     theta3(j,i,k) = pi/2 - real(pi + acos((112(j,i,k)*...
671     cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
672     14*cos(log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
673     112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
674     112(j,i,k)^2*exp(Ar(j)*1i)*...
675     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
676     exp(phi12(j,i,k)*1i) +...
677     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
678     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
679     exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) +...
680     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
681     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
682     exp(phi12(j,i,k)*1i) +...
683     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
684     exp(phi12(j,i,k)*1i)) + 2*13*14*exp(Ar(j)*1i)*...
685     exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
686     112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
687     112(j,i,k)^2*exp(Ar(j)*1i)*...
688     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*...
689     exp(phi12(j,i,k)*1i) +...
690     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
691     exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
692     14*r*exp(phi12(j,i,k)*1i))) *1i)/13));
693
694     if theta3(j,i,k) < - pi
695     theta3(j,i,k) = 2*pi + pi/2 -...
696     real(pi + acos((112(j,i,k)*cos(phi12(j,i,k)) -...
697     r*cos(Ar(j)) +...
698     14*cos(log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
699     112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
700     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
701     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
702     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
703     r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
704     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
705     (112(j,i,k)*r*exp(Ar(j)*2i) +...
706     112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
707     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
708     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
709     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
710     r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
711     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
712     112(j,i,k)*r*exp(Ar(j)*2i) -...
713     112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
714     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
715     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
716     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...

```

```

716             r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
717             (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
718             l4*r*exp(phi12(j,i,k)*1i)))*1i)/l3));
719         end
720     end
721 end
722
723 %if endpoint of second segment is in Q2
724 if ((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
725     (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0)...
726     || (passedX(j,i) == 1)
727
728     %indicate that the endpoint of second segment passed x-axis
729     passedX(j,i) = 1;
730     %angle pendulum w.r.t. positive x-axis, (CCW positive)
731     Ar(j) = (pi/2) - alpha(j);
732     %angle of segment 1 with respect to positive x-axis (CW positive)
733     theta1p(j,i) = theta1(j,i) - (pi/2);
734     %angle of imaginary connection (between the origin and the
735     %node at the end of the second segment) with respect to
736     %positive x-axis
737     %(clockwise positive)
738     theta12P(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
739     (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);
740
741     if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
742         theta12P(j,i,k) = theta12P(j,i,k) + pi;
743     end
744
745     %angle imaginary connection line origin and endpoint segment 2
746     phi12(j,i,k) = -theta12P(j,i,k);
747
748     %angle of segment 3 and segment 4, for given precision point &
749     %angle segment 1 & angle segment 2
750     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
751     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
752     exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
753     exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
754     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
755     exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
756     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
757     exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
758     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
759     exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
760     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
761     exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
762     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
763     exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
764     exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
765     exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
766     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
767     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
768     exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i - ...
769     l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i - ...
770     l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
771
772     theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
773     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
774     l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
775     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
776     exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
777     exp(theta12P(j,i,k)*1i) + ...
778     l12(j,i,k)*r*exp(Ar(j)*2i)*...
779     exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - ...
780     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
781     l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
782     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
783     exp(theta12P(j,i,k)*1i) + ...
784     2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
785     l12(j,i,k)*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i)))^(1/2) - ...
786     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
787     l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
788     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
789     exp(theta12P(j,i,k)*1i) + ...
790     l12(j,i,k)*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))/...
791     (2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i - ...
792     l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i);
793
794     %calculate the deviations in x and y of the coordinates of the compensator,
795     %respectively
796     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) + ...
797     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
798     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) + ...
799     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

```

```

799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882

```

```

%if the absolute value of any of these deviations transcends a
%certain threshold, then use alternative formulation for theta3
if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
    theta3(j,i,k) = pi + real( - asin((14*sin(log(-(112(j,i,k)*r + ...
        ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i))*...
        exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*2i) + 112(j,i,k)*r*exp(Ar(j)*2i))*...
        exp(theta12P(j,i,k)*2i)))^(1/2) - 112(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i))*...
        exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i))*...
        exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i - ...
        14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))^1i - ...
        112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/13));
end
end
end
end
%the expressions within this loop are valid for theta1 > 0
if theta1(j,i) >= 0
    %angle pendulum w.r.t. positive x-axis, (CCW positive)
    Ar(j) = (pi/2) - alpha(j);
    %angle segment 1 w.r.t. positive x-axis, (CCW positive)
    A1(j,i) = (pi/2) - theta1(j,i);
%lowerbound and upperbound of segment 2, respectively, for given
%precision point and angle of segment 1
theta20(j,i) = (pi/2) - real(-log((( - 11^2*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i) + ...
    12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 13^2*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i) + ...
    14^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i) + ...
    11*r*exp(A1(j,i)*2i) + 11*r*exp(Ar(j)*2i) - ...
    2*12*13*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    2*12*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i))*(- 11^2*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i) + 12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    14^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 11*r*exp(A1(j,i)*2i) + ...
    11*r*exp(Ar(j)*2i) + 2*12*13*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    2*12*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 2*13*14*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i)))^(1/2) - 11^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    14^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 11*r*exp(A1(j,i)*2i) + ...
    11*r*exp(Ar(j)*2i) + 2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i))/...
    (2*(11*12*exp(Ar(j)*1i) - 12*r*exp(A1(j,i)*1i))))*1i);
theta2f(j,i) = (pi/2) - real(-log((( - 11^2*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i) + ...
    12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    14^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 11*r*exp(A1(j,i)*2i) + ...
    11*r*exp(Ar(j)*2i) - 2*12*13*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    2*12*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i))*(- 11^2*exp(A1(j,i)*1i))*...
    exp(Ar(j)*1i) + 12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    14^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 11*r*exp(A1(j,i)*2i) + ...
    11*r*exp(Ar(j)*2i) + 2*12*13*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    2*12*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i)))^(1/2) - ...
    11^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - ...
    12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...
    13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + ...

```

```

883     l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
884     r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) +...
885     l1*r*exp(Ar(j)*2i) + 2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i))/...
886     (2*(l1*l2*exp(Ar(j)*1i) - l2*r*exp(A1(j,i)*1i)))*1i);
887
888     %compensate for erroneous results due to periodicity of the loop
889     %closure equations
890     if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
891         theta2f(j,i) = theta2f(j,i) + 2*pi;
892     end
893
894     %compensate for erroneous results due to periodicity of the loop
895     %closure equations
896     if (i>1) && (theta20(j,i) - theta20(j,i-1)) > pi
897         theta20(j,i) = theta20(j,i) - 2*pi;
898     end
899
900     %prevent the upperbound of segment 2 from being smaller
901     %than the lowerbound
902     if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
903         theta2f(j,i) = theta2f(j,i) + 2*pi;
904     end
905
906     %define boundaries segment 2 sweep
907     BEGIN2(j,i) = theta20(j,i);
908     END2(j,i) = theta2f(j,i);
909     %define stepsize segment 2 sweep
910     STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;
911
912     %start angle of segment 2 equal to lowerbound,increase with stepsize
913     theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;
914
915     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
916     A2(j,i,k) = (pi/2) - theta2(j,i,k);
917
918     %length of imaginary connection line between origin and end of segment 2
919     l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
920         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
921
922     %angle imaginary connection line origin and endpoint segment 2
923     phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
924         (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
925
926     %...and the same angle calculated by using other variables
927     phi12v(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
928         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
929
930     %if the node at the end of the second segment is located beneath the
931     %positive x-axis
932     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0
933         phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
934     end
935
936     %if endpoint of second segment is in Q3
937     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
938         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
939
940         %angle imaginary connection line origin and endpoint segment 2
941         phi12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))/...
942             abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi;
943     end
944
945     %compensate for erroneous results due to periodicity of the loop
946     %closure equations
947     if k>1 && (phi12(j,i,k)-phi12(j,i,k-1)) > pi
948         phi12(j,i,k) = phi12(j,i,k) - 2*pi;
949     end
950
951     %angle of segment 3 and segment 4, for given precision point &
952     %angle segment 1 & angle segment 2
953     theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
954         r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) +...
955         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
956         exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
957         l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
958         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
959         (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
960         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
961         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
962         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
963         exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
964         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
965         exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
966         l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...

```



```

967     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*14*exp(Ar(j)*1i) - ...
968     14*r*exp(phi12(j,i,k)*1i))))*1i)/13));
969
970
971 theta4(j,i,k) = pi/2 - real(-log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + ...
972     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
973     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
974     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
975     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
976     (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
977     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
978     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
979     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
980     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) - ...
981     l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + ...
982     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
983     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
984     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
985     (2*(l12(j,i,k)*14*exp(Ar(j)*1i) - 14*r*exp(phi12(j,i,k)*1i))))*1i);
986
987 if phi12(j,i,k) > pi/2
988     %angle connection line origin and endpoint segment 2
989     Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + ...
990         12*sin(theta2(j,i,k)))/...
991         (l1*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))));
992
993     %if endpoint of second segment is in Q4
994     if (l1*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0 &&...
995         (l1*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0
996
997         %angle connection line origin and endpoint segment 2
998         Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) + ...
999             12*cos(theta2(j,i,k)))/abs(l1*sin(theta1(j,i)) + ...
1000             12*sin(theta2(j,i,k)))) + pi/2;
1001 end
1002
1003 %angle of segment 3 and segment 4, for given precision point &
1004 %angle segment 1 & angle segment 2
1005 theta3(j,i,k) = real(asin((14*sin(log(-(l12(j,i,k)*r + ...
1006     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
1007     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i))*...
1008     exp(alpha(j)*1i) + 14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1009     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1010     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1011     l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
1012     (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
1013     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1014     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1015     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1016     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1017     l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
1018     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1019     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1020     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1021     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1022     l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1023     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - ...
1024     l12(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) + ...
1025     l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/13));
1026
1027 theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
1028     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1029     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1030     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1031     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1032     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1033     l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
1034     (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
1035     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*...
1036     exp(alpha(j)*1i) + 14^2*exp(Mtheta12(j,i,k)*1i)*...
1037     exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
1038     exp(alpha(j)*1i) + 2*13*14*exp(Mtheta12(j,i,k)*1i)*...
1039     exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
1040     exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
1041     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1042     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1043     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1044     l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1045     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - ...
1046     l12(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
1047
1048 %compensate for erroneous results due to periodicity of the loop
1049 %closure equations
1050 if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %ok<COMPNOT>

```

```

1051     theta4(j,i,k) = 2*pi + real(-log(-(l12(j,i,k)*r + ...
1052         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
1053         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
1054         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
1055         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
1056         exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
1057         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
1058         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
1059         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1060         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1061         l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1062         r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1063         2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1064         l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
1065         exp(alpha(j)*2i))^(1/2) - l12(j,i,k)^2*...
1066         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1067         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1068         l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1069         r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1070         l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1071         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - ...
1072         l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
1073 end
1074
1075 %calculate the deviations in x and y of the coordinates of the compensator,
1076 %respectively
1077 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) + ...
1078     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
1079 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) + ...
1080     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
1081
1082 %if the absolute value of any of these deviations transcends a
1083 %certain threshold, then use alternative formulation for theta3
1084 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
1085     theta3(j,i,k) = pi + real(-asin((l4*sin(log(-(l12(j,i,k)*r + ...
1086         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
1087         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
1088         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1089         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1090         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
1091         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
1092         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
1093         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
1094         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1095         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1096         exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
1097         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
1098         exp(theta12P(j,i,k)*2i))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
1099         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
1100         exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
1101         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1102         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
1103         exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i - ...
1104         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i - ...
1105         l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/l3));
1106 end
1107
1108 end
1109
1110 if phi12(j,i,k) < 0
1111     %angle pendulum w.r.t. positive x-axis, (CCW positive)
1112     Ar(j) = (pi/2) - alpha(j);
1113     %angle of pendulum with respect to positive x-axis (CW positive)
1114     theta1p(j,i) = theta1(j,i) - (pi/2);
1115
1116     %angle of imaginary connection (between the origin and the
1117     %node at the end of the second segment) with respect to
1118     %positive x-axis
1119     % (clockwise positive)
1120     theta12P(j,i,k) = - phi12(j,i,k);
1121
1122     %angle of segment 3 and segment 4, for given precision point &
1123     %angle segment 2
1124     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
1125         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
1126         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
1127         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1128         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1129         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
1130         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
1131         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - ...
1132         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
1133         l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1134         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...

```

```

1134     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1135     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1136     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1137     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1138     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
1139     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1140     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1141     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
1142     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i - 112(j,i,k)*...
1143     cos(theta12P(j,i,k)) + r*cos(Ar(j))/13));
1144
1145     theta4(j,i,k) = real(-log(-(112(j,i,k)*r + ((112(j,i,k)*r -...
1146     112(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
1147     13^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1148     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1149     exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1150     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1151     exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
1152     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1153     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1154     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1155     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1156     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1157     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1158     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1159     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
1160     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1161     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1162     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
1163     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i));
1164
1165     %calculate the deviations in x and y of the coordinates of the compensator,
1166     respectively
1167     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
1168     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
1169     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
1170     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
1171
1172     %if the absolute value of any of these deviations transcends a
1173     %certain threshold, then use alternative formulation for theta3
1174     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
1175         theta3(j,i,k) = pi + real(-asin((14*sin(log(-(112(j,i,k)*r +...
1176         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1177         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1178         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1179         exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1180         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1181         exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
1182         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1183         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1184         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1185         exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1186         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1187         exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1188         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1189         exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
1190         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1191         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1192         exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
1193         14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i -...
1194         112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/13));
1195     end
1196
1197     end
1198
1199     %calculate the deviations in x and y of the coordinates of the compensator, respectively
1200     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
1201     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
1202     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
1203     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
1204
1205     %if the absolute value of any of these deviations transcends a
1206     %certain threshold, then use alternative formulation for theta3
1207     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
1208         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((112(j,i,k)*...
1209         cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
1210         14*cos(log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
1211         112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1212         exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1213         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1214         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
1215         (112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1216         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...

```

```

1217     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1218     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1219     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
1220     112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
1221     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1222     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1223     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
1224     (2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
1225     14*r*exp(phi12(j,i,k)*1i)))*/13));
1226
1227     if theta3(j,i,k) > pi
1228         theta3(j,i,k) = pi/2 - real(pi + acos(((112(j,i,k)*...
1229             cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
1230             14*cos(log(-(((112(j,i,k)*r*exp(Ar(j)*2i) +...
1231             112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1232             112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1233             13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1234             14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1235             exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1236             exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) +...
1237             112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1238             112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1239             13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1240             exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1241             2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
1242             112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
1243             112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1244             13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1245             exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
1246             (2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
1247             14*r*exp(phi12(j,i,k)*1i)))*/13));
1248     end
1249 end
1250
1251 %compensate for erroneous results due to periodicity of the loop
1252 %closure equations
1253 if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %#ok<COMPNOT>
1254     theta4(j,i,k) = 2*pi + theta4(j,i,k);
1255 end
1256
1257 end
1258
1259 %in the case of a horizontally positioned segment 1, MATLAB solve() has
1260 %troubles finding a solution... Therefore, perturb by small amount to solve
1261 if theta1(j,i) == pi/2
1262     theta1(j,i) = pi/2 + STEP1(j);
1263 end
1264
1265 %the expressions within this loop are valid for theta1 > pi/2
1266 if theta1(j,i) > pi/2
1267     %angle pendulum w.r.t. positive x-axis, (CCW positive)
1268     Ar(j) = (pi/2) - alpha(j);
1269     %angle of segment 1 with respect to positive x-axis (CW positive)
1270     theta1p(j,i) = theta1(j,i) - (pi/2);
1271
1272 %lowerbound and upperbound of segment 2, respectively,
1273 %for given precision point and angle of segment 1
1274     theta20(j,i) = real(-log(-(11*r - ((11*r - 11^2*exp(Ar(j)*1i)*...
1275         exp(theta1p(j,i)*1i) + 12^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1276         13^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 14^2*exp(Ar(j)*1i)*...
1277         exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
1278         2*12*13*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*12*14*exp(Ar(j)*1i)*...
1279         exp(theta1p(j,i)*1i) + 2*13*14*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1280         11*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(11*r - 11^2*exp(Ar(j)*1i)*...
1281         exp(theta1p(j,i)*1i) + 12^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1282         13^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 14^2*exp(Ar(j)*1i)*...
1283         exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1284         2*12*13*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*12*14*exp(Ar(j)*1i)*...
1285         exp(theta1p(j,i)*1i) + 2*13*14*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1286         11*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))^(1/2) - 11^2*exp(Ar(j)*1i)*...
1287         exp(theta1p(j,i)*1i) - 12^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1288         13^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 14^2*exp(Ar(j)*1i)*...
1289         exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1290         2*13*14*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 11*r*exp(Ar(j)*2i)*...
1291         exp(theta1p(j,i)*2i))/(2*(11*12*exp(Ar(j)*1i)*1i - 12*r*exp(Ar(j)*2i)*...
1292         exp(theta1p(j,i)*1i)*1i));
1293
1294     theta2f(j,i) = real(-log(-(11*r + ((11*r - 11^2*exp(Ar(j)*1i)*...
1295         exp(theta1p(j,i)*1i) + 12^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1296         13^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 14^2*exp(Ar(j)*1i)*...
1297         exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
1298         2*12*13*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*12*14*exp(Ar(j)*1i)*...
1299         exp(theta1p(j,i)*1i) + 2*13*14*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1300         11*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(11*r - 11^2*exp(Ar(j)*1i)*...

```

```

1301     exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1302     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
1303     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1304     2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
1305     exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1306     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))^(1/2) - l1^2*exp(Ar(j)*1i)*...
1307     exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1308     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
1309     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
1310     2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
1311     exp(theta1p(j,i)*2i))/(2*(l1*l2*exp(Ar(j)*1i)*1i - l2*r*exp(Ar(j)*2i)*...
1312     exp(theta1p(j,i)*1i)*1i));
1313
1314     %compensate for erroneous results due to periodicity of the loop
1315     %closure equations
1316     if (i>1) && (theta20(j,i) - theta20(j,i-1)) > pi
1317         theta20(j,i) = theta20(j,i) - 2*pi;
1318     end
1319
1320     %compensate for erroneous results due to periodicity of the loop
1321     %closure equations
1322     if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
1323         theta2f(j,i) = theta2f(j,i) + 2*pi;
1324     end
1325
1326     %prevent the upperbound of segment 2 from being smaller than the lowerbound
1327     if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
1328         theta2f(j,i) = theta2f(j,i) + 2*pi;
1329     end
1330
1331     %define boundaries segment 2 sweep
1332     BEGIN2(j,i) = theta20(j,i);
1333     END2(j,i) = theta2f(j,i);
1334     %define stepsize segment 2 sweep
1335     STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;
1336
1337     %start angle of segment 2 equal to lowerbound,increase with stepsize
1338     theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;
1339
1340     %length of imaginary connection line between origin and end of segment 2
1341     l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
1342         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
1343
1344     %angle of imaginary connection (between the origin and the
1345     %node at the end of the second segment) with respect to positive x-axis
1346     % (clockwise positive)
1347     theta12P(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
1348         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);
1349
1350     if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
1351         theta12P(j,i,k) = theta12P(j,i,k) + pi;
1352     end
1353
1354     %if endpoint of second segment is in Q4
1355     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
1356         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
1357
1358         %angle of imaginary connection (between the origin and the
1359         %node at the end of the second segment) with respect to positive x-axis
1360         % (clockwise positive)
1361         theta12P(j,i,k) = -atan(abs(l1*cos(theta1(j,i)) +...
1362             l2*cos(theta2(j,i,k)))/abs(l1*sin(theta1(j,i)) +...
1363             l2*sin(theta2(j,i,k)))) - pi;
1364     end
1365
1366     %compensate for erroneous results due to periodicity of the loop
1367     %closure equations
1368     if k>1 && abs(theta12P(j,i,k)-theta12P(j,i,k-1)) > pi
1369         theta12P(j,i,k) = theta12P(j,i,k) + 2*pi;
1370     end
1371
1372     %angle imaginary connection line origin and endpoint segment 2
1373     phi12(j,i,k) = -theta12P(j,i,k);
1374
1375     %angle of segment 3 and segment 4, for given precision point &
1376     %angle segment 1 & angle segment 2
1377     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
1378         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
1379         l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1380         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
1381         2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
1382         exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
1383         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
1384         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...

```

```

1385     r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1386     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1387     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1388     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
1389     14^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1390     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1391     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
1392     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)) - 112(j,i,k)*...
1393     cos(theta12P(j,i,k)) + r*cos(Ar(j)))/13);
1394
1395     theta4(j,i,k) = real(-log(-(112(j,i,k)*r + ((112(j,i,k)*r - 112(j,i,k)^2*...
1396     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1397     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
1398     r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1399     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1400     exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1401     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
1402     14^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1403     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
1404     112(j,i,k)*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))^(1/2) -...
1405     112(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1406     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
1407     r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1408     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
1409     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i);
1410
1411     %if endpoint segment 2 is in Q3
1412     if theta12P(j,i,k) <= 0 && theta12P(j,i,k) > -pi/2
1413         %angle pendulum w.r.t. positive x-axis, (CCW positive)
1414         Ar(j) = (pi/2) - alpha(j);
1415         %angle segment 1 w.r.t. positive x-axis, (CCW positive)
1416         A1(j,i) = (pi/2) - theta1(j,i);
1417         %angle segment 2 w.r.t. positive x-axis, (CCW positive)
1418         A2(j,i,k) = (pi/2) - theta2(j,i,k);
1419         %angle imaginary connection line origin and endpoint segment 2
1420         phi12(j,i,k) = atan((11*sin(A1(j,i)) + 12*sin(A2(j,i,k)))/...
1421         (11*cos(A1(j,i)) + 12*cos(A2(j,i,k))));
1422
1423         %angle of segment 3 and segment 4, for given precision point &...
1424         %angle segment 1 & angle segment 2
1425         theta3(j,i,k) = pi/2 - real(pi - acos((112(j,i,k)*cos(phi12(j,i,k)) -...
1426         r*cos(Ar(j)) + 14*cos(log(-(((112(j,i,k)*r*exp(Ar(j)*2i) +...
1427         112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1428         exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1429         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1430         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1431         exp(phi12(j,i,k)*1i))*...
1432         (112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1433         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1434         13^2*exp(Ar(j)*1i)*...
1435         exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1436         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1437         exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
1438         112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
1439         exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1440         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
1441         exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
1442         14*r*exp(phi12(j,i,k)*1i))*1i))/13);
1443
1444         theta4(j,i,k) = pi/2 - real(-log(-(((112(j,i,k)*r*exp(Ar(j)*2i) +...
1445         112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1446         exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1447         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1448         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1449         exp(phi12(j,i,k)*1i))*...
1450         (112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1451         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1452         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1453         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1454         2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
1455         112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
1456         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1457         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1458         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
1459         (2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
1460         14*r*exp(phi12(j,i,k)*1i))*1i));
1461
1462         %compensate for erroneous results due to periodicity of the loop
1463         %closure equations
1464         if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %ok<COMPNOT>
1465             theta4(j,i,k) = 2*pi + pi/2 - real(-log(-(((112(j,i,k)*r*...
1466             exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1467             112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1468             13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...

```

```

1469     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1470     2*13*14*exp(Ar(j)*1i)* exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*...
1471     exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1472     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1473     13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1474     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1475     exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1476     exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*...
1477     exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
1478     112(j,i,k)^2*...
1479     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*...
1480     exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1481     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
1482     (2*(112(j,i,k)*14*exp(Ar(j)*1i) - 14*r*exp(phi12(j,i,k)*1i))*1i));
1483
1484     end
1485
1486     %calculate the deviations in x and y of the coordinates of the compensator,
1487     respectively
1488     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
1489     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
1490     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
1491     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
1492
1493     %if the absolute value of any of these deviations transcends a
1494     %certain threshold, then use alternative formulation for theta3
1495     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
1496         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((112(j,i,k)*...
1497         cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
1498         14*cos(log(-(((112(j,i,k)*r*exp(Ar(j)*2i) +...
1499         112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1500         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1501         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1502         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1503         2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*...
1504         exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1505         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1506         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1507         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1508         2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
1509         112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i)+...
1510         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1511         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1512         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
1513         (2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
1514         14*r*exp(phi12(j,i,k)*1i))*1i))/13));
1515
1516     if theta3(j,i,k) > pi
1517         theta3(j,i,k) = pi/2 - real(pi + acos((112(j,i,k)*...
1518         cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
1519         14*cos(log(-(((112(j,i,k)*r*exp(Ar(j)*2i) +...
1520         112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1521         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1522         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1523         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1524         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1525         2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(112(j,i,k)*...
1526         r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1527         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1528         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1529         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1530         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*13*14*...
1531         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*...
1532         r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
1533         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1534         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1535         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1536         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
1537         (2*(112(j,i,k)*14*exp(Ar(j)*1i) - 14*r*...
1538         exp(phi12(j,i,k)*1i))*1i))/13));
1539
1540     end
1541
1542     end
1543
1544     %if endpoint of second segment is in Q4
1545     if ((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0 &&...
1546     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0)
1547
1548     %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
1549     theta1n(j,i) = - theta1(j,i);
1550
1551     %length of imaginary connection line between origin and end of segment 2
1552     112(j,i,k) = sqrt((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))^2 +...
1553     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))^2);

```

```

1552
1553 %angle connection line origin and endpoint segment 2
1554 Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
1555 (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
1556
1557 %if endpoint of second segment is still in Q4
1558 if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
1559 (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
1560
1561 %angle connection line origin and endpoint segment 2
1562 Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) + ...
1563 l2*cos(theta2(j,i,k)))/abs(l1*sin(theta1(j,i)) + ...
1564 l2*sin(theta2(j,i,k)))) + pi/2;
1565
1566 end
1567
1568 %angle of segment 3 and segment 4
1569 %for given precision point & angle segment 1 & angle segment 2
1570 theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
1571 ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
1572 exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i)...
1573 + l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1574 r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*13*14*...
1575 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1576 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - ...
1577 l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1578 l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
1579 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1580 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
1581 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1582 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) - l12(j,i,k)^2*...
1583 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
1584 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
1585 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1586 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1587 (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
1588 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))*1i) + ...
1589 l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3));
1590
1591 theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
1592 l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1593 l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1594 l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1595 r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1596 2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1597 l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
1598 (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
1599 exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
1600 exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i)...
1601 - r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1602 2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1603 l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2)...
1604 - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1605 l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1606 l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1607 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1608 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(l4*r*...
1609 exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*...
1610 exp(alpha(j)*1i))*1i);
1611
1612 %calculate the deviations in x and y of the coordinates of the compensator,
1613 %respectively
1614 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) + ...
1615 l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
1616 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) + ...
1617 l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
1618
1619 %if the absolute value of any of these deviations transcends a
1620 %certain threshold, then use alternative formulation for theta3
1621 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
1622 theta3(j,i,k) = pi + real(- asin((l4*sin(log(-(l12(j,i,k)*r + ...
1623 ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
1624 exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
1625 exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
1626 exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
1627 exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i)*...
1628 exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
1629 exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
1630 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
1631 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
1632 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1633 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
1634 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1635 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) - ...

```



```

1636     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
1637     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
1638     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
1639     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
1640     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1641     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*...
1642     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) +...
1643     112(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/13));
1644     end
1645
1646 end
1647
1648 %calculate the deviations in x and y of the coordinates of the compensator, respectively
1649 DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
1650     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
1651 DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
1652     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
1653
1654 %if the absolute value of any of these deviations transcends a
1655 %certain threshold, then use alternative formulation for theta3
1656 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
1657     theta3(j,i,k) = pi + real(-asin((14*sin(log(-(112(j,i,k)*r +...
1658         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1659         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1660         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1661         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1662         exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
1663         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1664         exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
1665         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1666         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
1667         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1668         exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
1669         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1670         exp(theta12P(j,i,k)*2i)))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
1671         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
1672         exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
1673         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1674         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
1675         exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
1676         14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i) -...
1677     112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/13));
1678     end
1679
1680 %compensate for erroneous results due to periodicity of the loop
1681 %closure equations
1682 if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %#ok<*COMPNOT>
1683     theta4(j,i,k) = 2*pi + theta4(j,i,k);
1684     end
1685
1686 end
1687
1688 %initial relative angle of segment 1
1689 alpha10 = theta1i;
1690 %initial relative angle of segment 2
1691 alpha20 = theta2i - theta1i;
1692 %initial relative angle of segment 3
1693 alpha30 = theta3i - theta2i;
1694 %initial relative angle of segment 4
1695 alpha40 = theta4i - theta3i;
1696
1697 %angle of rotation torsion spring 1
1698 alpha1(j,i) = theta1(j,i) - alpha10;
1699 %angle of rotation torsion spring 2
1700 alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
1701 %angle of rotation torsion spring 3
1702 alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
1703 %angle of rotation torsion spring 4
1704 alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;
1705
1706 if nonlinearity == 0
1707     %potential energy spring 1
1708     V1(j,i) = ((k1/2)*alpha1(j,i)^2);
1709     %potential energy spring 2
1710     V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2);
1711     %potential energy spring 3
1712     V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2);
1713     %potential energy spring 4
1714     V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
1715     %total potential energy
1716     V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
1717 end
1718
1719 if nonlinearity == 1

```

```

1720 %potential energy spring 1
1721 V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
1722 %potential energy spring 2
1723 V2(j,i,k) = (A/3)*alpha2(j,i,k)^3 + (B/2)*alpha2(j,i,k)^2;
1724 %potential energy spring 3
1725 V3(j,i,k) = (A/3)*alpha3(j,i,k)^3 + (B/2)*alpha3(j,i,k)^2;
1726 %potential energy spring 4
1727 V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
1728 %total potential energy
1729 V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
1730 end
1731
1732 %calculate the deviations in x and y of the coordinates of the compensator, respectively
1733 DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
1734     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
1735 DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
1736     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
1737
1738 %calculate the distance from the endpoint of the second segment to the end
1739 %effector of the inverted pendulum
1740 d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
1741     (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);
1742
1743 %check condition upper loop closure
1744 if (l4-l3-d(j,i,k)) > 0
1745     %set the deviation in x...
1746     DEV11(j,i,k) = 0;
1747     %... and y to zero such that this scenario won't be flagged
1748     DEV22(j,i,k) = 0;
1749     %posture does not exist, so potential energy not a number
1750     V(j,i,k) = NaN;
1751
1752     %define the angles of the third and fourth segment to be no value;
1753     %the surface plots of these tensors (used for debugging) would
1754     %otherwise be nonsmooth
1755     theta3(j,i,k) = NaN;
1756     theta4(j,i,k) = NaN;
1757     %flag this event with variable "Count2" instead
1758     Count2 = Count2 + 1;
1759 end
1760
1761 %if segment 1 and segment 2 are not at their lowerbound
1762 if i>1 && k>1
1763     %if the angle of the third segment was previously - for the same angle
1764     %of the pendulum - NaN, then it will remain NaN for this angle of the
1765     %pendulum (infeasible solution space)
1766     if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)    %#ok<COMPNOP>
1767         theta3(j,i,k) = NaN;
1768
1769         %the potential energy and the angle of segment 4 should
1770         %consequently be NaN as well
1771         V(j,i,k) = NaN;
1772         theta4(j,i,k) = NaN;
1773     end
1774 end
1775
1776 %check condition upper loop closure
1777 if l4-l3+d(j,i,k) < 0
1778     %set the deviation in x...
1779     DEV11(j,i,k) = 0;
1780     %... and y to zero such that this scenario won't be flagged
1781     DEV22(j,i,k) = 0;
1782     %posture doesn't exist, so potential energy not a number
1783     V(j,i,k) = NaN;
1784
1785     %define the angles of the third and fourth segment to be no value;
1786     %the surface plots of these tensors (used for debugging) would
1787     %otherwise be nonsmooth
1788     theta3(j,i,k) = NaN;
1789     theta4(j,i,k) = NaN;
1790     %flag this event with variable "Count3" instead
1791     Count3 = Count3 + 1;
1792 end
1793
1794 %if the absolute value of any of these deviations transcends a
1795 %certain threshold, then increase the variable "Count" by one
1796 if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
1797     Count = Count + 1;
1798 end
1799
1800 %x - coordinate origin (and first spring)
1801 x0 = 0;
1802 %y - coordinate origin (and first spring)
1803 y0 = 0;

```

```

1804 %x - coordinate 2nd spring
1805 x1(j,i) = l1*sin(theta1(j,i));
1806 %y - coordinate 2nd spring
1807 y1(j,i) = l1*cos(theta1(j,i));
1808 %x - coordinate 3rd spring
1809 x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
1810 %y - coordinate 3rd spring
1811 y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
1812 %x - coordinate 4th spring
1813 x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
1814 %y - coordinate 4th spring
1815 y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
1816 %x - coordinate end effector
1817 x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
1818 %y - coordinate end effector
1819 y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));
1820
1821 end
1822 end
1823 end
1824
1825 if prestress == 1
1826
1827 %loop for segment 1 angle sweep for N1 different angles of segment 1
1828 for i = 1:1:N1
1829
1830 %loop for segment 2 angle sweep for N2 different angles of segment 2
1831 for k = 1:1:N2
1832
1833 %increase angle with steps equal to the stepsize STEP1pa(j)
1834 %formulation for balancer with spring 3 enabled, spring 2 locked
1835 theta1sw(j,i) = BEGIN1pa(j) + STEP1pa(j)*i;
1836
1837 %increase angle with steps equal to the stepsize STEP1pa2(j)
1838 %formulation for balancer with spring 2 enabled, spring 3 locked
1839 theta1sw2(j,i) = BEGIN1pa2(j) + STEP1pa2(j)*i;
1840
1841 %increase angle with steps equal to the stepsize STEP1(j)
1842 %formulation for balancer with all springs enabled
1843 theta1fa(j,i) = BEGIN1(j) + STEP1(j)*i;
1844
1845 %if both spring 2 and spring 3 are locked OR
1846 %if both segment 1 and 2 are at their lowerbound and the inverted pendulum
1847 %has made just 1 step
1848 if (M3lt(j,i,k) < M03 && M2lt(j,i,k) < M02) || (j == 1 && i == 1 && k == 1)
1849 %the angle of the first segment increases linearly with the angle of
1850 %the inverted pendulum
1851 theta1(j,i) = alpha(j)+theta1i;
1852 %the angle of the second segment increases linearly with the angle of
1853 %segment 1
1854 theta2(j,i,k) = theta1(j,i) + (theta2i-theta1i);
1855
1856 %the expressions within this loop are valid for theta1 < 0
1857 if theta1(j,i) < 0
1858 %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
1859 theta1n(j,i) = - theta1(j,i);
1860
1861 %angle connection line origin and endpoint segment 2
1862 Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
1863 (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
1864
1865 %length of imaginary connection line between origin and end of segment 2
1866 l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 + ...
1867 (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
1868
1869 %angle of segment 3 and segment 4, for given precision point &
1870 %angle segment 1 & angle segment 2
1871 theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
1872 ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
1873 exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1874 l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1875 r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1876 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1877 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
1878 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
1879 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
1880 exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
1881 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1882 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) - l12(j,i,k)^2*...
1883 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
1884 exp(alpha(j)*1i) - l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
1885 r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1886 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1887 (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...

```

```

1888     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))) *1i) +...
1889     l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3));
1890
1891 theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - l12(j,i,k)^2*...
1892     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
1893     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
1894     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1895     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
1896     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1897     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
1898     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
1899     l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
1900     l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1901     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
1902     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1903     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - l12(j,i,k)^2*...
1904     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
1905     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
1906     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1907     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
1908     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)/...
1909     (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
1910     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))) *1i);
1911
1912 %if the endpoint of segment 2 is located above the x-axis
1913 if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) > 0
1914     %angle pendulum w.r.t. positive x-axis, (CCW positive)
1915     Ar(j) = (pi/2) - alpha(j);
1916     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
1917     A1(j,i) = (pi/2) - theta1(j,i);
1918     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
1919     A2(j,i,k) = (pi/2) - theta2(j,i,k);
1920     %angle imaginary connection line origin and endpoint segment 2
1921     phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
1922     (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
1923
1924 %angle of segment 3 and segment 4
1925 %for given precision point & angle segment 1 & angle segment 2
1926 theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
1927     r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) +...
1928     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
1929     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1930     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1931     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
1932     exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
1933     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
1934     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1935     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1936     exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
1937     exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
1938     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
1939     exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1940     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
1941     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
1942     l4*r*exp(phi12(j,i,k)*1i))) *1i)/l3));
1943
1944 theta4(j,i,k) = pi/2 - real(-log(-((l12(j,i,k)*r*exp(Ar(j)*2i) +...
1945     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
1946     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1947     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
1948     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)*...
1949     (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
1950     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1951     l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1952     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
1953     2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
1954     l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
1955     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
1956     l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
1957     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)/...
1958     (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))) *1i);
1959
1960 end
1961
1962 %calculate the deviations in x and y of the coordinates of the compensator, respectively
1963 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
1964     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
1965 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
1966     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
1967
1968 %if the absolute value of any of these deviations transcends a
1969 %certain threshold, then use alternative formulation for theta3
1970 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-12
1971     theta3(j,i,k) = pi + real(-asin((l4*sin(log(-(l12(j,i,k)*r +...
1972     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...

```

```

1972     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*...
1973     exp(alpha(j)*1i) + 14^2*exp(Mtheta12(j,i,k)*1i)*...
1974     exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
1975     exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i)*...
1976     exp(alpha(j)*1i) + 112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
1977     exp(alpha(j)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
1978     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
1979     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 14^2*...
1980     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1981     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
1982     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
1983     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - 112(j,i,k)^2*...
1984     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
1985     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...
1986     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
1987     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
1988     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
1989     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*...
1990     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))) *1i) +...
1991     112(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/13));
1992 end
1993
1994 end
1995
1996 %the expressions within this loop are valid for theta1 > 0
1997 if theta1(j,i) >= 0
1998     %angle pendulum w.r.t. positive x-axis, (CCW positive)
1999     Ar(j) = (pi/2) - alpha(j);
2000     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
2001     A1(j,i) = (pi/2) - theta1(j,i);
2002     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
2003     A2(j,i,k) = (pi/2) - theta2(j,i,k);
2004
2005     %length of imaginary connection line between origin and end of segment 2
2006     l12(j,i,k) = sqrt((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))^2 +...
2007     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))^2);
2008
2009     %angle imaginary connection line origin and endpoint segment 2
2010     phi12(j,i,k) = atan((11*sin(A1(j,i)) + 12*sin(A2(j,i,k)))/...
2011     (11*cos(A1(j,i)) + 12*cos(A2(j,i,k))));
2012
2013
2014     %...and the same angle calculated by using other variables
2015     phi12v(j,i,k) = atan((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))/...
2016     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))));
2017
2018     %if the node at the end of the second segment is located left to the
2019     %positive y-axis
2020     if (11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0
2021         phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
2022     end
2023
2024     %angle of segment 3 and segment 4, for given precision point &
2025     %angle segment 1 & angle segment 2
2026     theta3(j,i,k) = pi/2 - real(pi - acos((112(j,i,k)*cos(phi12(j,i,k)) -...
2027     r*cos(Ar(j)) + 14*cos(log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
2028     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2029     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2030     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2031     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
2032     (112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
2033     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2034     exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
2035     r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2036     exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
2037     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
2038     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2039     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
2040     exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
2041     14*r*exp(phi12(j,i,k)*1i))) *1i)/13));
2042
2043     theta4(j,i,k) = pi/2 - real(-log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
2044     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2045     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2046     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2047     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
2048     (112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
2049     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2050     exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
2051     r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2052     exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
2053     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
2054     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2055     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...

```

```

2056     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
2057     l4*r*exp(phi12(j,i,k)*1i))))*1i);
2058
2059 % if angle of imaginary connection with respect to positive x-axis
2060 %(clockwise positive) is larger than 90 deg
2061 if phi12(j,i,k) > pi/2
2062     %angle connection line origin and endpoint segment 2
2063     Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + ...
2064     l2*sin(theta2(j,i,k)))/(l1*cos(theta1(j,i)) + ...
2065     l2*cos(theta2(j,i,k))));
2066
2067 %angle of segment 3 and segment 4, for given precision point &
2068 %angle segment 1 & angle segment 2
2069     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
2070     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
2071     exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
2072     exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
2073     exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
2074     exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i))*...
2075     exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
2076     exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2077     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
2078     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
2079     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2080     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
2081     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2082     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) - l12(j,i,k)^2*...
2083     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
2084     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
2085     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2086     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2087     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2088     (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
2089     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) + ...
2090     l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3));
2091
2092     theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
2093     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
2094     l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
2095     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2096     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
2097     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2098     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - ...
2099     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
2100     l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
2101     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2102     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
2103     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2104     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) - ...
2105     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
2106     l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
2107     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2108     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2109     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2110     (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
2111     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
2112 end
2113
2114 % if angle of imaginary connection with respect to positive x-axis
2115 %(clockwise positive) is negative
2116 if phi12(j,i,k) < 0
2117     %angle pendulum w.r.t. positive x-axis, (CCW positive)
2118     Ar(j) = (pi/2) - alpha(j);
2119     %angle of segment 1 with respect to positive x-axis (CW positive)
2120     theta1p(j,i) = theta1(j,i) - (pi/2);
2121
2122 %angle of imaginary connection (between the origin and the
2123 %node at the end of the second segment) with respect to positive x-axis
2124 %(clockwise positive)
2125     theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) + ...
2126     l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);
2127
2128 %angle of segment 3 and segment 4, for given precision point &
2129 %angle segment 1 & angle segment 2
2130     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
2131     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i))*...
2132     exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i))*...
2133     exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i))*...
2134     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i))*...
2135     exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i))*...
2136     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i))*...
2137     exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2138     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i))*...
2139     exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i))*...

```

```

2140     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2141     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2142     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2143     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2144     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2145     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
2146     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2147     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2148     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
2149     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i) - 112(j,i,k)*...
2150     cos(theta12P(j,i,k)) + r*cos(Ar(j))/13));
2151
2152     theta4(j,i,k) = real(-log(-(112(j,i,k)*r + ((112(j,i,k)*r -...
2153     112(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
2154     13^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2155     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2156     exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
2157     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2158     exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
2159     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2160     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2161     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2162     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2163     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2164     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2165     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2166     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
2167     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2168     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2169     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
2170     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i);
2171
2172
2173     %calculate the deviations in x and y of the coordinates of the compensator, respectively
2174     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
2175     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
2176
2177     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
2178     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
2179
2180     %if the absolute value of any of these deviations transcends a
2181     %certain threshold, then use alternative formulation for theta3
2182     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
2183         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((112(j,i,k)*...
2184         cos(phi12(j,i,k)) - r*cos(Ar(j)) + 14*cos(log(-((112(j,i,k)*r*...
2185         exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*...
2186         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2187         exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
2188         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
2189         exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*...
2190         exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2191         exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2192         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2193         exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2194         exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
2195         112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
2196         exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2197         14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
2198         exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
2199         14*r*exp(phi12(j,i,k)*1i))*1i)/13));
2200
2201     end
2202
2203     end
2204
2205     %in the case of a horizontally positioned segment 1, MATLAB solve() has
2206     %troubles finding a solution... Therefore, perturb by small amount to solve
2207     if theta1(j,i) == pi/2
2208         theta1(j,i) = pi/2 + STEP1(j);
2209     end
2210
2211     %the expressions within this loop are valid for theta1 > pi/2
2212     if theta1(j,i) > pi/2
2213         %angle pendulum w.r.t. positive x-axis, (CCW positive)
2214         Ar(j) = (pi/2) - alpha(j);
2215         %angle of segment 1 with respect to positive x-axis (CW positive)
2216         theta1p(j,i) = theta1(j,i) - (pi/2);
2217
2218         %length of imaginary connection line between origin and end of segment 2
2219         l12(j,i,k) = sqrt((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))^2 +...
2220         (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))^2);
2221
2222         %angle of imaginary connection (between the origin and the
2223         %node at the end of the second segment) with respect to positive x-axis
2224         % (clockwise positive)

```

```

2224 theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) + ...
2225     12*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);
2226
2227 %angle of segment 3 and segment 4, for given precision point &
2228 %angle segment 1 & angle segment 2
2229 theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
2230     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
2231     l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2232     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
2233     2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
2234     exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2235     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
2236     exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
2237     r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
2238     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2239     exp(theta12P(j,i,k)*2i))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2240     exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
2241     l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2242     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2243     exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
2244     l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i -...
2245     l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
2246
2247 theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
2248     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
2249     l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2250     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
2251     2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
2252     exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2253     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
2254     exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2255     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2256     exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
2257     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2258     exp(theta12P(j,i,k)*2i))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2259     exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
2260     l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2261     exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2262     exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
2263     l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);
2264
2265 if theta12P(j,i,k) < 0
2266     %angle pendulum w.r.t. positive x-axis, (CCW positive)
2267     Ar(j) = (pi/2) - alpha(j);
2268     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
2269     A1(j,i) = (pi/2) - theta1(j,i);
2270     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
2271     A2(j,i,k) = (pi/2) - theta2(j,i,k);
2272
2273 %angle imaginary connection line origin and endpoint segment 2
2274 phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
2275     (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
2276
2277 %angle of segment 3 and segment 4, for given precision point &
2278 %angle segment 1 & angle segment 2
2279 theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
2280     r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + ...
2281     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2282     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2283     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2284     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
2285     exp(phi12(j,i,k)*1i)))*...
2286     (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
2287     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
2288     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2289     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2290     2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
2291     l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + ...
2292     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
2293     l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2294     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
2295     (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
2296     l4*r*exp(phi12(j,i,k)*1i)))*1i)/l3));
2297
2298 theta4(j,i,k) = pi/2 - real(-log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + ...
2299     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2300     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2301     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2302     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
2303     exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + ...
2304     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2305     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2306     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2307     exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...

```



```

2308     exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
2309     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
2310     exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2311     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
2312     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
2313     l4*r*exp(phi12(j,i,k)*1i)))^1i);
2314
2315     %calculate the deviations in x and y of the coordinates of the compensator,
           respectively
2316     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
2317     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
2318     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
2319     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
2320
2321     %if the absolute value of any of these deviations transcends a
           %certain threshold, then use alternative formulation for theta3
2322     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
2323         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
2324         cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)...
2325         *r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
2326         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2327         l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2328         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2329         exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
2330         exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
2331         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
2332         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
2333         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2334         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2335         exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
2336         exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
2337         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
2338         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
2339         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2340         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
2341         exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
2342         l4*r*exp(phi12(j,i,k)*1i)))^1i)/l3);
2343     end
2344
2345     end
2346
2347     %calculate the deviations in x and y of the coordinates of the compensator, respectively
2348     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
2349     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
2350     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
2351     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
2352
2353     %if the absolute value of any of these deviations transcends a
           %certain threshold, then use alternative formulation for theta3
2354     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
2355         theta3(j,i,k) = pi + real(- asin((l4*sin(log(-l12(j,i,k)*r +...
2356         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2357         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
2358         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2359         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2360         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
2361         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2362         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2363         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
2364         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2365         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2366         exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
2367         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2368         exp(theta12P(j,i,k)*2i))^(1/2) - l12(j,i,k)^2*...
2369         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*...
2370         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
2371         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2372         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
2373         exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
2374         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))^1i -...
2375         l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/l3);
2376     end
2377
2378     end
2379
2380     %calculate the deviations in x and y of the coordinates of the compensator, respectively
2381     DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
2382     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
2383     DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
2384     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
2385
2386     %calculate the distance from the endpoint of the second segment to the end
           %effector of the inverted pendulum
2387     d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...

```

```

2391     (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);
2392
2393 %check condition upper loop closure
2394 if (l4-l3-d(j,i,k)) > 0
2395     %set the deviation in x...
2396     DEV11(j,i,k) = 0;
2397     %... and y to zero such that this scenario won't be flagged
2398     DEV22(j,i,k) = 0;
2399     %posture doesn't exist, so potential energy not a number
2400     V(j,i,k) = NaN;
2401
2402     %define the angles of the third and fourth segment to be no value;
2403     %the surface plots of these tensors (used for debugging) would
2404     %otherwise be nonsmooth
2405     theta3(j,i,k) = NaN;
2406     theta4(j,i,k) = NaN;
2407     %flag this event with variable "Count2" instead
2408     Count2 = Count2 + 1;
2409 end
2410
2411 %if segment 1 and segment 2 are not at their lowerbound
2412 if i>1 && k>1
2413     %if the angle of the third segment was previously - for the same angle
2414     %of the pendulum - NaN, then it will remain NaN for this angle of the
2415     %pendulum (infeasible solution space)
2416     if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)    %#ok<COMPNOP>
2417         theta3(j,i,k) = NaN;
2418
2419         %the potential energy and the angle of segment 4 should
2420         %consequently be NaN as well
2421         V(j,i,k) = NaN;
2422         theta4(j,i,k) = NaN;
2423     end
2424 end
2425
2426 %check condition upper loop closure
2427 if l4-l3+d(j,i,k) < 0
2428     %set the deviation in x...
2429     DEV11(j,i,k) = 0;
2430     %... and y to zero such that this scenario won't be flagged
2431     DEV22(j,i,k) = 0;
2432     %posture doesn't exist, so potential energy not a number
2433     V(j,i,k) = NaN;
2434
2435     %define the angles of the third and fourth segment to be no value;
2436     %the surface plots of these tensors (used for debugging) would
2437     %otherwise be nonsmooth
2438     theta3(j,i,k) = NaN;
2439     theta4(j,i,k) = NaN;
2440     %flag this event with variable "Count3" instead
2441     Count3 = Count3 + 1;
2442 end
2443
2444 %if the absolute value of any of these deviations transcends a
2445 %certain threshold, then increase the variable "Count" by one
2446 if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
2447     Count = Count + 1;
2448 end
2449
2450 %initial relative angle of segment 1
2451 alpha10 = theta1i;
2452 %initial relative angle of segment 2
2453 alpha20 = theta2i - theta1i;
2454 %initial relative angle of segment 3
2455 alpha30 = theta3i - theta2i;
2456 %initial relative angle of segment 4
2457 alpha40 = theta4i - theta3i;
2458
2459 %angle of rotation torsion spring 1
2460 alpha1(j,i) = theta1(j,i) - alpha10;
2461 %angle of rotation torsion spring 2
2462 alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
2463 %angle of rotation torsion spring 3
2464 alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
2465 %angle of rotation torsion spring 4
2466 alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;
2467
2468 if nonlinearity == 0
2469     %internal moment spring 1
2470     M1(j,i) = k1*alpha1(j,i);
2471     %internal moment spring 2
2472     M2(j,i,k) = k2*alpha2(j,i,k) + M02;
2473     %internal moment spring 3
2474     M3(j,i,k) = k3*alpha3(j,i,k) + M03;

```

```

2475 %internal moment spring 4
2476 M4(j,i,k) = k4*alpha4(j,i,k);
2477
2478 %potential energy spring 1
2479 V1(j,i) = ((k1/2)*alpha1(j,i)^2);
2480 %potential energy spring 2
2481 V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) + ...
2482 ((k2/2)*(M02/k2)^2);
2483 %potential energy spring 3
2484 V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) + ...
2485 ((k3/2)*(M03/k3)^2);
2486 %potential energy spring 4
2487 V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
2488 %total potential energy
2489 V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
2490 end
2491
2492 if nonlinearity == 1
2493 %first solution prestress angle: angle of rotation corresponding to
2494 %prestress spring 2
2495 alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
2496 %second solution prestress angle: angle of rotation corresponding to
2497 %prestress spring 2
2498 alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);
2499
2500 %allow only for nonnegative solutions; set to NaN if negative
2501 if alphastar1M2 < 0
2502     alphastar1M2 = NaN;
2503 end
2504
2505 %allow only for nonnegative solutions; set to NaN if negative
2506 if alphastar2M2 < 0
2507     alphastar2M2 = NaN;
2508 end
2509
2510 %store solutions prestress angle in array called "alphastarsM2"
2511 alphastarsM2 = [alphastar1M2, alphastar2M2];
2512
2513 %store the smallest solution for the prestress angle
2514 alphastarM2 = min(abs(alphastarsM2));
2515
2516 %first solution prestress angle: angle of rotation corresponding to
2517 %prestress spring 3
2518 alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
2519 %first solution prestress angle: angle of rotation corresponding to
2520 %prestress spring 3
2521 alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);
2522
2523 %allow only for nonnegative solutions; set to NaN if negative
2524 if alphastar1M3 < 0
2525     alphastar1M3 = NaN;
2526 end
2527
2528 %allow only for nonnegative solutions; set to NaN if negative
2529 if alphastar2M3 < 0
2530     alphastar2M3 = NaN;
2531 end
2532
2533 %store solutions prestress angle in array called "alphastarsM3"
2534 alphastarsM3 = [alphastar1M3, alphastar2M3];
2535
2536 %store the smallest solution for the prestress angle
2537 alphastarM3 = min(abs(alphastarsM3));
2538
2539 %internal moment spring 1
2540 M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
2541 %internal moment spring 2
2542 M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 + ...
2543 B*(alpha2(j,i,k)+alphastarM2);
2544 %internal moment spring 3
2545 M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 + ...
2546 B*(alpha3(j,i,k)+alphastarM3);
2547 %internal moment spring 4
2548 M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);
2549
2550 %potential energy spring 1
2551 V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
2552 %potential energy spring 2
2553 V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 + ...
2554 (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
2555 %potential energy spring 3
2556 V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 + ...
2557 (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
2558 %potential energy spring 4

```

```

2559     V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
2560     %total potential energy
2561     V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
2562 end
2563
2564 %x - coordinate origin (and first spring)
2565 x0 = 0;
2566 %y - coordinate origin (and first spring)
2567 y0 = 0;
2568 %x - coordinate 2nd spring
2569 x1(j,i) = l1*sin(theta1(j,i));
2570 %y - coordinate 2nd spring
2571 y1(j,i) = l1*cos(theta1(j,i));
2572 %x - coordinate 3rd spring
2573 x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
2574 %y - coordinate 3rd spring
2575 y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
2576 %x - coordinate 4th spring
2577 x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
2578 %y - coordinate 4th spring
2579 y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
2580 %x - coordinate end effector
2581 x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
2582 %y - coordinate end effector
2583 y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));
2584
2585 %magnitude reaction force y-direction
2586 F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
2587     (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))/...
2588     (-tan(theta4(j,i,k)))*...
2589     (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))*...
2590     + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));
2591
2592 %magnitude reaction force x-direction
2593 F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
2594     (l4*cos(theta4(j,i,k)));
2595
2596 %external moment on second spring (node 2)
2597 M2lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
2598     F1yt(j,i,k)*l1*sin(theta1(j,i));
2599
2600 %external moment on third spring (node 3)
2601 M3lt(j,i,k) = M1(j,i) +...
2602     F1xt(j,i,k)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))) -...
2603     F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
2604 end
2605
2606 %if spring 3 is activated and spring 2 is still locked
2607 if M3lt(j,i,k) >= M03 && M2lt(j,i,k) < M02
2608
2609     %formulation for angle segment 1 with spring 3 enabled, spring 2 locked
2610     theta1(j,i) = theta1sw(j,i);
2611
2612     %the angle of the second segment increases linearly with the angle of
2613     %the first segment
2614     theta2(j,i,k) = theta1(j,i) + (theta2i-theta1i);
2615
2616     %the expressions within this loop are valid for theta1 < 0
2617     if theta1(j,i) < 0
2618         %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
2619         theta1n(j,i) = - theta1(j,i);
2620
2621         %angle connection line origin and endpoint segment 2
2622         Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
2623             /(l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
2624
2625         %length of imaginary connection line between origin and end of segment 2
2626         l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
2627             (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
2628
2629         %angle of segment 3 and segment 4, for given precision point &
2630         %angle segment 1 & angle segment 2
2631         theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
2632             ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
2633             exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2634             l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2635             r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2636             2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2637             l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
2638             (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
2639             exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2640             l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2641             r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2642             2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...

```

```

2643     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) -...
2644     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2645     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2646     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2647     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2648     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2649     (2*(14*r*exp(Mtheta12(j,i,k)*1i) -...
2650     112(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) +...
2651     112(j,i,k)*sin(Mtheta12(j,i,k) + r*sin(alpha(j)))/13);
2652
2653 theta4(j,i,k) = real(-log(-(112(j,i,k)*r +...
2654     ((112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
2655     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2656     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2657     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2658     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2659     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
2660     (112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
2661     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2662     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2663     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2664     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2665     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)).^(1/2) -...
2666     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2667     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2668     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2669     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2670     112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2671     (2*(14*r*exp(Mtheta12(j,i,k)*1i) -...
2672     112(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i);
2673
2674 %if the endpoint of segment 2 is located above the x-axis
2675 if (11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) > 0
2676 %angle pendulum w.r.t. positive x-axis, (CCW positive)
2677 Ar(j) = (pi/2) - alpha(j);
2678 %angle segment 1 w.r.t. positive x-axis, (CCW positive)
2679 A1(j,i) = (pi/2) - theta1(j,i);
2680 %angle segment 2 w.r.t. positive x-axis, (CCW positive)
2681 A2(j,i,k) = (pi/2) - theta2(j,i,k);
2682 %angle imaginary connection line origin and endpoint segment 2
2683 phi12(j,i,k) = atan((11*sin(A1(j,i)) + 12*sin(A2(j,i,k)))/...
2684     (11*cos(A1(j,i)) + 12*cos(A2(j,i,k))));
2685
2686 %angle of segment 3 and segment 4, for given precision point &
2687 %angle segment 1 & angle segment 2
2688 theta3(j,i,k) = pi/2 - real(pi - acos((112(j,i,k)*...
2689     cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
2690     14*cos(log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
2691     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2692     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2693     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2694     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
2695     exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) +...
2696     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2697     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2698     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2699     exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2700     exp(phi12(j,i,k)*1i)).^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
2701     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
2702     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2703     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
2704     exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
2705     14*r*exp(phi12(j,i,k)*1i)))*1i)/13);
2706
2707 theta4(j,i,k) = pi/2 - real(-log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
2708     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2709     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2710     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2711     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
2712     exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) +...
2713     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2714     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2715     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2716     exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2717     exp(phi12(j,i,k)*1i)).^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
2718     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
2719     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2720     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2721     r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
2722     (2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
2723     14*r*exp(phi12(j,i,k)*1i)))*1i);
2724
2725 end
2726 %calculate the deviations in x and y of the coordinates of the compensator, respectively

```

```

2727 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
2728         l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
2729
2730 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
2731         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
2732
2733 %if the absolute value of any of these deviations transcends a
2734 %certain threshold, then use alternative formulation for theta3
2735 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-12
2736     theta3(j,i,k) = pi + real( - asin((l4*sin(log(-(l12(j,i,k)*r +...
2737         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
2738         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
2739         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
2740         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
2741         exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i))*...
2742         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
2743         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2744         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2745         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2746         l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2747         r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2748         2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2749         l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
2750         l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2751         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2752         l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
2753         r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
2754         l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2755         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
2756         l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
2757         l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3));
2758     end
2759 end
2760
2761 %the expressions within this loop are valid for theta1 > 0
2762 if theta1(j,i) >= 0
2763     %angle pendulum w.r.t. positive x-axis, (CCW positive)
2764     Ar(j) = (pi/2) - alpha(j);
2765     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
2766     A1(j,i) = (pi/2) - theta1(j,i);
2767
2768     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
2769     A2(j,i,k) = (pi/2) - theta2(j,i,k);
2770
2771     %length of imaginary connection line between origin and end of segment 2
2772     l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
2773         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
2774
2775     %angle imaginary connection line origin and endpoint segment 2
2776     phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
2777         (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
2778
2779     %...and the same angle calculated by using other variables
2780     phi12v(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
2781         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
2782
2783     %if the node at the end of the second segment is located beneath the
2784     %positive x-axis
2785     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0
2786         phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
2787     end
2788
2789     %angle of segment 3 and segment 4, for given precision point &
2790     %angle segment 1 & angle segment 2
2791     theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
2792         r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
2793         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i))*...
2794         exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2795         l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i))*...
2796         exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
2797         (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
2798         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2799         l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i))*...
2800         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2801         2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*...
2802         r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
2803         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
2804         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
2805         r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
2806         (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
2807         l4*r*exp(phi12(j,i,k)*1i))))*1i)/l3));
2808
2809     theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...

```

```

2811     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2812     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2813     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2814     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
2815     (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
2816     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2817     l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2818     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2819     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*...
2820     r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + ...
2821     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
2822     l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
2823     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
2824     (2*(l12(j,i,k)*14*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i)))*1i);
2825
2826 if phi12(j,i,k) > pi/2
2827     %angle connection line origin and endpoint segment 2
2828     Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + ...
2829         l2*sin(theta2(j,i,k)))/...
2830         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
2831
2832     %angle of segment 3 and segment 4, for given precision point &
2833     %angle segment 1 & angle segment 2
2834     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
2835         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
2836         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
2837         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
2838         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
2839         exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i))*...
2840         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
2841         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2842         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
2843         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
2844         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2845         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
2846         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2847         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - ...
2848         l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
2849         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
2850         l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
2851         r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
2852         l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2853         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*14*...
2854         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) + ...
2855         l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3));
2856
2857
2858     theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ...
2859         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
2860         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
2861         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
2862         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
2863         exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i))*...
2864         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
2865         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
2866         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
2867         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
2868         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2869         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
2870         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2871         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - l12(j,i,k)^2*...
2872         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
2873         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
2874         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
2875         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
2876         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
2877         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - ...
2878         l12(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i);
2879 end
2880
2881 if phi12(j,i,k) < 0
2882     %angle pendulum w.r.t. positive x-axis, (CCW positive)
2883     Ar(j) = (pi/2) - alpha(j);
2884     %angle of segment 1 with respect to positive x-axis (CW positive)
2885     theta1p(j,i) = theta1(j,i) - (pi/2);
2886     %angle of imaginary connection (between the origin and the
2887     %node at the end of the second segment) with respect to
2888     %positive x-axis
2889     %(clockwise positive)
2890     theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) + ...
2891         l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);
2892
2893     %angle of segment 3 and segment 4, for given precision point &
2894     %angle segment 1 & angle segment 2

```

```

2895     theta3(j,i,k) = real(asin((14*sin(log(-(112(j,i,k)*r + ...
2896         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2897         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2898         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2899         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2900         exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
2901         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2902         exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
2903         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2904         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2905         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2906         exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2907         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2908         exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2909         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2910         exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
2911         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2912         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2913         exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i - ...
2914         14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i) - ...
2915         112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/13));
2916
2917     theta4(j,i,k) = real(-log(-(112(j,i,k)*r + ...
2918         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2919         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2920         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2921         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2922         exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
2923         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2924         exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
2925         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2926         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2927         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2928         exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
2929         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2930         exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
2931         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
2932         exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
2933         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
2934         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
2935         exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*...
2936         exp(Ar(j)*1i)*1i - 14*r*exp(Ar(j)*2i)*...
2937         exp(theta12P(j,i,k)*1i)*1i))*1i);
2938
2939     end
2940
2941     %calculate the deviations in x and y of the coordinates of the compensator, respectively
2942     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) + ...
2943         13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
2944
2945     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) + ...
2946         13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
2947
2948     %if the absolute value of any of these deviations transcends a
2949     %certain threshold, then use alternative formulation for theta3
2950     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
2951         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((112(j,i,k)*...
2952         cos(phi12(j,i,k)) - r*cos(Ar(j)) + 14*cos(log(-((112(j,i,k)*...
2953         r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
2954         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2955         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2956         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
2957         2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*...
2958         exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
2959         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2960         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2961         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
2962         2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) - ...
2963         112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) + ...
2964         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
2965         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
2966         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
2967         (2*(112(j,i,k)*14*exp(Ar(j)*1i) - ...
2968         14*r*exp(phi12(j,i,k)*1i))*1i))/13));
2969
2970     end
2971
2972     end
2973
2974     %in the case of a horizontally positioned segment 1, MATLAB solve() has
2975     %troubles finding a solution... Therefore, perturb by small amount to solve
2976     if theta1(j,i) == pi/2
2977         theta1(j,i) = pi/2 + STEP1(j);
2978     end
2979
2980     %the expressions within this loop are valid for theta1 > pi/2

```



```

2979 if theta1(j,i) > pi/2
2980     %angle pendulum w.r.t. positive x-axis, (CCW positive)
2981     Ar(j) = (pi/2) - alpha(j);
2982     %angle of segment 1 with respect to positive x-axis (CW positive)
2983     theta1p(j,i) = theta1(j,i) - (pi/2);
2984
2985     %length of imaginary connection line between origin and end of segment 2
2986     l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 + ...
2987         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
2988
2989     %angle of imaginary connection (between the origin and the
2990     %node at the end of the second segment) with respect to positive x-axis
2991     % (clockwise positive)
2992     theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) + ...
2993         l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);
2994
2995     %angle of segment 3 and segment 4, for given precision point &
2996     %angle segment 1 & angle segment 2
2997     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ...
2998         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
2999         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
3000         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
3001         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3002         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
3003         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3004         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
3005         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
3006         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
3007         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3008         exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
3009         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3010         exp(theta12P(j,i,k)*2i))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
3011         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
3012         exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
3013         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3014         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3015         exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i - ...
3016         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i - ...
3017         l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
3018
3019     theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - ...
3020         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
3021         l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
3022         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3023         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
3024         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3025         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
3026         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
3027         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
3028         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3029         exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
3030         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3031         exp(theta12P(j,i,k)*2i))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
3032         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
3033         exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
3034         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3035         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3036         exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i - ...
3037         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i);
3038
3039     if theta12P(j,i,k) < 0
3040         %angle pendulum w.r.t. positive x-axis, (CCW positive)
3041         Ar(j) = (pi/2) - alpha(j);
3042         %angle segment 1 w.r.t. positive x-axis, (CCW positive)
3043         A1(j,i) = (pi/2) - theta1(j,i);
3044         %angle segment 2 w.r.t. positive x-axis, (CCW positive)
3045         A2(j,i,k) = (pi/2) - theta2(j,i,k);
3046         %angle imaginary connection line origin and endpoint segment 2
3047         phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
3048             (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
3049
3050         %angle of segment 3 and segment 4, for given precision point &...
3051         %angle segment 1 & angle segment 2
3052         theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*...
3053             cos(phi12(j,i,k)) - r*cos(Ar(j)) + ...
3054             l4*cos(log(-(l12(j,i,k)*r*exp(Ar(j)*2i) + ...
3055             l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
3056             exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
3057             l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3058             exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
3059             exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + ...
3060             l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
3061             exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
3062             l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...

```

```

3063     exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
3064     exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
3065     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
3066     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
3067     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
3068     exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
3069     14*r*exp(phi12(j,i,k)*1i)))^1i)/13));
3070
3071     theta4(j,i,k) = pi/2 - real(-log(-((112(j,i,k)*r*exp(Ar(j)*2i) +...
3072     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
3073     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
3074     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3075     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
3076     exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) +...
3077     112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
3078     exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
3079     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3080     exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
3081     exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
3082     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
3083     exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
3084     14^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
3085     exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
3086     14*r*exp(phi12(j,i,k)*1i)))^1i);
3087
3088     %calculate the deviations in x and y of the coordinates of the compensator,
3089     %respectively
3090     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
3091     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
3092     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
3093     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
3094
3095     %if the absolute value of any of these deviations transcends a
3096     %certain threshold, then use alternative formulation for theta3
3097     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
3098         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((112(j,i,k)*...
3099         cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
3100         14*cos(log(-((112(j,i,k)*r*exp(Ar(j)*2i) + 112(j,i,k)*r*...
3101         exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
3102         exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
3103         exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
3104         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3105         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
3106         exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*exp(Ar(j)*2i) +...
3107         112(j,i,k)*r*exp(phi12(j,i,k)*2i) - 112(j,i,k)^2*...
3108         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
3109         exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
3110         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3111         exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
3112         exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*r*exp(Ar(j)*2i) -...
3113         112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*...
3114         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*...
3115         exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
3116         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
3117         exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
3118         14*r*exp(phi12(j,i,k)*1i)))^1i)/13));
3119     end
3120
3121     end
3122
3123     %calculate the deviations in x and y of the coordinates of the compensator, respectively
3124     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
3125     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
3126
3127     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
3128     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
3129
3130     %if the absolute value of any of these deviations transcends a
3131     %certain threshold, then use alternative formulation for theta3
3132     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
3133         theta3(j,i,k) = pi + real(- asin((14*sin(log(-112(j,i,k)*r +...
3134         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
3135         exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
3136         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
3137         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3138         exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
3139         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
3140         exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
3141         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
3142         exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
3143         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3144         exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
3145         exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
3146         exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...

```

```

3146         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
3147         exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
3148         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
3149         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
3150         exp(theta12P(j,i,k)*2i)/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
3151         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i -...
3152         l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/l3);
3153     end
3154 end
3155 end
3156
3157 %calculate the deviations in x and y of the coordinates of the compensator, respectively
3158 DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
3159         l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
3160
3161 DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
3162         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
3163
3164 %calculate the distance from the endpoint of the second segment to the end
3165 %effector of the inverted pendulum
3166 d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
3167         (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);
3168
3169 %check condition upper loop closure
3170 if (l4-l3-d(j,i,k)) > 0
3171     %set the deviation in x...
3172     DEV11(j,i,k) = 0;
3173     %... and y to zero such that this scenario won't be flagged
3174     DEV22(j,i,k) = 0;
3175     %posture doesn't exist, so potential energy not a number
3176     V(j,i,k) = NaN;
3177
3178     %define the angles of the third and fourth segment to be no value;
3179     %the surface plots of these tensors (used for debugging) would
3180     %otherwise be nonsmooth
3181     theta3(j,i,k) = NaN;
3182     theta4(j,i,k) = NaN;
3183     %flag this event with variable "Count2" instead
3184     Count2 = Count2 + 1;
3185 end
3186
3187 %if segment 1 and segment 2 are not at their lowerbound
3188 if i>1 && k>1
3189     %if the angle of the third segment was previously - for the same angle
3190     %of the pendulum - NaN, then it will remain NaN for this angle of the
3191     %pendulum (infeasible solution space)
3192     if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)    %#ok<COMPNOP>
3193         theta3(j,i,k) = NaN;
3194
3195         %the potential energy and the angle of segment 4 should
3196         %consequently be NaN as well
3197         V(j,i,k) = NaN;
3198         theta4(j,i,k) = NaN;
3199     end
3200 end
3201
3202 %check condition upper loop closure
3203 if l4-l3+d(j,i,k) < 0
3204     %set the deviation in x...
3205     DEV11(j,i,k) = 0;
3206     %... and y to zero such that this scenario won't be flagged
3207     DEV22(j,i,k) = 0;
3208     %posture doesn't exist, so potential energy not a number
3209     V(j,i,k) = NaN;
3210
3211     %define the angles of the third and fourth segment to be no value;
3212     %the surface plots of these tensors (used for debugging) would
3213     %otherwise be nonsmooth
3214     theta3(j,i,k) = NaN;
3215     theta4(j,i,k) = NaN;
3216     %flag this event with variable "Count3" instead
3217     Count3 = Count3 + 1;
3218 end
3219
3220 %if the absolute value of any of these deviations transcends a
3221 %certain threshold, then increase the variable "Count" by one
3222 if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
3223     Count = Count + 1;
3224 end
3225
3226 %initial relative angle of segment 1
3227 alpha10 = theta1i;
3228 %initial relative angle of segment 2
3229 alpha20 = theta2i - theta1i;

```

```

3230 %initial relative angle of segment 3
3231 alpha30 = theta3i - theta2i;
3232 %initial relative angle of segment 4
3233 alpha40 = theta4i - theta3i;
3234
3235 %angle of rotation torsion spring 1
3236 alpha1(j,i) = theta1(j,i) - alpha10;
3237 %angle of rotation torsion spring 2
3238 alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
3239 %angle of rotation torsion spring 3
3240 alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
3241 %angle of rotation torsion spring 4
3242 alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;
3243
3244 if nonlinearity == 0
3245     %internal moment spring 1
3246     M1(j,i) = k1*alpha1(j,i);
3247     %internal moment spring 2
3248     M2(j,i,k) = k2*alpha2(j,i,k) + M02;
3249     %internal moment spring 3
3250     M3(j,i,k) = k3*alpha3(j,i,k) + M03;
3251     %internal moment spring 4
3252     M4(j,i,k) = k4*alpha4(j,i,k);
3253
3254     %potential energy spring 1
3255     V1(j,i) = ((k1/2)*alpha1(j,i)^2);
3256     %potential energy spring 2
3257     V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) + ...
3258         ((k2/2)*(M02/k2)^2);
3259     %potential energy spring 3
3260     V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) + ...
3261         ((k3/2)*(M03/k3)^2);
3262     %potential energy spring 4
3263     V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
3264     %total potential energy
3265     V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
3266 end
3267
3268 if nonlinearity == 1
3269     %first solution prestress angle: angle of rotation corresponding to
3270     %prestress spring 2
3271     alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
3272     %second solution prestress angle: angle of rotation corresponding to
3273     %prestress spring 2
3274     alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);
3275
3276     %allow only for nonnegative solutions; set to NaN if negative
3277     if alphastar1M2 < 0
3278         alphastar1M2 = NaN;
3279     end
3280
3281     %allow only for nonnegative solutions; set to NaN if negative
3282     if alphastar2M2 < 0
3283         alphastar2M2 = NaN;
3284     end
3285
3286     %store solutions prestress angle in array called "alphastarsM2"
3287     alphastarsM2 = [alphastar1M2,alphastar2M2];
3288
3289     %store the smallest solution for the prestress angle
3290     alphastarM2 = min(abs(alphastarsM2));
3291
3292     %first solution prestress angle: angle of rotation corresponding to
3293     %prestress spring 3
3294     alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
3295
3296     %first solution prestress angle: angle of rotation corresponding to
3297     %prestress spring 3
3298     alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);
3299
3300     %allow only for nonnegative solutions; set to NaN if negative
3301     if alphastar1M3 < 0
3302         alphastar1M3 = NaN;
3303     end
3304
3305     %allow only for nonnegative solutions; set to NaN if negative
3306     if alphastar2M3 < 0
3307         alphastar2M3 = NaN;
3308     end
3309
3310     %store solutions prestress angle in array called "alphastarsM3"
3311     alphastarsM3 = [alphastar1M3,alphastar2M3];
3312
3313     %store the smallest solution for the prestress angle

```

```

3314     alphastarM3 = min(abs(alphastarsM3));
3315
3316     %internal moment spring 1
3317     M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
3318     %internal moment spring 2
3319     M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 + ...
3320         B*(alpha2(j,i,k)+alphastarM2);
3321     %internal moment spring 3
3322     M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 + ...
3323         B*(alpha3(j,i,k)+alphastarM3);
3324     %internal moment spring 4
3325     M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);
3326
3327     %potential energy spring 1
3328     V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
3329     %potential energy spring 2
3330     V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 + ...
3331         (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
3332     %potential energy spring 3
3333     V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 + ...
3334         (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
3335     %potential energy spring 4
3336     V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
3337     %total potential energy
3338     V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
3339 end
3340
3341 %allow for nonnegative rotation of spring 2 only
3342 if alpha2(j,i,k) < 0
3343     V(j,i,k) = NaN;
3344 end
3345
3346 %allow for nonnegative rotation of spring 3 only
3347 if alpha3(j,i,k) < 0
3348     V(j,i,k) = NaN;
3349 end
3350
3351 %x - coordinate origin (and first spring)
3352 x0 = 0;
3353 %y - coordinate origin (and first spring)
3354 y0 = 0;
3355 %x - coordinate 2nd spring
3356 x1(j,i) = l1*sin(theta1(j,i));
3357 %y - coordinate 2nd spring
3358 y1(j,i) = l1*cos(theta1(j,i));
3359 %x - coordinate 3rd spring
3360 x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
3361 %y - coordinate 3rd spring
3362 y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
3363 %x - coordinate 4th spring
3364 x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
3365 %y - coordinate 4th spring
3366 y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
3367 %x - coordinate end effector
3368 x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
3369 %y - coordinate end effector
3370 y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));
3371
3372 %magnitude reaction force y-direction
3373 F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
3374     (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))/...
3375     (-tan(theta4(j,i,k))*(l1*cos(theta1(j,i)))+...
3376     l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))...
3377     + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));
3378
3379 %magnitude reaction force x-direction
3380 F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
3381     (l4*cos(theta4(j,i,k)));
3382
3383 %external moment on second spring (node 2)
3384 M2lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) - ...
3385     F1yt(j,i,k)*l1*sin(theta1(j,i));
3386
3387 %external moment on third spring (node 3)
3388 M3lt(j,i,k) = M1(j,i) + ...
3389     F1xt(j,i,k)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))) - ...
3390     F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
3391 end
3392
3393 %if spring 2 is activated and spring 3 is still locked
3394 if M3lt(j,i,k) < M03 && M2lt(j,i,k) >= M02
3395
3396     %formulation for angle segment 1 with spring 2 enabled, spring 3 locked
3397     theta1(j,i) = theta1sw2(j,i);

```

```

3398
3399 %angle of imaginary connection (between the node 2 and node 4)
3400 %with respect to the the second segment
3401 phi232 = acos((l2^2 + l23^2 - l3^2)/(2*l2*l23));
3402
3403 %the expressions within this loop are valid for theta1 < 0
3404 if theta1(j,i) < 0
3405     %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
3406     theta1n(j,i) = - theta1(j,i);
3407
3408 %formulation for theta4: elbow up
3409 theta4(j,i,k) = real(-log(-(l1*r + ((l1*r - l1^2*exp(theta1n(j,i)*1i))*...
3410     exp(alpha(j)*1i) + l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3411     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3412     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3413     2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3414     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))*...
3415     (l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3416     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3417     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3418     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3419     2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3420     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
3421     l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3422     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3423     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3424     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3425     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))/...
3426     (2*(l4*r*exp(theta1n(j,i)*1i) -...
3427     l1*l4*exp(theta1n(j,i)*2i)*exp(alpha(j)*1i))*1i);
3428
3429 %formulation for theta2: elbow up
3430 theta23(j,i) = real(asin((l4*sin(log(-(l1*r +...
3431     ((l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3432     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3433     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3434     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3435     2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3436     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))*(l1*r -...
3437     l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3438     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3439     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3440     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) + 2*l23*l4*...
3441     exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) + l1*r*exp(theta1n(j,i)*2i)*...
3442     exp(alpha(j)*2i)))^(1/2) - l1^2*exp(theta1n(j,i)*1i)*...
3443     exp(alpha(j)*1i) + l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3444     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) - r^2*...
3445     exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) + l1*r*exp(theta1n(j,i)*2i)*...
3446     exp(alpha(j)*2i))/(2*(l4*r*exp(theta1n(j,i)*1i) -...
3447     l1*l4*exp(theta1n(j,i)*2i)*exp(alpha(j)*1i))*1i) +...
3448     l1*sin(theta1n(j,i)) + r*sin(alpha(j))/l23));
3449
3450 theta2(j,i,k) = theta23(j,i) + phi232;
3451 theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
3452
3453 %calculate the deviations in x and y of the coordinates of the compensator, respectively
3454 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
3455     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
3456 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
3457     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
3458
3459 %if the absolute value of any of these deviations transcends a
3460 %certain threshold, then use alternative formulation for theta3
3461 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-12
3462     theta23(j,i) = pi + real( - asin((l4*sin(log(-(l1*r +...
3463     ((l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3464     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3465     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3466     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3467     2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3468     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))*...
3469     (l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3470     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3471     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3472     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3473     2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3474     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
3475     l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3476     l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3477     l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
3478     r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
3479     l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))/...
3480     (2*(l4*r*exp(theta1n(j,i)*1i) - l1*l4*exp(theta1n(j,i)*2i)*...
3481     exp(alpha(j)*1i))*1i) +...

```

```

3482         l1*sin(theta1n(j,i) + r*sin(alpha(j)))/l23));
3483
3484     theta2(j,i,k) = theta23(j,i) + phi232;
3485     theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
3486 end
3487
3488 end
3489
3490 %the expressions within this loop are valid for theta1 > 0
3491 if theta1(j,i) >= 0
3492     %angle pendulum w.r.t. positive x-axis, (CCW positive)
3493     Ar(j) = (pi/2) - alpha(j);
3494     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
3495     A1(j,i) = (pi/2) - theta1(j,i);
3496
3497     %formulation for theta4: elbow up
3498     theta4(j,i,k) = pi/2 - real(-log(-((l1*r*exp(Ar(j)*2i) +...
3499         l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3500         l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
3501         exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
3502         2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i))*(l1*r*exp(Ar(j)*2i) +...
3503         l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3504         l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
3505         exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3506         2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i)))^(1/2) -...
3507         l1*r*exp(Ar(j)*2i) - l1*r*exp(A1(j,i)*2i) + l1^2*exp(Ar(j)*1i)*...
3508         exp(A1(j,i)*1i) - l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3509         l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + r^2*exp(Ar(j)*1i)*...
3510         exp(A1(j,i)*1i))/...
3511         (2*(l1*l4*exp(Ar(j)*1i) - l4*r*exp(A1(j,i)*1i))))*1i);
3512
3513     %formulation for theta2 and theta3: elbow up
3514     theta23(j,i) = pi/2 - real(pi - acos((l1*cos(A1(j,i)) - r*cos(Ar(j)) +...
3515         l4*cos(log(-((l1*r*exp(Ar(j)*2i) + l1*r*exp(A1(j,i)*2i) -...
3516         l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l23^2*exp(Ar(j)*1i)*...
3517         exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
3518         r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) - 2*l23*l4*exp(Ar(j)*1i)*...
3519         exp(A1(j,i)*1i))*(l1*r*exp(Ar(j)*2i) + l1*r*exp(A1(j,i)*2i) -...
3520         l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l23^2*exp(Ar(j)*1i)*...
3521         exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
3522         r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*...
3523         exp(A1(j,i)*1i)))^(1/2) - l1*r*exp(Ar(j)*2i) -...
3524         l1*r*exp(A1(j,i)*2i) + l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
3525         l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
3526         exp(A1(j,i)*1i) + r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i))/...
3527         (2*(l1*l4*exp(Ar(j)*1i) - l4*r*exp(A1(j,i)*1i))))*1i)/l23));
3528
3529     theta2(j,i,k) = theta23(j,i) + phi232;
3530     theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
3531
3532     %calculate the deviations in x and y of the coordinates of the compensator, respectively
3533     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
3534         l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
3535
3536     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
3537         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
3538
3539     %if the absolute value of any of these deviations transcends a
3540     %certain threshold, then use alternative formulation for theta2 and
3541     %theta3
3542     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
3543         theta23(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(A1(j,i)) -...
3544             r*cos(Ar(j)) + l4*cos(log(-((l1*r*exp(Ar(j)*2i) +...
3545             l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3546             l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
3547             exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
3548             2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i))*(l1*r*exp(Ar(j)*2i) +...
3549             l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3550             l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
3551             exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3552             2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i)))^(1/2) -...
3553             l1*r*exp(Ar(j)*2i) - l1*r*exp(A1(j,i)*2i) + l1^2*exp(Ar(j)*1i)*...
3554             exp(A1(j,i)*1i) - l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
3555             l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + r^2*exp(Ar(j)*1i)*...
3556             exp(A1(j,i)*1i))/...
3557             (2*(l1*l4*exp(Ar(j)*1i) - l4*r*exp(A1(j,i)*1i))))*1i)/l23));
3558
3559         theta2(j,i,k) = theta23(j,i) + phi232;
3560         theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
3561     end
3562 end
3563
3564 end
3565 %in the case of a horizontally positioned segment 1, MATLAB solve() has

```

```

3566 %troubles finding a solution... Therefore, perturb by small amount to solve
3567 if theta1(j,i) == pi/2
3568     theta1(j,i) = pi/2 + STEP1(j);
3569 end
3570
3571 %the expressions within this loop are valid for theta1 > pi/2
3572 if theta1(j,i) > pi/2
3573     %angle pendulum w.r.t. positive x-axis, (CCW positive)
3574     Ar(j) = (pi/2) - alpha(j);
3575     %angle of segment 1 with respect to positive x-axis (CW positive)
3576     theta1p(j,i) = theta1(j,i) - (pi/2);
3577
3578 %formulation for theta4: elbow up
3579 theta4(j,i,k) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Ar(j)*1i)*...
3580     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3581     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3582     exp(theta1p(j,i)*1i) - 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3583     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
3584     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3585     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3586     exp(theta1p(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3587     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))^(1/2) - l1^2*exp(Ar(j)*1i)*...
3588     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
3589     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3590     exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))/...
3591     (2*(l1*l4*exp(Ar(j)*1i)*1i -...
3592     l4*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*1i)*1i)));
3593
3594 %formulation for theta2 and theta3: elbow up
3595 theta23(j,i) = real(asin((l4*sin(log(-(l1*r+((l1*r - l1^2*exp(Ar(j)*1i)*...
3596     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3597     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3598     exp(theta1p(j,i)*1i) - 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3599     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
3600     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3601     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3602     exp(theta1p(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3603     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))^(1/2) - l1^2*exp(Ar(j)*1i)*...
3604     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
3605     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3606     exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))/...
3607     (2*(l1*l4*exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
3608     exp(theta1p(j,i)*1i)*1i))) -...
3609     l1*cos(theta1p(j,i)) + r*cos(Ar(j))/l23));
3610
3611 theta2(j,i,k) = theta23(j,i) + phi232;
3612 theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
3613
3614 %calculate the deviations in x and y of the coordinates of the compensator, respectively
3615 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
3616     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
3617
3618 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
3619     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
3620
3621 %if the absolute value of any of these deviations transcends a
3622 %certain threshold, then use alternative formulation for theta2 and
3623 %theta3
3624 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
3625     theta23(j,i) = pi + real( - asin((l4*sin(log(-(l1*r +...
3626     ((l1*r - l1^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3627     l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
3628     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
3629     2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
3630     exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
3631     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
3632     l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
3633     exp(theta1p(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*...
3634     exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
3635     exp(theta1p(j,i)*2i))^(1/2) - l1^2*exp(Ar(j)*1i)*...
3636     exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*...
3637     exp(theta1p(j,i)*1i) - l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
3638     r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
3639     exp(theta1p(j,i)*2i))/(2*(l1*l4*exp(Ar(j)*1i)*1i -...
3640     l4*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*1i)*1i))) -...
3641     l1*cos(theta1p(j,i)) + r*cos(Ar(j))/l23));
3642
3643     theta2(j,i,k) = theta23(j,i) + phi232;
3644     theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
3645 end
3646
3647 end
3648
3649 %calculate the deviations in x and y of the coordinates of the compensator, respectively

```



```

3650 DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
3651     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
3652
3653 DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
3654     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
3655
3656 %calculate the distance from the endpoint of the second segment to the end
3657 %effector of the inverted pendulum
3658 d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
3659     (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);
3660
3661 %check condition upper loop closure
3662 if (l4-l3-d(j,i,k)) > 0
3663     %set the deviation in x...
3664     DEV11(j,i,k) = 0;
3665     %... and y to zero such that this scenario won't be flagged
3666     DEV22(j,i,k) = 0;
3667     %posture doesn't exist, so potential energy not a number
3668     V(j,i,k) = NaN;
3669
3670     %define the angles of the third and fourth segment to be no value;
3671     %the surface plots of these tensors (used for debugging) would
3672     %otherwise be nonsmooth
3673     theta3(j,i,k) = NaN;
3674     theta4(j,i,k) = NaN;
3675     %flag this event with variable "Count2" instead
3676     Count2 = Count2 + 1;
3677 end
3678
3679 %if segment 1 and segment 2 are not at their lowerbound
3680 if i>1 && k>1
3681     %if the angle of the third segment was previously - for the same angle
3682     %of the pendulum - NaN, then it will remain NaN for this angle of the
3683     %pendulum (infeasible solution space)
3684     if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)    %#ok<COMPNOP>
3685         theta3(j,i,k) = NaN;
3686
3687         %the potential energy and the angle of segment 4 should
3688         %consequently be NaN as well
3689         V(j,i,k) = NaN;
3690         theta4(j,i,k) = NaN;
3691     end
3692 end
3693
3694 %check condition upper loop closure
3695 if l4-l3+d(j,i,k) < 0
3696     %set the deviation in x...
3697     DEV11(j,i,k) = 0;
3698     %... and y to zero such that this scenario won't be flagged
3699     DEV22(j,i,k) = 0;
3700     %posture doesn't exist, so potential energy not a number
3701     V(j,i,k) = NaN;
3702
3703     %define the angles of the third and fourth segment to be no value;
3704     %the surface plots of these tensors (used for debugging) would
3705     %otherwise be nonsmooth
3706     theta3(j,i,k) = NaN;
3707     theta4(j,i,k) = NaN;
3708     %flag this event with variable "Count3" instead
3709     Count3 = Count3 + 1;
3710 end
3711
3712 %if the absolute value of any of these deviations transcends a
3713 %certain threshold, then increase the variable "Count" by one
3714 if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
3715     Count = Count + 1;
3716 end
3717
3718 %initial relative angle of segment 1
3719 alpha10 = theta1i;
3720 %initial relative angle of segment 2
3721 alpha20 = theta2i - theta1i;
3722 %initial relative angle of segment 3
3723 alpha30 = theta3i - theta2i;
3724 %initial relative angle of segment 4
3725 alpha40 = theta4i - theta3i;
3726
3727 %angle of rotation torsion spring 1
3728 alpha1(j,i) = theta1(j,i) - alpha10;
3729 %angle of rotation torsion spring 2
3730 alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
3731 %angle of rotation torsion spring 3
3732 alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
3733 %angle of rotation torsion spring 4

```

```

3734 alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;
3735
3736 if nonlinearity == 0
3737     %internal moment spring 1
3738     M1(j,i) = k1*alpha1(j,i);
3739     %internal moment spring 2
3740     M2(j,i,k) = k2*alpha2(j,i,k) + M02;
3741     %internal moment spring 3
3742     M3(j,i,k) = k3*alpha3(j,i,k) + M03;
3743     %internal moment spring 4
3744     M4(j,i,k) = k4*alpha4(j,i,k);
3745
3746     %potential energy spring 1
3747     V1(j,i) = ((k1/2)*alpha1(j,i)^2);
3748     %potential energy spring 2
3749     V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) + ...
3750         ((k2/2)*(M02/k2)^2);
3751     %potential energy spring 3
3752     V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) + ...
3753         ((k3/2)*(M03/k3)^2);
3754     %potential energy spring 4
3755     V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
3756     %total potential energy
3757     V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
3758 end
3759
3760 if nonlinearity == 1
3761     %first solution prestress angle: angle of rotation corresponding to
3762     %prestress spring 2
3763     alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
3764     %second solution prestress angle: angle of rotation corresponding to
3765     %prestress spring 2
3766     alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);
3767
3768     %allow only for nonnegative solutions; set to NaN if negative
3769     if alphastar1M2 < 0
3770         alphastar1M2 = NaN;
3771     end
3772
3773     %allow only for nonnegative solutions; set to NaN if negative
3774     if alphastar2M2 < 0
3775         alphastar2M2 = NaN;
3776     end
3777
3778     %store solutions prestress angle in array called "alphastarsM2"
3779     alphastarsM2 = [alphastar1M2,alphastar2M2];
3780
3781     %store the smallest solution for the prestress angle
3782     alphastarM2 = min(abs(alphastarsM2));
3783
3784     %first solution prestress angle: angle of rotation corresponding to
3785     %prestress spring 3
3786     alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
3787     %first solution prestress angle: angle of rotation corresponding to
3788     %prestress spring 3
3789     alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);
3790
3791     %allow only for nonnegative solutions; set to NaN if negative
3792     if alphastar1M3 < 0
3793         alphastar1M3 = NaN;
3794     end
3795
3796     %allow only for nonnegative solutions; set to NaN if negative
3797     if alphastar2M3 < 0
3798         alphastar2M3 = NaN;
3799     end
3800
3801     %store solutions prestress angle in array called "alphastarsM3"
3802     alphastarsM3 = [alphastar1M3,alphastar2M3];
3803
3804     %store the smallest solution for the prestress angle
3805     alphastarM3 = min(abs(alphastarsM3));
3806
3807     %internal moment spring 1
3808     M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
3809     %internal moment spring 2
3810     M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 + ...
3811         B*(alpha2(j,i,k)+alphastarM2);
3812     %internal moment spring 3
3813     M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 + ...
3814         B*(alpha3(j,i,k)+alphastarM3);
3815     %internal moment spring 4
3816     M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);
3817

```

```

3818 %potential energy spring 1
3819 V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
3820 %potential energy spring 2
3821 V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 + ...
3822 (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
3823 %potential energy spring 3
3824 V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 + ...
3825 (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
3826 %potential energy spring 4
3827 V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
3828 %total potential energy
3829 V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
3830 end
3831
3832 %x - coordinate origin (and first spring)
3833 x0 = 0;
3834 %y - coordinate origin (and first spring)
3835 y0 = 0;
3836 %x - coordinate 2nd spring
3837 x1(j,i) = l1*sin(theta1(j,i));
3838 %y - coordinate 2nd spring
3839 y1(j,i) = l1*cos(theta1(j,i));
3840 %x - coordinate 3rd spring
3841 x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
3842 %y - coordinate 3rd spring
3843 y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
3844 %x - coordinate 4th spring
3845 x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
3846 %y - coordinate 4th spring
3847 y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
3848 %x - coordinate end effector
3849 x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
3850 %y - coordinate end effector
3851 y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));
3852
3853 %magnitude reaction force y-direction
3854 F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
3855 (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))/...
3856 (-tan(theta4(j,i,k))*(l1*cos(theta1(j,i))+...
3857 l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))...
3858 + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));
3859
3860 %magnitude reaction force x-direction
3861 F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
3862 (l4*cos(theta4(j,i,k)));
3863
3864 %external moment on second spring (node 2)
3865 M2lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
3866 F1yt(j,i,k)*l1*sin(theta1(j,i));
3867
3868 %external moment on third spring (node 3)
3869 M3lt(j,i,k) = M1(j,i) + ...
3870 F1xt(j,i,k)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))) -...
3871 F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
3872
3873 end
3874
3875 %if spring 2 and 3 are both activated
3876 if M3lt(j,i,k) >= M03 && M2lt(j,i,k) >= M02
3877
3878 %formulation for angle segment 1 with spring 2 and 3 both enabled
3879 theta1(j,i) = theta1fa(j,i);
3880
3881 %the expressions within this loop are valid for theta1 < 0
3882 if theta1(j,i) < 0
3883 %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
3884 theta1n(j,i) = - theta1(j,i);
3885
3886 %lowerbound and upperbound of segment 2, respectively,
3887 %for given precision point and angle of segment 1
3888 theta20(j,i) = real(-log(-(l1*r - ((l1*r - l1^2*exp(alpha(j)*1i))*...
3889 exp(theta1n(j,i)*1i) + l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
3890 l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
3891 exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
3892 2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
3893 2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
3894 2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
3895 l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))*...
3896 (l1*r - l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
3897 l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l3^2*exp(alpha(j)*1i)*...
3898 exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
3899 r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
3900 2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
3901 2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...

```

```

3902     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3903     11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))^(1/2) - ...
3904     11^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - 12^2*exp(alpha(j)*1i)*...
3905     exp(theta1n(j,i)*1i) + 13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3906     14^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*...
3907     exp(theta1n(j,i)*1i) + 2*13*14*exp(alpha(j)*1i)*...
3908     exp(theta1n(j,i)*1i) + 11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))/...
3909     (2*(12*r*exp(theta1n(j,i)*1i) - ...
3910     11*12*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))) *1i);
3911
3912     theta2f(j,i) = real(-log(-((11*r - 11^2*exp(alpha(j)*1i)*...
3913     exp(theta1n(j,i)*1i) + 12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3914     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 14^2*exp(alpha(j)*1i)*...
3915     exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - ...
3916     2*12*13*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - 2*12*14*...
3917     exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 2*13*14*exp(alpha(j)*1i)*...
3918     exp(theta1n(j,i)*1i) + 11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))*...
3919     (11*r - 11^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3920     12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 13^2*exp(alpha(j)*1i)*...
3921     exp(theta1n(j,i)*1i) + 14^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - ...
3922     r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 2*12*13*exp(alpha(j)*1i)*...
3923     exp(theta1n(j,i)*1i) + 2*12*14*exp(alpha(j)*1i)*...
3924     exp(theta1n(j,i)*1i) + 2*13*14*exp(alpha(j)*1i)*...
3925     exp(theta1n(j,i)*1i) + 11*r*exp(alpha(j)*2i)*...
3926     exp(theta1n(j,i)*2i))^(1/2) + 11*r - 11^2*exp(alpha(j)*1i)*...
3927     exp(theta1n(j,i)*1i) - 12^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3928     13^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 14^2*exp(alpha(j)*1i)*...
3929     exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3930     2*13*14*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + ...
3931     11*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))/...
3932     (2*(12*r*exp(theta1n(j,i)*1i) - ...
3933     11*12*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))) *1i);
3934
3935     %compensate for erroneous results due to periodicity of the loop
3936     %closure equations
3937     if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
3938         theta2f(j,i) = theta2f(j,i) + 2*pi;
3939     end
3940
3941     %prevent the upperbound of segment 2 from being smaller than
3942     %the lowerbound
3943     if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
3944         theta2f(j,i) = theta2f(j,i) + 2*pi;
3945     end
3946
3947     %define boundaries segment 2 sweep
3948     BEGIN2(j,i) = theta20(j,i);
3949     END2(j,i) = theta2f(j,i);
3950     %define stepsize segment 2 sweep
3951     STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;
3952
3953     %start angle of segment 2 equal to lowerbound,increase with stepsize
3954     theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;
3955
3956     %angle connection line origin and endpoint segment 2
3957     Mtheta12(j,i,k) = - atan((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))/...
3958     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))));
3959
3960     %if endpoint of second segment is still in Q4
3961     if (11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0 &&...
3962     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0
3963
3964         %angle connection line origin and endpoint segment 2
3965         Mtheta12(j,i,k) = atan(abs(11*cos(theta1(j,i)) + ...
3966         12*cos(theta2(j,i,k)))/...
3967         abs(11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))) + pi/2;
3968     end
3969
3970     %length of imaginary connection line between origin and end of segment 2
3971     l12(j,i,k) = sqrt((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))^2 + ...
3972     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))^2);
3973
3974     %angle of segment 3 and segment 4, for given precision point &
3975     %angle segment 1 & angle segment 2
3976     theta3(j,i,k) = real(asin((14*sin(log(-(l12(j,i,k)*r + ...
3977     ((12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
3978     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
3979     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
3980     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*13*14*...
3981     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
3982     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
3983     (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
3984     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
3985     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...

```

```

3986     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
3987     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
3988     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - 112(j,i,k)^2*...
3989     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
3990     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...
3991     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
3992     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
3993     r^2*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(14*r*...
3994     exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*...
3995     exp(alpha(j)*1i))) *1i) +...
3996     112(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/13));
3997
3998 theta4(j,i,k) = real(-log(-112(j,i,k)*r +...
3999     ((112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
4000     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
4001     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
4002     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*13*14*...
4003     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4004     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
4005     (112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
4006     exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
4007     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
4008     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
4009     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4010     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - 112(j,i,k)^2*...
4011     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4012     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...
4013     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4014     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4015     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(14*r*...
4016     exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*exp(Mtheta12(j,i,k)*2i)*...
4017     exp(alpha(j)*1i))) *1i);
4018
4019 %compensate for erroneous results due to periodicity of the loop
4020 %closure equations
4021 if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %#ok<*COMPNOT>
4022     theta4(j,i,k) = 2*pi + real(-log(-112(j,i,k)*r +...
4023         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
4024         exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*...
4025         exp(alpha(j)*1i) + 14^2*exp(Mtheta12(j,i,k)*1i)*...
4026         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
4027         exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i)*...
4028         exp(alpha(j)*1i) + 112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
4029         exp(alpha(j)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4030         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4031         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 14^2*...
4032         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4033         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
4034         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4035         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - 112(j,i,k)^2*...
4036         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4037         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...
4038         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4039         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4040         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
4041         (2*(14*r*exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*...
4042         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))) *1i);
4043
4044 end
4045
4046 %calculate the deviations in x and y of the coordinates of the compensator, respectively
4047 DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
4048     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
4049 DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
4050     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
4051
4052 %if the absolute value of any of these deviations transcends a
4053 %certain threshold, then use alternative formulation for theta3
4054 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-12
4055     theta3(j,i,k) = pi + real(-asin((14*sin(log(-112(j,i,k)*r +...
4056         ((112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
4057         exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*...
4058         exp(alpha(j)*1i) + 14^2*exp(Mtheta12(j,i,k)*1i)*...
4059         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
4060         exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i)*...
4061         exp(alpha(j)*1i) + 112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
4062         exp(alpha(j)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4063         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4064         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 14^2*...
4065         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4066         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
4067         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4068         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - 112(j,i,k)^2*...
4069         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4070         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...

```

```

4070     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4071     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
4072     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
4073     (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
4074     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) +...
4075     l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3));
4076 end
4077
4078 %if endpoint of second segment is still in Q1
4079 if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
4080     (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) > 0
4081
4082     %angle pendulum w.r.t. positive x-axis, (CCW positive)
4083     Ar(j) = (pi/2) - alpha(j);
4084     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
4085     A1(j,i) = (pi/2) - theta1(j,i);
4086     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
4087     A2(j,i,k) = (pi/2) - theta2(j,i,k);
4088     %angle imaginary connection line origin and endpoint segment 2
4089     phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
4090         (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));
4091
4092 %angle of segment 3 and segment 4, for given precision point &
4093 %angle segment 1 & angle segment 2
4094 theta3(j,i,k) = pi/2 - ...
4095     real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) - r*cos(Ar(j)) + ...
4096         l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
4097         exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4098         exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4099         l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4100         exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
4101         exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + ...
4102         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4103         exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4104         l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4105         exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4106         exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...
4107         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4108         exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4109         l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4110         exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
4111         l4*r*exp(phi12(j,i,k)*1i))))*1i)/l3));
4112
4113 theta4(j,i,k) = pi/2 - real(-log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + ...
4114     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4115     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4116     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4117     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
4118     exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*...
4119     r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4120     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4121     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4122     exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4123     exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...
4124     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4125     exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4126     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4127     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
4128     l4*r*exp(phi12(j,i,k)*1i))))*1i);
4129
4130 %calculate the deviations in x and y of the coordinates of the compensator,
4131 %respectively
4132 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) + ...
4133     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
4134 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) + ...
4135     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
4136
4137 %if the absolute value of any of these deviations transcends a
4138 %certain threshold, then use alternative formulation for theta3
4139 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
4140     theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
4141         cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*...
4142         r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4143         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
4144         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4145         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4146         exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
4147         exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + ...
4148         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
4149         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4150         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4151         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4152         exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4153         exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...

```

```

4153     112(j,i,k)*r*exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*...
4154     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*...
4155     exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4156     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4157     exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i) -...
4158     14*r*exp(phi12(j,i,k)*1i))))*1i)/13));
4159
4160     if theta3(j,i,k) < - pi
4161         theta3(j,i,k) = 2*pi + pi/2 -...
4162         real(pi + acos((112(j,i,k)*cos(phi12(j,i,k)) -...
4163         r*cos(Ar(j)) + 14*cos(log(-((112(j,i,k)*r*...
4164         exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
4165         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
4166         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*...
4167         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*...
4168         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 2*13*14*...
4169         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*...
4170         exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
4171         112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
4172         13^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*...
4173         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*...
4174         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*13*14*...
4175         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) -...
4176         112(j,i,k)*r*exp(Ar(j)*2i) - 112(j,i,k)*r*...
4177         exp(phi12(j,i,k)*2i) + 112(j,i,k)^2*exp(Ar(j)*1i)*...
4178         exp(phi12(j,i,k)*1i) - 13^2*exp(Ar(j)*1i)*...
4179         exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4180         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4181         exp(phi12(j,i,k)*1i))/(2*(112(j,i,k)*14*...
4182         exp(Ar(j)*1i) -...
4183         14*r*exp(phi12(j,i,k)*1i))))*1i)/13));
4184     end
4185     end
4186 end
4187
4188 %if endpoint of second segment is still in Q2
4189 if ((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) >= 0 &&...
4190     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0) ||...
4191     (passedX(j,i) == 1)
4192
4193     %indicate that the endpoint of second segment passed x-axis
4194     passedX(j,i) = 1;
4195     %angle pendulum w.r.t. positive x-axis, (CCW positive)
4196     Ar(j) = (pi/2) - alpha(j);
4197     %angle of segment 1 with respect to positive x-axis (CW positive)
4198     theta1p(j,i) = theta1(j,i) - (pi/2);
4199
4200     %angle of imaginary connection (between the origin and the
4201     %node at the end of the second segment) with respect to
4202     %positive x-axis
4203     % (clockwise positive)
4204     theta12P(j,i,k) = atan((11*sin(theta1(j,i))+12*sin(theta2(j,i,k)))/...
4205     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))) - (pi/2);
4206
4207     if (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0
4208         theta12P(j,i,k) = theta12P(j,i,k) + pi;
4209     end
4210
4211     %angle imaginary connection line origin and endpoint segment 2
4212     phi12(j,i,k) = -theta12P(j,i,k);
4213
4214     %angle of segment 3 and segment 4, for given precision point &
4215     %angle segment 1 & angle segment 2
4216     theta3(j,i,k) = real(asin((14*sin(log(-112(j,i,k)*r +...
4217     ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4218     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4219     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4220     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4221     exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4222     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4223     exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4224     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4225     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4226     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4227     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4228     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4229     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4230     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4231     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4232     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4233     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4234     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
4235     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i) -...
4236     112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/13));

```

```

4237
4238     theta4(j,i,k) = real(-log(-(l12(j,i,k)*r +...
4239         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4240         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4241         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4242         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4243         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
4244         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
4245         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
4246         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4247         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4248         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4249         exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4250         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
4251         exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4252         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4253         exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
4254         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4255         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
4256         exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
4257         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i));
4258
4259     %calculate the deviations in x and y of the coordinates of the compensator,
         respectively
4260     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
4261         l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
4262     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
4263         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
4264
4265     %if the absolute value of any of these deviations transcends a...
4266     %certain threshold, then use alternative formulation for theta3
4267     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
4268         theta3(j,i,k) = pi + real(- asin((l4*sin(log(-(l12(j,i,k)*r +...
4269             ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4270             exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4271             exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4272             exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4273             exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
4274             exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
4275             exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
4276             exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4277             exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4278             exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4279             exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4280             exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
4281             exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*...
4282             exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4283             exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
4284             exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4285             exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
4286             exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*...
4287             exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
4288             exp(theta12P(j,i,k)*1i)*1i))*1i) - l12(j,i,k)*...
4289             cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3);
4290     end
4291
4292     end
4293
4294     end
4295
4296     %the expressions within this loop are valid for theta1 > 0
4297     if theta1(j,i) >= 0
4298         %angle pendulum w.r.t. positive x-axis, (CCW positive)
4299         Ar(j) = (pi/2) - alpha(j);
4300         %angle segment 1 w.r.t. positive x-axis, (CCW positive)
4301         A1(j,i) = (pi/2) - theta1(j,i);
4302
4303         %lowerbound and upperbound of segment 2, respectively,
4304         %for given precision point and angle of segment 1
4305         theta20(j,i) = (pi/2) -...
4306             real(-log((( - l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4307                 l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*...
4308                 exp(Ar(j)*1i) + l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*...
4309                 exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) + l1*r*...
4310                 exp(Ar(j)*2i) - 2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - 2*l2*l4*...
4311                 exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 2*l3*l4*exp(A1(j,i)*1i)*...
4312                 exp(Ar(j)*1i))*(- l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l2^2*...
4313                 exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4314                 l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*...
4315                 exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) +...
4316                 2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 2*l2*l4*exp(A1(j,i)*1i)*...
4317                 exp(Ar(j)*1i) + 2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i)))^(1/2) -...
4318                 l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - l2^2*exp(A1(j,i)*1i)*...
4319                 exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l4^2*...

```



```

4320     exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4321     11*r*exp(A1(j,i)*2i) + 11*r*exp(Ar(j)*2i) + 2*13*14*exp(A1(j,i)*1i)*...
4322     exp(Ar(j)*1i))/(2*(11*12*exp(Ar(j)*1i) - 12*r*exp(A1(j,i)*1i)))*)1i);
4323
4324     theta2f(j,i) = (pi/2) - real(-log((- ((- 11^2*exp(A1(j,i)*1i)*...
4325     exp(Ar(j)*1i) + 12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4326     13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 14^2*exp(A1(j,i)*1i)*...
4327     exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4328     11*r*exp(A1(j,i)*2i) + 11*r*exp(Ar(j)*2i) - 2*12*13*exp(A1(j,i)*1i)*...
4329     exp(Ar(j)*1i) - 2*12*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4330     2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i))*(- 11^2*exp(A1(j,i)*1i)*...
4331     exp(Ar(j)*1i) + 12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4332     13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 14^2*exp(A1(j,i)*1i)*...
4333     exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4334     11*r*exp(A1(j,i)*2i) + 11*r*exp(Ar(j)*2i) + 2*12*13*exp(A1(j,i)*1i)*...
4335     exp(Ar(j)*1i) + 2*12*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4336     2*13*14*exp(A1(j,i)*1i)*exp(Ar(j)*1i))^(1/2) - 11^2*exp(A1(j,i)*1i)*...
4337     exp(Ar(j)*1i) - 12^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4338     13^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 14^2*exp(A1(j,i)*1i)*...
4339     exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
4340     11*r*exp(A1(j,i)*2i) + 11*r*exp(Ar(j)*2i) + 2*13*14*exp(A1(j,i)*1i)*...
4341     exp(Ar(j)*1i))/(2*(11*12*exp(Ar(j)*1i) - 12*r*exp(A1(j,i)*1i)))*)1i);
4342
4343     %compensate for erroneous results due to periodicity of the loop
4344     %closure equations
4345     if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
4346         theta2f(j,i) = theta2f(j,i) + 2*pi;
4347     end
4348
4349     %compensate for erroneous results due to periodicity of the loop
4350     %closure equations
4351     if (i>1) && (theta20(j,i) - theta20(j,i-1)) > pi
4352         theta20(j,i) = theta20(j,i) - 2*pi;
4353     end
4354
4355     %prevent the upperbound of segment 2 from being smaller than
4356     %the lowerbound
4357     if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
4358         theta2f(j,i) = theta2f(j,i) + 2*pi;
4359     end
4360
4361     %define boundaries segment 2 sweep
4362     BEGIN2(j,i) = theta20(j,i);
4363     END2(j,i) = theta2f(j,i);
4364     %define stepsize segment 2 sweep
4365     STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;
4366
4367     %start angle of segment 2 equal to lowerbound,increase with stepsize
4368     theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;
4369
4370     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
4371     A2(j,i,k) = (pi/2) - theta2(j,i,k);
4372
4373     %length of imaginary connection line between origin and end of segment 2
4374     l12(j,i,k) = sqrt((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))^2 +...
4375     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)))^2);
4376
4377     %angle imaginary connection line origin and endpoint segment 2
4378     phi12(j,i,k) = atan((11*sin(A1(j,i)) + 12*sin(A2(j,i,k)))/...
4379     (11*cos(A1(j,i)) + 12*cos(A2(j,i,k))));
4380
4381     %...and the same angle calculated by using other variables
4382     phi12v(j,i,k) = atan((11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))/...
4383     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))));
4384
4385     %if the node at the end of the second segment is located left to the
4386     %positive y-axis
4387     if (11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0
4388         phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
4389     end
4390
4391     %if endpoint of second segment is in Q4
4392     if (11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k))) < 0 &&...
4393     (11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k))) < 0
4394
4395         %angle imaginary connection line origin and endpoint segment 2
4396         phi12(j,i,k) = atan(abs(11*cos(theta1(j,i))+12*cos(theta2(j,i,k)))/...
4397         abs(11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)))) + pi;
4398     end
4399
4400     %compensate for erroneous results due to periodicity of the loop
4401     %closure equations
4402     if k>1 && (phi12(j,i,k)-phi12(j,i,k-1)) > pi
4403         phi12(j,i,k) = phi12(j,i,k) - 2*pi;

```

```

4404     end
4405
4406     %angle of segment 3 and segment 4, for given precision point &
4407     %angle segment 1 & angle segment 2
4408     theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) - ...
4409     r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + ...
4410     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4411     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4412     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4413     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
4414     (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4415     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4416     exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
4417     r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4418     exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...
4419     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4420     exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4421     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4422     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
4423     l4*r*exp(phi12(j,i,k)*1i)))*1i)/l3);
4424
4425     theta4(j,i,k) = pi/2 - real(-log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + ...
4426     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4427     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4428     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4429     exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
4430     (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4431     l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4432     exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*...
4433     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4434     exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...
4435     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4436     exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4437     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4438     exp(phi12(j,i,k)*1i))/...
4439     (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i)))*1i);
4440
4441     %angle imaginary connection line origin and endpoint segment 2
4442     if phi12(j,i,k) > pi/2
4443         %angle connection line origin and endpoint segment 2
4444         Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + ...
4445         l2*sin(theta2(j,i,k)))/...
4446         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
4447
4448         %if endpoint of second segment is in Q4
4449         if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
4450         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
4451
4452         %angle connection line origin and endpoint segment 2
4453         Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) + ...
4454         l2*cos(theta2(j,i,k)))/...
4455         abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi/2;
4456     end
4457
4458     %angle of segment 3 and segment 4, for given precision point &
4459     %angle segment 1 & angle segment 2
4460     theta3(j,i,k) = real(asin((l4*sin(log(-l12(j,i,k)*r + ...
4461     ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
4462     exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
4463     exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
4464     exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
4465     exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
4466     exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
4467     exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
4468     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
4469     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
4470     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4471     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
4472     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
4473     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - ...
4474     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4475     l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
4476     l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
4477     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4478     l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
4479     (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
4480     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) + ...
4481     l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3);
4482
4483
4484     theta4(j,i,k) = real(-log(-l12(j,i,k)*r + ((l12(j,i,k)*r - ...
4485     l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4486     l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4487     l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...

```

```

4488     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*13*14*...
4489     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4490     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(112(j,i,k)*r - ...
4491     112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4492     13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4493     14^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - ...
4494     r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4495     2*13*14*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4496     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - 112(j,i,k)^2*...
4497     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4498     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...
4499     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4500     exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4501     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
4502     (2*(14*r*exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*...
4503     exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))^1i);
4504
4505     if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %#ok<*COMPNOT>
4506         theta4(j,i,k) = 2*pi + real(-log(-(112(j,i,k)*r + ...
4507             ((112(j,i,k)*r - 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
4508                 exp(alpha(j)*1i) + 13^2*exp(Mtheta12(j,i,k)*1i)*...
4509                 exp(alpha(j)*1i) + 14^2*exp(Mtheta12(j,i,k)*1i)*...
4510                 exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
4511                 exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i)*...
4512                 exp(alpha(j)*1i) + 112(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
4513                 exp(alpha(j)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4514                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 13^2*...
4515                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 14^2*...
4516                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4517                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
4518                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4519                 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) - ...
4520                 112(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + ...
4521                 13^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 14^2*...
4522                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
4523                 exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 112(j,i,k)*r*...
4524                 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
4525                 (2*(14*r*exp(Mtheta12(j,i,k)*1i) - 112(j,i,k)*14*...
4526                 exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))^1i);
4527     end
4528
4529     %calculate the deviations in x and y of the coordinates of the compensator,
4530     %respectively
4531     DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) + ...
4532     13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
4533     DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) + ...
4534     13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
4535
4536     %if the absolute value of any of these deviations transcends a...
4537     %certain threshold, then use alternative formulation for theta3
4538     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
4539         theta3(j,i,k) = pi + real( - asin((14*sin(log(-(112(j,i,k)*r + ...
4540             ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4541                 exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4542                 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4543                 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4544                 exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4545                 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4546                 exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4547                 exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4548                 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4549                 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4550                 exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4551                 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4552                 exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4553                 exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4554                 exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4555                 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4556                 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4557                 exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*...
4558                 exp(Ar(j)*1i)*1i - 14*r*exp(Ar(j)*2i)*...
4559                 exp(theta12P(j,i,k)*1i)*1i)))^1i) - ...
4560         112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/13);
4561     end
4562
4563     end
4564
4565     if phi12(j,i,k) < 0
4566         %angle pendulum w.r.t. positive x-axis, (CCW positive)
4567         Ar(j) = (pi/2) - alpha(j);
4568         %angle of segment 1 with respect to positive x-axis (CW positive)
4569         theta1p(j,i) = theta1(j,i) - (pi/2);
4570
4571         %angle of imaginary connection (between the origin and the

```

```

4571 %node at the end of the second segment) with respect to
4572 %positive x-axis
4573 %(clockwise positive)
4574 theta12P(j,i,k) = - phi12(j,i,k);
4575
4576 %angle of segment 3 and segment 4, for given precision point &
4577 %angle segment 1 & angle segment 2
4578 theta3(j,i,k) = real(asin((14*sin(log(-(112(j,i,k)*r +...
4579 ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4580 exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4581 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4582 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4583 exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4584 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4585 exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4586 exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4587 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4588 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4589 exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4590 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4591 exp(theta12P(j,i,k)*2i)))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4592 exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4593 exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4594 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4595 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4596 exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
4597 14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i -...
4598 112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/13));
4599
4600
4601 theta4(j,i,k) = real(-log(-(112(j,i,k)*r + ((112(j,i,k)*r -...
4602 112(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
4603 13^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4604 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4605 exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4606 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4607 exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4608 exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4609 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4610 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4611 exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4612 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4613 exp(theta12P(j,i,k)*2i)))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4614 exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4615 exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4616 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4617 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4618 exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i -...
4619 14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i))*1i);
4620
4621 %calculate the deviations in x and y of the coordinates of the compensator,
4622 %respectively
4623 DEV1(j,i,k) = 11*sin(theta1(j,i)) + 12*sin(theta2(j,i,k)) +...
4624 13*sin(theta3(j,i,k)) + 14*sin(theta4(j,i,k)) - r*sin(alpha(j));
4625 DEV2(j,i,k) = 11*cos(theta1(j,i)) + 12*cos(theta2(j,i,k)) +...
4626 13*cos(theta3(j,i,k)) + 14*cos(theta4(j,i,k)) - r*cos(alpha(j));
4627
4628 %if the absolute value of any of these deviations transcends a
4629 %certain threshold, then use alternative formulation for theta3
4630 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
4631 theta3(j,i,k) = pi + real(- asin((14*sin(log(-(112(j,i,k)*r +...
4632 ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4633 exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4634 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4635 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4636 exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4637 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4638 exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4639 exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4640 exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4641 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4642 exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4643 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4644 exp(theta12P(j,i,k)*2i)))^(1/2) - 112(j,i,k)^2*...
4645 exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4646 exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4647 exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4648 exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4649 exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*...
4650 exp(Ar(j)*1i)*1i - 14*r*exp(Ar(j)*2i)*...
4651 exp(theta12P(j,i,k)*1i)*1i))*1i - ...
4652 112(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/13));
4653 end

```

```

4654     end
4655
4656     %calculate the deviations in x and y of the coordinates of the compensator, respectively
4657     DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) + ...
4658         l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
4659     DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) + ...
4660         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
4661
4662     %if the absolute value of any of these deviations transcends a
4663     %certain threshold, then use alternative formulation for theta3
4664     if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
4665         theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
4666             cos(phi12(j,i,k)) - r*cos(Ar(j)) + ...
4667             l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
4668             exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i))*...
4669             exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4670             l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4671             exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
4672             exp(phi12(j,i,k)*1i))*...
4673             (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4674             l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4675             l3^2*exp(Ar(j)*1i)*...
4676             exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
4677             r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4678             exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...
4679             l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4680             exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4681             l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4682             exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
4683             l4*r*exp(phi12(j,i,k)*1i))))*1i)/l3);
4684
4685         if theta3(j,i,k) > pi
4686             theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
4687                 cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*...
4688                 r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4689                 l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4690                 l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4691                 l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
4692                 r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
4693                 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
4694                 exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4695                 l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4696                 l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4697                 exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4698                 exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
4699                 exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) - ...
4700                 l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
4701                 exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
4702                 exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4703                 exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4704                 exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - ...
4705                 l4*r*exp(phi12(j,i,k)*1i))))*1i)/l3);
4706         end
4707     end
4708
4709     %compensate for erroneous results due to periodicity of the loop
4710     %closure equations
4711     if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %ok<*COMPNOT>
4712         theta4(j,i,k) = 2*pi + theta4(j,i,k);
4713     end
4714
4715 end
4716
4717 %in the case of a horizontally positioned segment 1, MATLAB solve() has
4718 %troubles finding a solution... Therefore, perturb by small amount to solve
4719 if theta1(j,i) == pi/2
4720     theta1(j,i) = pi/2 + STEP1(j);
4721 end
4722
4723 %the expressions within this loop are valid for theta1 > pi/2
4724 if theta1(j,i) > pi/2
4725     %angle pendulum w.r.t. positive x-axis, (CCW positive)
4726     Ar(j) = (pi/2) - alpha(j);
4727     %angle of segment 1 with respect to positive x-axis (CW positive)
4728     theta1p(j,i) = theta1(j,i) - (pi/2);
4729
4730     %lowerbound and upperbound of segment 2, respectively,
4731     %for given precision point and angle of segment 1
4732     theta20(j,i) = real(-log(-(l1*r - ((l1*r - l1^2*exp(Ar(j)*1i))*...
4733         exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + ...
4734         l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
4735         exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - ...
4736         2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*l2*l4*exp(Ar(j)*1i)*...
4737         exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + ...

```

```

4738     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
4739     exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4740     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
4741     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4742     2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
4743     exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4744     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))^(1/2) - l1^2*exp(Ar(j)*1i)*...
4745     exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4746     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
4747     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4748     2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
4749     exp(theta1p(j,i)*2i))/(2*(l1*l2*exp(Ar(j)*1i)*1i -...
4750     l2*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*1i)*1i));
4751
4752
4753
4754     theta2f(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Ar(j)*1i)*...
4755     exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4756     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
4757     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
4758     2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*l2*l4*exp(Ar(j)*1i)*...
4759     exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4760     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
4761     exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4762     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
4763     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4764     2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
4765     exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4766     l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))^(1/2) - l1^2*exp(Ar(j)*1i)*...
4767     exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4768     l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
4769     exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
4770     2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
4771     exp(theta1p(j,i)*2i))/...
4772     (2*(l1*l2*exp(Ar(j)*1i)*1i - l2*r*exp(Ar(j)*2i)*...
4773     exp(theta1p(j,i)*1i)*1i));
4774
4775     %compensate for erroneous results due to periodicity of the loop
4776     %closure equations
4777     if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
4778         theta2f(j,i) = theta2f(j,i) + 2*pi;
4779     end
4780
4781     %prevent the upperbound of segment 2 from being
4782     %smaller than the lowerbound
4783     if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
4784         theta2f(j,i) = theta2f(j,i) + 2*pi;
4785     end
4786
4787     %define boundaries segment 2 sweep
4788     BEGIN2(j,i) = theta20(j,i);
4789     END2(j,i) = theta2f(j,i);
4790     %define stepsize segment 2 sweep
4791     STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;
4792
4793     %start angle of segment 2 equal to lowerbound,increase with stepsize
4794     theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;
4795
4796     %length of imaginary connection line between origin and end of segment 2
4797     l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
4798     (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
4799
4800     %angle of imaginary connection (between the origin and the
4801     %node at the end of the second segment) with respect to positive x-axis
4802     % (clockwise positive)
4803     theta12P(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
4804     (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);
4805
4806     if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
4807         theta12P(j,i,k) = theta12P(j,i,k) + pi;
4808     end
4809
4810     %if endpoint of second segment is in Q4
4811     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
4812     (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
4813
4814     %angle of imaginary connection (between the origin and the
4815     %node at the end of the second segment) with respect to
4816     %positive x-axis
4817     % (clockwise positive)
4818     theta12P(j,i,k) = -atan(abs(l1*cos(theta1(j,i)) +...
4819     l2*cos(theta2(j,i,k)))/...
4820     abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) - pi;
4821     end

```

```

4822 %compensate for erroneous results due to periodicity of the loop
4823 %closure equations
4824 if k>1 && abs(theta12P(j,i,k)-theta12P(j,i,k-1)) > pi
4825     theta12P(j,i,k) = theta12P(j,i,k) + 2*pi;
4826 end
4827
4828 %angle imaginary connection line origin and endpoint segment 2
4829 phi12(j,i,k) = -theta12P(j,i,k);
4830
4831 %angle of segment 3 and segment 4, for given precision point &
4832 %angle segment 1 & angle segment 2
4833 theta3(j,i,k) = real(asin((14*sin(log(-(112(j,i,k)*r + ...
4834     ((112(j,i,k)*r - 112(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + ...
4835     13^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4836     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - ...
4837     2*13*14*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*...
4838     exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4839     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4840     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4841     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4842     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4843     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4844     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4845     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4846     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4847     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4848     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4849     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i - ...
4850     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i - 112(j,i,k)*...
4851     cos(theta12P(j,i,k) + r*cos(Ar(j)))/13));
4852
4853 theta4(j,i,k) = real(-log(-(112(j,i,k)*r + ((112(j,i,k)*r - 112(j,i,k)^2*...
4854     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4855     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4856     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4857     exp(theta12P(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4858     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4859     exp(theta12P(j,i,k)*2i))*(112(j,i,k)*r - 112(j,i,k)^2*...
4860     exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4861     exp(theta12P(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4862     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4863     exp(theta12P(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4864     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4865     exp(theta12P(j,i,k)*2i))^(1/2) - 112(j,i,k)^2*exp(Ar(j)*1i)*...
4866     exp(theta12P(j,i,k)*1i) + 13^2*exp(Ar(j)*1i)*...
4867     exp(theta12P(j,i,k)*1i) - 14^2*exp(Ar(j)*1i)*...
4868     exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4869     exp(theta12P(j,i,k)*1i) + 112(j,i,k)*r*exp(Ar(j)*2i)*...
4870     exp(theta12P(j,i,k)*2i))/(2*(112(j,i,k)*14*exp(Ar(j)*1i)*1i - ...
4871     14*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);
4872
4873 %if endpoint segment 2 is in Q3
4874 if theta12P(j,i,k) <= 0 && theta12P(j,i,k) > -pi/2
4875     %angle pendulum w.r.t. positive x-axis, (CCW positive)
4876     Ar(j) = (pi/2) - alpha(j);
4877     %angle segment 1 w.r.t. positive x-axis, (CCW positive)
4878     A1(j,i) = (pi/2) - theta1(j,i);
4879     %angle segment 2 w.r.t. positive x-axis, (CCW positive)
4880     A2(j,i,k) = (pi/2) - theta2(j,i,k);
4881     %angle imaginary connection line origin and endpoint segment 2
4882     phi12(j,i,k) = atan((11*sin(A1(j,i)) + 12*sin(A2(j,i,k)))/...
4883         (11*cos(A1(j,i)) + 12*cos(A2(j,i,k))));
4884
4885 %angle of segment 3 and segment 4, for given precision point &...
4886 %angle segment 1 & angle segment 2
4887 theta3(j,i,k) = pi/2 - real(pi - acos((112(j,i,k)*...
4888     cos(phi12(j,i,k)) - r*cos(Ar(j)) + 14*cos(log(-(112(j,i,k)*...
4889     r*exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4890     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 13^2*...
4891     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4892     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - ...
4893     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(112(j,i,k)*r*...
4894     exp(Ar(j)*2i) + 112(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
4895     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 13^2*...
4896     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4897     exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + ...
4898     2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))^(1/2) - 112(j,i,k)*...
4899     r*exp(Ar(j)*2i) - 112(j,i,k)*r*exp(phi12(j,i,k)*2i) + ...
4900     112(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 13^2*...
4901     exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 14^2*exp(Ar(j)*1i)*...
4902     exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
4903     (2*(112(j,i,k)*14*exp(Ar(j)*1i) - ...
4904     14*r*exp(phi12(j,i,k)*1i)))*1i)/13));
4905

```

```

4906 theta4(j,i,k) = pi/2 - real(-log(-((l12(j,i,k)*r*exp(Ar(j)*2i) +...
4907     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4908     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
4909     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4910     exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4911     exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*...
4912     r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4913     exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
4914     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4915     exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4916     exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
4917     l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4918     exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
4919     l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4920     exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*14*exp(Ar(j)*1i) -...
4921     l4*r*exp(phi12(j,i,k)*1i))) *1i);
4922
4923 %compensate for erroneous results due to periodicity of the loop
4924 %closure equations
4925 if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %ok<<COMPNOT>
4926     theta4(j,i,k) = 2*pi + pi/2 - real(-log(-((l12(j,i,k)*r*...
4927         exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
4928         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
4929         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4930         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4931         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4932         exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
4933         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
4934         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4935         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4936         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*...
4937         1i) + 2*13*14*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
4938         l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*...
4939         exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
4940         exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*...
4941         1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*...
4942         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*14*...
4943         exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))) *1i);
4944 end
4945
4946 %calculate the deviations in x and y of the coordinates of the compensator,
4947     respectively
4948 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
4949     l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
4950 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
4951     l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
4952
4953 %if the absolute value of any of these deviations transcends a
4954 %certain threshold, then use alternative formulation for theta3
4955 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
4956     theta3(j,i,k) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
4957         cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-((l12(j,i,k)*...
4958         r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
4959         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
4960         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4961         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4962         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4963         exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
4964         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
4965         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4966         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4967         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4968         exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4969         exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
4970         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
4971         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
4972         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4973         exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4974         exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*14*exp(Ar(j)*1i) -...
4975         l4*r*exp(phi12(j,i,k)*1i))) *1i)/13);
4976
4977 if theta3(j,i,k) > pi
4978     theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
4979         cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
4980         l4*cos(log(-((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
4981         *exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
4982         exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4983         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
4984         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4985         exp(phi12(j,i,k)*1i) - 2*13*14*exp(Ar(j)*1i)*...
4986         exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
4987         l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
4988         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
4989         exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...

```



```

4989         exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
4990         exp(phi12(j,i,k)*1i) + 2*13*14*exp(Ar(j)*1i)*...
4991         exp(phi12(j,i,k)*1i))^(1/2) - l12(j,i,k)*r*...
4992         exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
4993         l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
4994         l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*...
4995         exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
4996         exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
4997         l4*r*exp(phi12(j,i,k)*1i)))*1i)/l3);
4998     end
4999 end
5000
5001 end
5002
5003 if theta12P(j,i,k) < 0 && theta12P(j,i,k) < -pi/2
5004
5005     %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
5006     theta1n(j,i) = - theta1(j,i);
5007
5008
5009     %length of imaginary connection line between origin and end of segment 2
5010     l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
5011         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
5012
5013     %angle connection line origin and endpoint segment 2
5014     Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) +...
5015         l2*sin(theta2(j,i,k)))/...
5016         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));
5017
5018     %if endpoint of second segment is still in Q4
5019     if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
5020         (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
5021
5022         %angle connection line origin and endpoint segment 2
5023         Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
5024             l2*cos(theta2(j,i,k)))/...
5025             abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi/2;
5026     end
5027
5028     %angle of segment 3 and segment 4
5029     %for given precision point & angle segment 1 & angle segment 2
5030     theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
5031         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
5032         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
5033         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
5034         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
5035         exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i))*...
5036         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
5037         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
5038         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
5039         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
5040         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
5041         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
5042         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
5043         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) -...
5044         l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
5045         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
5046         l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
5047         r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
5048         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
5049         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
5050         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i)))*1i) +...
5051         l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3);
5052
5053
5054     theta4(j,i,k) = real(-log(-(l12(j,i,k)*r +...
5055         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
5056         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i))*...
5057         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i))*...
5058         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i))*...
5059         exp(alpha(j)*1i) - 2*13*14*exp(Mtheta12(j,i,k)*1i))*...
5060         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i))*...
5061         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
5062         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
5063         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
5064         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
5065         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*13*14*...
5066         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
5067         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))^(1/2) -...
5068         l12(j,i,k)^2*exp(alpha(j)*1i) + l3^2*...
5069         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
5070         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
5071         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
5072         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...

```

```

5073         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
5074         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))) *1i);
5075
5076 %calculate the deviations in x and y of the coordinates of
5077 %the compensator, respectively
5078 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
5079             l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
5080
5081 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
5082             l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
5083
5084 %if the absolute value of any of these deviations transcends a
5085 %certain threshold, then use alternative formulation for theta3
5086 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
5087     theta3(j,i,k) = pi + real(- asin((l4*sin(log(-(l12(j,i,k)*r +...
5088         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i))*...
5089         exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
5090         exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
5091         exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
5092         exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
5093         exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
5094         exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
5095         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
5096         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
5097         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
5098         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
5099         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
5100         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
5101         l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
5102         l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
5103         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
5104         exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
5105         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
5106         (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
5107         exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))) *1i) +...
5108         l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j))/l3);
5109     end
5110
5111 end
5112
5113 %calculate the deviations in x and y of the coordinates of the compensator, respectively
5114 DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
5115             l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
5116 DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
5117             l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
5118
5119 %if the absolute value of any of these deviations transcends a
5120 %certain threshold, then use alternative formulation for theta3
5121 if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
5122     theta3(j,i,k) = pi + real(- asin((l4*sin(log(-(l12(j,i,k)*r +...
5123         ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i))*...
5124         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
5125         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
5126         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
5127         exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
5128         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
5129         exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
5130         exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
5131         exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
5132         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
5133         exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
5134         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
5135         exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
5136         exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
5137         exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
5138         exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
5139         exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
5140         exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
5141         l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)) *1i) -...
5142         l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j))/l3);
5143     end
5144
5145 %compensate for erroneous results due to periodicity of the loop
5146 %closure equations
5147 if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi) %ok<=COMPNOT>
5148     theta4(j,i,k) = 2*pi + theta4(j,i,k);
5149 end
5150
5151 end
5152
5153 %calculate the deviations in x and y of the coordinates of the compensator, respectively
5154 DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
5155             l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
5156

```

```

5157 DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
5158         l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
5159
5160 %calculate the distance from the endpoint of the second segment to the end
5161 %effector of the inverted pendulum
5162 d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
5163         (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);
5164
5165 %check condition upper loop closure
5166 if (l4-l3-d(j,i,k)) > 0
5167     %set the deviation in x...
5168     DEV11(j,i,k) = 0;
5169     %... and y to zero such that this scenario won't be flagged
5170     DEV22(j,i,k) = 0;
5171     %posture doesn't exist, so potential energy not a number
5172     V(j,i,k) = NaN;
5173
5174     %define the angles of the third and fourth segment to be no value;
5175     %the surface plots of these tensors (used for debugging) would
5176     %otherwise be nonsmooth
5177     theta3(j,i,k) = NaN;
5178     theta4(j,i,k) = NaN;
5179     %flag this event with variable "Count2" instead
5180     Count2 = Count2 + 1;
5181 end
5182
5183 %if segment 1 and segment 2 are not at their lowerbound
5184 if i>1 && k>1
5185     %if the angle of the third segment was previously - for the same angle
5186     %of the pendulum - NaN, then it will remain NaN for this angle of the
5187     %pendulum (infeasible solution space)
5188     if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)           %#ok<COMPNOP>
5189         theta3(j,i,k) = NaN;
5190
5191         %the potential energy and the angle of segment 4 should
5192         %consequently be NaN as well
5193         V(j,i,k) = NaN;
5194         theta4(j,i,k) = NaN;
5195     end
5196 end
5197
5198 %check condition upper loop closure
5199 if l4-l3+d(j,i,k) < 0
5200     %set the deviation in x...
5201     DEV11(j,i,k) = 0;
5202     %... and y to zero such that this scenario won't be flagged
5203     DEV22(j,i,k) = 0;
5204     %posture doesn't exist, so potential energy not a number
5205     V(j,i,k) = NaN;
5206
5207     %define the angles of the third and fourth segment to be no value;
5208     %the surface plots of these tensors (used for debugging) would
5209     %otherwise be nonsmooth
5210     theta3(j,i,k) = NaN;
5211     theta4(j,i,k) = NaN;
5212     %flag this event with variable "Count3" instead
5213     Count3 = Count3 + 1;
5214 end
5215
5216 %if the absolute value of any of these deviations transcends a
5217 %certain threshold, then increase the variable "Count" by one
5218 if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
5219     Count = Count + 1;
5220 end
5221
5222 %initial relative angle of segment 1
5223 alpha10 = theta1i;
5224 %initial relative angle of segment 2
5225 alpha20 = theta2i - theta1i;
5226 %initial relative angle of segment 3
5227 alpha30 = theta3i - theta2i;
5228 %initial relative angle of segment 4
5229 alpha40 = theta4i - theta3i;
5230
5231 %angle of rotation torsion spring 1
5232 alpha1(j,i) = theta1(j,i) - alpha10;
5233 %angle of rotation torsion spring 2
5234 alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
5235 %angle of rotation torsion spring 3
5236 alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
5237 %angle of rotation torsion spring 4
5238 alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;
5239
5240 if nonlinearity == 0

```

```

5241 %internal moment spring 1
5242 M1(j,i) = k1*alpha1(j,i);
5243 %internal moment spring 2
5244 M2(j,i,k) = k2*alpha2(j,i,k) + M02;
5245 %internal moment spring 3
5246 M3(j,i,k) = k3*alpha3(j,i,k) + M03;
5247 %internal moment spring 4
5248 M4(j,i,k) = k4*alpha4(j,i,k);
5249
5250 %potential energy spring 1
5251 V1(j,i) = ((k1/2)*alpha1(j,i)^2);
5252 %potential energy spring 2
5253 V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) + ...
5254 ((k2/2)*(M02/k2)^2);
5255 %potential energy spring 3
5256 V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) + ...
5257 ((k3/2)*(M03/k3)^2);
5258 %potential energy spring 4
5259 V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
5260 %total potential energy
5261 V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
5262 end
5263
5264 if nonlinearity == 1
5265 %first solution prestress angle: angle of rotation corresponding to
5266 %prestress spring 2
5267 alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
5268 %second solution prestress angle: angle of rotation corresponding to
5269 %prestress spring 2
5270 alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);
5271
5272 %allow only for nonnegative solutions; set to NaN if negative
5273 if alphastar1M2 < 0
5274     alphastar1M2 = NaN;
5275 end
5276
5277 %allow only for nonnegative solutions; set to NaN if negative
5278 if alphastar2M2 < 0
5279     alphastar2M2 = NaN;
5280 end
5281
5282 %store solutions prestress angle in array called "alphastarsM2"
5283 alphastarsM2 = [alphastar1M2,alphastar2M2];
5284
5285 %store the smallest solution for the prestress angle
5286 alphastarM2 = min(abs(alphastarsM2));
5287
5288 %first solution prestress angle: angle of rotation corresponding to
5289 %prestress spring 3
5290 alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
5291 %first solution prestress angle: angle of rotation corresponding to
5292 %prestress spring 3
5293 alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);
5294
5295 %allow only for nonnegative solutions; set to NaN if negative
5296 if alphastar1M3 < 0
5297     alphastar1M3 = NaN;
5298 end
5299
5300 %allow only for nonnegative solutions; set to NaN if negative
5301 if alphastar2M3 < 0
5302     alphastar2M3 = NaN;
5303 end
5304
5305 %store solutions prestress angle in array called "alphastarsM3"
5306 alphastarsM3 = [alphastar1M3,alphastar2M3];
5307
5308 %store the smallest solution for the prestress angle
5309 alphastarM3 = min(abs(alphastarsM3));
5310
5311 %internal moment spring 1
5312 M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
5313 %internal moment spring 2
5314 M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 + ...
5315     B*(alpha2(j,i,k)+alphastarM2);
5316 %internal moment spring 3
5317 M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 + ...
5318     B*(alpha3(j,i,k)+alphastarM3);
5319 %internal moment spring 4
5320 M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);
5321
5322 %potential energy spring 1
5323 V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
5324 %potential energy spring 2

```

```

5325     V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 +...
5326             (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
5327     %potential energy spring 3
5328     V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 +...
5329             (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
5330     %potential energy spring 4
5331     V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
5332     %total potential energy
5333     V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
5334 end
5335
5336 %allow only for nonnegative solutions; set to NaN if negative
5337 if alpha2(j,i,k) < 0
5338     V(j,i,k) = NaN;
5339 end
5340
5341 %allow only for nonnegative solutions; set to NaN if negative
5342 if alpha3(j,i,k) < 0
5343     V(j,i,k) = NaN;
5344 end
5345
5346 %x - coordinate origin (and first spring)
5347 x0 = 0;
5348 %y - coordinate origin (and first spring)
5349 y0 = 0;
5350 %x - coordinate 2nd spring
5351 x1(j,i) = l1*sin(theta1(j,i));
5352 %y - coordinate 2nd spring
5353 y1(j,i) = l1*cos(theta1(j,i));
5354 %x - coordinate 3rd spring
5355 x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
5356 %y - coordinate 3rd spring
5357 y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
5358 %x - coordinate 4th spring
5359 x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
5360 %y - coordinate 4th spring
5361 y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
5362 %x - coordinate end effector
5363 x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
5364 %y - coordinate end effector
5365 y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));
5366
5367 %magnitude reaction force y-direction
5368 F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
5369             (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))/...
5370             (-tan(theta4(j,i,k))*(l1*cos(theta1(j,i))+...
5371             l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))...
5372             + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));
5373
5374 %magnitude reaction force x-direction
5375 F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
5376             (l4*cos(theta4(j,i,k)));
5377
5378 %external moment on second spring (node 2)
5379 M2lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
5380             F1yt(j,i,k)*l1*sin(theta1(j,i));
5381
5382 %external moment on third spring (node 3)
5383 M3lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*(l1*cos(theta1(j,i))+...
5384             l2*cos(theta2(j,i,k))) -...
5385             F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
5386 end
5387
5388 end
5389 end
5390 end
5391 end
5392 end
5393
5394
5395 %find the minimum value of the potential energy for each precision point
5396 %and store the linear index
5397 [Vmin,I] = min(V,[],[2 3],"linear");
5398
5399 %convert linear index to j,i,k indices
5400 ind = I; %linear index
5401 sz = [M N1 N2]; %size of the V tensor
5402 %convert the linear index into 3 indices for j, i & k
5403 [I1,I2,I3] = ind2sub(sz,ind);
5404
5405 %coordinates nodes in initial (relaxed) configuration
5406
5407 %x - coordinate origin (and first spring)
5408 x00 = 0;

```

```

5409 %x - coordinate origin (and first spring)
5410 y00 = 0;
5411 %x - coordinate 2nd spring
5412 x10 = l1*sin(theta1i);
5413 %y - coordinate 2nd spring
5414 y10 = l1*cos(theta1i);
5415 %x - coordinate 3rd spring
5416 x20 = x10 + l2*sin(theta2i);
5417 %y - coordinate 3rd spring
5418 y20 = y10 + l2*cos(theta2i);
5419 %x - coordinate 4th spring
5420 x30 = x20 + l3*sin(theta3i);
5421 %y - coordinate 4th spring
5422 y30 = y20 + l3*cos(theta3i);
5423 %x - coordinate end effector
5424 x40 = x30 + l4*sin(theta4i);
5425 %y - coordinate end effector
5426 y40 = y30 + l4*cos(theta4i);
5427
5428
5429 %create new figure to plot the lowest energy configurations
5430 figure(2); %create figure
5431 %plot following plot commands in that same figure
5432 hold on
5433 axis equal
5434 title("Lowest energy configurations")
5435
5436 %start a loop throughout all precision points
5437 for j = 1:M %divide the 90 deg range of motion into equally sized segments
5438 alpha(j) = (pi/2)*(j/M);
5439
5440 %plot connection line between spring 1 and 2 in black
5441 plot([x0 x1(j,I2(j))],[y0 y1(j,I2(j))],'k')
5442 %plot connection line between spring 2 and 3 in black
5443 plot([x1(j,I2(j)) x2(j,I2(j),I3(j))],[y1(j,I2(j)) y2(j,I2(j),I3(j))],'k')
5444 %plot connection line between spring 3 and 4 in black
5445 plot([x2(j,I2(j),I3(j)) x3(j,I2(j),I3(j))],...
5446 [y2(j,I2(j),I3(j)) y3(j,I2(j),I3(j))],'k')
5447 %plot connection line between spring 4 and end - effector in black
5448 plot([x3(j,I2(j),I3(j)) x4(j,I2(j),I3(j))],...
5449 [y3(j,I2(j),I3(j)) y4(j,I2(j),I3(j))],'k')
5450 %plot the location of the end effector of the pendulum with a circle
5451 plot(r*sin(alpha(j)),r*cos(alpha(j)),"b--o")
5452
5453 %coordinates nodes in initial (relaxed) configuration
5454
5455 %plot connection line between spring 1 and 2 in black
5456 plot([x00 x10],[y00 y10],'r')
5457 %plot connection line between spring 2 and 3 in black
5458 plot([x10 x20],[y10 y20],'r')
5459 %plot connection line between spring 3 and 4 in black
5460 plot([x20 x30],[y20 y30],'r')
5461 %plot connection line between spring 4 and end - effector in black
5462 plot([x30 x40],[y30 y40],'r')
5463 %plot the location of the end effector of the pendulum with a circle
5464 plot(r*sin(0),r*cos(0),"r--o")
5465
5466 %minimum value of alpha1 per precision point
5467 alpha1m(j) = alpha1(j,I2(j));
5468 %minimum value of alpha2 per precision point
5469 alpha2m(j) = alpha2(j,I2(j),I3(j));
5470 %minimum value of alpha3 per precision point
5471 alpha3m(j) = alpha3(j,I2(j),I3(j));
5472 %minimum value of alpha4 per precision point
5473 alpha4m(j) = alpha4(j,I2(j),I3(j));
5474
5475 if prestress == 0 && nonlinearity == 0
5476 %minimum moment in torsion spring 1 per precision point
5477 M1m(j) = k1*alpha1(j,I2(j));
5478 %minimum moment in torsion spring 2 per precision point
5479 M2m(j) = k2*alpha2(j,I2(j),I3(j));
5480 %minimum moment in torsion spring 3 per precision point
5481 M3m(j) = k3*alpha3(j,I2(j),I3(j));
5482 %minimum moment in torsion spring 4 per precision point
5483 M4m(j) = k4*alpha4(j,I2(j),I3(j));
5484 end
5485
5486 %if springs are nonlinear
5487 if prestress == 0 && nonlinearity == 1
5488 %minimum moment in torsion spring 1 per precision point
5489 M1m(j) = A*alpha1(j,I2(j))^2 + B*alpha1(j,I2(j));
5490 %minimum moment in torsion spring 2 per precision point
5491 M2m(j) = A*alpha2(j,I2(j),I3(j))^2 + B*alpha2(j,I2(j),I3(j));
5492 %minimum moment in torsion spring 3 per precision point

```

```

5493     M3m(j) = A*alpha3(j,I2(j),I3(j))^2 + B*alpha3(j,I2(j),I3(j));
5494     %minimum moment in torsion spring 4 per precision point
5495     M4m(j) = A*alpha4(j,I2(j),I3(j))^2 + B*alpha4(j,I2(j),I3(j));
5496 end
5497
5498 %if springs are prestressed
5499 if prestress == 1 && nonlinearity == 0
5500     %minimum moment in torsion spring 1 per precision point
5501     M1m(j) = k1*alpha1(j,I2(j));
5502     %minimum moment in torsion spring 2 per precision point
5503     M2m(j) = k2*alpha2(j,I2(j),I3(j)) + M02;
5504     %minimum moment in torsion spring 3 per precision point
5505     M3m(j) = k3*alpha3(j,I2(j),I3(j)) + M03;
5506     %minimum moment in torsion spring 4 per precision point
5507     M4m(j) = k4*alpha4(j,I2(j),I3(j));
5508 end
5509
5510 %if springs are prestressed and nonlinear
5511 if prestress == 1 && nonlinearity == 1
5512     %minimum moment in torsion spring 1 per precision point
5513     M1m(j) = A*alpha1(j,I2(j))^2 + B*alpha1(j,I2(j));
5514     %minimum moment in torsion spring 2 per precision point
5515     M2m(j) = A*(alpha2(j,I2(j),I3(j))+alphastarM2)^2 +...
5516             B*(alpha2(j,I2(j),I3(j))+alphastarM2);
5517     %minimum moment in torsion spring 3 per precision point
5518     M3m(j) = A*(alpha3(j,I2(j),I3(j))+alphastarM3)^2 +...
5519             B*(alpha3(j,I2(j),I3(j))+alphastarM3);
5520     %minimum moment in torsion spring 4 per precision point
5521     M4m(j) = A*alpha4(j,I2(j),I3(j))^2 + B*alpha4(j,I2(j),I3(j));
5522 end
5523
5524 %minimum value of theta1 per precision point
5525 theta1m(j) = theta1(j,I2(j));
5526 %minimum value of theta2 per precision point
5527 theta2m(j) = theta2(j,I2(j),I3(j));
5528 %minimum value of theta3 per precision point
5529 theta3m(j) = theta3(j,I2(j),I3(j));
5530 %minimum value of theta4 per precision point
5531 theta4m(j) = theta4(j,I2(j),I3(j));
5532
5533 if objective == "sinus"
5534     Vm(j) = mg*r*cos(alpha(j)); %potential energy : height energy
5535     Mobj(j) = mg*r*sin(alpha(j)); %the moment around the origin caused by mass
5536 end
5537
5538 if objective == "Laevo"
5539     Vm(j) = 0.05022*alpha(j)^5 - 0.33575*alpha(j)^4 +...
5540           0.97*alpha(j)^3 - 1.412*alpha(j)^2 + 0.006501*alpha(j) + 1;
5541     Mobj(j) = -0.2511*alpha(j)^4 + 1.343*alpha(j)^3 -...
5542           2.91*alpha(j)^2 + 2.824*alpha(j) - 0.006501;
5543 end
5544
5545 if objective == "stiffening"
5546     Vm(j) = sin(alpha(j)) - alpha(j);
5547     Mobj(j) = -cos(alpha(j))+1;
5548 end
5549
5550 if objective == "sqrt"
5551     Vm(j) = - (2/3)*alpha(j)^(3/2);
5552     Mobj(j) = sqrt(alpha(j));
5553 end
5554
5555 if objective == "quadratic"
5556     Vm(j) = - (1/3)*alpha(j)^(3);
5557     Mobj(j) = alpha(j)^2;
5558 end
5559
5560 if objective == "hardening-softening"
5561     Vm(j) = 0.25*cos(2*alpha(j)-pi/2) - 0.5*alpha(j);
5562     Mobj(j) = (sin(2*alpha(j)-pi/2)+1)/2;
5563 end
5564
5565 if objective == "hardening-softening2"
5566     Vm(j) = -0.5*alpha(j)+(-0.333333+0.424413*alpha(j))*...
5567           atan(2.41421-3.07387*alpha(j))+...
5568           0.0690356*log(9.8696-21.4521*alpha(j)+13.6569*alpha(j)^2);
5569     Mobj(j) = 0.5 + (4/(3*pi))*atan( tan((3*pi)/8)*((4/pi)*alpha(j)-1));
5570 end
5571
5572 if objective == "softening-hardening"
5573     Vm(j) = 0.5*log(cos(alpha(j)-pi/4)) - 0.5*alpha(j);
5574     Mobj(j) = 0.5*tan(alpha(j)-pi/4) + 0.5;
5575 end
5576

```

```

5577 if objective == "softening-hardening2"
5578     Vm(j) = -0.5*alpha(j) - 0.0690356*log(1 + tan(1.1781 - 1.5*alpha(j))^2);
5579     Mobj(j) = 0.5*tan(1.5*(alpha(j)-pi/4))/tan(1.5*(pi/4))+0.5;
5580 end
5581
5582 if objective == "sinuspi"
5583     Vm(j) = 0.5*cos(2*alpha(j));
5584     Mobj(j) = sin(2*alpha(j));
5585 end
5586
5587 %vertical reaction force at segment 1 (positive upwards)
5588 F1y(j) = (M1m(j) - M4m(j) + (-M4m(j)/(14*cos(theta4m(j))))*...
5589     (11*cos(theta1m(j))+12*cos(theta2m(j))+13*cos(theta3m(j))))/...
5590     (-tan(theta4m(j))*(11*cos(theta1m(j))+...
5591     12*cos(theta2m(j))+13*cos(theta3m(j))))...
5592     + (11*sin(theta1m(j))+12*sin(theta2m(j))+13*sin(theta3m(j))));
5593
5594 %horizontal reaction force at segment 1 (positive to the right)
5595 F1x(j) = (-M4m(j) + F1y(j)*14*sin(theta4m(j)))/(14*cos(theta4m(j)));
5596
5597 %the external load (moment) on nodes 2 and 3...
5598 %...(where springs 2 and 3 are located), respectively
5599 M2l(j) = M1m(j) + F1x(j)*11*cos(theta1m(j)) - F1y(j)*11*sin(theta1m(j));
5600 M3l(j) = M1m(j) + F1x(j)*(11*cos(theta1m(j))+12*cos(theta2m(j))) -...
5601     F1y(j)*(11*sin(theta1m(j))+12*sin(theta2m(j)));
5602 end
5603
5604 %print the root mean square error (objective function)
5605 e = sqrt(mean((M1m - Mobj).^2)) %#ok<NOPTS>
5606 %integrate the residual moment - angle plot to obtain the required work
5607 IntM = trapz(alpha,abs(M1m-Mobj));
5608 %integrate the original moment - angle plot to obtain the required work
5609 IntMo = trapz(alpha,Mobj);
5610
5611 %arrays with data exported from SAM
5612 %moment and potential energy, respectively
5613 MSAM = [-0.03520,-0.05665,-0.07593,-0.09171,-0.11565,-0.12151,-0.12093,...
5614     -0.11309,-0.09704,-0.07212,-0.03772,0.00668,0.06148,0.12702,0.20346];
5615 VSAM = [0.99866,0.99414,0.98723,0.97789,0.96674,0.95425,0.94145,0.92896,...
5616     0.91733,0.90790,0.90161,0.89939,0.90228,0.91169,0.92878];
5617
5618 figure(3)
5619 hold on
5620 plot(alpha*180/pi,M1m)
5621 plot(alpha*180/pi,Mobj)
5622 plot(alpha*180/pi,M1m - Mobj)
5623 xlabel("Angle of rotation pendulum (deg)")
5624 ylabel("Moment around suspension-point 1 (Nm)")
5625 legend("Moment in spring 1", "Moment objective", "Error in moment",...
5626     "location","northwest")
5627
5628 figure(4)
5629 hold on
5630 plot(alpha*180/pi,Vmin+transpose(Vm))
5631 xlabel("Angle of rotation pendulum (deg)")
5632 ylabel("Total potential energy in system (J)")
5633
5634 if prestress == 1
5635     figure(5)
5636     hold on
5637     plot(alpha*180/pi,F1x)
5638     plot(alpha*180/pi,F1y)
5639     xlabel("Angle of rotation pendulum (deg)")
5640     ylabel("Reaction force in point 1 (N)")
5641     legend("F1x","F1y","location","northwest")
5642
5643     figure(6)
5644     hold on
5645     plot(alpha*180/pi,M2m)
5646     plot(alpha*180/pi,M3m)
5647     plot(alpha*180/pi,M2l)
5648     plot(alpha*180/pi,M3l)
5649     xlabel("Angle of rotation pendulum (deg)")
5650     ylabel("Reaction moment in nodes (Nm)")
5651     legend("M2m","M3m","M2l","M3l","location","northwest")
5652 end

```


Bibliography

- [1] D. Duan, N. Goemans, S. Takeda, E. Mercuri, and A. Aartsma-Rus, "Duchenne muscular dystrophy," *Nature Reviews Disease Primers* 2021 7:1, vol. 7, no. 1, pp. 1–19, 2 2021. [Online]. Available: <https://www.nature.com/articles/s41572-021-00248-3>
- [2] L. F. Cardoso, S. Tomázio, and J. L. Herder, "Conceptual Design of a Passive Arm Orthosis," *Proceedings of the ASME Design Engineering Technical Conference*, vol. 5 B, pp. 747–756, 6 2008.
- [3] R. L. Smith, J. Lobo-Prat, H. Van Der Kooij, and A. H. Stienen, "Design of a perfect balance system for active upper-extremity exoskeletons," *IEEE International Conference on Rehabilitation Robotics*, 2013. [Online]. Available: https://www.researchgate.net/publication/258255023_Design_of_a_perfect_balance_system_for_active_upper-extremity_exoskeletons
- [4] H. Zhang, K. Albee, and S. K. Agrawal, "A spring-loaded compliant neck brace with adjustable supports," *Mechanism and Machine Theory*, vol. 125, pp. 34–44, 7 2018.
- [5] D. F. P. Granados, H. Kadone, and K. Suzuki, "Unpowered Lower-Body Exoskeleton with Torso Lifting Mechanism for Supporting Sit-to-Stand Transitions," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2755–2761, 12 2018.
- [6] J. Lee and M. A. Nussbaum, "Experienced workers exhibit distinct torso kinematics/kinetics and patterns of task dependency during repetitive lifts and lowers," <http://dx.doi.org/10.1080/00140139.2012.723139>, vol. 55, no. 12, pp. 1535–1547, 12 2012. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/00140139.2012.723139>
- [7] "Laevo Exoskeletons." [Online]. Available: <https://www.laevo-exoskeletons.com/>
- [8] J. Herder, "Energy- free Systems: Theory, conception and design of statically balanced spring mechanisms," Ph.D. dissertation, TU Delft, Delft, 11 2001.
- [9] Y. Chheta R. and R. Joshi M., "A Review on Passive Gravity Compensation," in *International Conference on Electronics, Communication and Aerospace Technology*. Coimbatore: IEEE, 4 2017, pp. 184–189.
- [10] G. J. Walsh, D. A. Streit, and B. J. Gilmore, "Spatial spring equilibrators theory," *Mechanism and Machine Theory*, vol. 26, no. 2, pp. 155–170, 3 1991.
- [11] F. L. S. Te Riele and J. L. Herder, "Perfect static balance with normal springs," in *ASME 2001 Design Engineering Technical Conferences*. New York: ASME, 11 2001, pp. 571–578.
- [12] S. J. Baltrusch, J. H. van Dieën, A. S. Koopman, M. B. Näf, C. Rodriguez-Guerrero, J. Babič, and H. Houdijk, "SPEXOR passive spinal exoskeleton decreases metabolic cost during symmetric repetitive lifting," *European journal of applied physiology*, vol. 120, no. 2, pp. 401–412, 2 2020. [Online]. Available: <https://pubmed-ncbi-nlm-nih-gov.tudelft.idm.oclc.org/31828480/>
- [13] M. B. Näf, A. S. Koopman, S. Baltrusch, C. Rodriguez-Guerrero, B. Vanderborght, and D. Lefeber, "Passive back support exoskeleton improves range of motion using flexible beams," *Frontiers Robotics AI*, vol. 5, no. JUN, p. 72, 2018.
- [14] A. S. Koopman, M. Näf, S. J. Baltrusch, I. Kingma, C. Rodriguez-Guerrero, J. Babič, M. P. de Looze, and J. H. van Dieën, "Biomechanical evaluation of a new passive back support exoskeleton," *Journal of Biomechanics*, vol. 105, p. 109795, 5 2020.
- [15] M. Abdoli-Eramaki, J. M. Stevenson, S. A. Reid, and T. J. Bryant, "Mathematical and empirical proof of principle for an on-body personal lift augmentation device (PLAD)," *Journal of Biomechanics*, vol. 40, no. 8, pp. 1694–1700, 2007.

- [16] M. Abdoli-E, M. J. Agnew, and J. M. Stevenson, "An on-body personal lift augmentation device (PLAD) reduces EMG amplitude of erector spinae during lifting tasks," *Clinical Biomechanics*, vol. 21, no. 5, pp. 456–465, 6 2006.
- [17] M. Abdoli-E and J. M. Stevenson, "The effect of on-body lift assistive device on the lumbar 3D dynamic moments and EMG during asymmetric freestyle lifting," *Clinical Biomechanics*, vol. 23, no. 3, pp. 372–380, 3 2008.
- [18] K. Knitel, J. van Eck, T. Bosch, and M. de Looze, "Innovatieve rugondersteuning: de effecten van een passief exoskelet op spieractiviteit en volhoudtijd," *Tijdschrift voor Human Factors*, vol. 40, no. 40, p. 4, 2015. [Online]. Available: <https://repository.tno.nl/islandora/object/uuid%3A1d02f5a2-ea4d-42f1-bd80-626b1a4b4471>
- [19] L. L. Howell, *Compliant Mechanisms*. Wiley, 2001.
- [20] L. L. Howell, S. P. Magleby, and B. M. B. M. Olsen, *Handbook of compliant mechanisms*. Wiley, 2013.
- [21] "Home - SAM - The Ultimate Mechanism Designer - Artas Engineering." [Online]. Available: <https://www.artas.nl/nl/>
- [22] "SOLIDWORKS." [Online]. Available: <https://www.solidworks.com/>
- [23] "What is LabVIEW? Graphical Programming for Test & Measurement - NI." [Online]. Available: <https://www.ni.com/nl-nl/shop/labview.html>
- [24] J. K. Vandiver, "Engineering Dynamics | Mechanical Engineering | MIT OpenCourseWare," 2011. [Online]. Available: <https://ocw.mit.edu/courses/2-003sc-engineering-dynamics-fall-2011/>
- [25] "Thorlabs - MB3060/M Aluminum Breadboard, 300 mm x 600 mm x 12.7 mm, M6 Taps." [Online]. Available: <https://www.thorlabs.com/thorproduct.cfm?partnumber=MB3060/M>
- [26] "Thorlabs - XE25L225/M 25 mm Square Construction Rail, 225 mm Long, M6 Taps." [Online]. Available: <https://www.thorlabs.com/thorproduct.cfm?partnumber=XE25L225/M>
- [27] "Angle Position Sensor AN8 - ZF Switches & Sensors US." [Online]. Available: <https://switches-sensors.zf.com/us/product/angle-position-sensor-an8/>
- [28] "Miniature S-Beam Jr. Load Cell LSB200 : FSH00102." [Online]. Available: <https://www.futek.com/store/legacy-sensors-and-instruments/miniature-s-beam-LSB200/FSH00102>
- [29] "907: SCS 0,6X6X8." [Online]. Available: <https://catalog.lesjoforsab.com/product/907-scs-0-6x6x8>
- [30] "903: SCS 0,5X5X8." [Online]. Available: <https://catalog.lesjoforsab.com/product/903-scs-0-5x5x8>
- [31] "908: SCS 0,7X4X5." [Online]. Available: <https://catalog.lesjoforsab.com/product/908-scs-0-7x4x5>
- [32] "608-2Z | RS PRO Miniature Ball Bearing - Plain Race Type, 8mm I.D, 22mm O.D | RS." [Online]. Available: <https://nl.rs-online.com/web/p/ball-bearings/6190036>
- [33] "DDL1280ZZMTHA5P24LY121 | NMB Radial Ball Bearing - Plain Race Type, 8mm I.D, 12mm O.D | RS." [Online]. Available: <https://nl.rs-online.com/web/p/ball-bearings/0747800>
- [34] "DDL-1060ZZMTP24LY121 | NMB Radial Ball Bearing - Plain Race Type, 6mm I.D, 10mm O.D | RS." [Online]. Available: <https://nl.rs-online.com/web/p/ball-bearings/0747771>
- [35] "Glijlager Kunststof Polyamide (nylon) 6.6 8 (8715492869371) | Fabory." [Online]. Available: <https://www.fabory.com/nl/glijlager-kunststof-polyamide-%28nylon%29-6-6-8/p/56806080001>
- [36] "GAMMA | GAMMA sluitring M8 nylon 15 stuks kopen? | ringen." [Online]. Available: <https://www.gamma.nl/assortiment/gamma-sluitring-m8-nylon-15-stuks/p/B458519>
- [37] "Vlakke sluitring DIN 125-1A Staal Elektrolytisch verzinkt 140 HV M8 (8715492085696) | Fabory." [Online]. Available: <https://www.fabory.com/nl/vlakke-sluitring-din-125-1a-staal-elektrolytisch-verzinkt-140-hv-m8/p/38130080001>

- [38] “Vlakke sluitring DIN 125-1A Staal Elektrolytisch verzinkt 140 HV M6 (8715492085658) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/vlakke-sluitring-din-125-1a-staal-elektrolytisch-verzinkt-140-hv-m6/p/38130060001>
- [39] “Vlakke sluitring DIN 125-1A Staal Elektrolytisch verzinkt 140 HV M5 (8715492085641) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/vlakke-sluitring-din-125-1a-staal-elektrolytisch-verzinkt-140-hv-m5/p/38130050001>
- [40] “Vlakke sluitring DIN 125-1A Staal Elektrolytisch verzinkt 140 HV M3 (8715492085666) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/vlakke-sluitring-din-125-1a-staal-elektrolytisch-verzinkt-140-hv-m3/p/38130030001>
- [41] “Snelborger voor assen Verenstaal 8MM (8715492757616) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/snelborger-voor-assen-verenstaal-8mm/p/36230080001>
- [42] “Zeskantmoer DIN 934 Staal Elektrolytisch verzinkt |8| M3 (8715492042033) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/zeskantmoer-din-934-staal-elektrolytisch-verzinkt-%7c8%7c-m3/p/01300030001>
- [43] “Zeskantmoer grootverpakking DIN 934 Staal Elektrolytisch verzinkt |8| grootverpakking M5 (8715492042521) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/zeskantmoer-grootverpakking-din-934-staal-elektrolytisch-verzinkt-%7c8%7c-grootverpakking-m5/p/01310050001>
- [44] “Cilinderschroef met binnenzeskant DIN 912 Staal Elektrolytisch verzinkt 8.8 M6X20 (8715492071361) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/cilinderschroef-met-binnenzeskant-din-912-staal-elektrolytisch-verzinkt-8-8-m6x20/p/07160060020?q=071630.060.020%3Arelevance>
- [45] “Cilinderschroef met binnenzeskant DIN 912 Staal Elektrolytisch verzinkt 8.8 M5X20 (8715492071149) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/cilinderschroef-met-binnenzeskant-din-912-staal-elektrolytisch-verzinkt-8-8-m5x20/p/07160050020?q=071630.050.020%3Arelevance>
- [46] “Cilinderschroef met binnenzeskant DIN 912 Staal Elektrolytisch verzinkt 8.8 M3X20 (8715492070814) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/cilinderschroef-met-binnenzeskant-din-912-staal-elektrolytisch-verzinkt-8-8-m3x20/p/07160030020?q=071630.030.020%3Arelevance>
- [47] “Cilinderschroef met binnenzeskant DIN 912 Staal Elektrolytisch verzinkt 8.8 M3X12 (8715492070777) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/cilinderschroef-met-binnenzeskant-din-912-staal-elektrolytisch-verzinkt-8-8-m3x12/p/07160030012?q=071630.030.012%3Arelevance>
- [48] “GAMMA | GAMMA metaalschroef M3 x 30mm platkop verzinkt 15 stuks kopen? | bouten.” [Online]. Available: <https://www.gamma.nl/assortiment/gamma-metaalschroef-m3-x-30mm-platkop-verzinkt-15-stuks/p/B458167>
- [49] “Stelschroef met binnenzeskant en afschuining DIN 913 Staal Blank 45H M3X12 (8715492158345) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/stelschroef-met-binnenzeskant-en-afschuining-din-913-staal-blank-45h-m3x12/p/07810030012>
- [50] “Draadstang lengte 1 meter DIN 976-1A Staal Elektrolytisch verzinkt 4.8 M3 (8715492606778) | Fabory.” [Online]. Available: <https://www.fabory.com/nl/draadstang-lengte-1-meter-din-976-1a-staal-elektrolytisch-verzinkt-4-8-m3/p/20300030001?q=20300.003.001%3Arelevance>
- [51] “Flat Spiral Springs: Calculation Formulas | Tokai Spring industries, Inc.” [Online]. Available: <https://www.tokaibane.com/en/spring-design/flat-spiral-springs-formulas#Section1>
- [52] “MITcalc - Geometrical design and strength check of 15 springs types.” [Online]. Available: <https://www.mitcalc.com/doc/springs/help/en/springs.htm>

-
- [53] "AllroundLine universele testmachine | ZwickRoell." [Online]. Available: <https://www.zwickroell.com/nl/producten/statische-materiaaltestmachines/universele-testmachines-voor-statische-toepassingen/allroundline/>
- [54] "Download PuTTY - a free SSH and telnet client for Windows." [Online]. Available: <https://www.putty.org/>
- [55] "WinSCP :: Official Site :: Free SFTP and FTP client for Windows." [Online]. Available: <https://winscp.net/eng/index.php>