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

Delft University of Technology

Challenge the future

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

TU Delft

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

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.

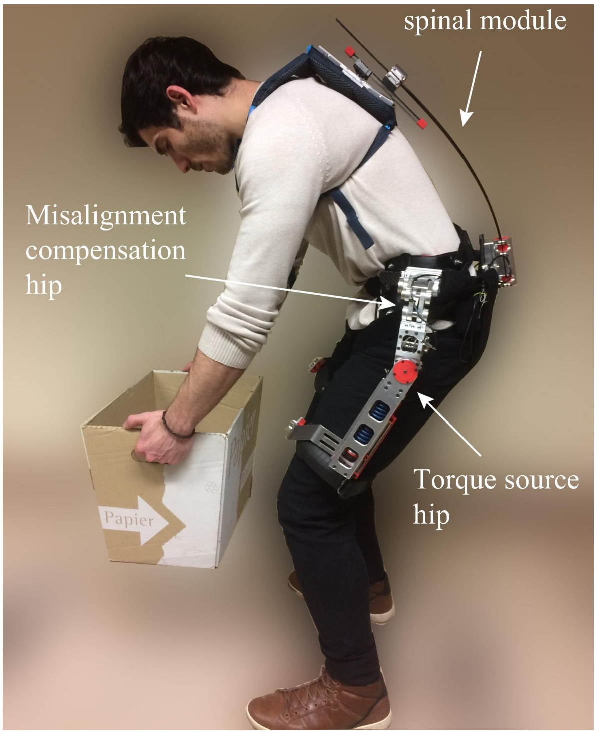
In most cases, static balance is achieved with use of countermasses or springs. Countermasses are frequently used in gravity balancers, where the gravitational force and induced moment of another mass are balanced. Disadvantages of systems that comprise these countermasses are increased volume, mass 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.

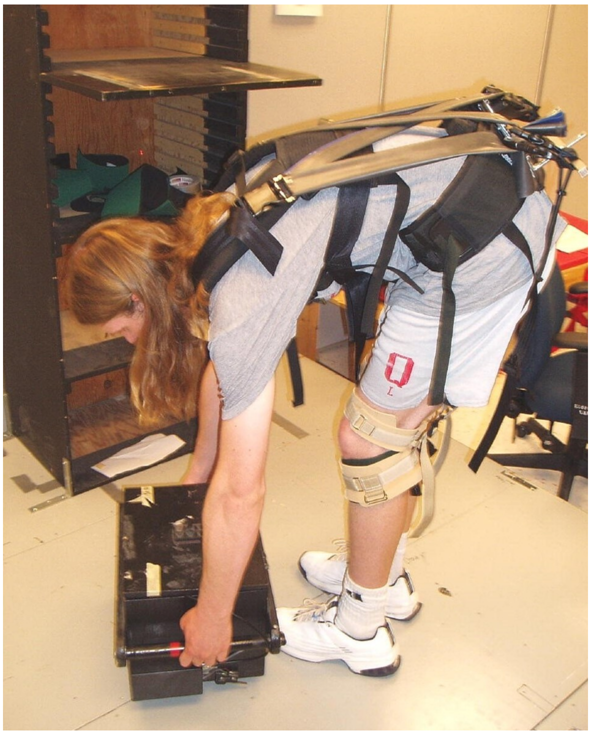
(a) Industrial application Laevo exoskeleton

(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

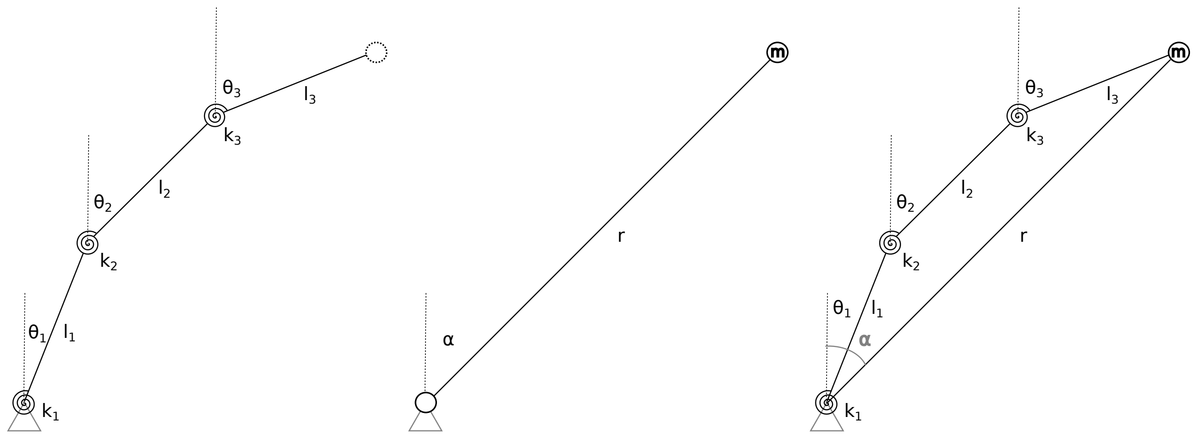
Although the presented exoskeletons are able to provide support and conformability to the human body, the working principle of the Laevo and SPEXOR devices are based on a pulley that is used as a transmission of forces and displacements. The latter increases the complexity and the size of the exoskeleton. 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.



(a) Three segment balancer (b) Inverted pendulum (c) System including balancer and pendulum

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.

Linssen 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
Doornenbal[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

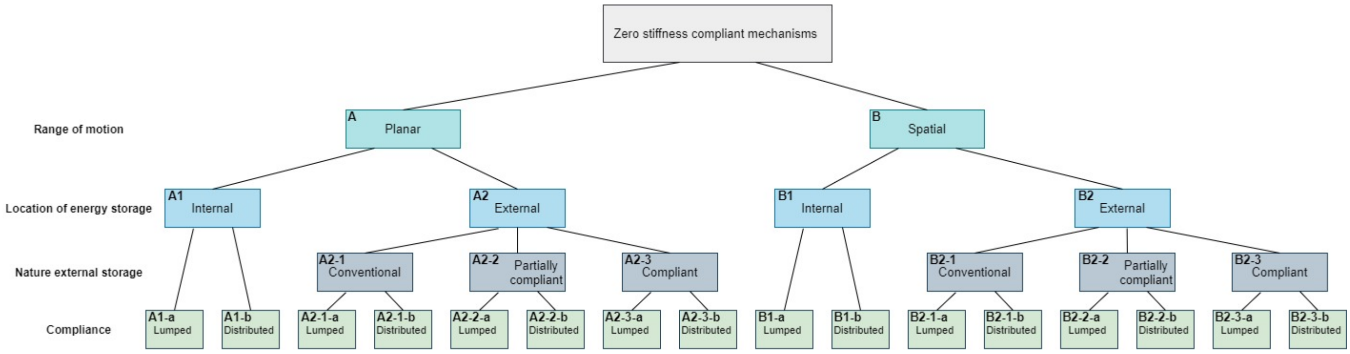


Fig. 1: Division of zero stiffness compliant mechanisms into (sub-) categories

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 it's 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.

$$k_{red} = 100\frac{k_{sb} - k_p}{k_p} \tag{1}$$

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 it's 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 it's 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			
Location energy storage	Internal	External			Internal	External		
Nature external storage		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.

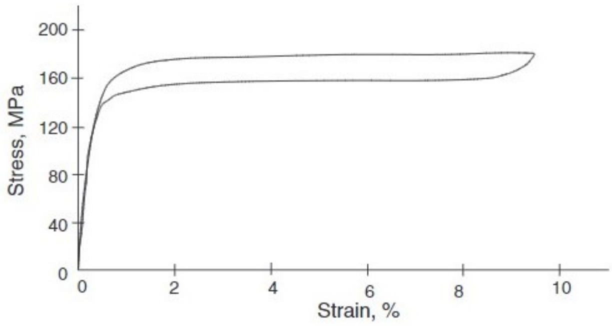


Fig. 3: Common stress- strain characteristic of a pseudoelastic material [26]

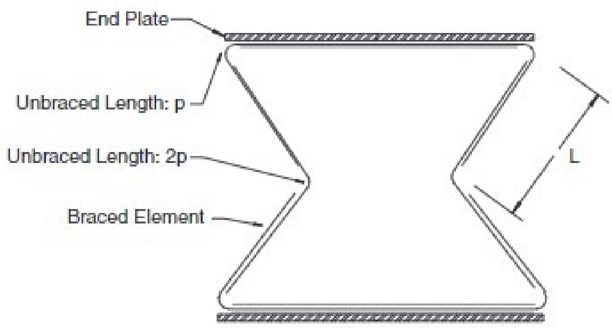


Fig. 4: Constant force spring by Soroushian et al. [26]

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-

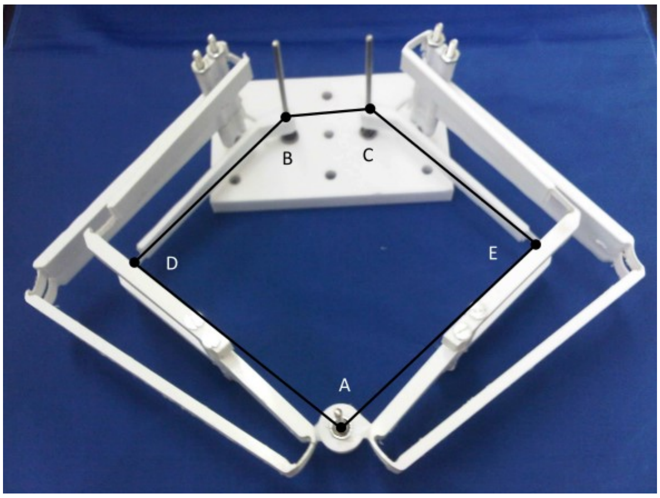


Fig. 5: Neutrally stable compliant pantograph by Merriam et al. [30]

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°.

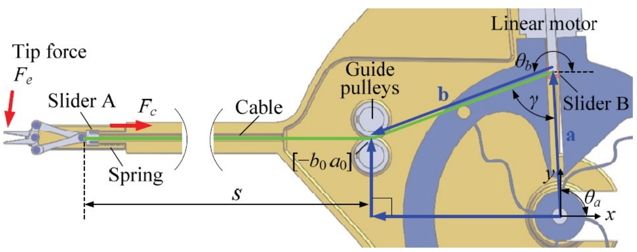


Fig. 6: Medical constant force forceps by Lan and Wang [31]

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.

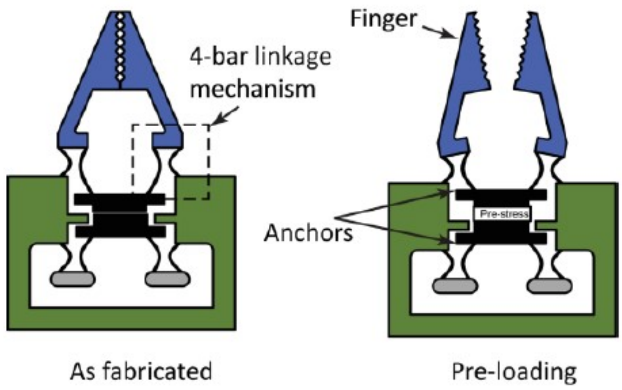


Fig. 7: Statically balanced gripper by Nguyen et al. [32]

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 eachother 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

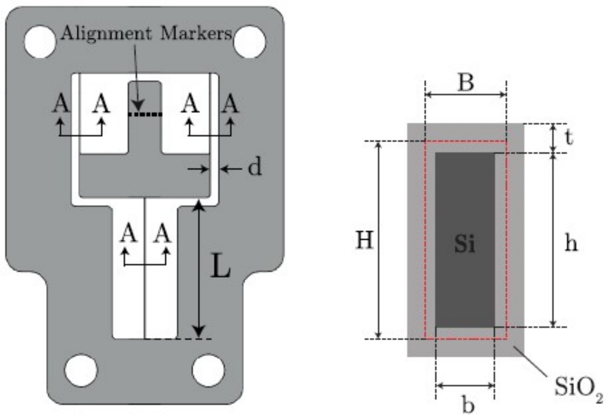


Fig. 8: The statically balanced linear motion stage (left) and the cross- section of the lower flexure (right) by Kuppens et al. [33]

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.

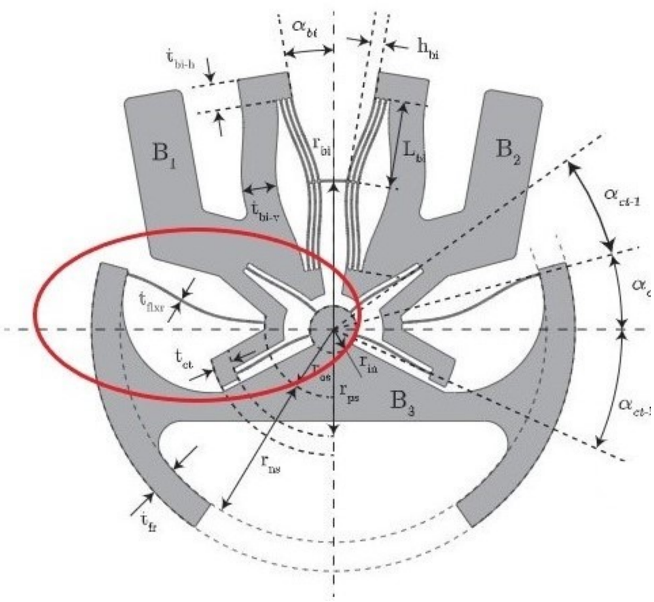


Fig. 9: Statically balanced compliant mechanism by Kuppens et al. [34]

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 +/- 30°.

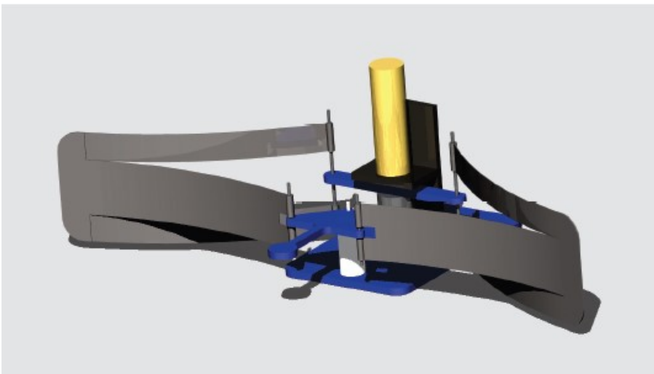
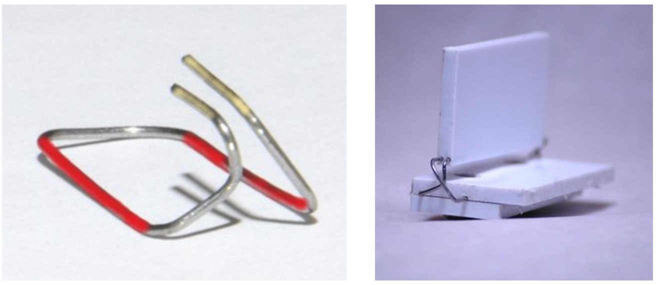


Fig. 10: CAD model of spring butterfly- based haptic interface device by Leishman et al. [36]

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 eachother. 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



(a) (b)

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.

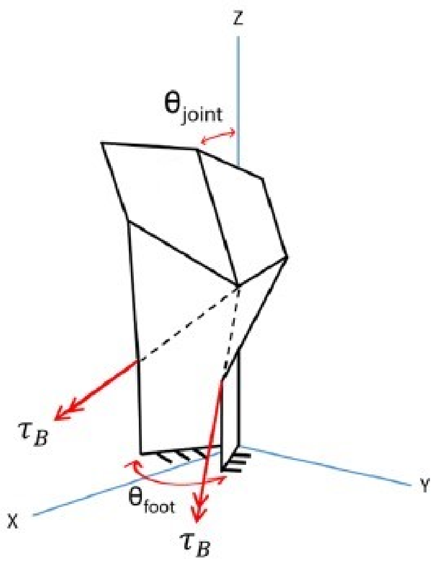


Fig. 13: PRBM of single vertex compliant facet origami mechanism presented by Rommers et al. [39]

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

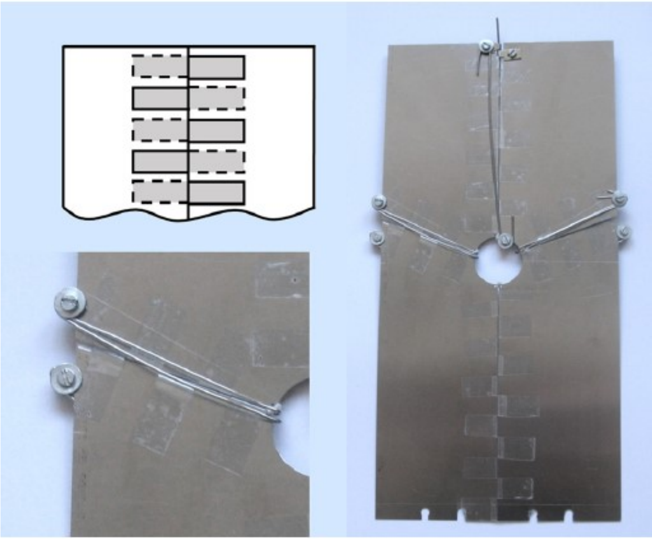


Fig. 14: Design of constant moment joint by Rommers et al. [40]

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 Pallegrino 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

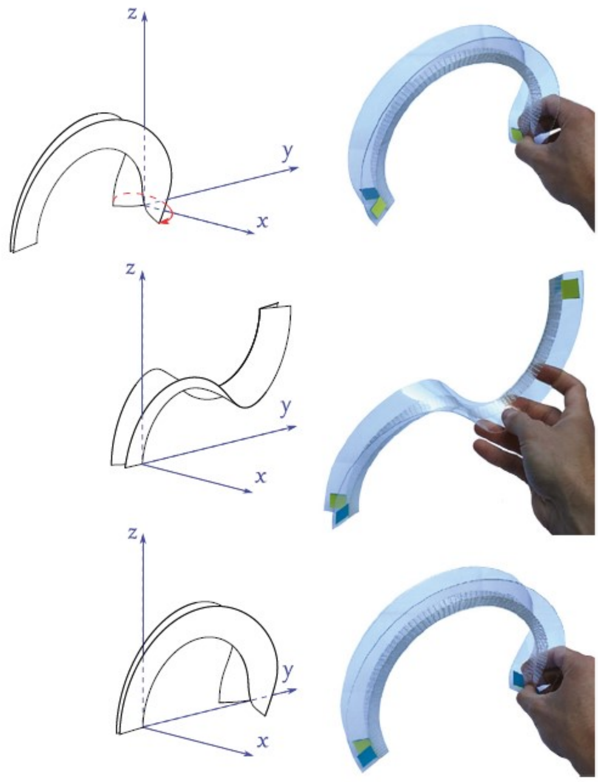
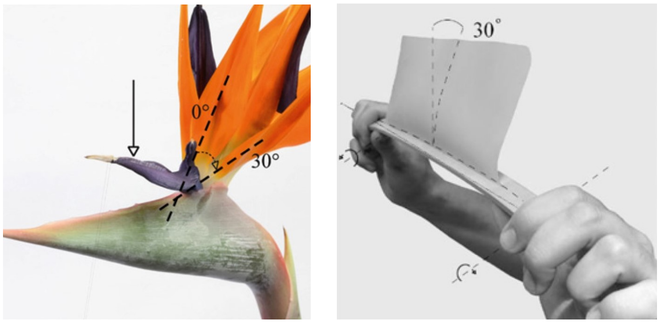


Fig. 15: Motion of neutrally stable shell structure by Kok et al. [41]

Fig. 16: Neutrally stable tape spring by Murphey and Pallegrino [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. It's 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 it's high tensile strength and low bending stiffness.



(a) (b)

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 **u**. The same point is subjected to a force vector **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 perturbance by

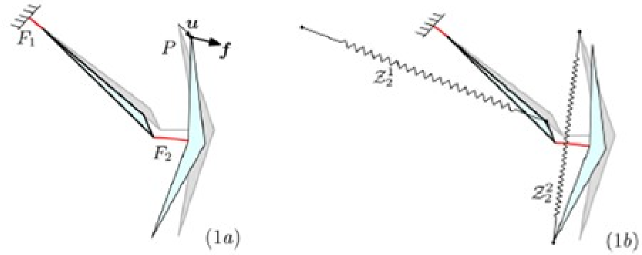


Fig. 18: The 2- DOF probe presented by Deepak et al. [44]

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.

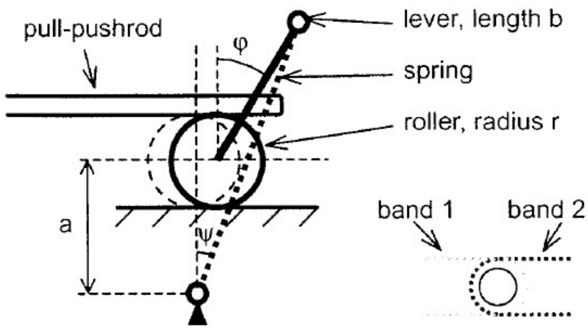


Fig. 19: The rolling- link balancing mechanism by Herder and Van den Berg [21]

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.

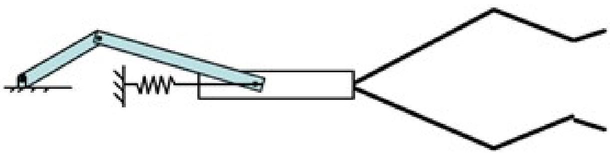


Fig. 20: The ophthalmic forceps including a static balancing mechanism designed by Powell and Frecker [46]

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.

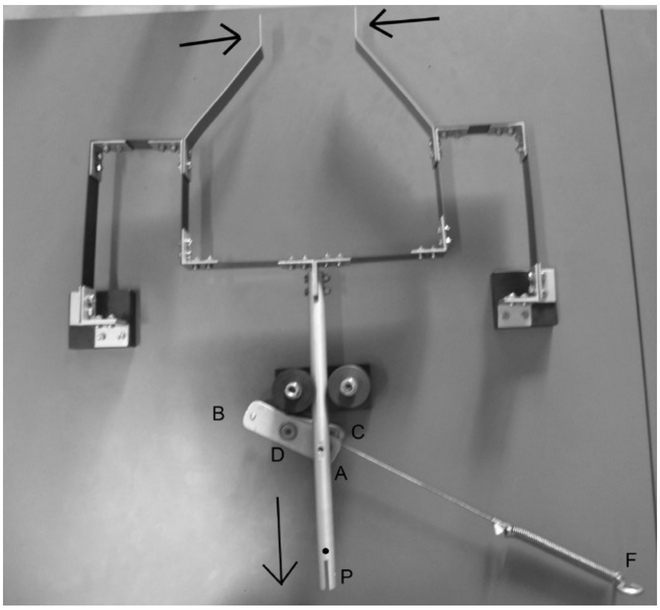


Fig. 21: The compliant gripper by Deepak et al. [44]

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 Dijksman [48], it was decided to use compressed cantilever leaf springs as compensation mechanism. In contrast to Dijksman'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

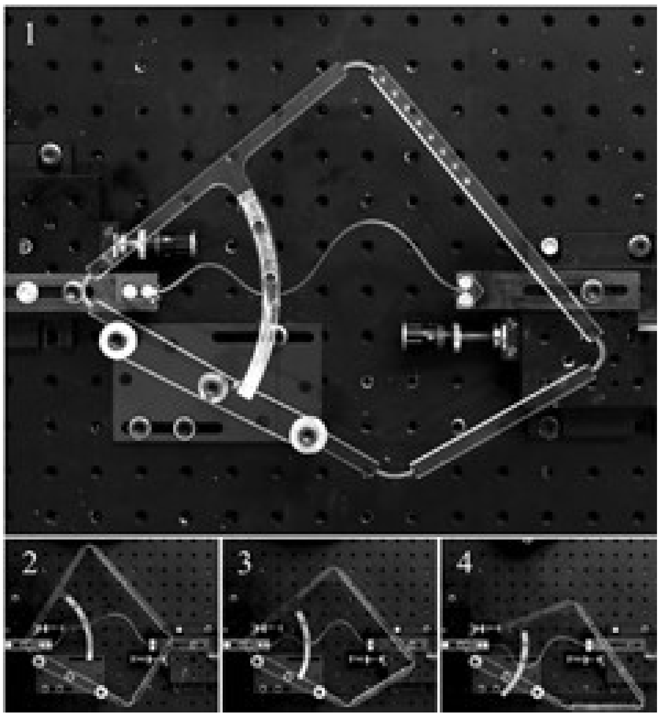


Fig. 22: Prototype of the statically balanced four- bar mechanism by Berntsen et al. [47]

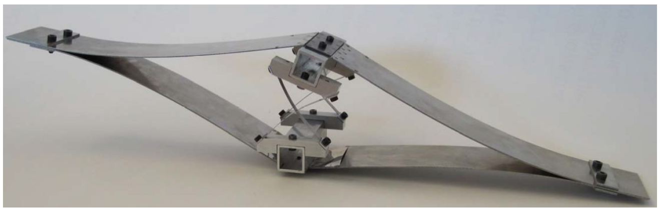


Fig. 23: Zero stiffness compliant joint by Morsch and Herder [49]

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].

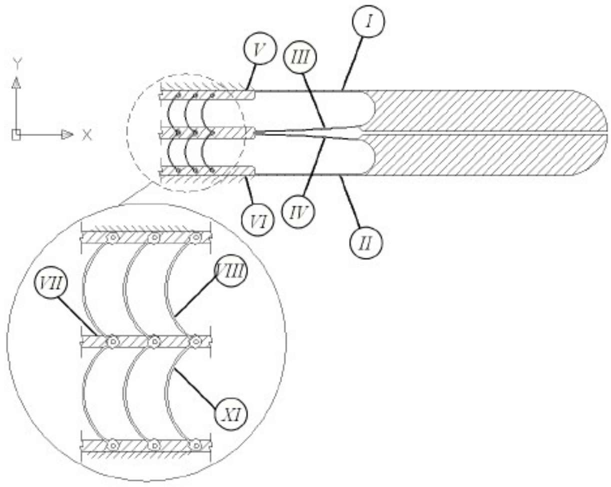


Fig. 24: The statically balanced grasper presented by Tolou and Herder [50]

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

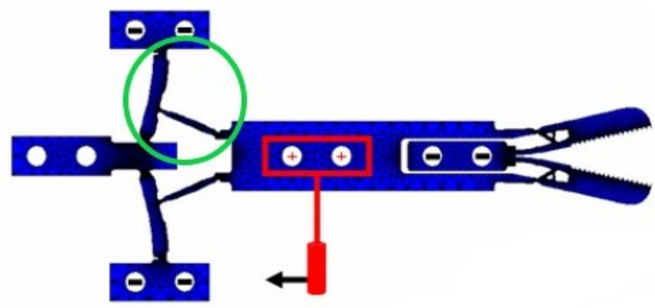


Fig. 25: Statically balanced compliant grasper designed by De Lange et al. using topology optimisation [51]

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.

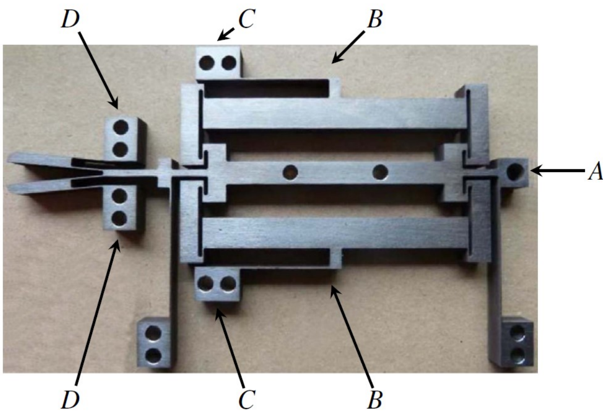


Fig. 26: Statically balanced compliant grasper by Lamers et al. [53]

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 it's 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%.

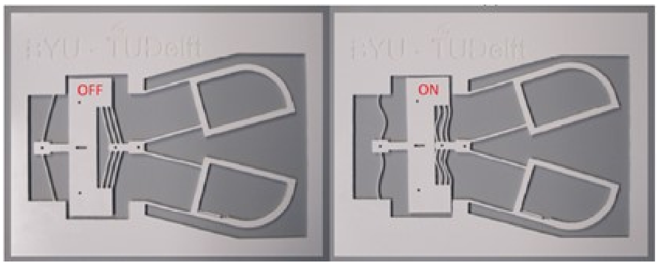


Fig. 27: Compliant mechanism with "static balancing-switch" by Pluimers et al. [35]

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.

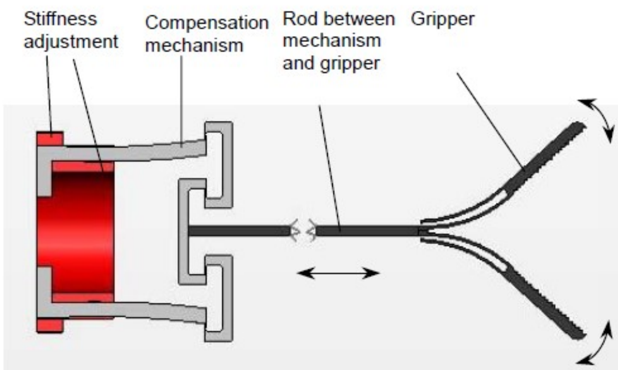


Fig. 28: Schematic of the laparoscopic grasper by Stapel and Herder [52]

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 eachother. 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.

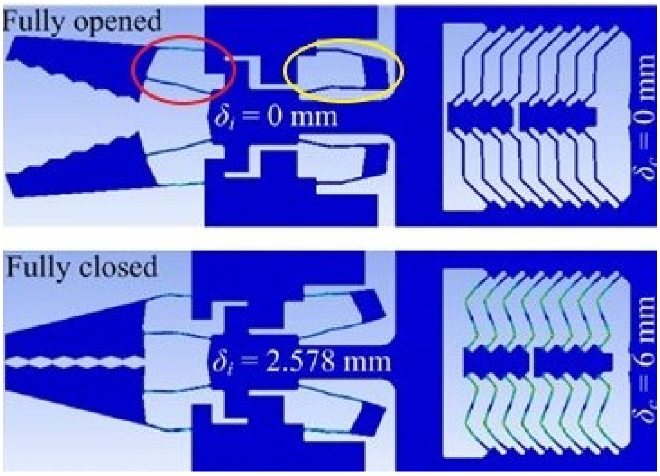


Fig. 29: Constant force compliant gripper by Wang and Lan [54]

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

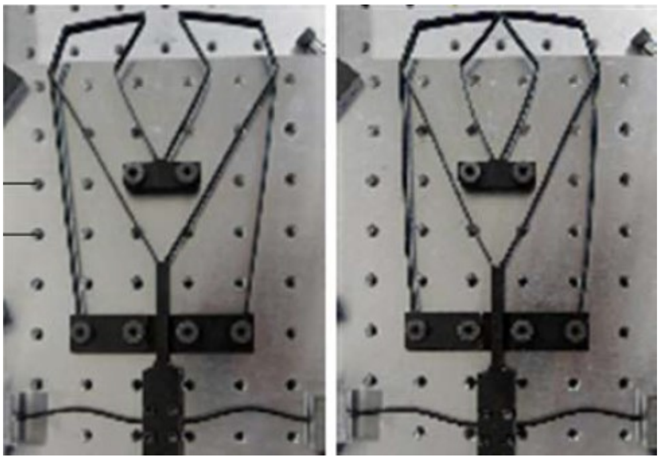


Fig. 30: Statically balanced compliant gripper designed by Hoetmer et al. using the extended building block approach [55]

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

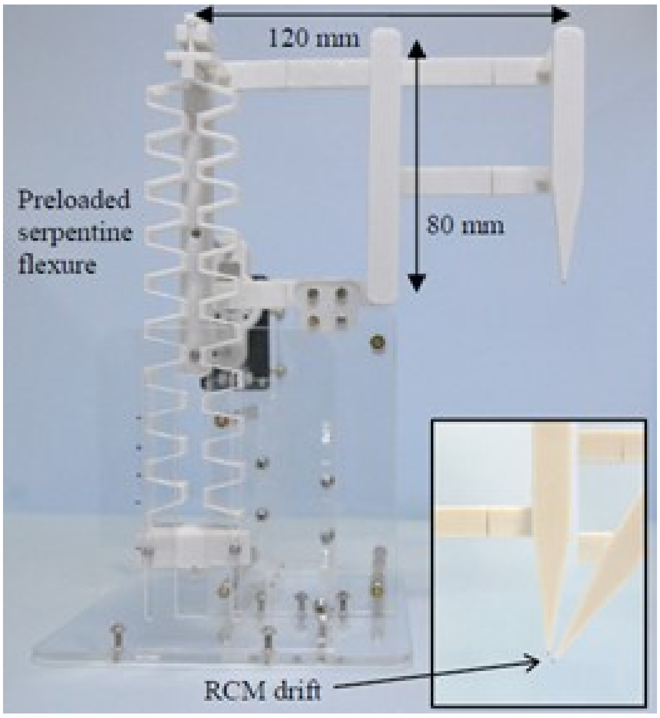


Fig. 31: Remote center of rotation statically balanced surgical tool by Chandrasekaran et al. [57]

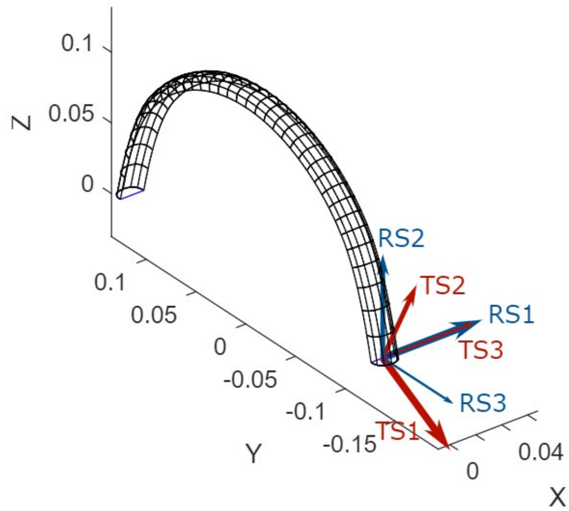


Fig. 32: Shell analysed for zero stiffness behaviour by Dekens [58]

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 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.

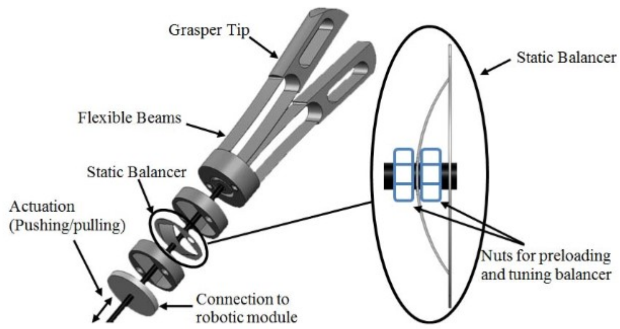


Fig. 33: Statically balanced end effector by Lassooij et al. [59]

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 it's 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 2mm 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.

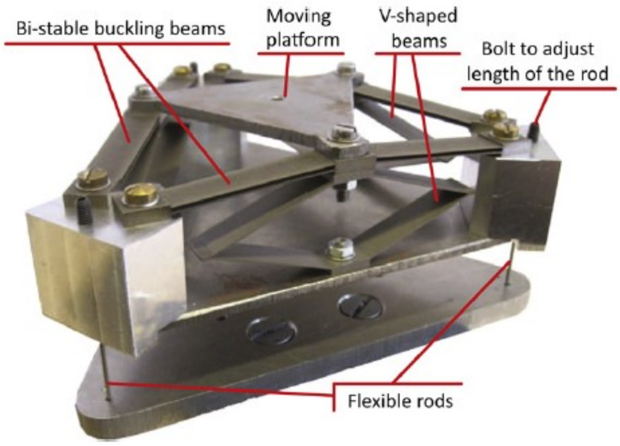


Fig. 34: Six DOF precision stage by Dunning et al. [60]

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 Pallegrino [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	$T_x = T_y = T_z = 2$mm, $R_x = R_y = 10$mrad, $R_z = 15$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 it's 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 of 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 (SAMOII) 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. Doornenbal, "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. Dijksman, "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/BIOMED/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. Lassooij, 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.

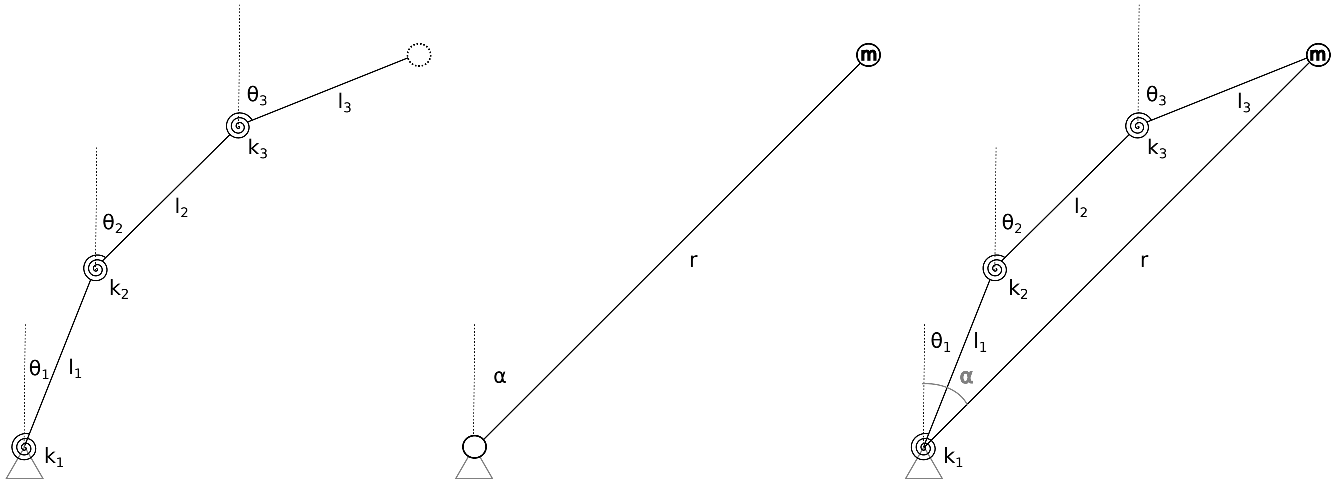
Radaelli and Herder used isogeometric shape optimization to statically balance an inverted pendulum with a prestressed compliant beam [20]. Eventually, a prototype with a carbon fibre composite beam was developed, which illustrated a work reduction of 96.98% for a 180 degree rotation of the inverted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



(a) Three segment balancer (b) Inverted pendulum (c) System including balancer and pendulum

Figure 1: Schematic overview of the proposed system

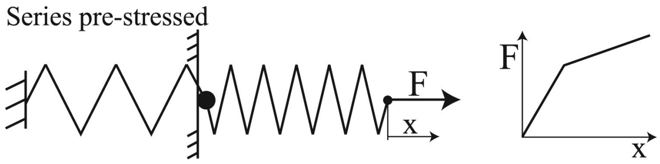
II. Fundamentals

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.

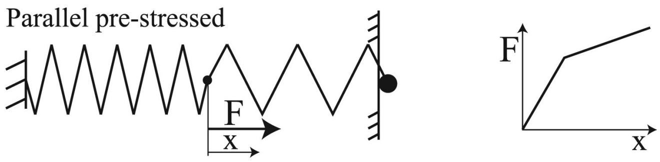
$$k_\mathrm{T} = \sum_{i=1}^{N} k_i \tag{1}$$

$$\frac{1}{k_\mathrm{T}} = \sum_{i=1}^{N} \frac{1}{k_i} \tag{2}$$

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.



(a) Prestressed series connection with contact



(b) Prestressed parallel connection with contact

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.

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

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

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.

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

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.

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

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

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 nxm 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.

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

$$\alpha_{1ji} = \theta_{1ji} - \theta_{1_0} \quad (9)$$

$$\alpha_{2ji} = \theta_{2ji} - \theta_{1ji} - \left(\theta_{2_0} - \theta_{1_0}\right) \quad (10)$$

$$\alpha_{3ji} = \theta_{3ji} - \theta_{2ji} - \left(\theta_{3_0} - \theta_{2_0}\right) \quad (11)$$

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.

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

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_{1_0}, θ_{2_0} and θ_{3_0}, 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.

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

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

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

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

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.

$$f_1 = \frac{1}{mgr}\sqrt{\frac{\sum_{j=1}^{N}\left(M_{1_j} - M_{\text{obj}_j}\right)^2}{N}} \tag{18}$$

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.

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

$$\left(\Delta V_{12_j}\right)^2 = \left(V_{1\text{m}_j} - V_{2\text{m}_j}\right)^2 \tag{20}$$

$$\left(\Delta V_{13_j}\right)^2 = \left(V_{1\text{m}_j} - V_{3\text{m}_j}\right)^2 \tag{21}$$

$$\left(\Delta V_{23_j}\right)^2 = \left(V_{2\text{m}_j} - V_{3\text{m}_j}\right)^2 \tag{22}$$

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.

$$M_\text{p} = -mgr\sin(\alpha) \tag{23}$$

$$M_{\text{obj}_\text{p}} = mgr\sin(\alpha) \tag{24}$$

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 $mg = 1$.

$$M_{\text{obj}_\text{h}} = -mgr\cos(\alpha) + mgr \tag{25}$$

$$M_{\text{obj}_\text{hs}} = 0.5 + \frac{4}{3\pi}\arctan\left(\tan\left(\frac{3\pi}{8}\right)\left(\frac{4}{\pi}\alpha - 1\right)\right) \tag{26}$$

$$M_{\text{obj}_\text{sh}} = 0.5\frac{\tan\left(1.5\left(\alpha - \frac{\pi}{4}\right)\right)}{\tan\left(1.5\left(\frac{\pi}{4}\right)\right)} + 0.5 \tag{27}$$

$$M_{\text{obj}_\text{L}} = -0.25\alpha^4 + 1.34\alpha^3 - 2.91\alpha^2 + 2.82\alpha - 0.01 \tag{28}$$

$$M_{\text{obj}_\text{s}} = \sin(2\alpha) \tag{29}$$

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.

$$\min_{k_1,k_2,k_3,l_1,l_2,l_3} \sqrt{\frac{\sum_{j=1}^{N}\left(M_{1_j}-M_{\text{obj}_j}\right)^2}{N}}$$
$$s.t. 0 \leq k_i \leq \frac{3}{2}$$
$$\frac{1}{3} \leq l_i \leq \frac{1}{2}$$

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

$$\min_{k_1,k_2,k_3,l_1,l_2,l_3} \frac{1}{mgrN}\sum_{j=1}^{N}\sqrt{\left(\Delta V_{12_j}\right)^2+\left(\Delta V_{13_j}\right)^2+\left(\Delta V_{23_j}\right)^2}$$
$$\left(\Delta V_{12_j}\right)^2 = \left(V_{1\text{m}_j}-V_{2\text{m}_j}\right)^2$$
$$\left(\Delta V_{13_j}\right)^2 = \left(V_{1\text{m}_j}-V_{3\text{m}_j}\right)^2$$
$$\left(\Delta V_{23_j}\right)^2 = \left(V_{2\text{m}_j}-V_{3\text{m}_j}\right)^2$$
$$s.t. 0 \leq k_i \leq \frac{3}{2}$$
$$\frac{1}{3} \leq l_i \leq \frac{1}{2}$$

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_2.

$$\min_{x_1,x_2,x_3,x_4,x_5,x_6} \sqrt{\left(\frac{k_1+k_2}{k_3+k_4}-r_1\right)^2+\left(\frac{k_5+k_6}{k_3+k_4}-r_2\right)^2}$$
$$k_1 = v_k\left(x_1\right)$$
$$k_2 = v_k\left(x_2\right)$$
$$k_3 = v_k\left(x_3\right)$$
$$k_4 = v_k\left(x_4\right)$$
$$k_5 = v_k\left(x_5\right)$$
$$k_6 = v_k\left(x_6\right)$$
$$s.t. 1 \leq x_i \leq 13$$

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.

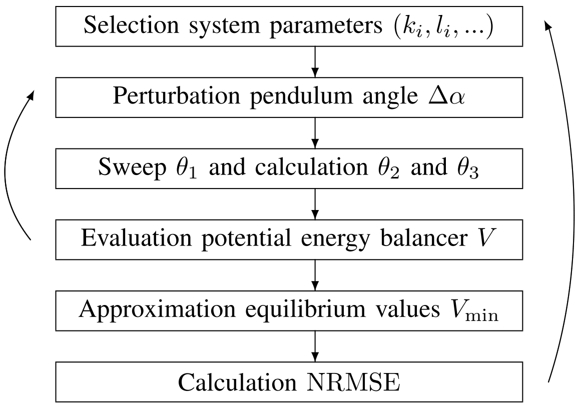


Figure 3: Modeling scheme standard three segment balancer

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-cutted 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.

$$r_1 = \frac{k_1}{k_2} = \frac{1.27}{0.85} = 1.50 \tag{30}$$

$$r_2 = \frac{k_3}{k_2} = \frac{4.11}{0.85} = 4.85 \tag{31}$$

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

$$r_{1\mathrm{p}} = \frac{k_{1\mathrm{p}}}{k_{2\mathrm{p}}} = \frac{0.037}{0.025} = 1.51 \tag{32}$$

$$r_{2\mathrm{p}} = \frac{k_{3\mathrm{p}}}{k_{2\mathrm{p}}} = \frac{0.12}{0.025} = 4.84 \tag{33}$$

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_{1_0}	0.66	0.66	rad
r	1.00	0.50	m

Table I: System parameters corresponding to prototype

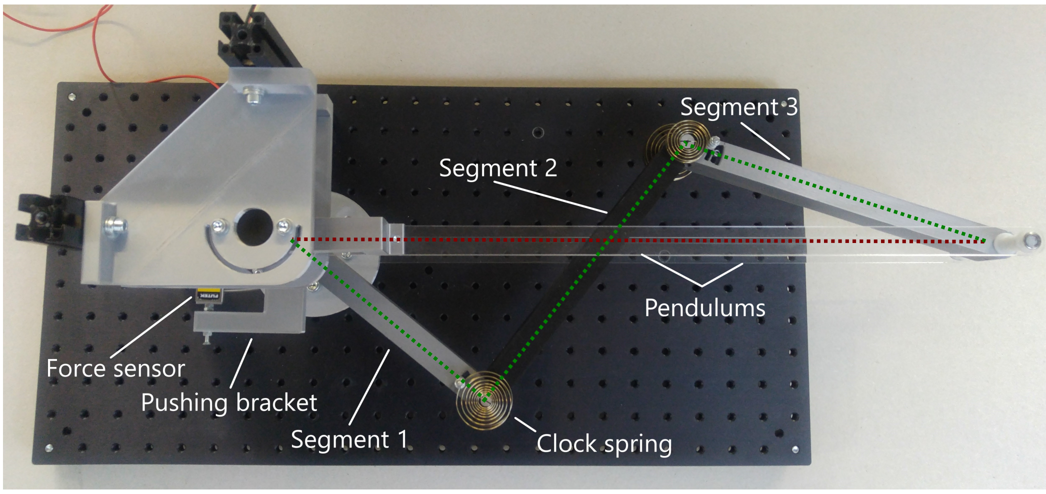
IV. Results

A. Model

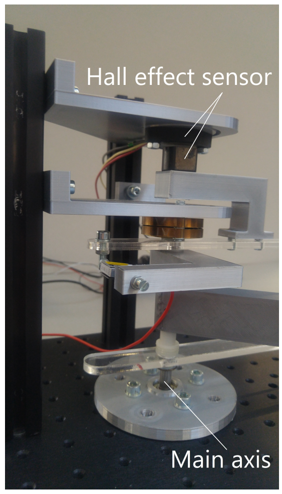
The optimization results for the various configurations of the three segment balancer are depicted in figure 5. The distinct objective functions are shown along the x-axis, whereas the y-axis illustrates the best obtained work reduction for each system. The work in the balanced and reference configurations 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.

$$W_{\mathrm{red}} = 100\frac{W_{\mathrm{ref}} - W_{\mathrm{bal}}}{W_{\mathrm{ref}}} \tag{34}$$

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



(a) General overview prototype



(b) Connection main axis

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

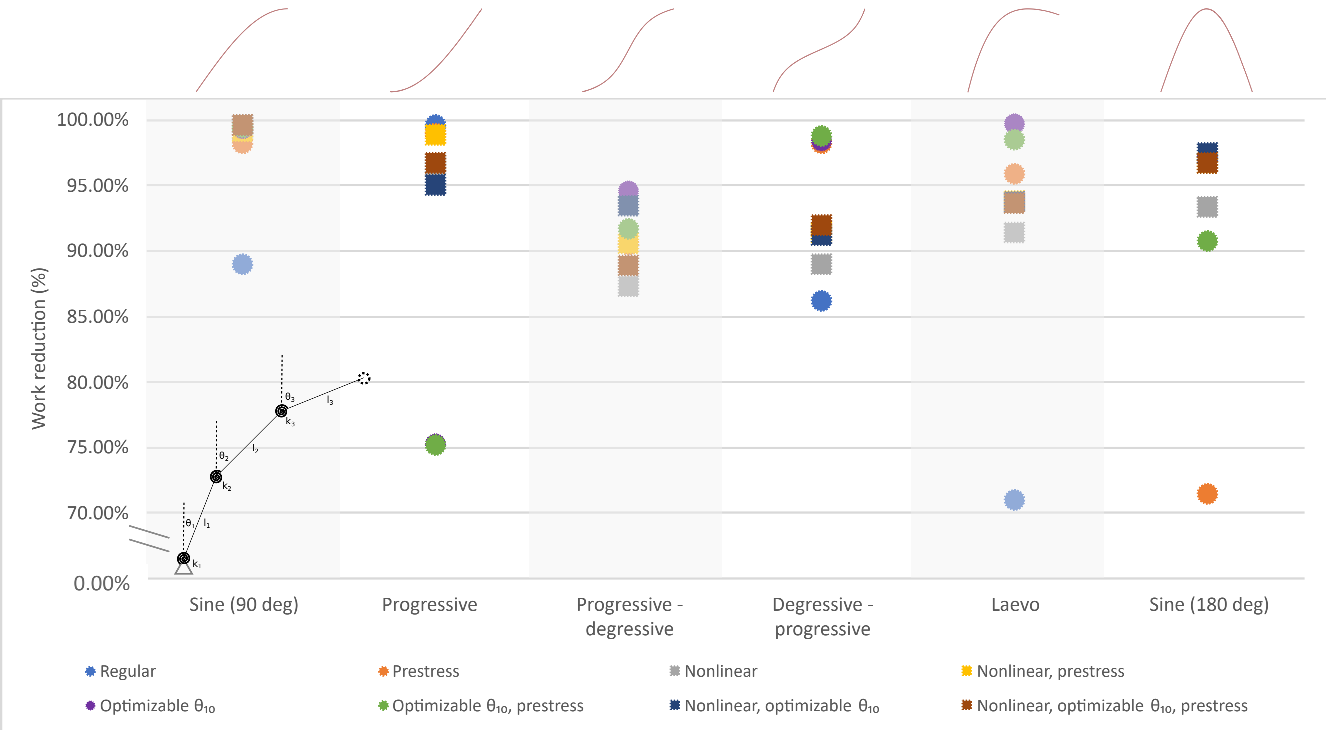


Figure 5: Modeling results three segment balancer

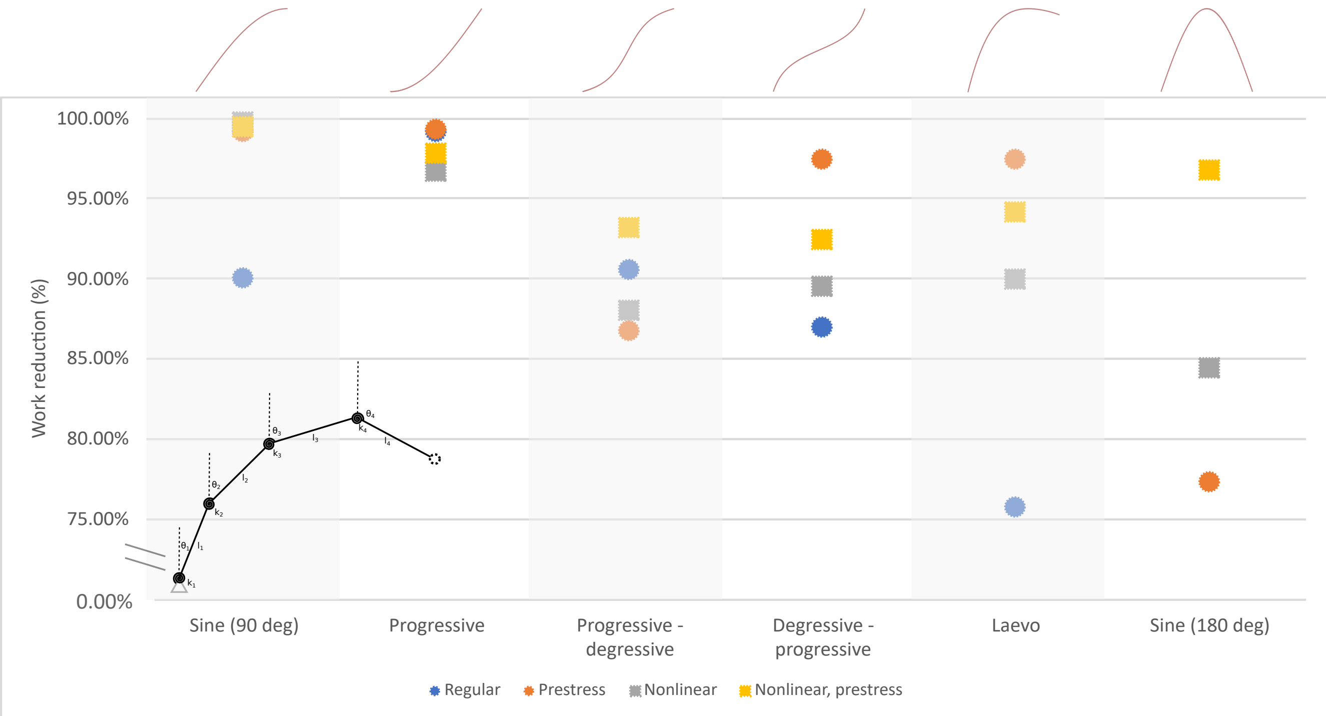


Figure 6: Modeling results four segment balancer

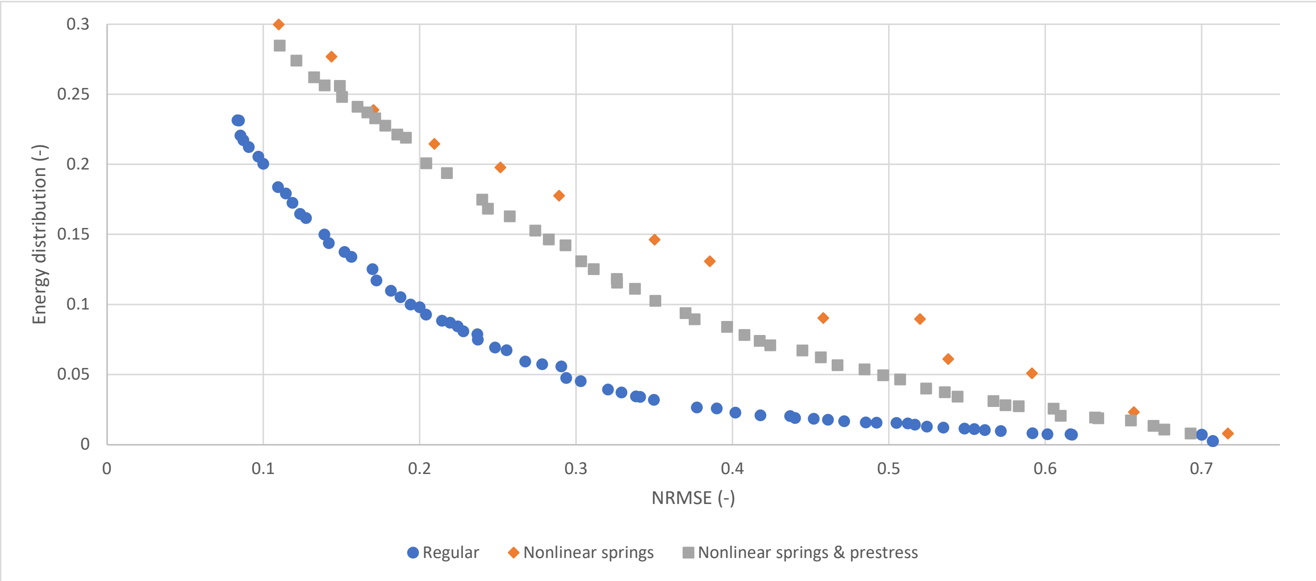


Figure 7: Pareto sets three segment balancer

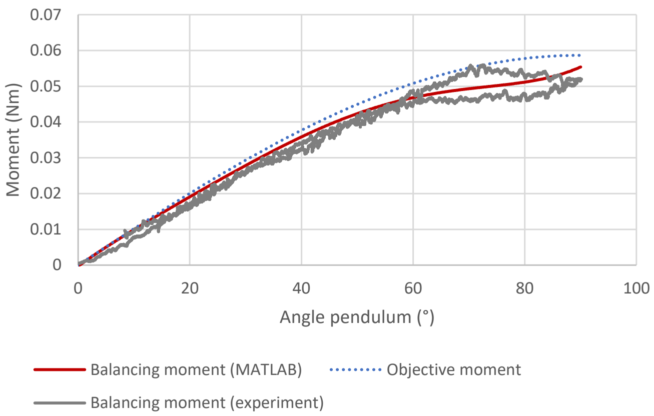


Figure 8: Measurement results experimental setup

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 optimization, 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 equilibrator 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 equilibrator 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 be 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 DOF, 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.

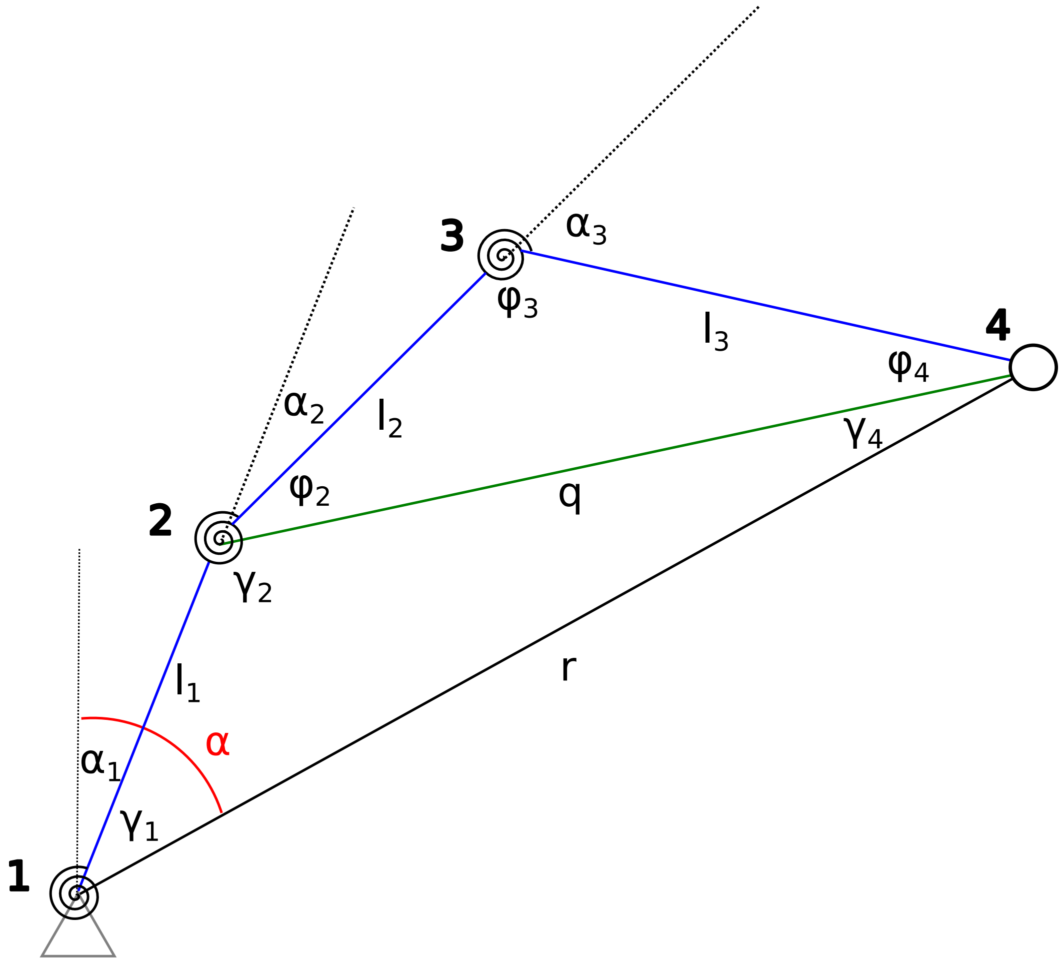


Figure A.1: Definition of angles

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

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

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

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

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

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

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

B

FBD three segment balancer

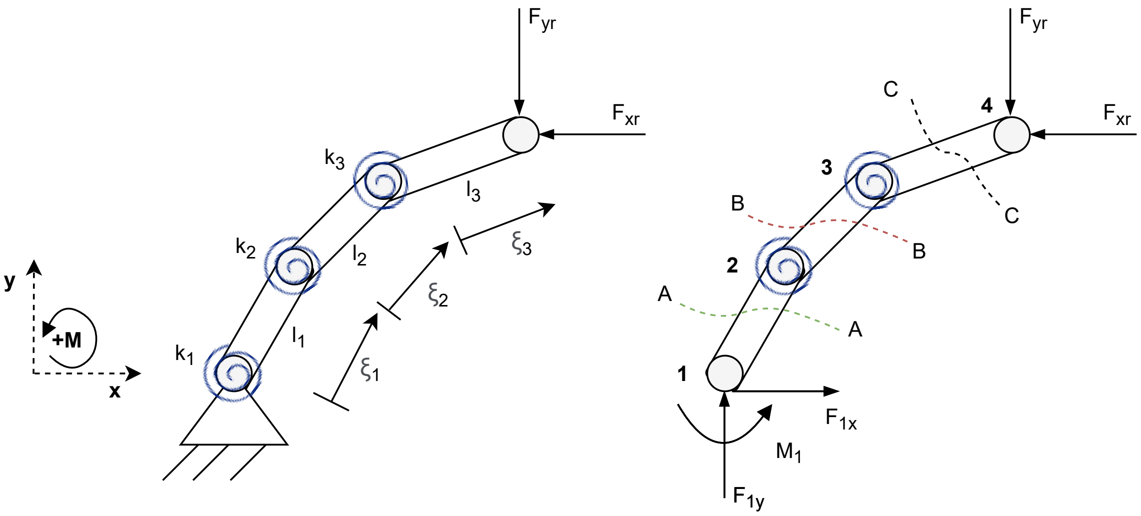


Figure B.1: Schematic of compensation mechanism (left) and setup for Free Body Diagrams (right)

B.1. Segment 1

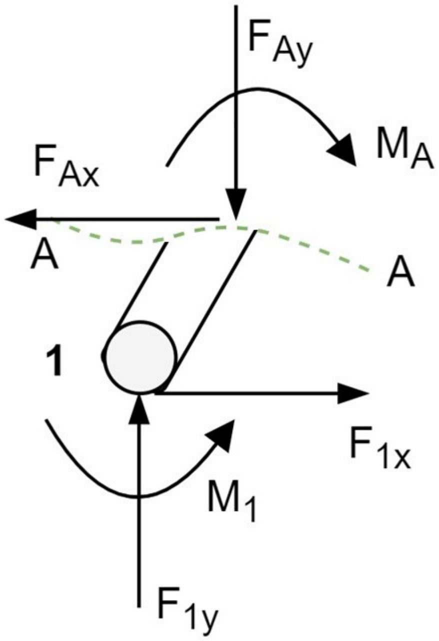


Figure B.2: FBD first segment

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

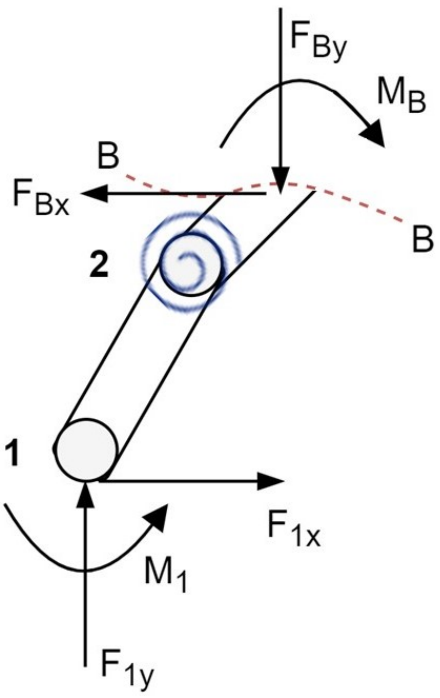


Figure B.3: FBD second segment

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

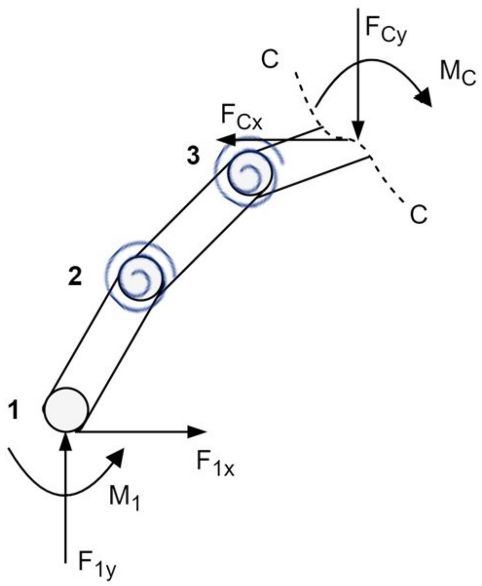


Figure B.4: FBD third segment

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

C

FBD four segment balancer

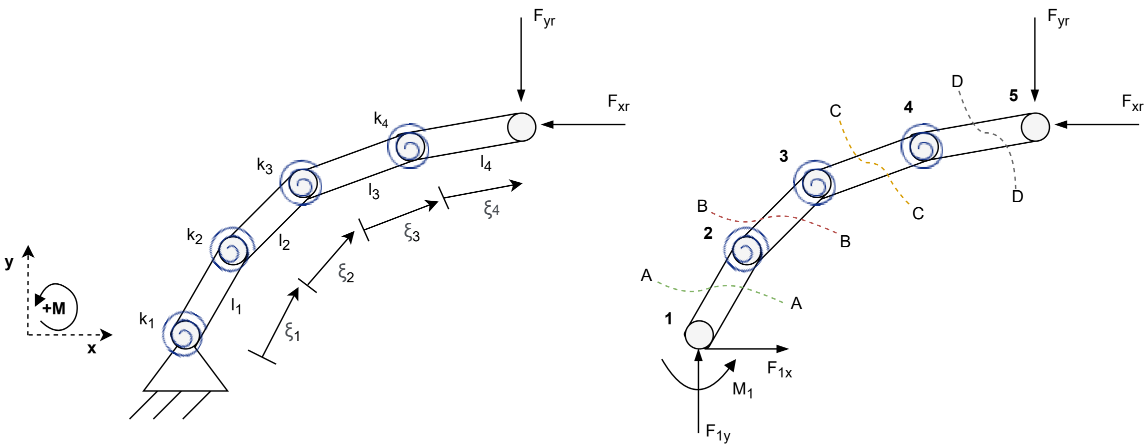


Figure C.1: Schematic of compensation mechanism (left) and setup for Free Body Diagrams (right)

C.1. Segment 4

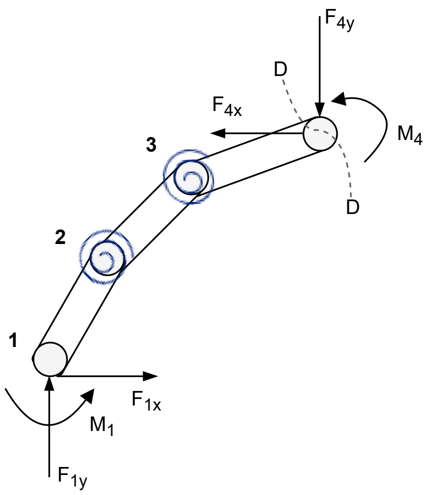


Figure C.2: FBD segment 4

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:

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

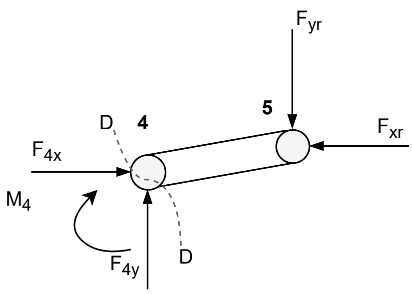


Figure C.3: FBD segment 4

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

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

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

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

After expanding this equation:

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

After rewriting this equation:

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

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

D

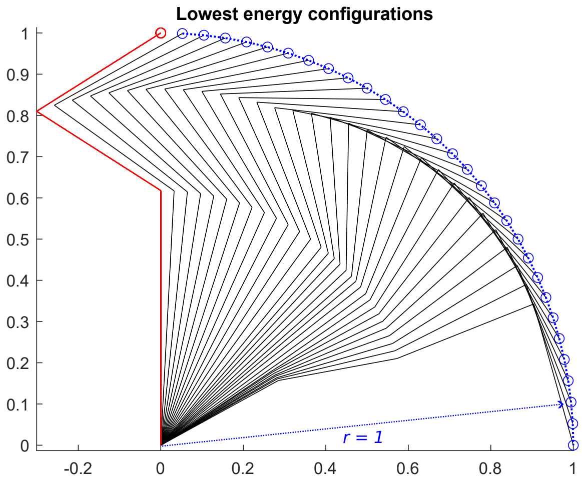
Release of contact

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 $r = 1$m.

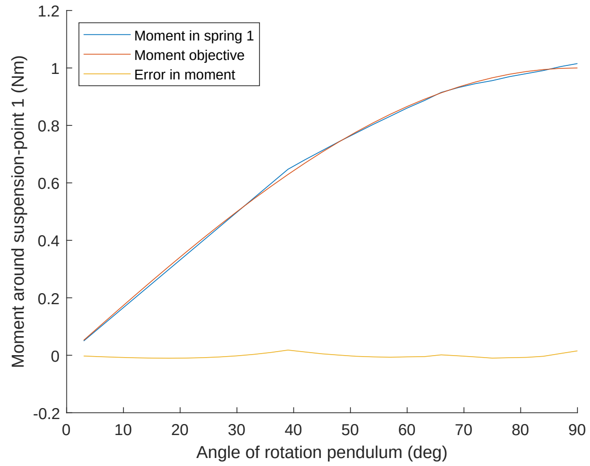
Parameter	Value	Unit
k_1	0.95	Nm/rad
k_2	0.21	Nm/rad
k_3	0.14	Nm/rad
k_4	0.06	Nm/rad
M_{3_0}	0.25	Nm
M_{2_0}	0.61	Nm
l_1	0.33	m
l_2	0.29	m
l_3	0.36	m
l_4	0.36	m

Table D.1: Optimization minimizers

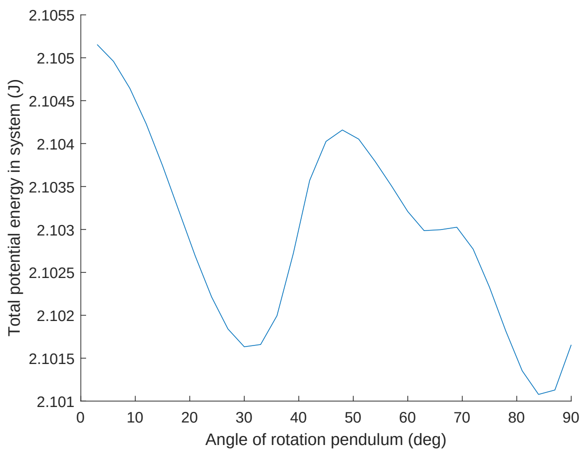
Figure D.1 depicts the four plots that are made with the obtained optimization results of the balancer. The kinematics of the balancer are visualized in figure D.1a. The inverted pendulum itself is omitted from this figure. The red posture is the relaxed initial configuration, whereas the blue circles 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 M3m, the internal load of the third spring, is initially constant and intersects M3l, 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.



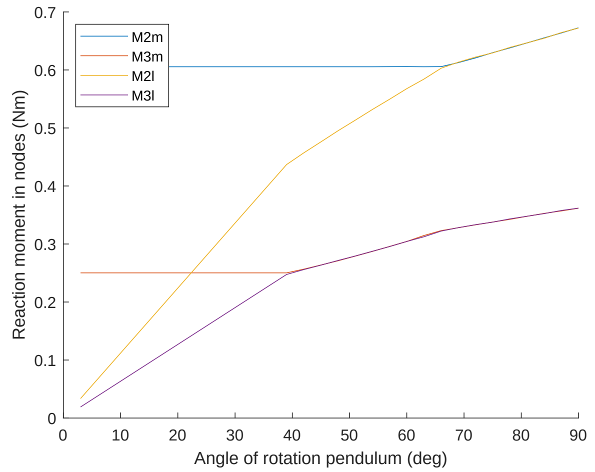
(a) Balancer kinematics



(b) Moment-angle characteristic of the balancer



(c) Total potential energy of the system



(d) Internal and external loads spring 2 and 3

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

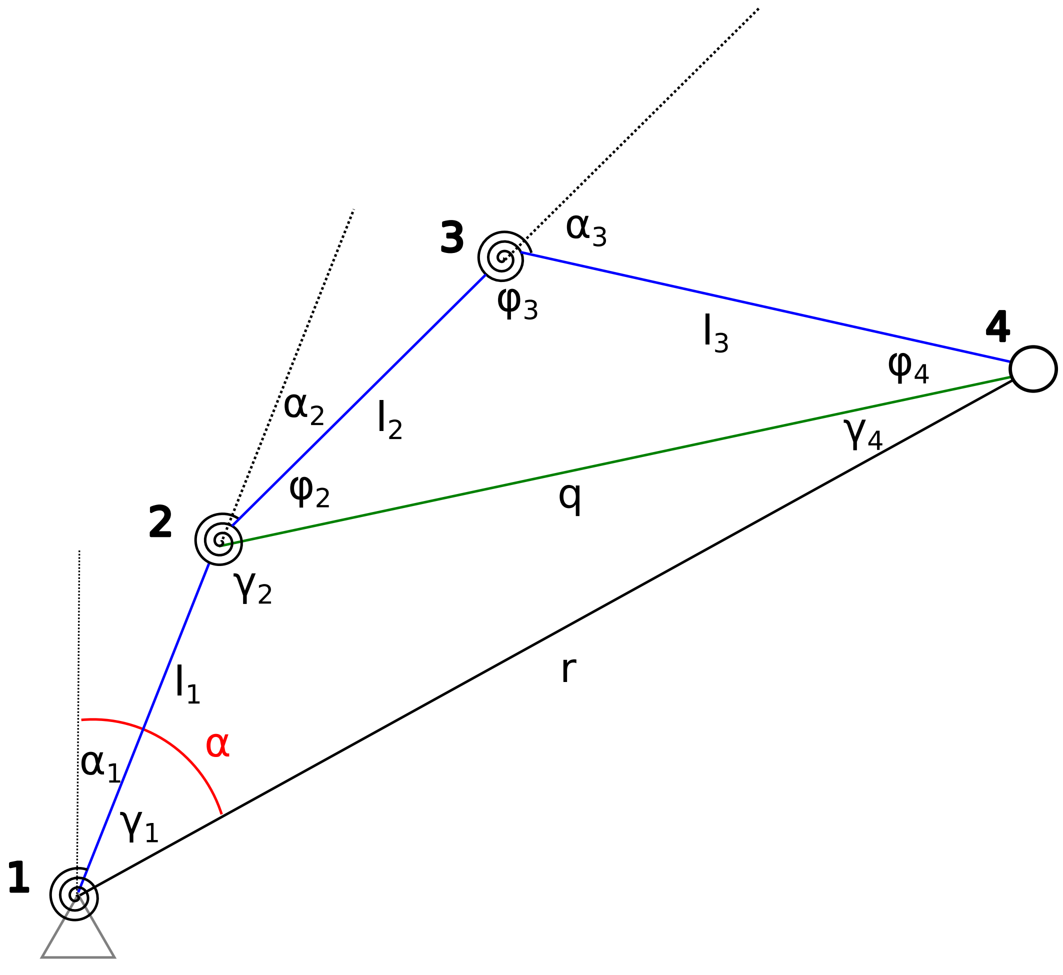


Figure E.1: Definition of angles

Requirements Lagrange: generalized coordinates should be [24]:

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

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

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.

$$\frac{\partial V}{\partial q_j} = 0 \tag{E.2}$$

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

$$\frac{\partial V}{\partial q_j} = 0 \begin{cases} \frac{\partial V}{\partial \alpha} = 0 \\ \frac{\partial V}{\partial \alpha_1} = 0 \end{cases} \tag{E.3}$$

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

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

$$\alpha_1 = \theta_1 - \theta_{1_0} \tag{E.5}$$

$$\alpha_2 = \pi - \gamma_2 - \phi_2 - \left(\theta_{2_0} - \theta_{1_0}\right) \tag{E.6}$$

$$\alpha_3 = \pi - \phi_3 - \left(\theta_{3_0} - \theta_{2_0}\right) \tag{E.7}$$

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

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

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

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

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.

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

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

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

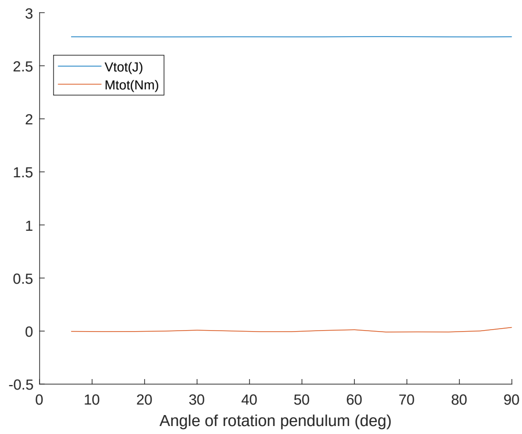
F

SAM

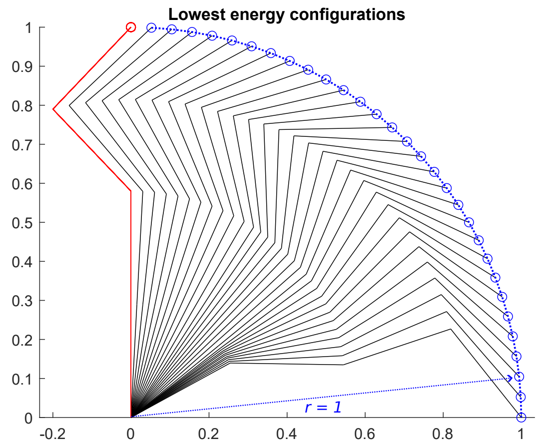
Early in the MATLAB modeling phase, the results obtained by the written MATLAB scripts were verified with use of Artas SAM software [21]. The results correspond to a four segment balancer with segment lengths equal to 29% of the length of the pendulum. The fourth spring is omitted and therefore has a 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 F.1: MATLAB analysis four segment balancer with prestress on springs 2 and 3

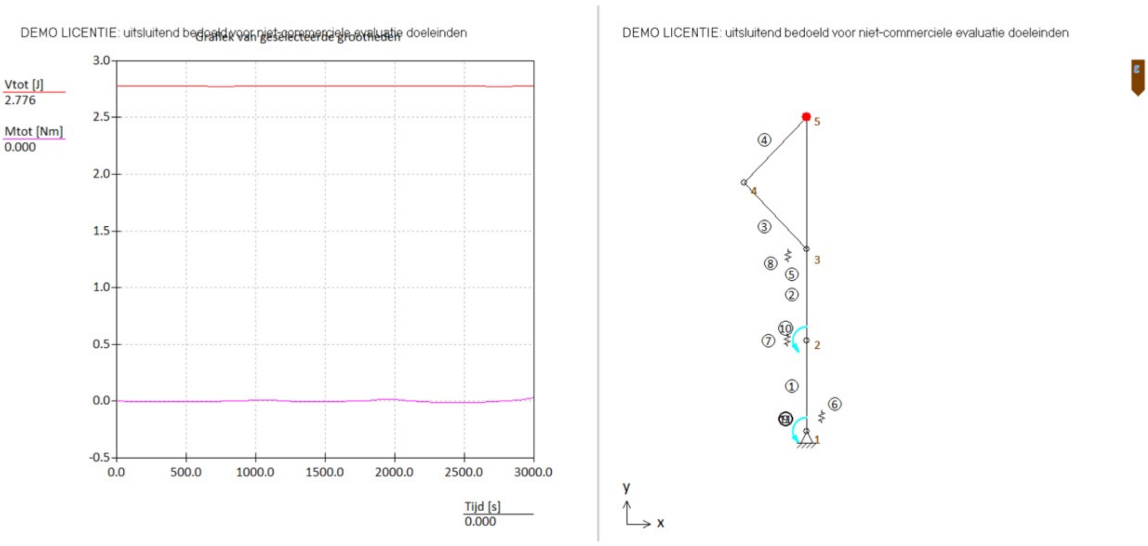


Figure F.2: SAM analysis four segment balancer with prestress on springs 2 and 3

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.

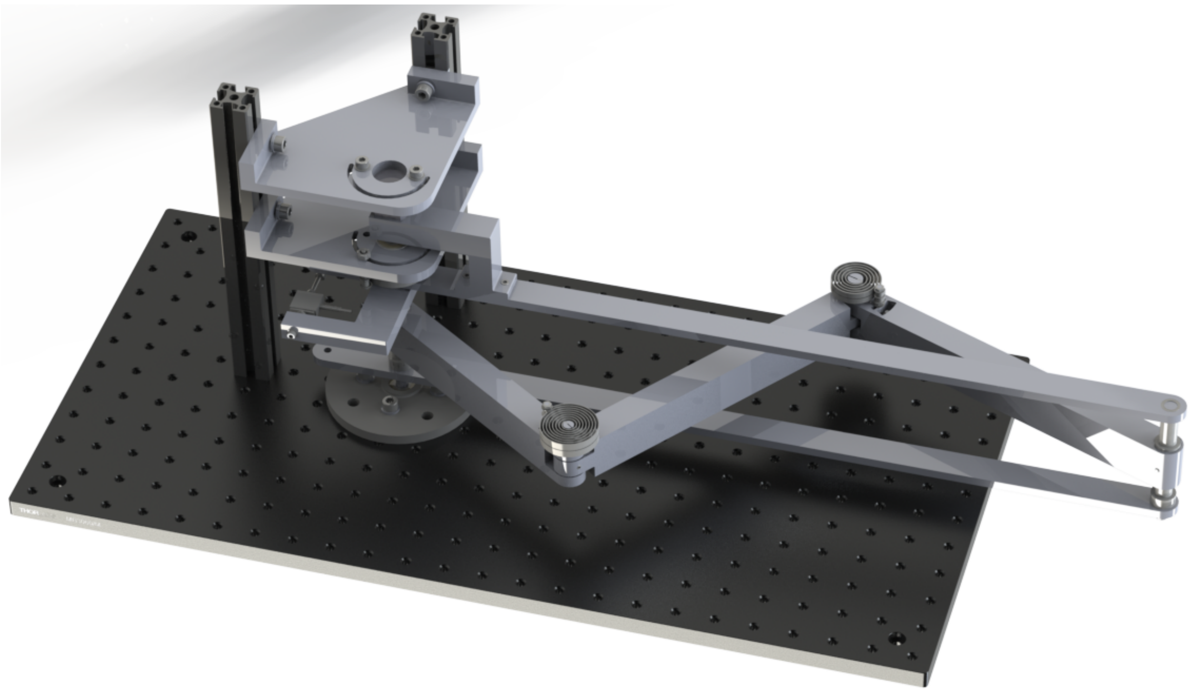


Figure G.1: Overview SW model

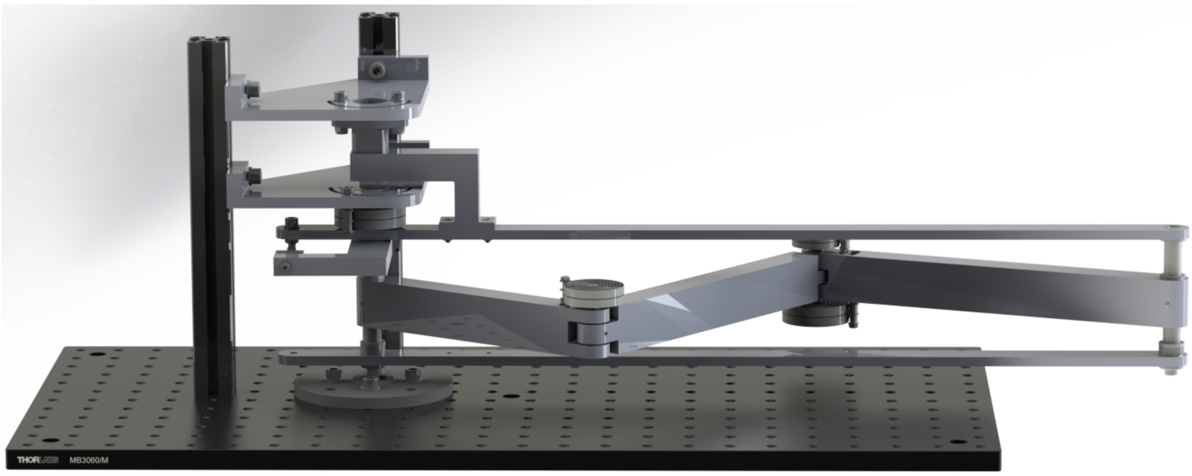


Figure G.2: Side view SW model

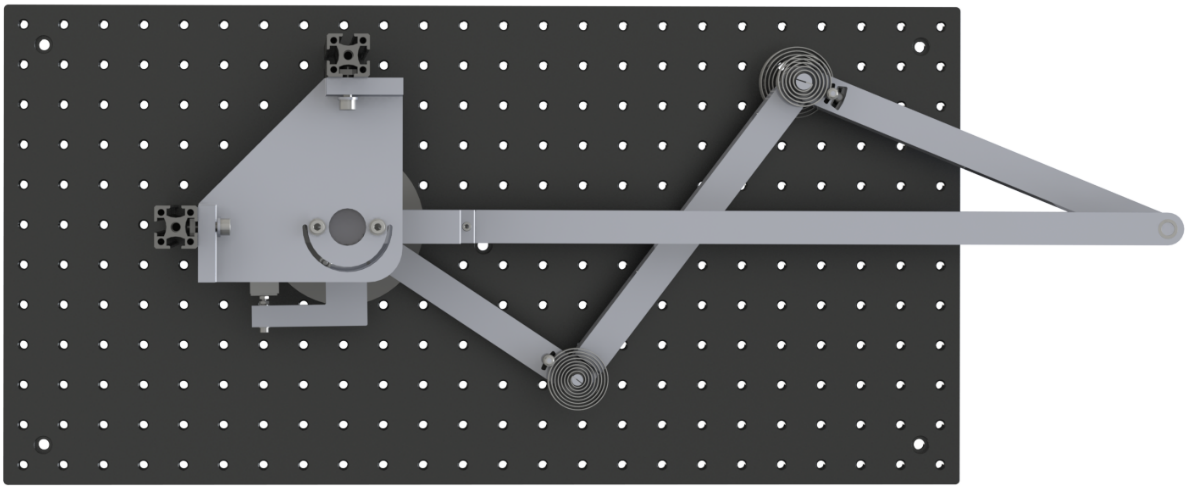


Figure G.3: Top view SW model

G.2. Solidworks model parts

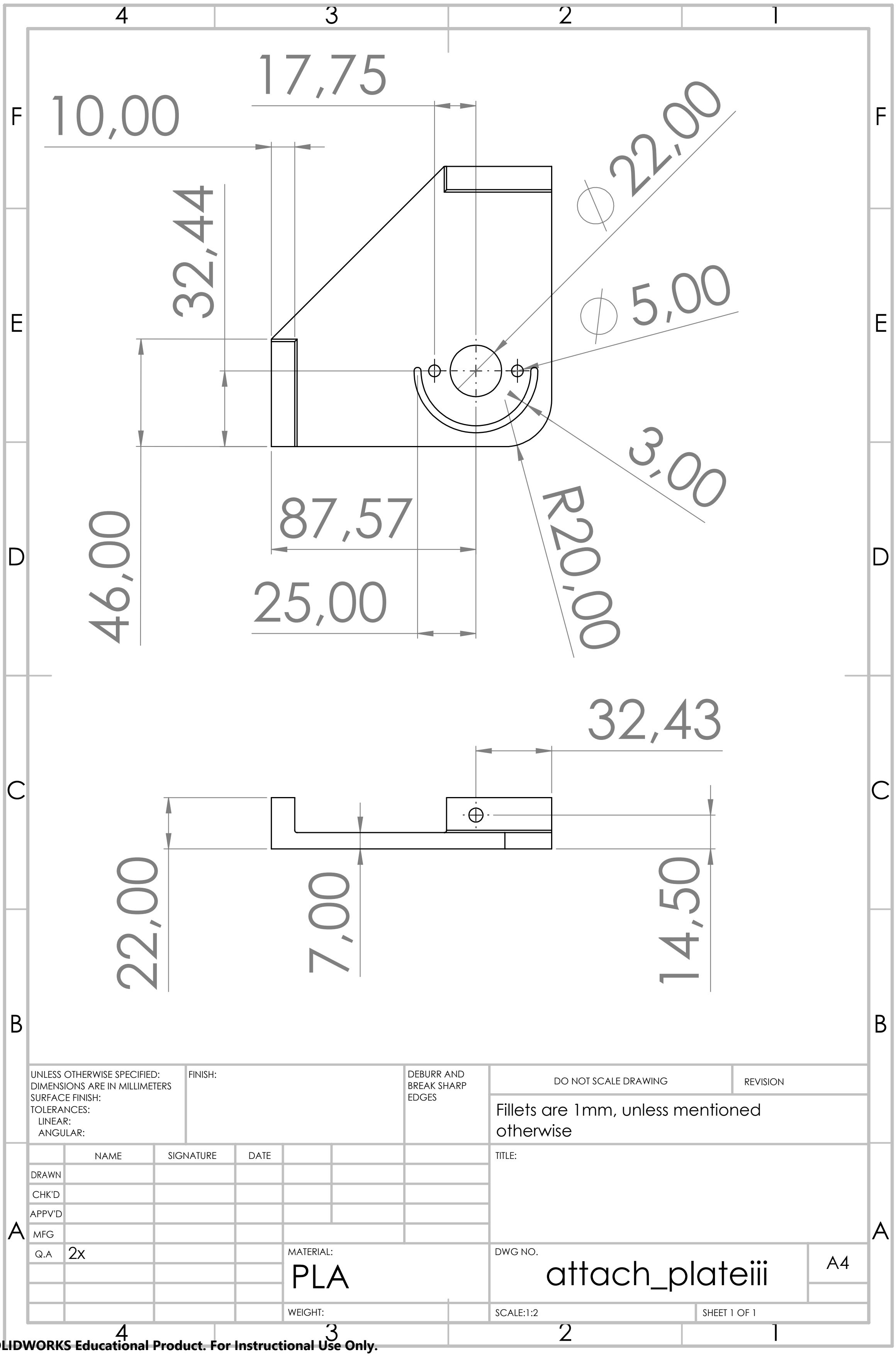
Off-the-shelf parts:

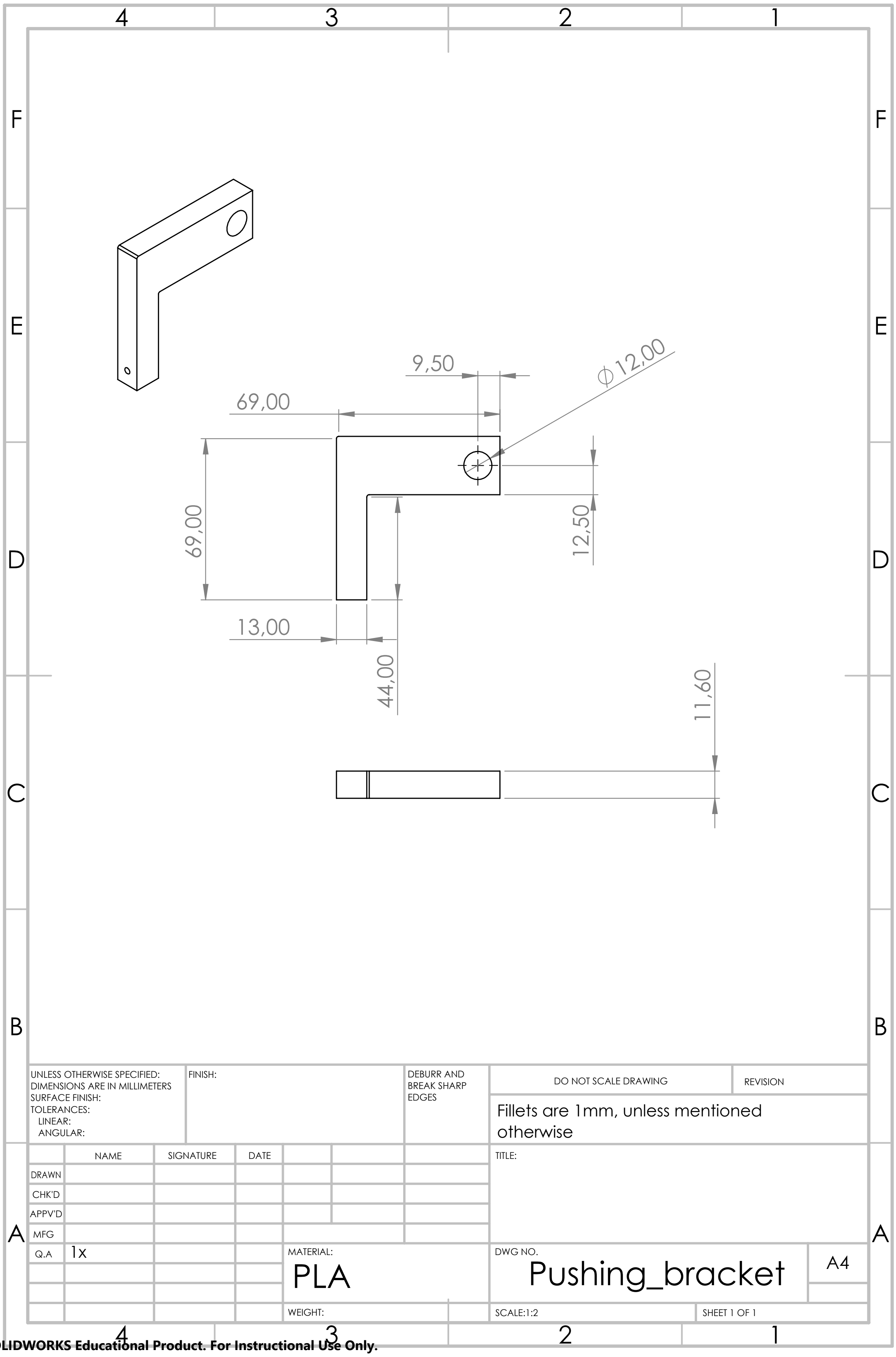
- 1x Thorlabs MB3060/M [25]
- 2x Thorlabs XE25L225/M construction rail [26]
- 1x Cherry AN8 angle position sensor [27]
- 1x FUTEK LSB200 FSH00102 load cell [28]
- 4x 907 Lesjöfors clock spring [29]
- 2x 903 Lesjöfors clock spring [30]
- 2x 908 Lesjöfors clock spring [31]
- 2x RS PRO 8mm-22mm miniature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

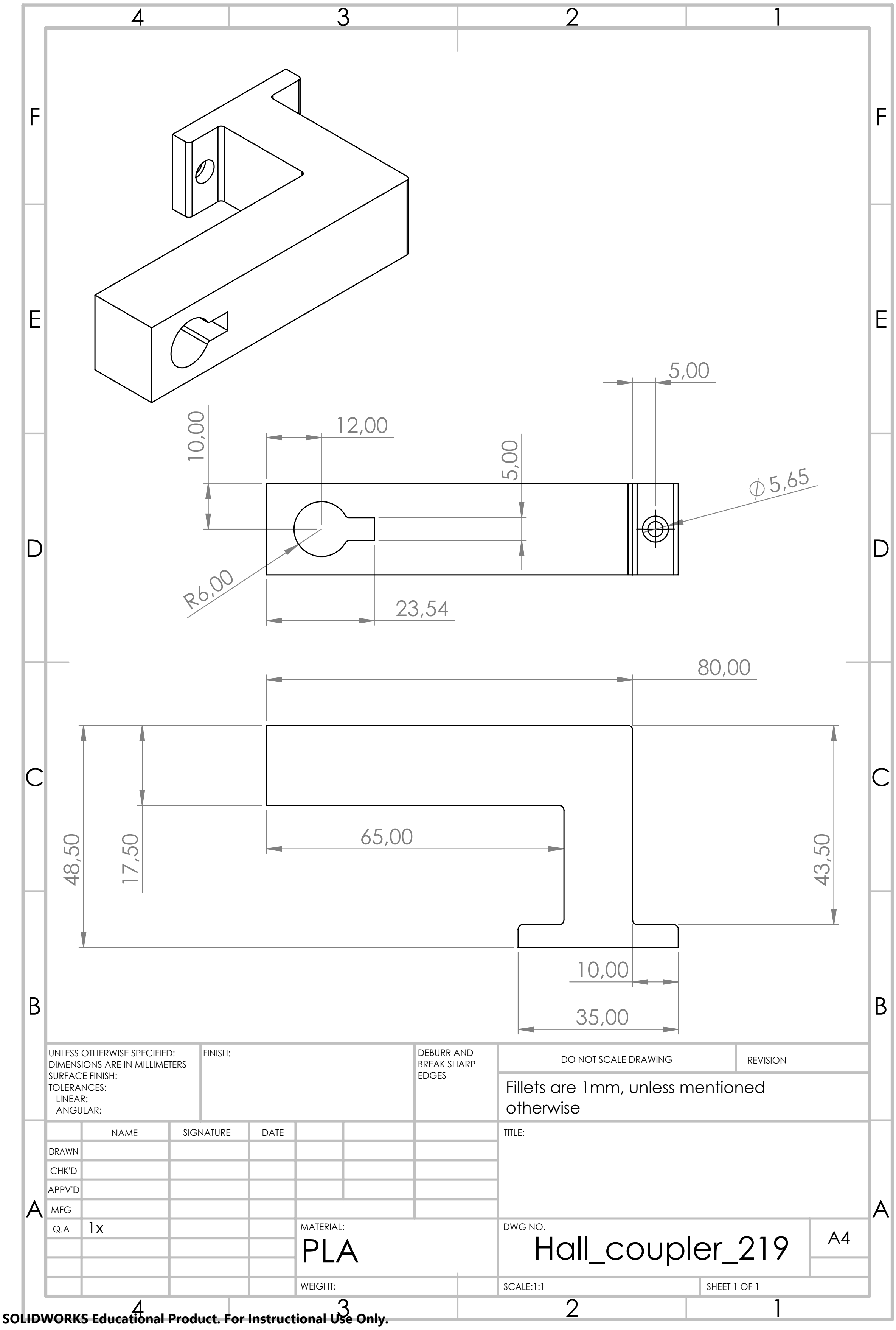
- 5x nylon 8mm bearing [35]
- 9x nylon M8 washer [36]
- 2x metal M8 washer [37]
- 8x metal M6 washer [38]
- 3x metal M5 washer [39]
- 20x metal M3 washer [40]
- 2x M8 Starlock [41]
- 17x M3 hexagon nut [42]
- 3x M5 hexagon nut [43]
- 8x M6 20mm cylinder head screw [44]
- 3x M5 20mm cylinder head screw [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:

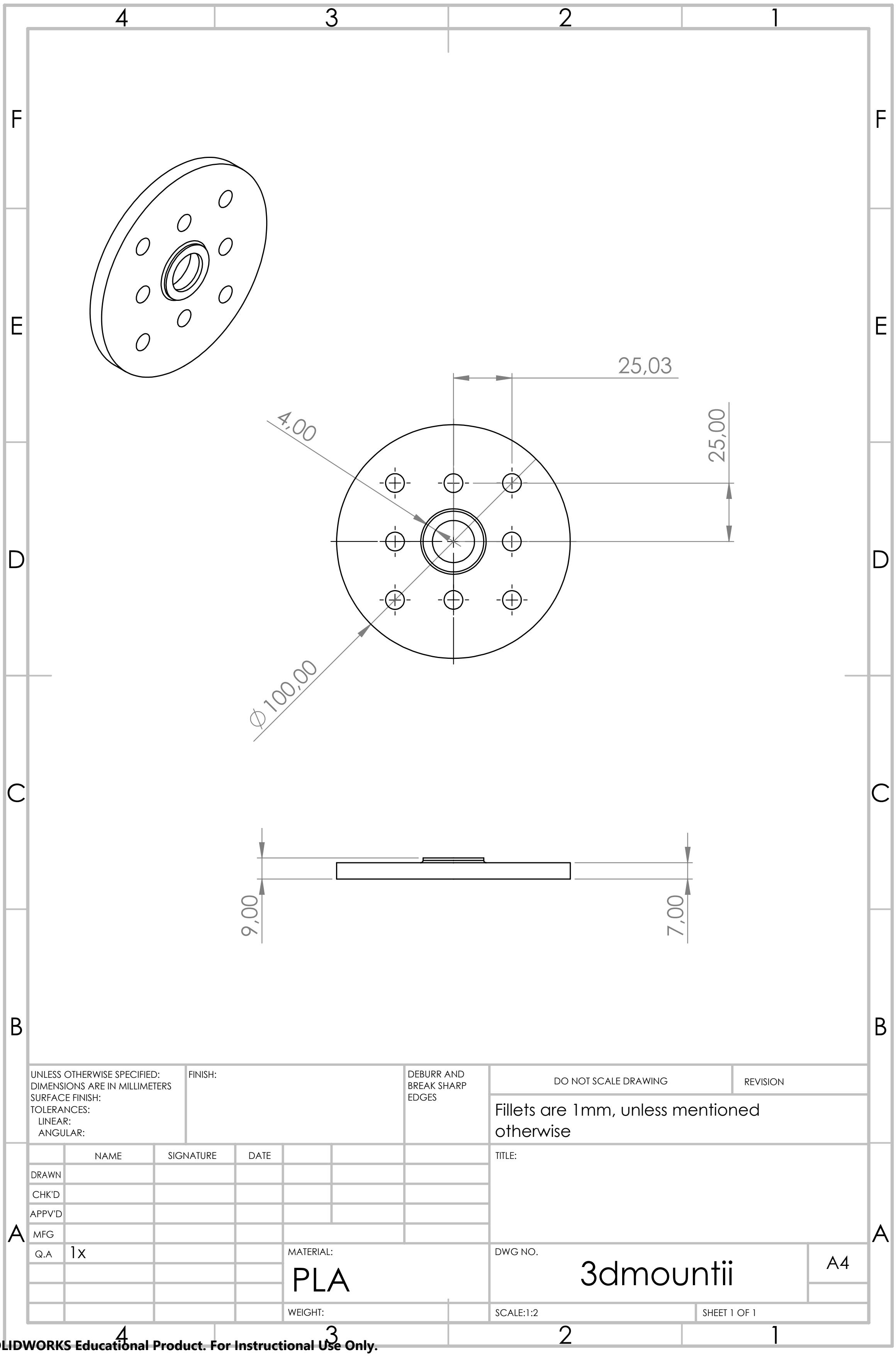
- 2x PLA “Attach plate”
- 1x PLA “Pushing bracket”
- 1x PLA “Hall coupler”
- 1x PLA “3dmount”
- 1x PLA “Segment1”
- 1x PLA “Segment2”
- 1x PLA “Segment3”
- 2x PMMA “Pendulum”
- 1x steel “Armaturerod1”
- 1x steel “Armaturerod2”
- 1x steel “Armaturerod3”
- 1x steel “Pendulumrod”

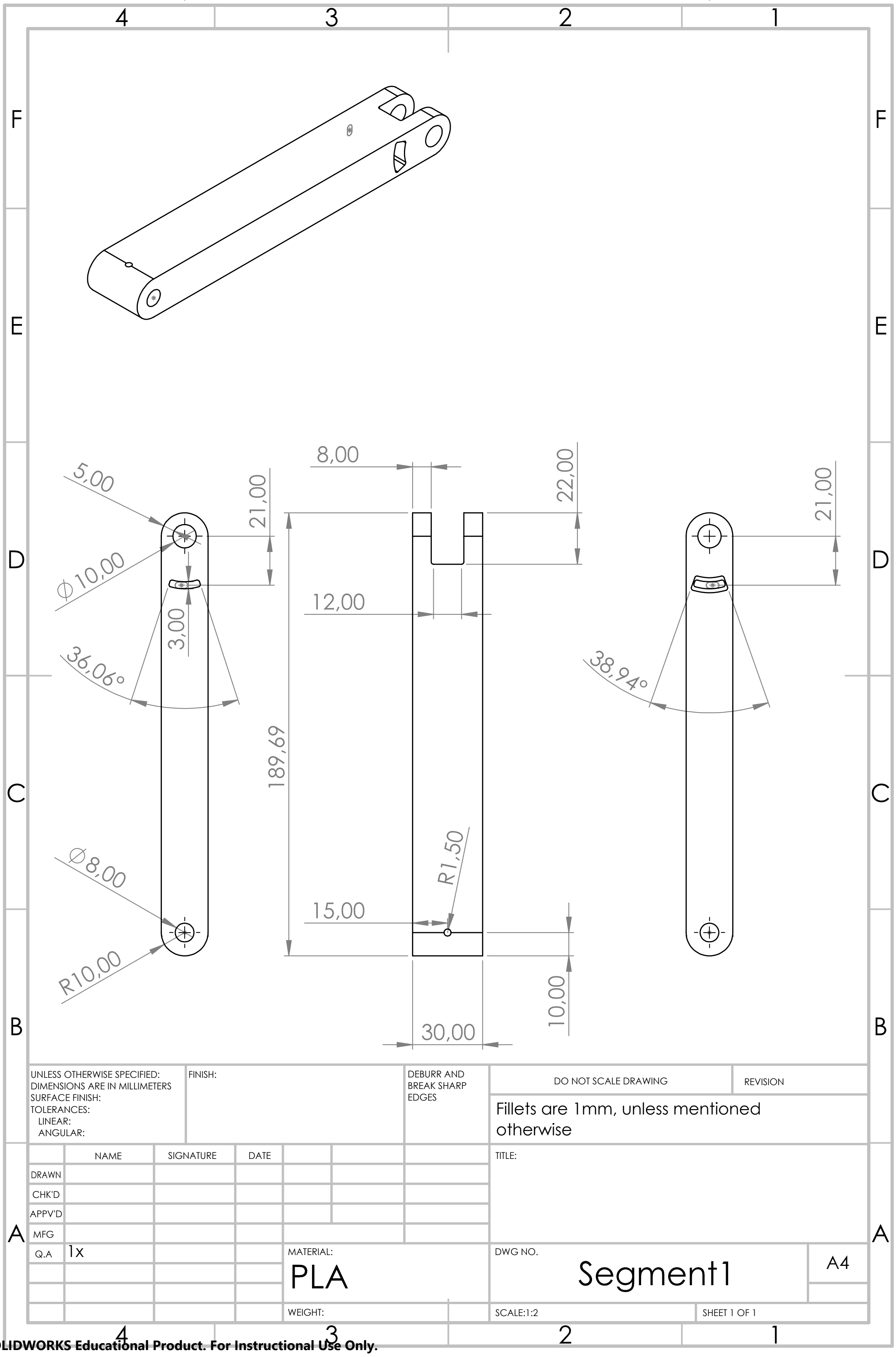

17,75
10,00
22,00
32,44
5,00
3,00
R20,00
87,57
46,00
25,00
32,43
22,00
7,00
14,50
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
2x
MATERIAL:
PLA
DWG NO.
attach_plateiii
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

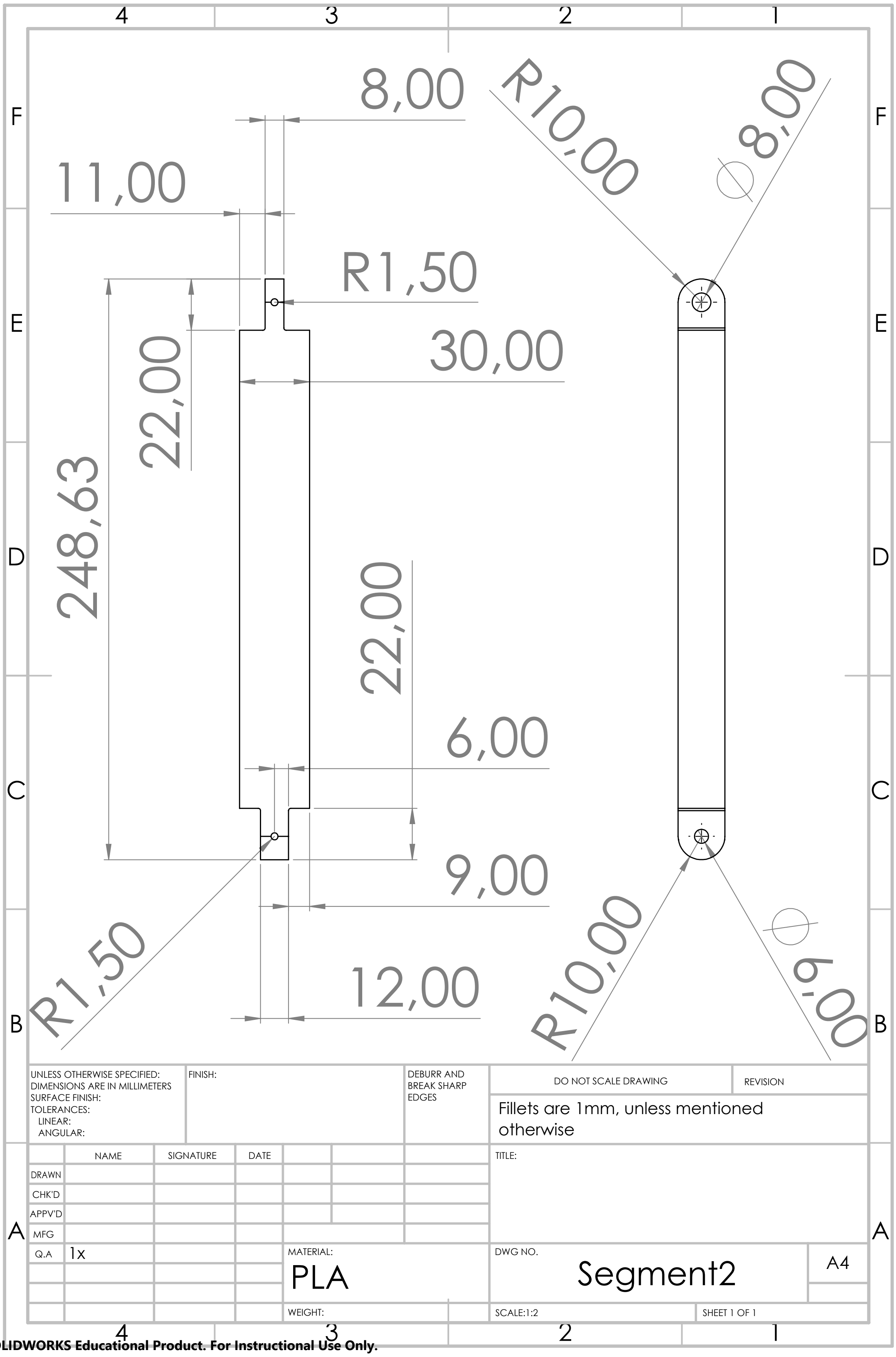

9,50
69,00
⌀12,00
69,00
12,50
13,00
44,00
11,60
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
TITLE:
MATERIAL:
PLA
DWG NO.
Pushing_bracket
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

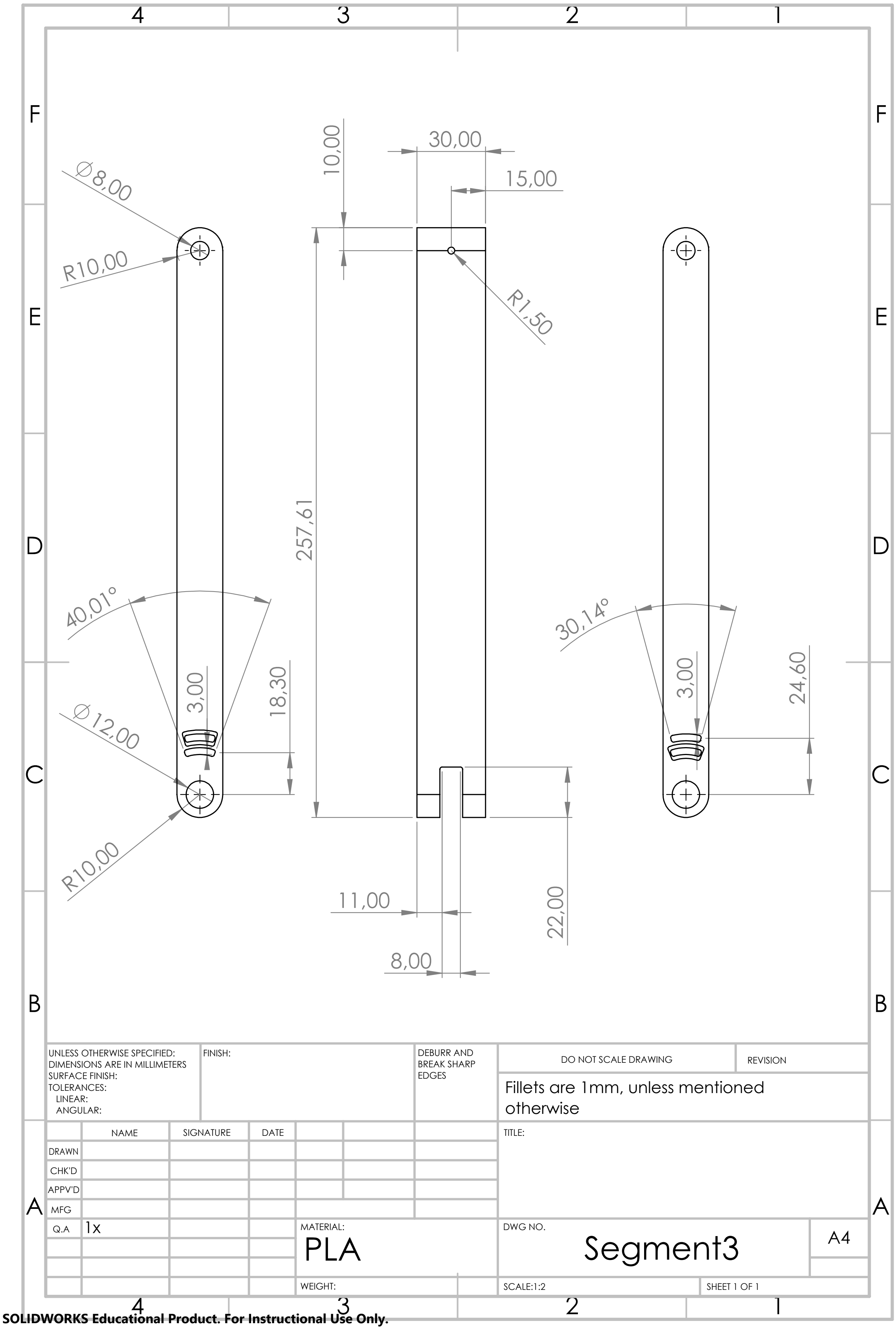


5,00
12,00
10,00
5,00
⌀5,65
R6,00
23,54
80,00
65,00
48,50
17,50
43,50
10,00
35,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
MATERIAL:
PLA
DWG NO.
Hall_coupler_219
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

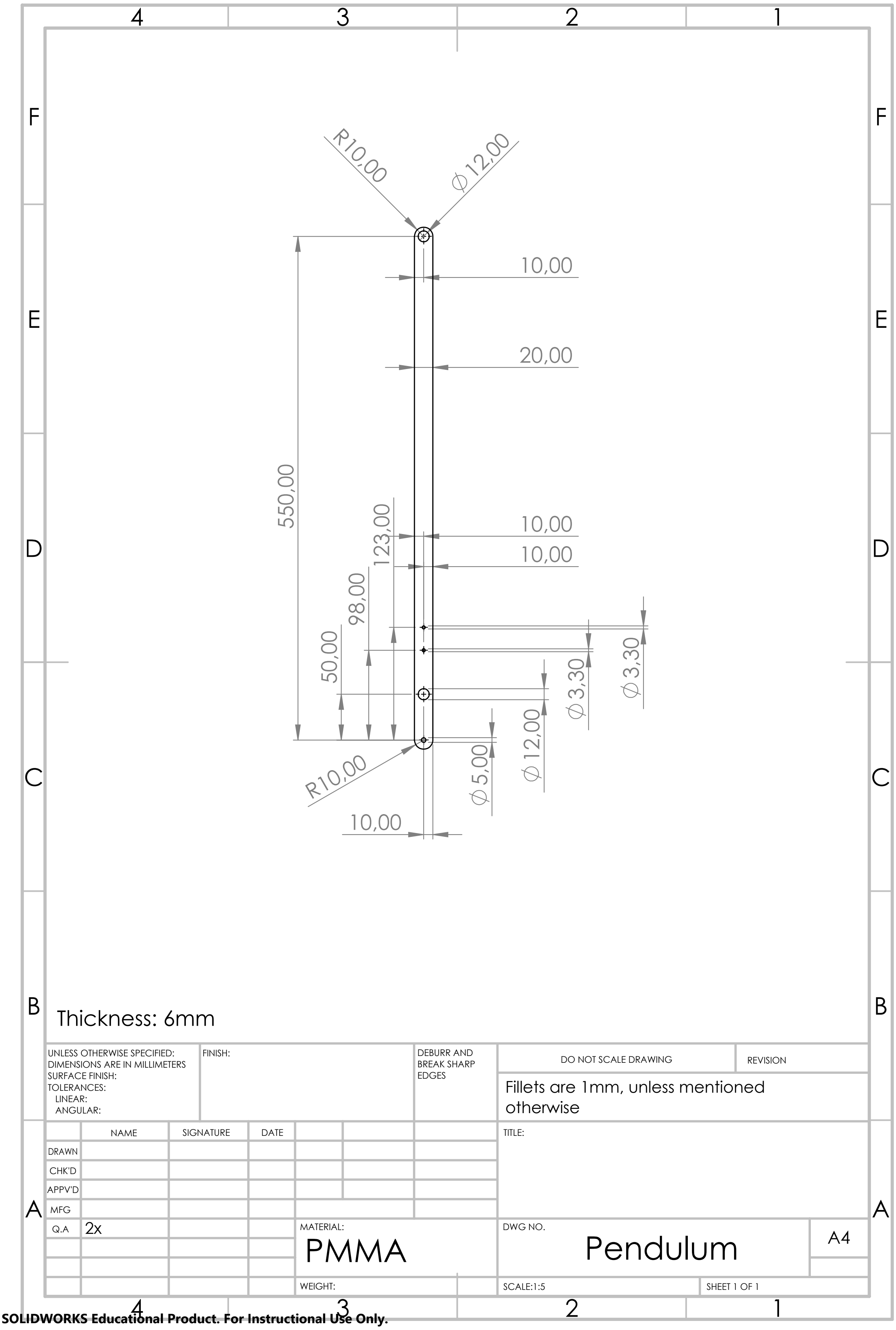

25,03
25,00
4,00
⌀100,00
9,00
7,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
MATERIAL:
PLA
DWG NO.
3dmountii
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

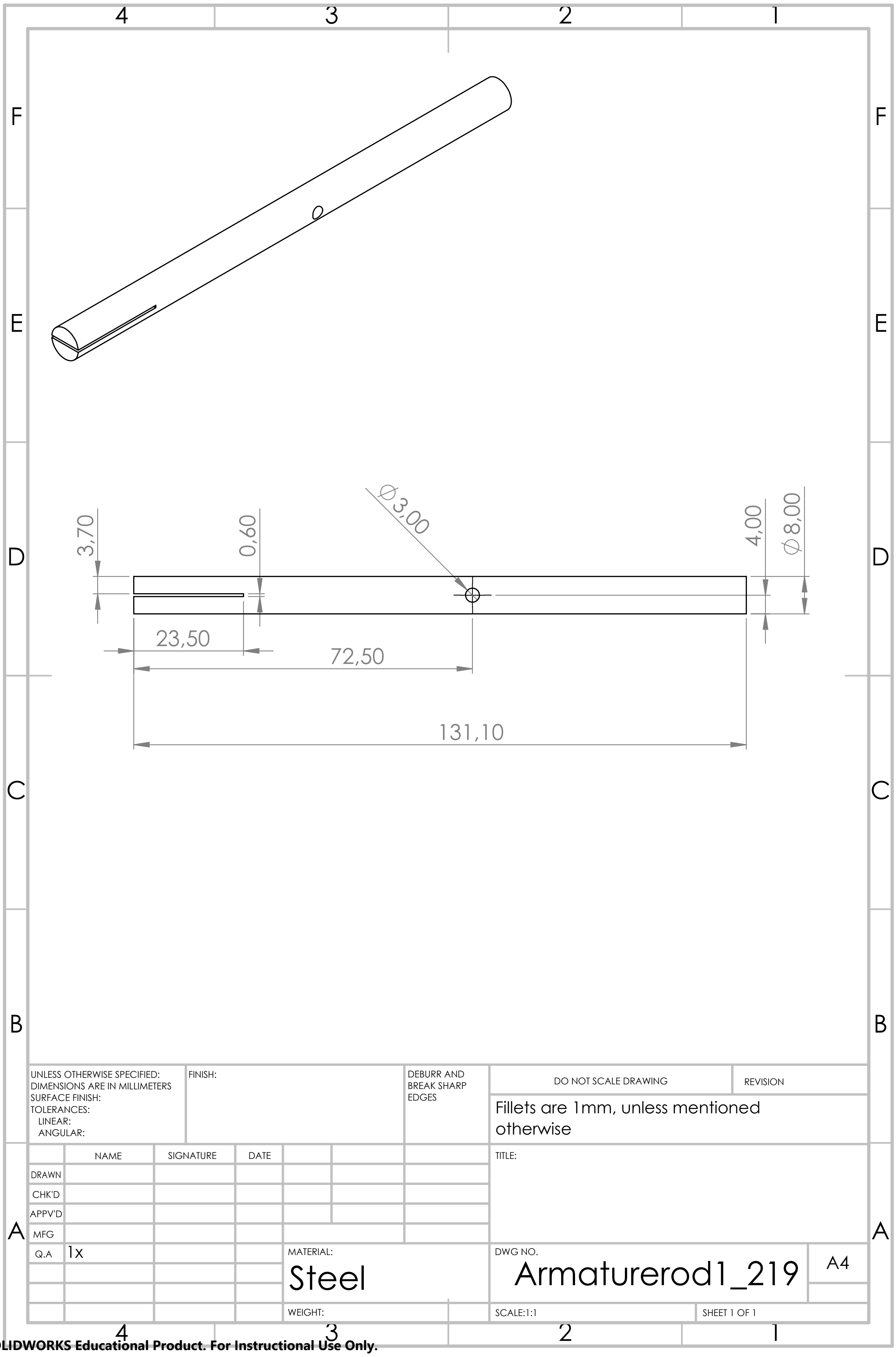

4
3
2
1
F
E
D
C
B
A
5,00
⌀10,00
21,00
3,00
36,06°
⌀8,00
R10,00
8,00
12,00
22,00
189,69
R1,50
15,00
10,00
30,00
21,00
38,94°
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
MATERIAL:
PLA
DWG NO.
Segment1
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

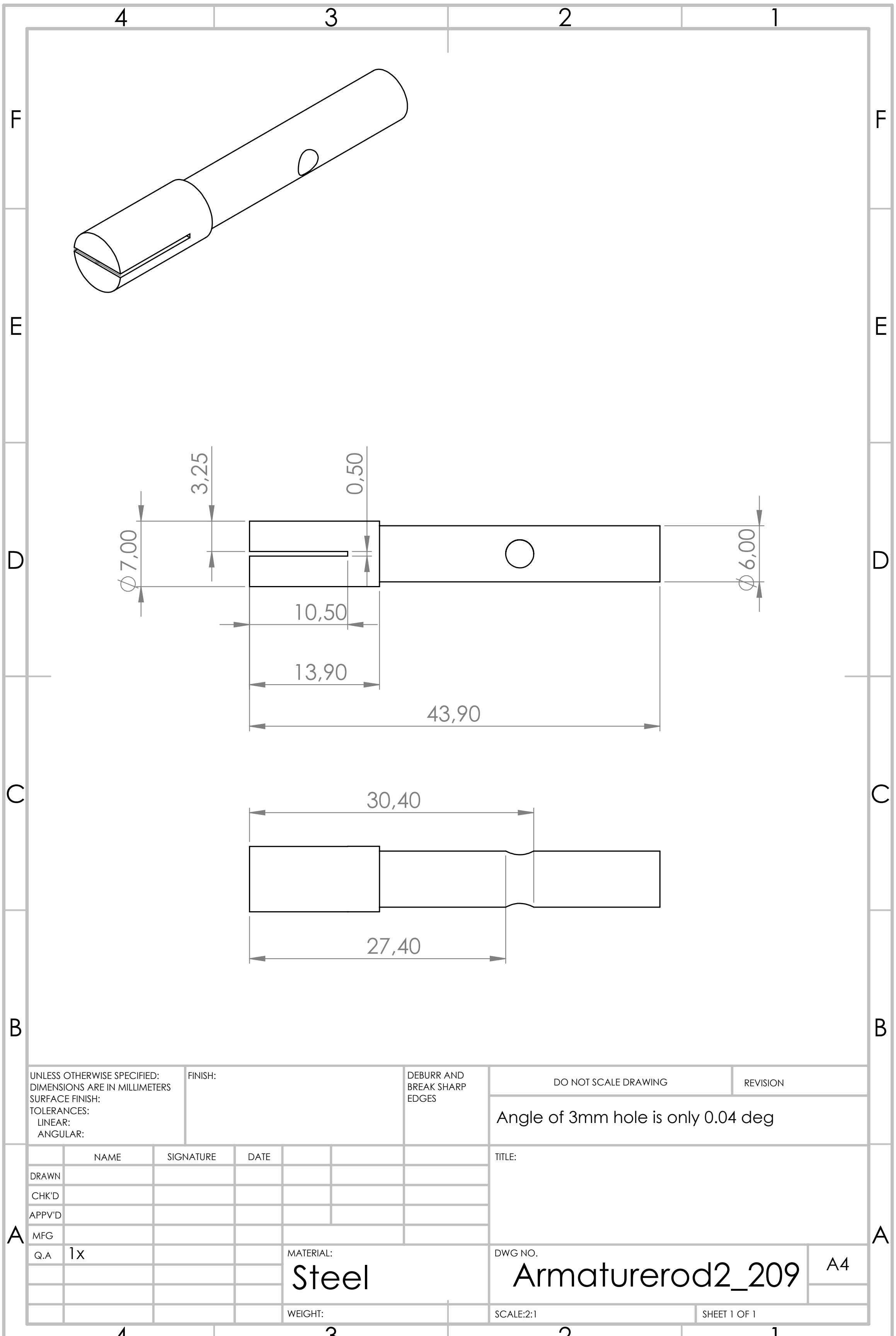

8,00
11,00
R1,50
30,00
22,00
248,63
22,00
6,00
9,00
12,00
R1,50
R10,00
⌀8,00
R10,00
⌀6,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
MATERIAL:
PLA
DWG NO.
Segment2
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

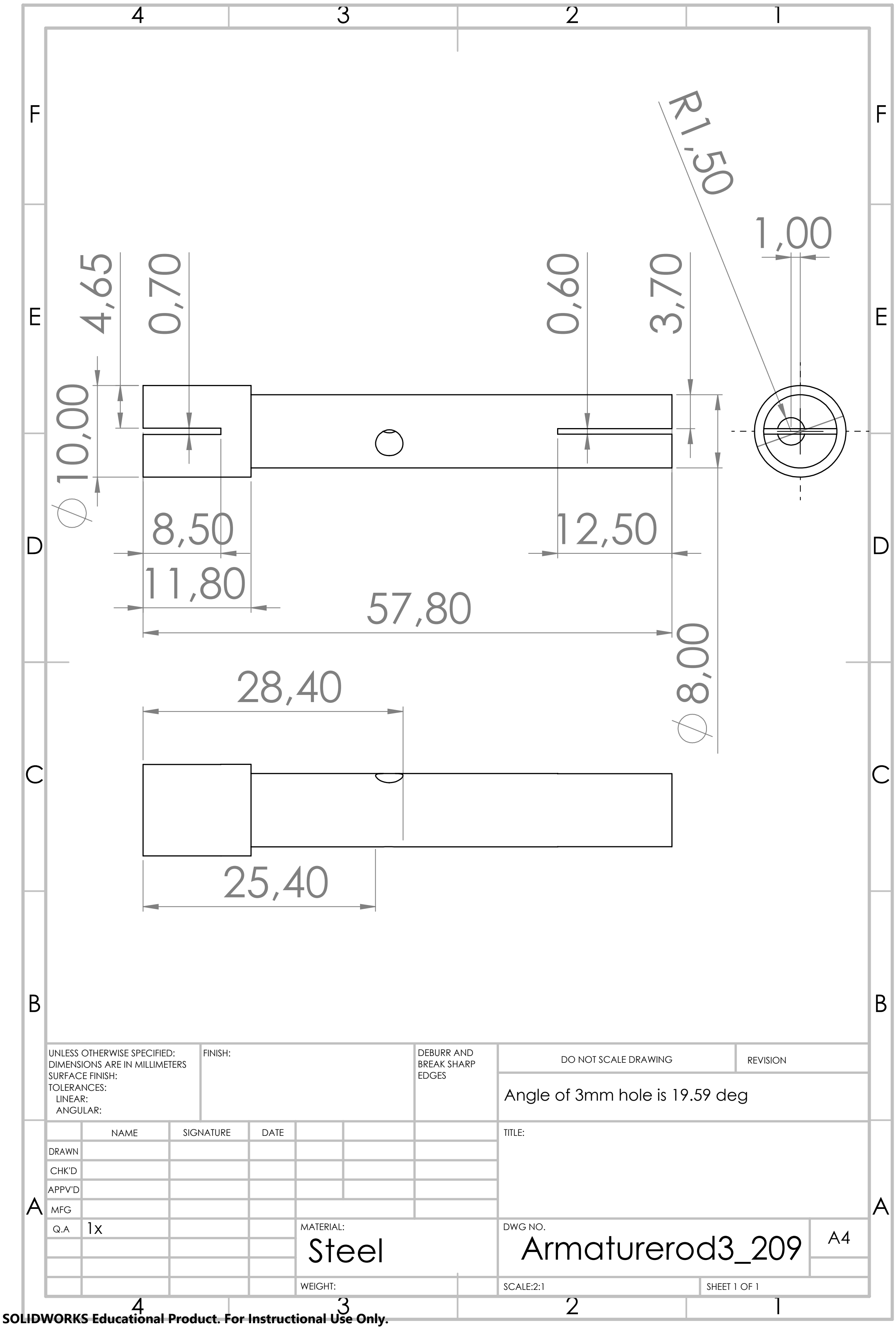


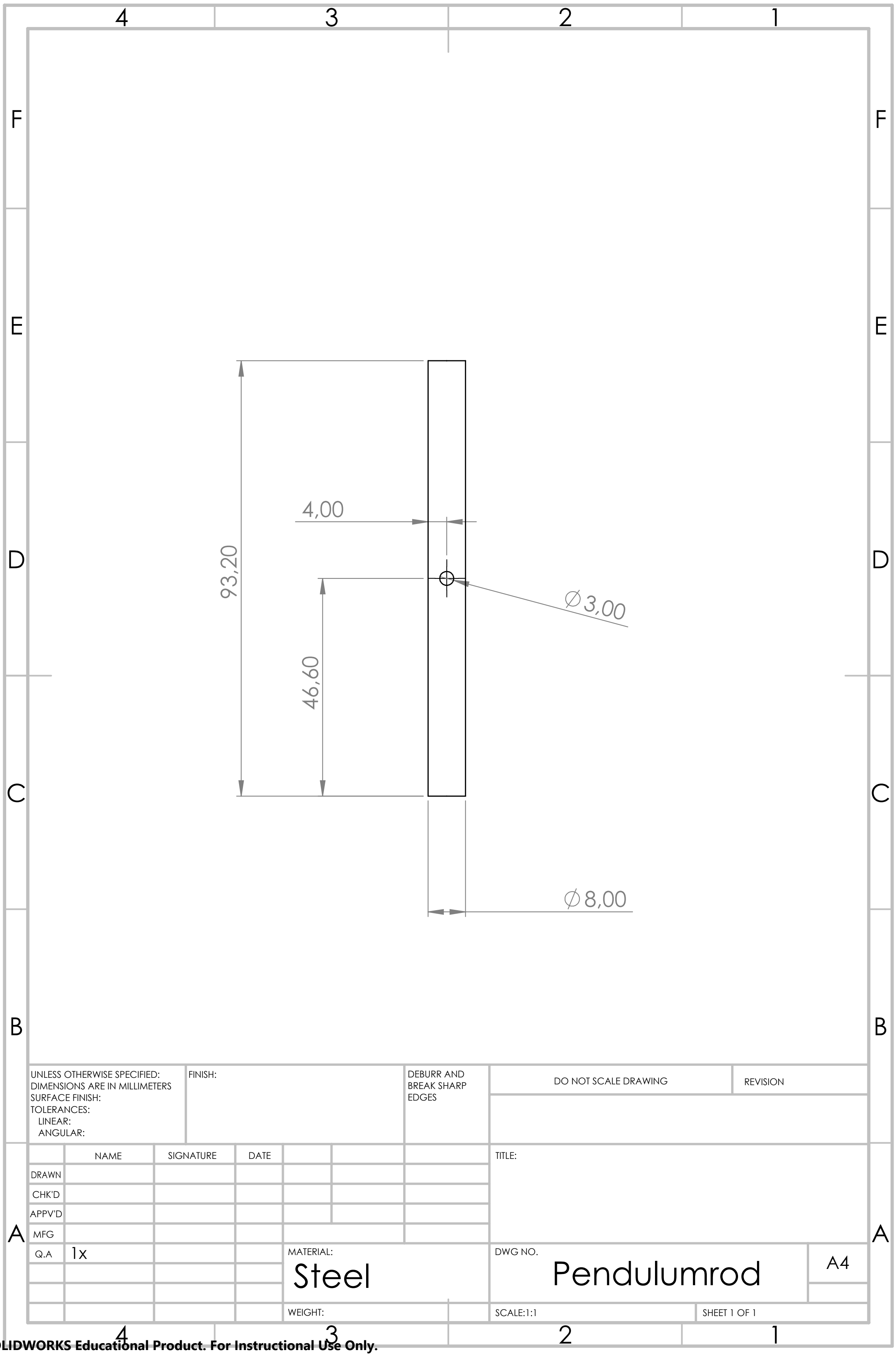
∅8,00
R10,00
10,00
30,00
15,00
R1,50
257,61
40,01°
3,00
18,30
∅12,00
R10,00
11,00
8,00
22,00
30,14°
3,00
24,60
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
Fillets are 1mm, unless mentioned otherwise
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
TITLE:
MATERIAL:
PLA
DWG NO.
Segment3
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


R10,00
⌀12,00
10,00
20,00
550,00
123,00
98,00
50,00
10,00
10,00
⌀3,30
⌀3,30
⌀12,00
⌀5,00
R10,00
10,00
Thickness: 6mm
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
APPV'D
MFG
Q.A
2x
MATERIAL:
PMMA
DWG NO.
Pendulum
A4
WEIGHT:
SCALE:1:5
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


Ø3,00
3,70
0,60
4,00
Ø8,00
23,50
72,50
131,10
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
APPV'D
MFG
Q.A
1x
MATERIAL:
Steel
DWG NO.
Armaturerod1_219
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1


3,25
0,50
Ø 7,00
Ø 6,00
10,50
13,90
43,90
30,40
27,40
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 only 0.04 deg
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
MATERIAL:
Steel
DWG NO.
Armaturerod2_209
A4
WEIGHT:
SCALE:2:1
SHEET 1 OF 1


R1,50
1,00
4,65
0,70
0,60
3,70
∅10,00
8,50
12,50
11,80
57,80
∅8,00
28,40
25,40
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
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
TITLE:
MATERIAL:
Steel
DWG NO.
Armaturerod3_209
A4
WEIGHT:
SCALE:2:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


4
3
2
1
F
E
D
C
B
A
4,00
93,20
46,60
∅3,00
∅8,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
1x
TITLE:
MATERIAL:
Steel
DWG NO.
Pendulumrod
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

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.

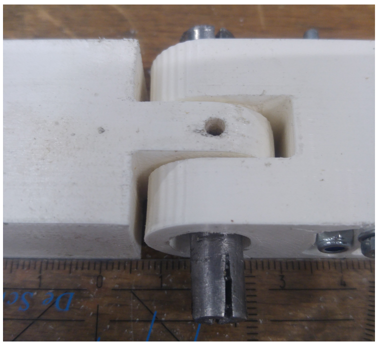
H.1. Installation axes

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.

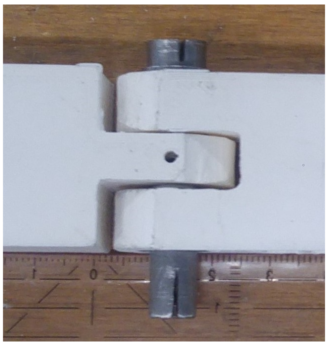
- Heat the PLA locally with use of a heat source
- Design for larger holes in the segments
- File the holes before inserting the axes
- Drill segment holes with correct diameter

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.

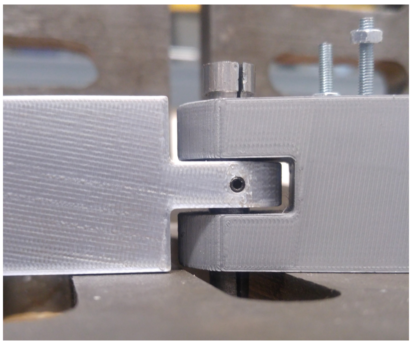
Two additional remarks should be made regarding the alignment of the segments and the axes. By inserting an axis between segment 1 and 2 or between segment 2 and 3, the fingers might deform. This is illustrated in figure H.1, where figure H.1a depicts the resulting misaligned axis. This deformation was 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.

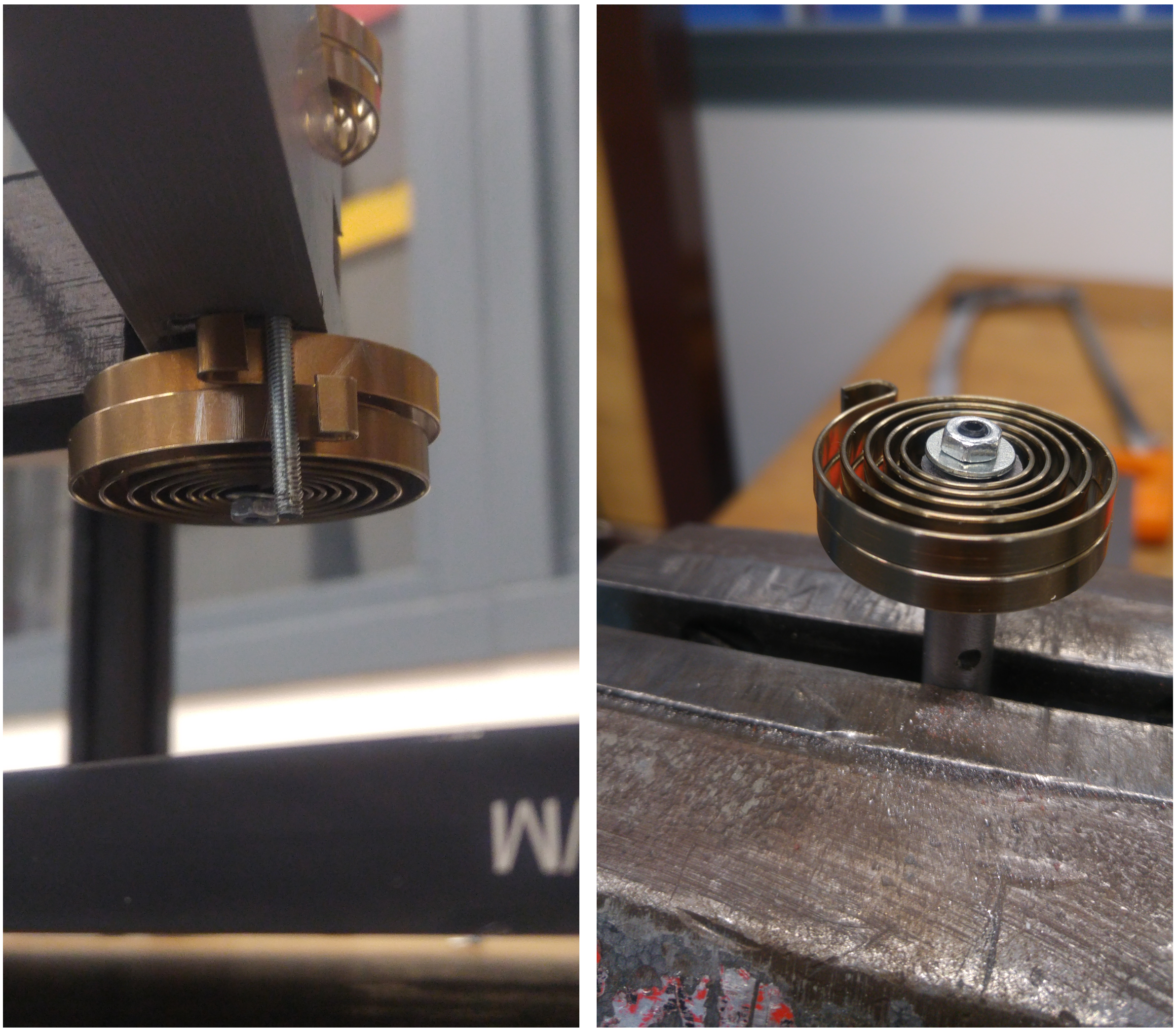
To realize a correct orientation of the main axis, it is recommended to not immediately tighten the bolts in the bottom connection part. This cylindrically shaped PLA part is seen in the left bottom corners of figure G.1 and figure G.2. Instead, it is advisable to first assemble the system without mounting 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

I

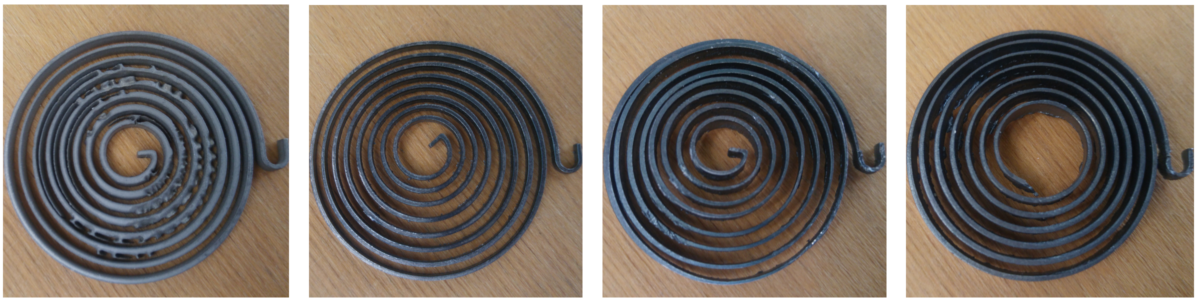
Spring design

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.

$$k_c = \frac{Ebt^3}{12L} \tag{I.1}$$

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.

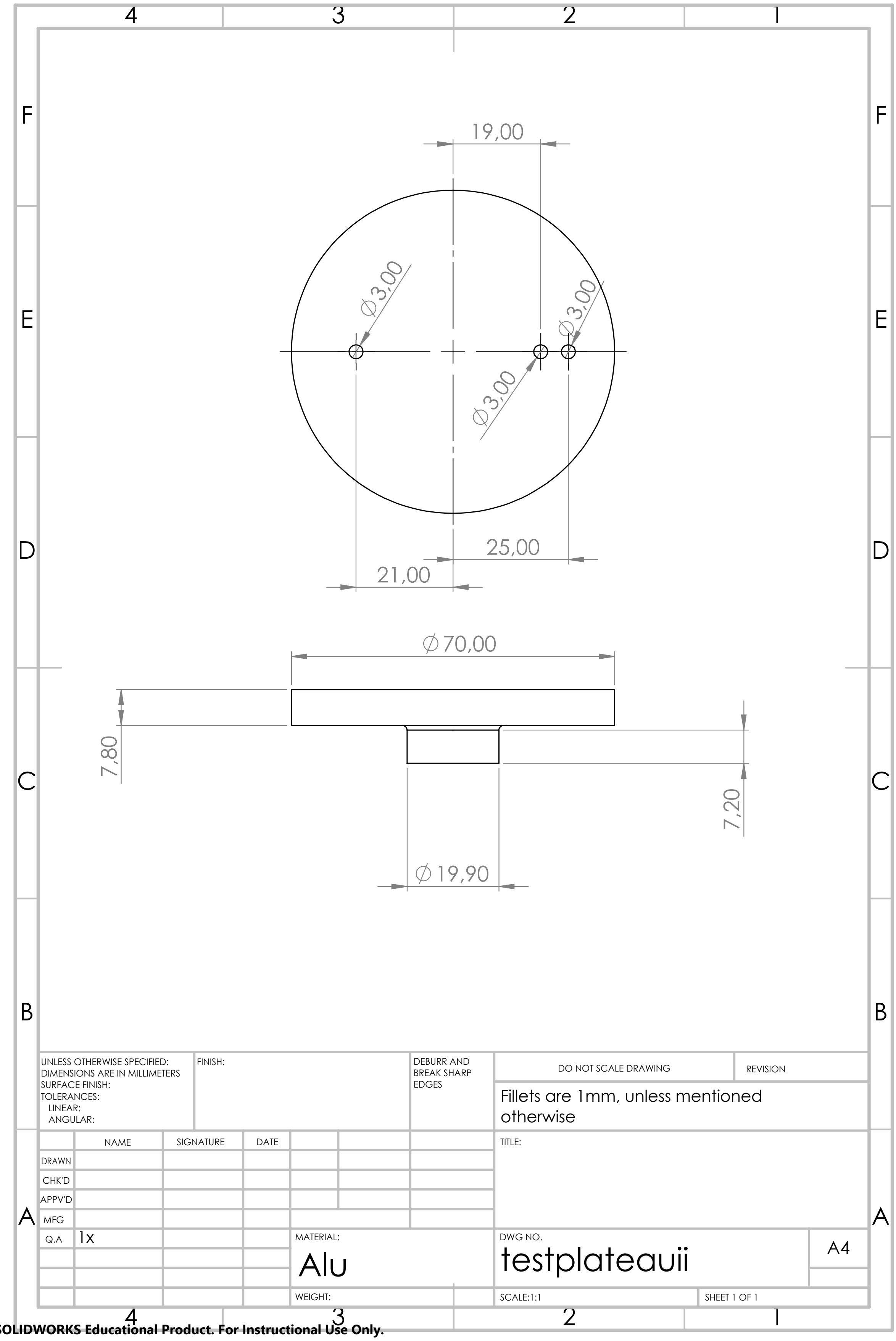
Spring	Stiffness (Nm/rad)	Amount of windings (-)	Thickness (mm)	Width (mm)	Inner radius (mm)	Outer radius (mm)
907	0.037	8	0.6	6.0	4.0	25
903	0.025	8	0.5	5.0	3.5	21
908	0.082	5	0.7	4.0	5.0	19

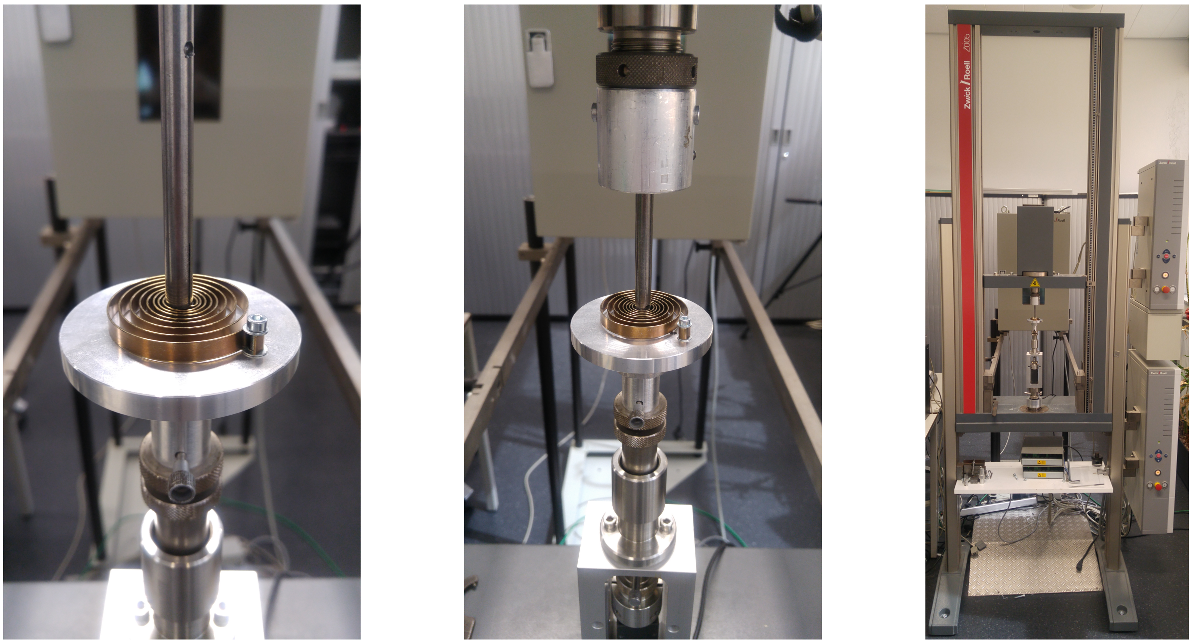
Table J.1: Properties distinct clock springs

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.


19,00
∅3,00
∅3,00
∅3,00
25,00
21,00
∅70,00
7,80
7,20
∅19,90
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
APPV'D
MFG
Q.A
1x
MATERIAL:
Alu
DWG NO.
testplateauii
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1



(a) Mounting plateau with clock spring

(b) Rotating head and fixed base

(c) Overview testing device

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.

K

LabVIEW code

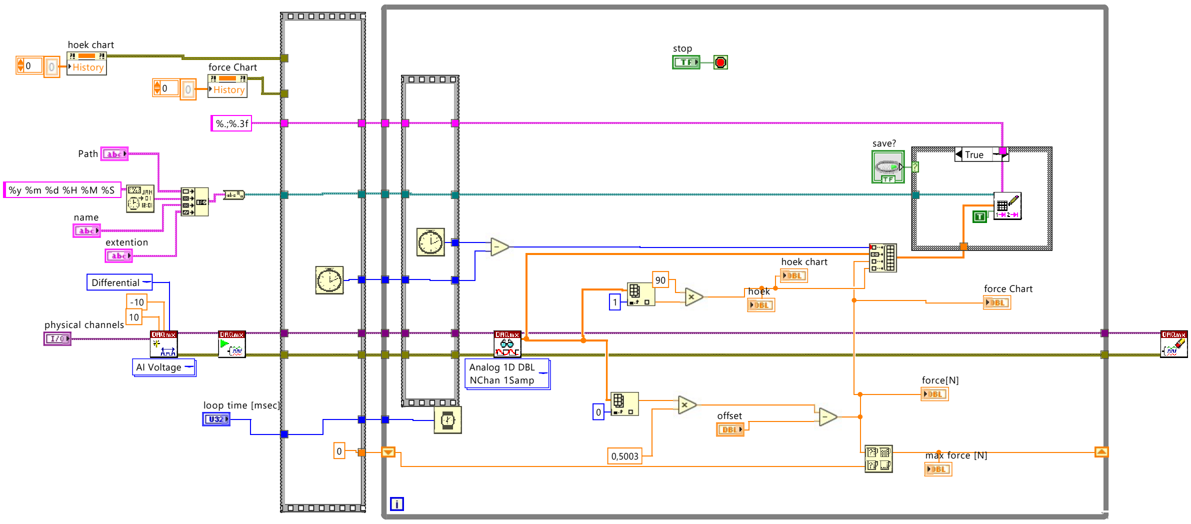


Figure K.1: LabVIEW code block diagram

L
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".

```
#!/bin/sh

#request 20 processors on 1 node
#PBS -l nodes=1:ppn=20

#define the name of the job
#PBS -N Optimization_Sjors

#provide mail adress
#PBS -M a.c.vannes@student.tudelft.nl

#give mail preferences
#PBS -m abe

# Make sure I'm the only one that can read my output
umask 0077

#change to the working directory
cd $PBS_O_WORKDIR

#load MATLAB
module load matlab/2020b

#define the name of the MATLAB file
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_{1_0}	Opt. θ_{1_0}, prestress	Nonlinear, opt. θ_{1_0}	Nonlinear, opt. θ_{1_0}, 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_{1_0}	Opt. θ_{1_0}, prestress	NL, opt. θ_{1_0}	NL, opt. θ_{1_0}, 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

Linear springs

	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	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	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

	A	B	M03	M02	l1	l2	l3	l4	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.	-0.6047	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

Linear springs

	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	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

	A	B	M03	M02	l1	l2	l3	l4	theta10	RMSE	Work red.
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.	-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

Linear springs

	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	0	0	0	0	0	0	0.1	0.1	0.1	0.1			
UB	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

	A	B	M03	M02	l1	l2	l3	l4	RMSE	Work red.
LB	-2	-2	0	0	0.1	0.1	0.1	0.1		
UB	2	2	1	1	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	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	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	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	l1	l2	l3	l4	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	l1	l2	l3	l4	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	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	l1	l2	l3	l4	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	l1	l2	l3	l4	contactan	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	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	l1	l2	l3	l4	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															
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	contactan	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	1	1	0.5-0.375	0.5-0.375	0.5-0.375	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														
	A	B			M03	M02	l1	l2	l3	l4	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.	0.6425	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													
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	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.	0.9601	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												
	A	B			M03	M02	l1	l2	l3	l4	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.	0.2203	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														
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	1	1	0.5-0.375	0.5-0.375	0.5-0.375	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													
	A	B			M03	M02	l1	l2	l3	l4	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.	0.849	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	l1	l2	l3	l4	theta10	contactan	RMSE	Best theor. Approx.	Work red.
LB	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	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	l1	l2	l3	l4	theta10	contactan	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.	0.0275	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	l1	l2	l3	l4	theta10	contactan	RMSE	Best theor. Approx.	Work red.
LB	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	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	l1	l2	l3	l4	theta10	contactan	RMSE		Work red.
LB	-4.5	-4.5			0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	4.5	4.5			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	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	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.	0.6324	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	l1	l2	l3	l4	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.	-0.2911	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

	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375				
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

	A	B			M03	M02	l1	l2	l3	l4	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.	-0.2247	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

	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	contactan	RMSE	Best theor. Approx.	Work red.
LB	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	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
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

	A	B			M03	M02	l1	l2	l3	l4	theta10	contactan	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.	-0.3131	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

	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	contactan	RMSE	Best theor. Approx.	Work red.
LB	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	1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
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

	A	B			M03	M02	l1	l2	l3	l4	theta10	contactan	RMSE		Work red.
LB	-4.5	-4.5			0	0	0.33-0.25	0.33-0.25	0.33-0.25	0.25					
UB	4.5	4.5			1	1	0.5-0.375	0.5-0.375	0.5-0.375	0.375					
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													
Linear springs													
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	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.	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												
	A	B			M03	M02	l1	l2	l3	l4	RMSE	Work red.
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.	-1.1842	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														
Linear springs														
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	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													
	A	B			M03	M02	l1	l2	l3	l4	theta10	RMSE	Work red.
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.	-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													
Linear springs													
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	0	0	0	0	0	0	0.1	0.1	0.1	0.1			
UB	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												
	A	B			M03	M02	l1	l2	l3	l4	RMSE	Work red.
LB	-3	-3			0	0	0.1	0.1	0.1	0.1		
UB	3	3			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												
	Linear springs												
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	RMSE	Best theor. Approx.	Work red.
LB	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											
	A	B			M03	M02	l1	l2	l3	l4	RMSE	Work red.
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													
	Linear springs													
	k1	k2	k3	k4	M03	M02	l1	l2	l3	l4	theta10	RMSE	Best theor. Approx.	Work red.
LB	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	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												
	A	B			M03	M02	l1	l2	l3	l4		RMSE	Work red.
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

```
clc                                                 %clear command window
clear variables                                     %empty workspace
close all                                           %close all windows

%the lowest value the minimizer can attain is equal to 1:
%the first entry of the vector with springs
lb = [1,1,1,1,1,1];
%the highest value the minimizer can attain is equal to 13:
%the last entry of the vector with springs
ub = [13,13,13,13,13,13];
%six optimizers, i.e. six vacancies for clock springs
nvars = 6;
%all design variables should be integer-valued
intcon = [1 2 3 4 5 6];

%do the optimization "n" times
n = 100;
%preallocate a matrix to store the minimizer values
testvectorX = zeros(nvars,n);
%preallocate a matrix to store the objective function value for each run
testvectorY = zeros(1,n);

%do a genetic algorithm optimization "n" times
for i = 1:1:n
%[output] = ga(input)
%objective function "f135", see bottom of script
%nvars is te dimension of the objective function
%refer to lower (lb) and upper (ub) bounds
%the variables listed in intcon should be integer values
[xvec,fval,exitflag,output,population,scores] = ga(@f315,nvars...
        ,[],[],[],[],lb,ub,[],intcon);

%store the minimizer solution values in the i-th column of the
%"testvectorX" matrix...
testvectorX(:,i) = transpose(xvec);
%...and write the objective function value to the corresponding column
%of the "testvectorY" vector
testvectorY(1,i) = fval;
end

%load the vector with spring stiffnesses, obtained from spring supplier
load Rvec
%store the lowest possible objective function value and call this value
%"testvectorYopt"
%save the corresponding index as well
[testvectorYopt,I] = min(testvectorY);
%the "testvectorXopt" vector contains the values of the minimizers for
%the lowest possible objective function value
testvectorXopt = testvectorX(:,I);
```

```
%translate the integer-valued minimizers into a corresponding spring
%stiffnesses by reading the entries from the spring stiffness vector
stiffnesses = [R(testvectorXopt(1)) R(testvectorXopt(2))...
    R(testvectorXopt(3)) R(testvectorXopt(4))...
    R(testvectorXopt(5)) R(testvectorXopt(6))];

%the objective function
function e = f315(x)
%the to be approximated ratio between the nett stiffness on axis 1 w.r.t.
%the nett stiffness on axis 2
A = 1.498937760758821;
%the to be approximated ratio between the nett stiffness on axis 3 w.r.t.
%the nett stiffness on axis 2
B = 4.847839773931474;
%load the vector with spring stiffnesses again
load Rvec %#ok<LOAD>

%calculate the objective function as a measure of a resultant
%deviaton from the required spring stiffness ratios
e = sqrt((((R(x(1))+R(x(2)))/(R(x(3))+R(x(4))))-A)^2 +...
    (((R(x(5))+R(x(6)))/(R(x(3))+R(x(4))))-B)^2);
end
```

N.2. Work reduction

```
close all                    %close all windows
clear variables              %empty workspace

load Angle_lower             %load angle data lower part of hysteresis loop
load Angle_upper             %load angle data upper part of hysteresis loop
load Moment_lower            %load moment data lower part of hysteresis loop
load Moment_upper            %load moment data upper part of hysteresis loop

%M_obj is a fit of the expected moment-angle curve
%Below, the arrays "Angle_upper" and "Angle_lower" are inserted as its
%Argument to calculate the corresponding fit value
M_obj_u = ((0.000000000044283)*Angle_upper.^5) -...
    ((0.0000000082536)*Angle_upper.^4) +...
    ((0.00000048168)*Angle_upper.^3) - ((0.000014378)*Angle_upper.^2) +...
    0.0011225*Angle_upper - 0.00029428;

M_obj_l = ((0.000000000044283)*Angle_lower.^5) -...
    ((0.0000000082536)*Angle_lower.^4) +...
    ((0.00000048168)*Angle_lower.^3) - ((0.000014378)*Angle_lower.^2) +...
    0.0011225*Angle_lower - 0.00029428;

figure
hold on
%plot(Angle_lower,M_obj_l)
plot(Angle_upper,M_obj_u)
plot(Angle_lower,Moment_lower)
plot(Angle_upper,Moment_upper)
xlabel('Angle pendulum (deg)')
ylabel('Moment (Nm)')


%Obtain the work corresponding to the lower part of the hysteresis loop
Work_lower = trapz(Angle_lower(117:23479),...
    abs(Moment_lower(117:23479)-M_obj_l(117:23479)));

%Obtain the work corresponding to the upper part of the hysteresis loop
Work_upper = trapz(Angle_upper(167:33411),...
    abs(Moment_upper(167:33411)-M_obj_u(167:33411)));

%Obtain the work corresponding to the reference configuration
Work_ref = trapz(Angle_upper(167:33411),M_obj_u(167:33411))+...
            trapz(Angle_lower(117:23479),M_obj_l(117:23479));

%Calculate the work reduction
Work_red = (((Work_lower+Work_upper)-Work_ref)/Work_ref)*100;
```

N.3. Three segment balancer

```
clc                                             %clear command window
clear variables                                 %empty workspace
close all                                       %close all windows

%prompt that asks for desired configurations and saves...
%...preference by means of variable X
prompt = 'Knee up (1) or knee down (0)?';
X = input(prompt);
```

```

%with (1) or without (0) prestress on springs
prestress = 0;
%with (1) or without (0) nonlinear springs
nonlinearity = 0;
%type objective function between quotation marks...
objective = "sinus";

%select manually: elbow up solution (X = 1;) or elbow down (X = 0;)
%X = 1;
M = 90;                                         %amount of precision points
%amount of configurations per precision point
N = 1000;

%stiffness spring 1 (Nm/rad)
k1 = 0.03732627 + 0.03755394;
%stiffness spring 2 (Nm/rad)
k2 = 0.0217042 + 0.021610197;
%stiffness spring 3 (Nm/rad)
k3 = 0.035792513 + 0.03678242 + 0.071335373 + 0.068828657;

l1 = 0.16968851561841500;                       %length first segment (m)
l2 = 0.22863398856099401;                       %length second segment (m)
l3 = 0.23760765730766202;                       %length third segment (m)

%angle at which contact is enabled at spring 2 (rad)
contactan = 100;
%initial angle of segment 1 (rad)
theta10 = 0.664071011410003;
%length pendulum (m)
r = 0.5;

%constant mass times grav. constant (N)
mg = 0.029321787763455*2*2;

%angle between segment 3 and pendulum when segment 2 and 3 are aligned
D3 = acos((r^2 + (l2+l3)^2 - l1^2)/(2*r*(l2+l3)));
%angle between segment 1 and segment 2 " "
D2 = acos((l1^2 + (l2+l3)^2 - r^2)/(2*l1*(l2+l3)));

%angle of pendulum and first segment in case of deviating...
%...definitions of angles
phir0 = (pi/2);
phi10 = (pi/2) - theta10;
Mtheta10 = -theta10;

%if initial angle of first segment is greater than zero or equal to zero
if theta10 >= 0
   %initial angle segment 2
   theta20ku = pi/2 - real(pi - acos((l1*cos(phi10) - r*cos(phir0) +...
       l3*cos(log(-(((l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i)-...
       l1^2*exp(phir0*1i)*exp(phi10*1i) + l2^2*exp(phir0*1i)*exp(phi10*1i) +...
       l3^2*exp(phir0*1i)*exp(phi10*1i) -...
       r^2*exp(phir0*1i)*exp(phi10*1i) - 2*l2*l3*exp(phir0*1i)*exp(phi10*1i))*...
       (l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) -...
       l1^2*exp(phir0*1i)*exp(phi10*1i) + l2^2*exp(phir0*1i)*exp(phi10*1i) +...
       l3^2*exp(phir0*1i)*exp(phi10*1i) -...
       r^2*exp(phir0*1i)*exp(phi10*1i) + 2*l2*l3*exp(phir0*1i)*...
       exp(phi10*1i)))^(1/2) - l1*r*exp(phir0*2i) - l1*r*exp(phi10*2i) +...
       l1^2*exp(phir0*1i)*exp(phi10*1i) - l2^2*exp(phir0*1i)*exp(phi10*1i) +...
       l3^2*exp(phir0*1i)*exp(phi10*1i) +...
       r^2*exp(phir0*1i)*exp(phi10*1i))/(2*(l1*l3*exp(phir0*1i) -...
       l3*r*exp(phi10*1i)))))*1i))/l2));
   %initial angle segment 3
   theta30ku = pi/2 - real(-log(-(((l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) -...
       l1^2*exp(phir0*1i)*exp(phi10*1i) +...
       l2^2*exp(phir0*1i)*exp(phi10*1i) + l3^2*exp(phir0*1i)*exp(phi10*1i) -...
       r^2*exp(phir0*1i)*exp(phi10*1i) -...
       2*l2*l3*exp(phir0*1i)*exp(phi10*1i))*(l1*r*exp(phir0*2i) +...
       l1*r*exp(phi10*2i) - l1^2*exp(phir0*1i)*exp(phi10*1i) +...
       l2^2*exp(phir0*1i)*exp(phi10*1i) + l3^2*exp(phir0*1i)*exp(phi10*1i) -...
       r^2*exp(phir0*1i)*exp(phi10*1i) +...
       2*l2*l3*exp(phir0*1i)*exp(phi10*1i)))^(1/2) - l1*r*exp(phir0*2i) -...
       l1*r*exp(phi10*2i) + l1^2*exp(phir0*1i)*exp(phi10*1i) -...
       l2^2*exp(phir0*1i)*exp(phi10*1i) + l3^2*exp(phir0*1i)*exp(phi10*1i) +...
       r^2*exp(phir0*1i)*exp(phi10*1i))/(2*(l1*l3*exp(phir0*1i) -...
       l3*r*exp(phi10*1i))))*1i);

   %calculate the deviations in x and y of the coordinates of the compensator, respectively
   DEV10 = l1*sin(theta10) + l2*sin(theta20ku) + l3*sin(theta30ku) - r*sin(0);
   DEV20 = l1*cos(theta10) + l2*cos(theta20ku) + l3*cos(theta30ku) - r*cos(0);

   %if any of the deviations is greater than its treshold
   if abs(DEV10) > 10^-12 || abs(DEV20) > 10^-8
```

```
       %other formulation initial angle segment 2
       theta20ku = pi/2 - real(pi + acos((l1*cos(phi10) - r*cos(phir0) +...
           l3*cos(log(-(((l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) -...
           l1^2*exp(phir0*1i)*exp(phi10*1i) + l2^2*exp(phir0*1i)*exp(phi10*1i) +...
           l3^2*exp(phir0*1i)*exp(phi10*1i) -...
           r^2*exp(phir0*1i)*exp(phi10*1i) - 2*l2*l3*exp(phir0*1i)*exp(phi10*1i))*...
           (l1*r*exp(phir0*2i) + l1*r*exp(phi10*2i) -...
           l1^2*exp(phir0*1i)*exp(phi10*1i) + l2^2*exp(phir0*1i)*exp(phi10*1i) +...
           l3^2*exp(phir0*1i)*exp(phi10*1i) -...
           r^2*exp(phir0*1i)*exp(phi10*1i) + 2*l2*l3*exp(phir0*1i)*...
           exp(phi10*1i)))^(1/2) - l1*r*exp(phir0*2i) - l1*r*exp(phi10*2i) +...
           l1^2*exp(phir0*1i)*exp(phi10*1i) - l2^2*exp(phir0*1i)*exp(phi10*1i) +...
           l3^2*exp(phir0*1i)*exp(phi10*1i) +...
           r^2*exp(phir0*1i)*exp(phi10*1i))/(2*(l1*l3*exp(phir0*1i) -...
           l3*r*exp(phi10*1i))))*1i))/l2));
   end
end

%if initial angle of first segment is smaller than zero
if theta10 < 0
   %initial angle segment 2
   theta20ku = real(asin((l3*sin(log(-(l1*r + ((l1*r - l1^2*exp(Mtheta10*1i)*...
       exp(0*1i) + l2^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) -...
       2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) +...
       l1*r*exp(Mtheta10*2i)*exp(0*2i))*(l1*r - l1^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l2^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) +...
       2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) +...
       l1*r*exp(Mtheta10*2i)*exp(0*2i))).^(1/2) - l1^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l2^2*exp(Mtheta10*1i)*exp(0*1i) -...
       l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l1*r*exp(Mtheta10*2i)*exp(0*2i))/(2*(l3*r*exp(Mtheta10*1i) -...
       l1*l3*exp(Mtheta10*2i)*exp(0*1i))))*1i) + l1*sin(Mtheta10) + r*sin(0))/l2));
   %initial angle segment 3
   theta30ku = real(-log(-(l1*r + ((l1*r - l1^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l2^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) -...
       2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) +...
       l1*r*exp(Mtheta10*2i)*exp(0*2i))*(l1*r - l1^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l2^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) +...
       2*l2*l3*exp(Mtheta10*1i)*exp(0*1i) +...
       l1*r*exp(Mtheta10*2i)*exp(0*2i)))^(1/2) - l1^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l2^2*exp(Mtheta10*1i)*exp(0*1i) -...
       l3^2*exp(Mtheta10*1i)*exp(0*1i) - r^2*exp(Mtheta10*1i)*exp(0*1i) +...
       l1*r*exp(Mtheta10*2i)*exp(0*2i))/(2*(l3*r*exp(Mtheta10*1i) -...
       l1*l3*exp(Mtheta10*2i)*exp(0*1i))))*1i);
end

%confine with elbow up solutions for the initial angles for now
theta20 = theta20ku;
theta30 = theta30ku;

%preallocate all variables for better performance...
%...vectors
alpha = zeros(1,M);
alpha1m = zeros(1,M);
alpha2m = zeros(1,M);
alpha3m = zeros(1,M);
theta100 = zeros(1,M);
theta1ff = zeros(1,M);
theta1m = zeros(1,M);
theta2m = zeros(1,M);
theta3m = zeros(1,M);
BEGIN = zeros(1,M);
END = zeros(1,M);
STEP = zeros(1,M);
M1m = zeros(1,M);
M2m = zeros(1,M);
M3m = zeros(1,M);
Mobj = zeros(1,M);
phir = zeros(1,M);
fit = zeros(1,M);
Vm = zeros(1,M);
V1m = zeros(1,M);
V2m = zeros(1,M);
V3m = zeros(1,M);
Vtm = zeros(1,M);
F1y = zeros(1,M);
F1x = zeros(1,M);
F2y = zeros(1,M);
F2x = zeros(1,M);
F1r = zeros(1,M);
```

```
M1l = zeros(1,M);
M2l = zeros(1,M);
M3l = zeros(1,M);
phi = zeros(1,M);
Tlb = zeros(1,M);
Tub = zeros(1,M);
Start = zeros(1,M);
Stop = zeros(1,M);
%...matrices
DEV1 = zeros(M,N);
DEV2 = zeros(M,N);
theta1 = zeros(M,N);
theta2 = zeros(M,N);
theta3 = zeros(M,N);
theta2kd = zeros(M,N);
theta3kd = zeros(M,N);
theta2ku = zeros(M,N);
theta3ku = zeros(M,N);
phi1 = zeros(M,N);
Mtheta1 = zeros(M,N);
theta1P = zeros(M,N);
d = zeros(M,N);
alpha1 = zeros(M,N);
alpha2 = zeros(M,N);
alpha3 = zeros(M,N);
V = zeros(M,N);
V1 = zeros(M,N);
V2 = zeros(M,N);
V3 = zeros(M,N);
M1 = zeros(M,N);
M2 = zeros(M,N);
M3 = zeros(M,N);
x1 = zeros(M,N);
x2 = zeros(M,N);
x3 = zeros(M,N);
y1 = zeros(M,N);
y2 = zeros(M,N);
y3 = zeros(M,N);
F1xt = zeros(M,N);
F1yt = zeros(M,N);
F2xt = zeros(M,N);
F2yt = zeros(M,N);
M2lt = zeros(M,N);
M3lt = zeros(M,N);

Count = 0;                                              %error counter
Count2 = 0;                                             %second error counter

%activation = 0 (only spring 1 active) or activation = 1 (all springs active)
activation = 0;
theta1ln = pi/2;
alphaln = pi/2;

%start a loop throughout all precision points
for j = 1:1:M
%divide the 90 deg range of motion into equally sized segments
alpha(j) = (pi/2)*(j/M);

%lowerbound of theta1 such that precision point (j) is still reached
theta100(j) = alpha(j) - pi + D2 + D3;
%upperbound of theta1 such that precision point (j) is still reached
theta1ff(j) = alpha(j) - (-pi + D2 + D3);

%if arm,consisting of segment 2 and segment 3,can not be fully stretched...
%...segment 1 should be given a full rotation for sweep as no phyiscal
%lower and upperbound exist

%check whether segment 2 and 3 can be aligned (stretched arm)
if (l1+r) <= (l2+l3)
    %alternative formulation lowerbound of theta1 if arm cannot be
    %stretched
    theta100(j) = alpha(j) - pi;
    %alternative formulation upperbound of theta1 if arm cannot be
    %stretched
    theta1ff(j) = alpha(j) + pi;
end

BEGIN(j) = theta100(j);                                 %begin interval
END(j) = theta1ff(j);                                   %end interval

STEP(j) = (END(j)-BEGIN(j))/N;                          %stepsize

%loop for segment 1 angle sweep
for i = 1:1:N
```

```
%increase angle with steps equal to the stepsize STEP(j)
if prestress == 0
    theta1(j,i) = BEGIN(j) + STEP(j)*i;
end


if prestress == 1
    %springs 2 and 3 not involved yet
    if activation == 0
    %the angle of the first segment when only spring 1 is active
        theta1(j,i) = alpha(j)+theta10;
    end

    %springs 2 and 3 engaged
    if activation == 1
        %increase angle with steps equal to the stepsize STEP(j)
        theta1(j,i) = BEGIN(j) + STEP(j)*i;
    end
end

%the following holds when contact is not engaged
if alpha2(j,i) < contactan

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %Mtheta1(j,i) is used instead of theta1(j,i) for practical reasons
    Mtheta1(j,i) = - theta1(j,i);

    %formulation for angle segment 3: elbow up
    theta3ku(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
        l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i) -...
        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i)))))*1i);
    %formulation for angle segment 2: elbow up
    theta2ku(j,i) = real(asin((l3*sin(log(-(l1*r +...
        ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))).^(1/2) -...
        l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i)-...
        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) +...
        l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));

    %formulation for angle segment 3: elbow down
    theta3kd(j,i) = real(-log((- l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) -...
        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) +...
        l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i)-...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
        (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i);
```

```
    %formulation for angle segment 2: elbow down
    theta2kd(j,i) = pi + real(- asin((l3*sin(log((- l1*r +...
        ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))).^(1/2) +...
        l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
        (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) +...
        l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));

    %select the angles for segment 2 and 3 corresponding to elbow up if X = 1 is selected
    if X == 1
        theta2(j,i) = theta2ku(j,i);
        theta3(j,i) = theta3ku(j,i);
    end

    %select the angles for segment 2 and 3 corresponding to elbow down if X = 0 is selected
    if X == 0
        theta2(j,i) = theta2kd(j,i);
        theta3(j,i) = theta3kd(j,i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
        r*sin(alpha(j));
    DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
        r*cos(alpha(j));

    %if the absolute value of any of these deviations transcends a certain threshold,
    %then use alternative formulations for theta2
    if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
        if X == 1    %elbow up
        %other formulation for angle segment 2
            theta2(j,i) = pi + real( - asin((l3*sin(log(-(l1*r +...
                ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
                (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
                l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
                (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
                exp(alpha(j)*1i))))*1i) +...
                l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));

            %even other formulation for angle segment 2 if angle is larger than 180 deg
                if theta2(j,i) > pi
                    theta2(j,i) = -pi + real( - asin((l3*sin(log(-(l1*r +...
                        ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
                        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
                        l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
```

```matlab
                        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
                        (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
                        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) +...
                        l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));
                end
        end

        if X == 0    %elbow down
        %other formulation for angle segment 2
            theta2(j,i) = real(asin((l3*sin(log((- l1*r +...
                ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
                (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) +...
                l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
                (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
                exp(alpha(j)*1i))))*1i) + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));
        end
    end
end

%the expressions within this loop are valid for theta1 >= 0
if theta1(j,i) >= 0
    %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
    phir(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (anti-clockwise positive)
    phi1(j,i) = (pi/2) - theta1(j,i);

    %formulation for angle segment 3: elbow up
    theta3ku(j,i) = pi/2 - real(-log(-(((l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) -...
        l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
        l3*r*exp(phi1(j,i)*1i))))*1i);

    %formulation for angle segment 2: elbow up
    theta2ku(j,i) = pi/2 - real(pi - acos((l1*cos(phi1(j,i)) - r*cos(phir(j)) +...
        l3*cos(log(-(((l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) - l1*r*exp(phir(j)*2i) -...
        l1*r*exp(phi1(j,i)*2i) + l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
        l3*r*exp(phi1(j,i)*1i))))*1i))/l2));

    %formulation for angle segment 3: elbow down
    theta3kd(j,i) = pi/2 - real(-log((((l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
```

```
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) +...
        l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
        l3*r*exp(phi1(j,i)*1i))))*1i);

    %formulation for angle segment 2: elbow down
    theta2kd(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
        r*cos(phir(j)) + l3*cos(log((((l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) + l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) -...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
        l3*r*exp(phi1(j,i)*1i))))*1i))/l2));

    %select the angles of segment 2 and 3 corresponding to elbow up if X = 1 is selected
    if X == 1
        theta2(j,i) = theta2ku(j,i);
        theta3(j,i) = theta3ku(j,i);
    end

    %select the angles of segment 2 and 3 corresponding to elbow down if X = 0 is selected
    if X == 0
        theta2(j,i) = theta2kd(j,i);
        theta3(j,i) = theta3kd(j,i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
        r*sin(alpha(j));
    DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
        r*cos(alpha(j));

    %if the absolute value of any of these deviations transcends a certain threshold,
    %then use alternative formulations for theta2
    if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-8
        if X == 1   %elbow up
        %other formulation for angle segment 2
            theta2(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
                r*cos(phir(j)) + l3*cos(log(-(((l1*r*exp(phir(j)*2i) +...
                l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
                (l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
                l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i)))^(1/2) - l1*r*exp(phir(j)*2i) -...
                l1*r*exp(phi1(j,i)*2i) + l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + r^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
                l3*r*exp(phi1(j,i)*1i))))*1i))/l2));

            %even other formulation for angle segment 2 if angle is larger
            %than 180 deg
            if theta2(j,i) > pi
                    theta2(j,i) = pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
                        r*cos(phir(j)) + l3*cos(log(-(((l1*r*exp(phir(j)*2i) +...
                        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
```

```
                        exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
                        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i)))^(1/2) - l1*r*exp(phir(j)*2i) -...
                        l1*r*exp(phi1(j,i)*2i) + l1^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + r^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
                        l3*r*exp(phi1(j,i)*1i))))*1i))/l2));
            end
        end

        if X == 0   %elbow down
        %other formulation for angle segment 2
            theta2(j,i) = pi/2 - real(pi - acos((l1*cos(phi1(j,i)) -...
                r*cos(phir(j)) + l3*cos(log((((l1*r*exp(phir(j)*2i) +...
                l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                2*l2*l3*exp(phir(j))*1i)*exp(phi1(j,i)*1i))*...
                (l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
                l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) +...
                l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
                l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
                l3*r*exp(phi1(j,i)*1i))))*1i))/l2));
        end
    end
end

%in the case of a horizontally positioned segment 1, MATLAB solve() has
%troubles finding a solution... Therefore, perturb by small amount to solve
if theta1(j,i) == pi/2
   theta1(j,i) = pi/2 + STEP(j);
end

%the expressions within this loop are valid for theta1 > pi/2
if theta1(j,i) > pi/2
    %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
    phir(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (clockwise positive)
    theta1P(j,i) = theta1(j,i) - pi/2;

    %formulation for angle segment 3: elbow up
    theta3ku(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
        (2*(l1*l3*exp(phir(j)*1i)*1i -...
        l3*r*exp(phir(j)*2i)*exp(theta1P(j,i)*1i)*1i)))*1i);

    %formulation for angle segment 2: elbow up
    theta2ku(j,i) = real(asin((l3*sin(log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
        (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
        exp(theta1P(j,i)*1i)*1i)))*1i) - l1*cos(theta1P(j,i)) +...
        r*cos(phir(j)))/l2));

```

```
    %formulation for angle segment 3: elbow down
    theta3kd(j,i) = real(-log((- l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) + l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
        (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
        exp(theta1P(j,i)*1i)*1i)))*1i);

    %formulation for angle segment 2: elbow down
    theta2kd(j,i) = pi +...
        real( - asin((l3*sin(log((- l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) +...
        l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*...
        exp(theta1P(j,i)*2i))/(2*(l1*l3*exp(phir(j)*1i)*1i -...
        l3*r*exp(phir(j)*2i)*exp(theta1P(j,i)*1i)*1i)))*1i) -...
        l1*cos(theta1P(j,i)) + r*cos(phir(j)))/l2));

    %select the angles of the second and third segment corresponding to elbow/ up...
    %...if X = 1 is selected
    if X == 1
        theta2(j,i) = theta2ku(j,i);
        theta3(j,i) = theta3ku(j,i);
    end

    %select the angles of the second and third segment corresponding to elbow/ down...
    %...if X = 0 is selected
    if X == 0
        theta2(j,i) = theta2kd(j,i);
        theta3(j,i) = theta3kd(j,i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
        r*sin(alpha(j));
    DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
        r*cos(alpha(j));

    %if the absolute value of any of these deviations transcends a certain threshold,
    %then use alternative formulations for theta2
    if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
        if X == 1   %elbow up
        %other formulation for angle segment 2
            theta2(j,i) = pi + real( - asin((l3*sin(log(-(l1*r +...
                ((l1*r - l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
                2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*...
                exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) +...
                2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) -...
                l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
                (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
                exp(theta1P(j,i)*1i)*1i)))*1i) -...
                l1*cos(theta1P(j,i)) + r*cos(phir(j)))/l2));
        end

        if X == 0   %elbow down
        %other formulation for angle segment 2
            theta2(j,i) = real(asin((l3*sin(log((- l1*r +...
                ((l1*r - l1^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
```

```
                l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
                r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*...
                (l1*r - l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) +...
                l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*...
                exp(theta1P(j,i)*2i))/(2*(l1*l3*exp(phir(j)*1i)*1i -...
                l3*r*exp(phir(j)*2i)* exp(theta1P(j,i)*1i)*1i)))*1i) -...
                l1*cos(theta1P(j,i)) + r*cos(phir(j)))/l2));
        end
    end
end

end

%if the current angle of deformation of the second spring is larger than...
%...the contact angle - while the angle of the previous posture is not -...
%...contact is just engaged
if  i > 1 && (alpha2(j,i) > contactan) && (alpha2(j,i-1) < contactan)
    %save angle of segment 1 corresponding to contact
    theta1ln = theta1(j,i);
    %save angle of pendulum corresponding to contact
    alphaln = alpha(j);
end

%initial relative angle of segment 2
alpha20 = theta20 - theta10;
%angle of rotation torsion spring 2
alpha2(j,i) = theta2(j,i) - theta1(j,i) - alpha20;

%if the angle of deformation of the second spring is larger than the...
%...contact angle
if  alpha2(j,i) > contactan
    %retrieve angle of segment 1
    theta1(j,i) = theta1ln + (alpha(j)-alphaln);

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %Mtheta1(j,i) is used instead of theta1(j,i) for practical reasons
    Mtheta1(j,i) = - theta1(j,i);

    %formulation for angle third segment: elbow up
    theta3ku(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
        l1^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) + l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
        (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i);

    %formulation for angle second segment: elbow up
    theta2ku(j,i) = real(asin((l3*sin(log(-(l1*r +...
        ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + 2*l2*l3*...
        exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l1*r*exp(Mtheta1(j,i)*2i)*...
        exp(alpha(j)*2i))).^(1/2) - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l3^2*exp(Mtheta1(j,i)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i) -...
        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) + l1*sin(Mtheta1(j,i)) +...
        r*sin(alpha(j)))/l2));

```

```
%formulation for angle third segment: elbow down
theta3kd(j,i) = real(-log((- l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
    (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) +...
    l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
    l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
    l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i) -...
    l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i);

%formulation for angle second segment: elbow down
theta2kd(j,i) = pi +...
    real(- asin((l3*sin(log((- l1*r + ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) - 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
    (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - r^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + 2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))).^(1/2) +...
    l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l2^2*exp(Mtheta1(j,i)*1i)*...
    exp(alpha(j)*1i) + l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
    r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) - l1*r*exp(Mtheta1(j,i)*2i)*...
    exp(alpha(j)*2i))/(2*(l3*r*exp(Mtheta1(j,i)*1i) -...
    l1*l3*exp(Mtheta1(j,i)*2i)*...
    exp(alpha(j)*1i))))*1i) + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));

%select the angles for the second and third segment corresponding to elbow up...
%...if X = 1 is selected
if X == 1
    theta2(j,i) = theta2ku(j,i);
    theta3(j,i) = theta3ku(j,i);
end

%select the angles for the second and third segment corresponding to elbow down...
%...if X = 0 is selected
if X == 0
    theta2(j,i) = theta2kd(j,i);
    theta3(j,i) = theta3kd(j,i);
end

%calculate the deviations in x and y of the coordinates of the compensator, respectively
DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
    r*sin(alpha(j));
DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
    r*cos(alpha(j));

%if the absolute value of any of these deviations transcends a certain threshold,
%then use alternative formulations for theta2
if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
    if X == 1    %elbow up
    %other formulation for angle segment 2
        theta2(j,i) = pi + real( - asin((l3*sin(log(-(l1*r +...
            ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
            2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
            (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
            l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
            l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
            l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
            (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
            exp(alpha(j)*1i))))*1i) + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));

            %even other formulation for angle segment 2 if angle is
            %larger than 180 deg
```

```
                if theta2(j,i) > pi
                    theta2(j,i) = -pi + real( - asin((l3*sin(log(-(l1*r +...
                        ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
                        (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^...
                        (1/2) - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                        r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                        l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
                        (2*(l3*r*exp(Mtheta1(j,i)*1i) -...
                        l1*l3*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*1i))))*1i) +...
                        l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));
                end
        end

        if X == 0    %elbow down
        %other formulation for angle segment 2
            theta2(j,i) = real(asin((l3*sin(log((- l1*r +...
                ((l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))*...
                (l1*r - l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                2*l2*l3*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) +...
                l1^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                l2^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) +...
                r^2*exp(Mtheta1(j,i)*1i)*exp(alpha(j)*1i) -...
                l1*r*exp(Mtheta1(j,i)*2i)*exp(alpha(j)*2i))/...
                (2*(l3*r*exp(Mtheta1(j,i)*1i) - l1*l3*exp(Mtheta1(j,i)*2i)*...
                exp(alpha(j)*1i))))*1i) + l1*sin(Mtheta1(j,i)) + r*sin(alpha(j)))/l2));
        end
    end
end

%the expressions within this loop are valid for theta1 >= 0
if theta1(j,i) >= 0
    %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
    phir(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (anti-clockwise positive)
    phi1(j,i) = (pi/2) - theta1(j,i);

    %formulation for angle third segment: elbow up
    theta3ku(j,i) = pi/2 - real(-log(-(((l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) -...
        l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
        l3*r*exp(phi1(j,i)*1i)))))*1i);

    %formulation for angle second segment: elbow up
    theta2ku(j,i) = pi/2 - real(pi - acos((l1*cos(phi1(j,i)) - r*cos(phir(j)) +...
        l3*cos(log(-(((l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
        exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...
```

```
            exp(phi1(j,i)*1i)))^(1/2) - l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
            l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
            l3*r*exp(phi1(j,i)*1i))))*1i))/l2));

        %formulation for angle third segment: elbow down
        theta3kd(j,i) = pi/2 - real(-log((((l1*r*exp(phir(j)*2i) +...
            l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
            2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
            l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) + l1*r*exp(phir(j)*2i) +...
            l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
            (2*(l1*l3*exp(phir(j)*1i) - l3*r*exp(phi1(j,i)*1i))))*1i);

        %formulation for angle second segment: elbow down
        theta2kd(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
            r*cos(phir(j)) + l3*cos(log((((l1*r*exp(phir(j)*2i) +...
            l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
            2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
            l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) + l1*r*exp(phir(j)*2i) +...
            l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
            l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
            exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
            (2*(l1*l3*exp(phir(j)*1i) - l3*r*exp(phi1(j,i)*1i))))*1i))/l2));

        %select the angles of the second and third segment corresponding to elbow up...
        %...if X = 1 is selected
        if X == 1
            theta2(j,i) = theta2ku(j,i);
            theta3(j,i) = theta3ku(j,i);
        end

        %select the angles of the second and third segment corresponding to elbow down...
        %...if X = 0 is selected
        if X == 0
            theta2(j,i) = theta2kd(j,i);
            theta3(j,i) = theta3kd(j,i);
        end

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
            r*sin(alpha(j));
        DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
            r*cos(alpha(j));

        %if the absolute value of any of these deviations transcends a certain threshold,
        %then use alternative formulations for theta2
        if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-8
            %display("Alternative formulation for pos theta1 active")
            if X == 1   %elbow up
            %other formulation for angle segment 2
                theta2(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
                    r*cos(phir(j)) + l3*cos(log(-(((l1*r*exp(phir(j)*2i) +...
                    l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                    l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                    exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                    2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*(l1*r*exp(phir(j)*2i) +...
                    l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                    l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                    exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                    2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) -...
                    l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
                    l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l2^2*exp(phir(j)*1i)*...
                    exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                    r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/(2*(l1*l3*exp(phir(j)*1i) -...
                    l3*r*exp(phi1(j,i)*1i))))*1i))/l2));

                    %even other formulation for angle segment 2 if angle is
                    %larger than 180 deg
                    if theta2(j,i) > pi
                        theta2(j,i) = pi/2 - real(pi + acos((l1*cos(phi1(j,i)) -...
                            r*cos(phir(j)) + l3*cos(log(-(((l1*r*exp(phir(j)*2i) +...
```

```
                        l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*...
                        exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
                        (l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
                        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                        2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i)))^(1/2) -...
                        l1*r*exp(phir(j)*2i) - l1*r*exp(phi1(j,i)*2i) +...
                        l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                        l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                        l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                        r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
                        (2*(l1*l3*exp(phir(j)*1i) -...
                        l3*r*exp(phi1(j,i)*1i))))*1i))/l2));
                end
        end

        if X == 0   %elbow down
        %other formulation for angle segment 2
            theta2(j,i) = pi/2 - real(pi - acos((l1*cos(phi1(j,i)) -...
                r*cos(phir(j)) + l3*cos(log((((l1*r*exp(phir(j)*2i) +...
                l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                2*l2*l3*exp(phir(j)*1i)*exp(phi1(j,i)*1i))*...
                (l1*r*exp(phir(j)*2i) + l1*r*exp(phi1(j,i)*2i) -...
                l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + l2^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) + l3^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) -...
                r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i)))^(1/2) + l1*r*exp(phir(j)*2i) +...
                l1*r*exp(phi1(j,i)*2i) - l1^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) +...
                l2^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i) - l3^2*exp(phir(j)*1i)*...
                exp(phi1(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(phi1(j,i)*1i))/...
                (2*(l1*l3*exp(phir(j)*1i) - l3*r*exp(phi1(j,i)*1i))))*1i))/l2));
        end
    end
end

%in the case of a horizontally positioned segment 1, MATLAB solve() has
%troubles finding a solution... Therefore, perturb by small amount to solve
if theta1(j,i) == pi/2
   theta1(j,i) = pi/2 + STEP(j);
end

%the expressions within this loop are valid for theta1 > pi/2
if theta1(j,i) > pi/2
    %angle of pendulum with respect to positive x-axis (anti-clockwise positive)
    phir(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (clockwise positive)
    theta1P(j,i) = theta1(j,i) - pi/2;

    %formulation for angle of third segment: elbow up
    theta3ku(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
        r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*...
        (l1*r - l1^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + l3^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
        (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
        exp(theta1P(j,i)*1i)*1i)))*1i);

    %formulation for angle of second segment: elbow up
    theta2ku(j,i) = real(asin((l3*sin(log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
        l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) - l1^2*exp(phir(j)*1i)*...
        exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
        l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
```

```
            exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
            (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
            exp(theta1P(j,i)*1i)*1i)))*1i) - l1*cos(theta1P(j,i)) + r*cos(phir(j)))/l2));

        %formulation for angle of third segment: elbow down
        theta3kd(j,i) = real(-log((- l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) + l1^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
            (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
            exp(theta1P(j,i)*1i)*1i)))*1i);

        %formulation for angle of second segment: elbow down
        theta2kd(j,i) = pi +...
            real( - asin((l3*sin(log((- l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i)))^(1/2) + l1^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
            l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
            exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
            (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
            exp(theta1P(j,i)*1i)*1i)))*1i) - l1*cos(theta1P(j,i)) + r*cos(phir(j)))/l2));

        %select the angles of the second and third segment corresponding to elbow up...
        %...if X = 1 is selected
        if X == 1
            theta2(j,i) = theta2ku(j,i);
            theta3(j,i) = theta3ku(j,i);
        end

        %select the angles of the second and third segment corresponding to elbow down...
        %...if X = 0 is selected
        if X == 0
            theta2(j,i) = theta2kd(j,i);
            theta3(j,i) = theta3kd(j,i);
        end

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
            r*sin(alpha(j));
        DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
            r*cos(alpha(j));

        %if the absolute value of any of these deviations transcends a certain threshold,
        %then use alternative formulations for theta2
        if abs(DEV1(j,i)) > 10^-12 || abs(DEV2(j,i)) > 10^-12
            if X == 1   %elbow up
            %other formulation for angle segment 2
                theta2(j,i) = pi +...
                    real( - asin((l3*sin(log(-(l1*r + ((l1*r - l1^2*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                    l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))*...
                    (l1*r - l1^2*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                    l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) +2*l2*l3*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) +l1*r*exp(phir(j)*2i)*...
                    exp(theta1P(j,i)*2i)))^(1/2) -l1^2*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) -...
                    l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                    exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
                    (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
                    exp(theta1P(j,i)*1i)*1i)))*1i) - l1*cos(theta1P(j,i)) +...
                    r*cos(phir(j)))/l2));
            end

            if X == 0   %elbow down
            %other formulation for angle segment 2
```

```
            theta2(j,i) = real(asin((l3*sin(log((- l1*r +...
                ((l1*r - l1^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - 2*l2*l3*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*...
                exp(theta1P(j,i)*2i))*(l1*r - l1^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) - r^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + 2*l2*l3*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) + l1*r*exp(phir(j)*2i)*...
                exp(theta1P(j,i)*2i)))^(1/2) + l1^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - l2^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) +...
                l3^2*exp(phir(j)*1i)*exp(theta1P(j,i)*1i) + r^2*exp(phir(j)*1i)*...
                exp(theta1P(j,i)*1i) - l1*r*exp(phir(j)*2i)*exp(theta1P(j,i)*2i))/...
                (2*(l1*l3*exp(phir(j)*1i)*1i - l3*r*exp(phir(j)*2i)*...
                exp(theta1P(j,i)*1i)*1i)))*1i) - l1*cos(theta1P(j,i)) +...
                r*cos(phir(j)))/l2));
        end
    end
end

end

%if the angle of the third segment is smaller than -90 deg...
%...do a phase shift of 360 deg
if theta3(j,i) < -pi/2
    theta3(j,i) = theta3(j,i) + 2*pi;
end

%evaluate the deviations in x and y again, respectively
DEV1(j,i) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i)) + l3*sin(theta3(j,i)) -...
    r*sin(alpha(j));
DEV2(j,i) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i)) + l3*cos(theta3(j,i)) -...
    r*cos(alpha(j));

d(j,i) = sqrt((r*sin(alpha(j))-l1*cos(theta1(j,i)))^2 +...
    (r*cos(alpha(j))-l1*sin(theta1(j,i)))^2);

%check condition upper loop closure
if l3-l2+d(j,i) < 0
    %mark error with variable "Count2"
    Count2 = Count2 + 1;
end

%check condition upper loop closure
if (l3-l2-d(j,i)) > 0
    %mark error with variable "Count2"
    Count2 = Count2 + 1;
end

%if any of these deviations transcends a certain treshold then throw an error
if abs(DEV1(j,i)) > 10^-10 || abs(DEV2(j,i)) > 10^-10
        Count = Count + 1;
end

%initial relative angle of segment 1
alpha10 = theta10;
%initial relative angle of segment 2
alpha20 = theta20 - theta10;
%initial relative angle of segment 3
alpha30 = theta30 - theta20;

%angle of rotation torsion spring 1
alpha1(j,i) = theta1(j,i) - alpha10;
%angle of rotation torsion spring 2
alpha2(j,i) = theta2(j,i) - theta1(j,i) - alpha20;
%angle of rotation torsion spring 3
alpha3(j,i) = theta3(j,i) - theta2(j,i) - alpha30;

if prestress == 0 && nonlinearity == 0
    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i) = ((k2/2)*alpha2(j,i)^2);
    %potential energy spring 3
    V3(j,i) = ((k3/2)*alpha3(j,i)^2);
    %total potential energy
    V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);


    %the spring moment in spring 1
    M1(j,i) = k1*alpha1(j,i);
    %the spring moment in spring 2
```

```
    M2(j,i) = k2*alpha2(j,i);
    %the spring moment in spring 3
    M3(j,i) = k3*alpha3(j,i);
end

%if springs are prestressed
if prestress == 1 && nonlinearity == 0
    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i) = ((k2/2)*alpha2(j,i)^2) + M0*alpha2(j,i) + ((k2/2)*(M0/k2)^2);
    %potential energy spring 3
    V3(j,i) = ((k3/2)*alpha3(j,i)^2);
    %total potential energy
    V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);


    %the spring moment in spring 1
    M1(j,i) = k1*alpha1(j,i);
    %the spring moment in spring 2
    M2(j,i) = k2*alpha2(j,i) + M0;
    %the spring moment in spring 3
    M3(j,i) = k3*alpha3(j,i);
end

%if springs are nonlinear
if prestress == 0 && nonlinearity == 1
    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
    V2(j,i) = (A/3)*alpha2(j,i)^3 + (B/2)*alpha2(j,i)^2;
    %potential energy spring 3
    V3(j,i) = (A/3)*alpha3(j,i)^3 + (B/2)*alpha3(j,i)^2;
    %total potential energy
    V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);
end

%if springs are prestressed and nonlinear
if prestress == 1 && nonlinearity == 1
    %first solution prestress angle: angle of rotation corresponding to prestress
    alphastar1 = (-B + sqrt(B^2 + 4*M0*A))/(2*A);
    %second solution prestress angle: angle of rotation corresponding to prestress
    alphastar2 = (-B - sqrt(B^2 + 4*M0*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1 < 0
        alphastar1 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2 < 0
        alphastar2 = NaN;
    end

    %store solutions prestress angle in array called "alphastars"
    alphastars = [alphastar1,alphastar2];

    %store the smallest solution for the prestress angle
    alphastar = min(abs(alphastars));

    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
    V2(j,i) = (A/3)*(alpha2(j,i)+alphastar)^3 + (B/2)*(alpha2(j,i)+alphastar)^2;
    %potential energy spring 3
    V3(j,i) = (A/3)*alpha3(j,i)^3 + (B/2)*alpha3(j,i)^2;
    %total potential energy springs
    V(j,i) = V1(j,i) + V2(j,i) + V3(j,i);


    %the spring moment in spring 1
    M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
    %the spring moment in spring 2
    M2(j,i) = A*(alpha2(j,i)+alphastar)^2 + B*(alpha2(j,i)+alphastar);
    %the spring moment in spring 3
    M3(j,i) = A*alpha3(j,i)^2 + B*alpha3(j,i);
end

%coordinates of nodes
%x - coordinate origin (and first spring)
x0 = 0;
%y - coordinate origin (and first spring)
y0 = 0;
%x - coordinate 2nd spring
```

```
x1(j,i) = l1*sin(theta1(j,i));
%y - coordinate 2nd spring
y1(j,i) = l1*cos(theta1(j,i));
%x - coordinate 3rd spring
x2(j,i) = x1(j,i) + l2*sin(theta2(j,i));
%y - coordinate 3rd spring
y2(j,i) = y1(j,i) + l2*cos(theta2(j,i));
%x - coordinate end effector
x3(j,i) = x2(j,i) + l3*sin(theta3(j,i));
%y - coordinate end effector
y3(j,i) = y2(j,i) + l3*cos(theta3(j,i));

%do the plotting for the initial configuration
%x - coordinate origin (and first spring)
x00 = 0;
%y - coordinate origin (and first spring)
y00 = 0;
%x - coordinate 2nd spring
x10 = l1*sin(theta10);
%y - coordinate 2nd spring
y10 = l1*cos(theta10);
%x - coordinate 3rd spring
x20 = x10 + l2*sin(theta20);
%y - coordinate 3rd spring
y20 = y10 + l2*cos(theta20);
%x - coordinate end effector
x30 = x20 + l3*sin(theta30);
%y - coordinate end effector
y30 = y20 + l3*cos(theta30);

%vertical reaction force at segment 1 (positive upwards)
F1yt(j,i) = (M1(j,i) - M3(j,i) - (M3(j,i)/(l3*cos(theta3(j,i))))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i))))/...
    ((l1*sin(theta1(j,i))+l2*sin(theta2(j,i))) - tan(theta3(j,i))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i))));

%horizontal reaction force at segment 1 (positive to the right)
F1xt(j,i) = (-M3(j,i) + F1yt(j,i)*l3*sin(theta3(j,i)))/(l3*cos(theta3(j,i)));

%the load (moment) on nodes 2 and 3 (where springs 2 and 3 are located), respectively
M2lt(j,i) = M1(j,i) + F1xt(j,i)*l1*cos(theta1(j,i)) - F1yt(j,i)*l1*sin(theta1(j,i));
M3lt(j,i) = M1(j,i) + F1xt(j,i)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i))) -...
    F1yt(j,i)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i)));

if prestress == 1
    %if the load (moment) on node 2 transcends the pretension...
    %...the spring will be activated
    if M2lt(j,i) > M0
        activation = 1;
    end
end

end
end

%find minimum potential energy and it's corresponding index
[Vmin,I] = min(V,[],2);
figure(1)                                               %create figure
%plot following plot commands in that same figure
hold on
axis equal
title('Lowest energy configurations')

%start a loop throughout all precision points
for j = 1:1:M
%divide the 90 deg range of motion into equally sized segments
alpha(j) = (pi/2)*(j/M);

%plot connection line between spring 1 and 2 in black
plot([x0 x1(j,I(j))],[y0 y1(j,I(j))],'k')
%plot connection line between spring 2 and 3 in black
plot([x1(j,I(j)) x2(j,I(j))],[y1(j,I(j)) y2(j,I(j))],'k')
%plot connection line between spring 3 and end effector in black
plot([x2(j,I(j)) x3(j,I(j))],[y2(j,I(j)) y3(j,I(j))],'k')
%plot the location of the end effector of the pendulum with a circle
plot(r*sin(alpha(j)),r*cos(alpha(j)),'b--o')

%plot connection line between spring 1 and 2 in black
plot([x00 x10],[y00 y10],'red')
%plot connection line between spring 2 and 3 in black
plot([x10 x20],[y10 y20],'red')
%plot connection line between spring 3 and end effector in black
plot([x20 x30],[y20 y30],'red')
%plot the location of the end effector of the pendulum with a circle
```

```
plot(r*sin(0),r*cos(0),'r--o')

%angle of spring 1 w.r.t. vertical, corresponding to equilibrium
theta1m(j) = theta1(j,I(j));
%angle of spring 2 w.r.t. vertical, corresponding to equilibrium
theta2m(j) = theta2(j,I(j));
%angle of spring 3 w.r.t. vertical, corresponding to equilibrium
theta3m(j) = theta3(j,I(j));

%deformation angle of spring 1, corresponding to equilibrium
alpha1m(j) = alpha1(j,I(j));
%deformation angle of spring 2, corresponding to equilibrium
alpha2m(j) = alpha2(j,I(j));
%deformation angle of spring 3, corresponding to equilibrium
alpha3m(j) = alpha3(j,I(j));


if prestress == 0 && nonlinearity == 0
    %internal moment spring 1, corresponding to equilibrium
    M1m(j) = k1*alpha1(j,I(j));
    %internal moment spring 2, corresponding to equilibrium
    M2m(j) = k2*alpha2(j,I(j));
    %internal moment spring 3, corresponding to equilibrium
    M3m(j) = k3*alpha3(j,I(j));


    %potential energy spring 1, corresponding to equilibrium
    V1m(j) = V1(j,I(j));
    %potential energy spring 2, corresponding to equilibrium
    V2m(j) = V2(j,I(j));
    %potential energy spring 3, corresponding to equilibrium
    V3m(j) = V3(j,I(j));
    %total potential energy springs, corresponding to equilibrium
    Vtm(j) = V1m(j) + V2m(j) + V3m(j);
end

%if springs are prestressed
if prestress == 1 && nonlinearity == 0
    %internal moment spring 1, corresponding to equilibrium
    M1m(j) = k1*alpha1(j,I(j));
    %internal moment spring 2, corresponding to equilibrium
    M2m(j) = k2*alpha2(j,I(j)) + M0;
    %internal moment spring 3, corresponding to equilibrium
    M3m(j) = k3*alpha3(j,I(j));
end

%if springs are nonlinear
if prestress == 0 && nonlinearity == 1
    %internal moment spring 1, corresponding to equilibrium
    M1m(j) = A*alpha1(j,I(j))^2 + B*alpha1(j,I(j));
    %internal moment spring 2, corresponding to equilibrium
    M2m(j) = A*alpha2(j,I(j))^2 + B*alpha2(j,I(j));
    %internal moment spring 3, corresponding to equilibrium
    M3m(j) = A*alpha3(j,I(j))^2 + B*alpha3(j,I(j));


    %potential energy spring 1, corresponding to equilibrium
    V1m(j) = (A/3)*alpha1(j,I(j))^3 + (B/2)*alpha1(j,I(j))^2;
    %potential energy spring 2, corresponding to equilibrium
    V2m(j) = (A/3)*alpha2(j,I(j))^3 + (B/2)*alpha2(j,I(j))^2;
    %potential energy spring 3, corresponding to equilibrium
    V3m(j) = (A/3)*alpha3(j,I(j))^3 + (B/2)*alpha3(j,I(j))^2;
end

%if springs are prestressed and nonlinear
if prestress == 1 && nonlinearity == 1
    %internal moment spring 1, corresponding to equilibrium
    M1m(j) = A*alpha1(j,I(j))^2 + B*alpha1(j,I(j));
    %internal moment spring 2, corresponding to equilibrium
    M2m(j) = A*(alpha2(j,I(j))+alphastar)^2 + B*(alpha2(j,I(j))+alphastar);
    %internal moment spring 3, corresponding to equilibrium
    M3m(j) = A*alpha3(j,I(j))^2 + B*alpha3(j,I(j));
end

%vertical reaction force at segment 1 (positive upwards)
F1y(j) = (M1m(j) - M3m(j) - (M3m(j)/(l3*cos(theta3(j,I(j))))))*...
    (l1*cos(theta1(j,I(j)))+l2*cos(theta2(j,I(j)))))/...
    ((l1*sin(theta1(j,I(j)))+l2*sin(theta2(j,I(j)))) - tan(theta3(j,I(j)))*...
    (l1*cos(theta1(j,I(j)))+l2*cos(theta2(j,I(j)))));

%horizontal reaction force at segment 1 (positive to the right)
F1x(j) = (-M3m(j) + F1y(j)*l3*sin(theta3(j,I(j))))/(l3*cos(theta3(j,I(j))));

T1b(j) = ((alpha(j)-pi/2)*180/pi);
```

```
Tub(j) = ((alpha(j)+pi/2)*180/pi);

%the external load (moment) on nodes 1, 2 and 3...
%...(where springs 1, 2 and 3 are located), respectively
M1l(j) = F1y(j)*r*sin(alpha(j)) - F1x(j)*r*cos(alpha(j));
M2l(j) = M1m(j) + F1x(j)*l1*cos(theta1(j,I(j))) - F1y(j)*l1*sin(theta1(j,I(j)));
M3l(j) = M1m(j) + F1x(j)*(l1*cos(theta1(j,I(j)))+l2*cos(theta2(j,I(j)))) -...
    F1y(j)*(l1*sin(theta1(j,I(j)))+l2*sin(theta2(j,I(j))));

%objective moment-angle characteristics
if objective == "sinus"
    Vm(j) = mg*r*cos(alpha(j));           %original value
    Mobj(j) = mg*r*sin(alpha(j));         %original value
end

if objective == "Laevo"
    Vm(j) = (0.05022*alpha(j)^5 - 0.33575*alpha(j)^4 + 0.97*alpha(j)^3 -...
        1.412*alpha(j)^2 + 0.006501*alpha(j) + 1);
    Mobj(j) = (-0.2511*alpha(j)^4 + 1.343*alpha(j)^3 - 2.91*alpha(j)^2 +...
        2.824*alpha(j) - 0.006501);
end

if objective == "stiffening"
    Vm(j) = sin(alpha(j)) - alpha(j);
    Mobj(j) = -cos(alpha(j))+1;
end

if objective == "sqrt"
    Vm(j) = - (2/3)*alpha(j)^(3/2);
    Mobj(j) = sqrt(alpha(j));
end

if objective == "quadratic"
    Vm(j) = - (1/3)*alpha(j)^(3);
    Mobj(j) = alpha(j)^2;
end

if objective == "hardening-softening"
    Vm(j) = 0.25*cos(2*alpha(j)-pi/2) - 0.5*alpha(j);
    Mobj(j) = (sin(2*alpha(j)-pi/2)+1)/2;
end

if objective == "hardening-softening2"
    Vm(j) = -0.5*alpha(j)+(-0.333333+0.424413*alpha(j))*atan(2.41421-3.07387...
        *alpha(j))+0.0690356*log(9.8696-21.4521*alpha(j)+13.6569*alpha(j)^2);
    Mobj(j) = 0.5 + (4/(3*pi))*atan(tan((3*pi)/8)*((4/pi)*alpha(j)-1));
end

if objective == "softening-hardening"
    Vm(j) = 0.5*log(cos(alpha(j)-pi/4)) - 0.5*alpha(j);
    Mobj(j) = 0.5*tan(alpha(j)-pi/4) + 0.5;
end

if objective == "softening-hardening2"
    Vm(j) = -0.5*alpha(j) - 0.0690356*log(1 + tan(1.1781 - 1.5*alpha(j))^2);
    Mobj(j) = 0.5*tan(1.5*(alpha(j)-pi/4))/tan(1.5*(pi/4))+0.5;
end

if objective == "sinuspi"
    Vm(j) = 0.5*cos(2*alpha(j));
    Mobj(j) = sin(2*alpha(j));
end

end

%print the root mean square error (objective function)
e = sqrt(mean((M1m - Mobj).^2)) %#ok<NOPTS>
vd = mean(sqrt((V1m-V2m).^2 + (V1m-V3m).^2 + (V2m-V3m).^2));

IntM = trapz(alpha,abs(M1m-Mobj));
IntMo = trapz(alpha,Mobj);

figure(2)
hold on
plot(alpha*180/pi,M1m)
plot(alpha*180/pi,Mobj)
plot(alpha*180/pi,M1m - Mobj)
xlabel('Angle of rotation pendulum from vertical (deg)')
ylabel('Moment around suspension-point 1 (Nm)')
legend('Moment in spring 1','Objective moment','Error in moment',...
    'location','northwest')

figure(3)
hold on
```

```
plot(alpha*180/pi,Vmin+transpose(Vm))
xlabel('Angle of rotation pendulum from vertical (deg)')
ylabel('Total potential energy in system (J)')

figure(4)
hold on
plot(alpha*180/pi,V1m)
plot(alpha*180/pi,V2m)
plot(alpha*180/pi,V3m)
xlabel('Angle of rotation pendulum from vertical (deg)')
ylabel('Energy storage in springs (J)')
legend('Energy spring 1','Energy spring 2','Energy spring 3','location','northwest')

figure(5)
hold on
plot(alpha*180/pi,M2m)
plot(alpha*180/pi,M3m)
plot(alpha*180/pi,M21)
plot(alpha*180/pi,M31)
xlabel('Angle of rotation pendulum from vertical (deg)')
ylabel('Reaction moments in springs (Nm)')
legend('M2m','M3m','M21','M31','location','northwest')

figure(6)
hold on
plot(alpha*180/pi,F1x)
plot(alpha*180/pi,F1y)
xlabel('Angle of rotation pendulum from vertical (deg)')
ylabel('Reaction force in point 1 (N)')
legend('F1x','F1y','location','northwest')

figure(7)
hold on
plot(alpha*180/pi,alpha1m*180/pi)
xlabel('Angle of rotation pendulum from vertical (deg)')
ylabel('Angle of rotation spring 1 (deg)')
```

N.4. Four segment balancer

```
clc                                            %clear command window
clear variables                                %empty workspace
close all                                      %close all windows

%with (1) or without (0) prestress on springs
prestress = 1;
%with (1) or without (0) nonlinear springs
nonlinearity = 0;
%type objective function between quotation marks...
objective = "sinus";

M = 15;                                        %amount of precision points

%amount of configurations of segment 1 per precision point
N1 = 150;
%number of configurations of segment 2 per precision point
N2 = 150;

k1 = 0.971;                                  %stiffness spring 1 (Nm/rad)
k2 = 0.105;                                  %stiffness spring 2 (Nm/rad)
k3 = 0.310;                                  %stiffness spring 3 (Nm/rad)
k4 = 0;                                      %stiffness spring 4 (Nm/rad)
M02 = 0.596;                                 %preload spring 2 (Nm)
M03 = 0.225;                                 %preload spring 3 (Nm)
l1 = 0.29;                                   %length first segment (m)
l2 = 0.29;                                   %length second segment (m)
l3 = 0.29;                                   %length third segment (m)
l4 = 0.29;                                   %length fourth segment (m)

%length pendulum (m)
r = 1;
%constant mass times grav. constant (N)
mg = 1;

Count = 0;                                     %error counter
Count2 = 0;                                    %second error counter
Count3 = 0;                                    %third error counter
%-----------------------------------------------------------------------------
%definition of angles in the initial (relaxed) configuration
theta1i = 0*pi/180;          %independent variable: initial angle segment 1
%dependent angles initial configuration
theta1ni = -theta1i;
Ari = (pi/2);
A1i = (pi/2) - theta1i;
```

```
theta1pi = theta1i - (pi/2);

%if initial angle of first segment is greater than zero or equal to zero
if theta1i >= 0
    %initial upperbound angle segment 2
    theta2fi = (pi/2) - real(-log((- ((- l1^2*exp(A1i*1i)*exp(Ari*1i) +...
        l2^2*exp(A1i*1i)*exp(Ari*1i) + l3^2*exp(A1i*1i)*exp(Ari*1i) +...
        l4^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
        l1*r*exp(A1i*2i) + l1*r*exp(Ari*2i) -...
        2*l2*l3*exp(A1i*1i)*exp(Ari*1i) - 2*l2*l4*exp(A1i*1i)*exp(Ari*1i) +...
        2*l3*l4*exp(A1i*1i)*exp(Ari*1i))*(- l1^2*exp(A1i*1i)*exp(Ari*1i) +...
        l2^2*exp(A1i*1i)*exp(Ari*1i) + l3^2*exp(A1i*1i)*exp(Ari*1i) +...
        l4^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
        l1*r*exp(A1i*2i) + l1*r*exp(Ari*2i) +...
        2*l2*l3*exp(A1i*1i)*exp(Ari*1i) + 2*l2*l4*exp(A1i*1i)*exp(Ari*1i) +...
        2*l3*l4*exp(A1i*1i)*exp(Ari*1i)))^(1/2) -...
        l1^2*exp(A1i*1i)*exp(Ari*1i) - l2^2*exp(A1i*1i)*exp(Ari*1i) +...
        l3^2*exp(A1i*1i)*exp(Ari*1i) + l4^2*exp(A1i*1i)*exp(Ari*1i) -...
        r^2*exp(A1i*1i)*exp(Ari*1i) + l1*r*exp(A1i*2i) + l1*r*exp(Ari*2i) +...
        2*l3*l4*exp(A1i*1i)*exp(Ari*1i))/...
        (2*(l1*l2*exp(Ari*1i) - l2*r*exp(A1i*1i))))*1i);
    %initial lowerbound angle segment 2
    theta20i = (pi/2) - real(-log((((- l1^2*exp(A1i*1i)*exp(Ari*1i) +...
        l2^2*exp(A1i*1i)*exp(Ari*1i) + l3^2*exp(A1i*1i)*exp(Ari*1i) +...
        l4^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
        l1*r*exp(A1i*2i) + l1*r*exp(Ari*2i) -...
        2*l2*l3*exp(A1i*1i)*exp(Ari*1i) - 2*l2*l4*exp(A1i*1i)*exp(Ari*1i) +...
        2*l3*l4*exp(A1i*1i)*exp(Ari*1i))*(- l1^2*exp(A1i*1i)*exp(Ari*1i) +...
        l2^2*exp(A1i*1i)*exp(Ari*1i) + l3^2*exp(A1i*1i)*exp(Ari*1i) +...
        l4^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
        l1*r*exp(A1i*2i) + l1*r*exp(Ari*2i) +...
        2*l2*l3*exp(A1i*1i)*exp(Ari*1i) + 2*l2*l4*exp(A1i*1i)*exp(Ari*1i) +...
        2*l3*l4*exp(A1i*1i)*exp(Ari*1i)))^(1/2) -...
        l1^2*exp(A1i*1i)*exp(Ari*1i) -...
        l2^2*exp(A1i*1i)*exp(Ari*1i) + l3^2*exp(A1i*1i)*exp(Ari*1i) +...
        l4^2*exp(A1i*1i)*exp(Ari*1i) - r^2*exp(A1i*1i)*exp(Ari*1i) +...
        l1*r*exp(A1i*2i) + l1*r*exp(Ari*2i) +...
        2*l3*l4*exp(A1i*1i)*exp(Ari*1i))/...
        (2*(l1*l2*exp(Ari*1i) - l2*r*exp(A1i*1i))))*1i);
end

%if initial angle of first segment is smaller than zero
if theta1i < 0
    %initial upperbound angle segment 2
    theta2fi = real(-log(-(((l1*r - l1^2*exp(0*1i)*exp(theta1ni*1i) +...
        l2^2*exp(0*1i)*exp(theta1ni*1i) + l3^2*exp(0*1i)*exp(theta1ni*1i) +...
        l4^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) -...
        2*l2*l3*exp(0*1i)*exp(theta1ni*1i) -...
        2*l2*l4*exp(0*1i)*exp(theta1ni*1i) +...
        2*l3*l4*exp(0*1i)*exp(theta1ni*1i) +...
        l1*r*exp(0*2i)*exp(theta1ni*2i))*...
        (l1*r - l1^2*exp(0*1i)*exp(theta1ni*1i) +...
        l2^2*exp(0*1i)*exp(theta1ni*1i) + l3^2*exp(0*1i)*exp(theta1ni*1i) +...
        l4^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) +...
        2*l2*l3*exp(0*1i)*exp(theta1ni*1i) +...
        2*l2*l4*exp(0*1i)*exp(theta1ni*1i) +...
        2*l3*l4*exp(0*1i)*exp(theta1ni*1i) +...
        l1*r*exp(0*2i)*exp(theta1ni*2i)))^(1/2) +...
        l1*r - l1^2*exp(0*1i)*exp(theta1ni*1i) -...
        l2^2*exp(0*1i)*exp(theta1ni*1i) + l3^2*exp(0*1i)*exp(theta1ni*1i) +...
        l4^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) +...
        2*l3*l4*exp(0*1i)*exp(theta1ni*1i) + l1*r*exp(0*2i)*exp(theta1ni*2i))/...
        (2*(l2*r*exp(theta1ni*1i) - l1*l2*exp(0*1i)*exp(theta1ni*2i))))*1i);
    %initial lowerbound angle segment 2
    theta20i = real(-log(-(l1*r - ((l1*r - l1^2*exp(0*1i)*exp(theta1ni*1i) +...
        l2^2*exp(0*1i)*exp(theta1ni*1i) + l3^2*exp(0*1i)*exp(theta1ni*1i) +...
        l4^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) -...
        2*l2*l3*exp(0*1i)*exp(theta1ni*1i) -...
        2*l2*l4*exp(0*1i)*exp(theta1ni*1i) +...
        2*l3*l4*exp(0*1i)*exp(theta1ni*1i) +...
        l1*r*exp(0*2i)*exp(theta1ni*2i))*...
        (l1*r - l1^2*exp(0*1i)*exp(theta1ni*1i) +...
        l2^2*exp(0*1i)*exp(theta1ni*1i) + l3^2*exp(0*1i)*exp(theta1ni*1i) +...
        l4^2*exp(0*1i)*exp(theta1ni*1i) - r^2*exp(0*1i)*exp(theta1ni*1i) +...
        2*l2*l3*exp(0*1i)*exp(theta1ni*1i) +...
        2*l2*l4*exp(0*1i)*exp(theta1ni*1i) +...
        2*l3*l4*exp(0*1i)*exp(theta1ni*1i) +...
        l1*r*exp(0*2i)*exp(theta1ni*2i)))^(1/2) -...
        l1^2*exp(0*1i)*exp(theta1ni*1i) - l2^2*exp(0*1i)*exp(theta1ni*1i) +...
        l3^2*exp(0*1i)*exp(theta1ni*1i) + l4^2*exp(0*1i)*exp(theta1ni*1i) -...
        r^2*exp(0*1i)*exp(theta1ni*1i) + 2*l3*l4*exp(0*1i)*exp(theta1ni*1i) +...
        l1*r*exp(0*2i)*exp(theta1ni*2i))/...
        (2*(l2*r*exp(theta1ni*1i) - l1*l2*exp(0*1i)*exp(theta1ni*2i))))*1i);
end
```

```
%initial angle of the second segment
theta2i = 0;

%initial angle of the second segment, (CCW positive) with respect to the
%positive x-axis
A2i = (pi/2) - theta2i;

%the length of the imaginary connection line between the origin and the
%node at the end of the second segment...
l12i = sqrt((l1*sin(theta1i) + l2*sin(theta2i))^2 +...
    (l1*cos(theta1i) + l2*cos(theta2i))^2);
%...and its angle with respect to the vertical
phi12i = (pi/2) - atan((l1*sin(theta1i) +...
    l2*sin(theta2i))/(l1*cos(theta1i) + l2*cos(theta2i)));

Mtheta12i = - atan((l1*sin(theta1i) + l2*sin(theta2i))/...
    (l1*cos(theta1i) + l2*cos(theta2i)));

%the angle of the third segment, corresponding to the system in its initial
%configuration
theta3i = pi/2 - real(pi - acos((l12i*cos(phi12i) - r*cos(Ari) +...
    l4*cos(log(-(((l12i*r*exp(Ari*2i) + l12i*r*exp(phi12i*2i) -...
    l12i^2*exp(Ari*1i)*exp(phi12i*1i) + l3^2*exp(Ari*1i)*exp(phi12i*1i) +...
    l4^2*exp(Ari*1i)*exp(phi12i*1i) - r^2*exp(Ari*1i)*exp(phi12i*1i) -...
    2*l3*l4*exp(Ari*1i)*exp(phi12i*1i))*(l12i*r*exp(Ari*2i) +...
    l12i*r*exp(phi12i*2i) - l12i^2*exp(Ari*1i)*exp(phi12i*1i) +...
    l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) -...
    r^2*exp(Ari*1i)*exp(phi12i*1i) +...
    2*l3*l4*exp(Ari*1i)*exp(phi12i*1i)))^(1/2) - l12i*r*exp(Ari*2i) -...
    l12i*r*exp(phi12i*2i) + l12i^2*exp(Ari*1i)*exp(phi12i*1i) -...
    l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) +...
    r^2*exp(Ari*1i)*exp(phi12i*1i))/...
    (2*(l12i*l4*exp(Ari*1i) - l4*r*exp(phi12i*1i))))*1i))/l3));

%the angle of the fourth segment, corresponding to the system in its initial
%configuration
theta4i = pi/2 - real(-log(-(((l12i*r*exp(Ari*2i) + l12i*r*exp(phi12i*2i) -...
    l12i^2*exp(Ari*1i)*exp(phi12i*1i) + l3^2*exp(Ari*1i)*exp(phi12i*1i) +...
    l4^2*exp(Ari*1i)*exp(phi12i*1i) - r^2*exp(Ari*1i)*exp(phi12i*1i) -...
    2*l3*l4*exp(Ari*1i)*  exp(phi12i*1i))*(l12i*r*exp(Ari*2i) +...
    l12i*r*exp(phi12i*2i) - l12i^2*exp(Ari*1i)*exp(phi12i*1i) +...
    l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) -...
    r^2*exp(Ari*1i)*exp(phi12i*1i) +...
    2*l3*l4*exp(Ari*1i)*exp(phi12i*1i)))^(1/2) - l12i*r*exp(Ari*2i) -...
    l12i*r*exp(phi12i*2i) + l12i^2*exp(Ari*1i)*exp(phi12i*1i) -...
    l3^2*exp(Ari*1i)*exp(phi12i*1i) + l4^2*exp(Ari*1i)*exp(phi12i*1i) +...
    r^2*exp(Ari*1i)*exp(phi12i*1i))/...
    (2*(l12i*l4*exp(Ari*1i) - l4*r*exp(phi12i*1i))))*1i);

%if the node at the end of the second segment is located left to the y-axis
%in the initial configuration
if (l1*sin(theta1i) + l2*sin(theta2i)) < 0
    %define alternative formulation initial angle segment 3
    theta3i =  real(asin((l4*sin(log(-(l12i*r +...
        ((l12i*r - l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l4^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        r^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        2*l3*l4*exp(Mtheta12i*1i)*exp(0*1i) +...
        l12i*r*exp(Mtheta12i*2i)*exp(0*2i))*(l12i*r -...
        l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l4^2*exp(Mtheta12i*1i)*exp(0*1i) - r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        2*l3*l4*exp(Mtheta12i*1i)*exp(0*1i) +...
        l12i*r*exp(Mtheta12i*2i)*exp(0*2i))).^(1/2) -...
        l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l3^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        l4^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l12i*r*exp(Mtheta12i*2i)*exp(0*2i))/(2*(l4*r*exp(Mtheta12i*1i) -...
        l12i*l4*exp(Mtheta12i*2i)*exp(0*1i))))*1i) +...
        l12i*sin(Mtheta12i) + r*sin(0))/l3));

    %define alternative formulation initial angle segment 4
    theta4i =  real(-log(-(l12i*r + ((l12i*r - l12i^2*exp(Mtheta12i*1i)*...
        exp(0*1i) + l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l4^2*exp(Mtheta12i*1i)*exp(0*1i) - r^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        2*l3*l4*exp(Mtheta12i*1i)*exp(0*1i) +...
        l12i*r*exp(Mtheta12i*2i)*exp(0*2i))*...
        (l12i*r - l12i^2*exp(Mtheta12i*1i)*...
        exp(0*1i) + l3^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l4^2*exp(Mtheta12i*1i)*...
        exp(0*1i) - r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
```

```
        2*l3*l4*exp(Mtheta12i*1i)*...
        exp(0*1i) + l12i*r*exp(Mtheta12i*2i)*exp(0*2i)))^(1/2) -...
        l12i^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l3^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        l4^2*exp(Mtheta12i*1i)*exp(0*1i) -...
        r^2*exp(Mtheta12i*1i)*exp(0*1i) +...
        l12i*r*exp(Mtheta12i*2i)*exp(0*2i))/(2*(l4*r*exp(Mtheta12i*1i) -...
        l12i*l4*exp(Mtheta12i*2i)*exp(0*1i))))*1i);
end
%-----------------------------------------------------------------------------
%preallocate all variables for better performance...
%...vectors
alpha = zeros(1,M);
theta100 = zeros(1,M);
theta1ff = zeros(1,M);
theta100pa = zeros(1,M);
theta1ffpa = zeros(1,M);
theta100pa2 = zeros(1,M);
theta1ffpa2 = zeros(1,M);
BEGIN1 = zeros(1,M);
END1 = zeros(1,M);
STEP1 = zeros(1,M);
BEGIN1pa = zeros(1,M);
END1pa = zeros(1,M);
STEP1pa = zeros(1,M);
BEGIN1pa2 = zeros(1,M);
END1pa2 = zeros(1,M);
STEP1pa2 = zeros(1,M);
Ar = zeros(1,M);
alpha1m = zeros(1,M);
alpha2m = zeros(1,M);
alpha3m = zeros(1,M);
alpha4m = zeros(1,M);
theta1m = zeros(1,M);
theta2m = zeros(1,M);
theta3m = zeros(1,M);
theta4m = zeros(1,M);
M1m = zeros(1,M);
M2m = zeros(1,M);
M3m = zeros(1,M);
M4m = zeros(1,M);
Mobj = zeros(1,M);
F1x = zeros(1,M);
F1y = zeros(1,M);
M2l = zeros(1,M);
M3l = zeros(1,M);
Vm = zeros(1,M);
fit = zeros(1,M);

%preallocation of matrices (number of rows = M, number of columns = N1)
theta1 = zeros(M,N1);
theta1n = zeros(M,N1);
theta2f = zeros(M,N1);
theta20 = zeros(M,N1);
BEGIN2 = zeros(M,N1);
END2 = zeros(M,N1);
STEP2 = zeros(M,N1);
A1 = zeros(M,N1);
theta1p = zeros(M,N1);
theta1sw = zeros(M,N1);
theta1sw2 = zeros(M,N1);
theta1fa = zeros(M,N1);
alpha1 = zeros(M,N1);
V1 = zeros(M,N1);
M1 = zeros(M,N1);
x1 = zeros(M,N1);
y1 = zeros(M,N1);
passedX = zeros(M,N1);
theta23 = zeros(M,N1);

%preallocation of tensors (number of rows = M, number of columns = N1,
%number of "pages" = N2)
theta2 = zeros(M,N1,N2);
theta3 = zeros(M,N1,N2);
theta4 = zeros(M,N1,N2);
DEV1 = zeros(M,N1,N2);
DEV2 = zeros(M,N1,N2);
DEV11 = zeros(M,N1,N2);
DEV22 = zeros(M,N1,N2);
A2 = zeros(M,N1,N2);
alpha2 = zeros(M,N1,N2);
alpha3 = zeros(M,N1,N2);
alpha4 = zeros(M,N1,N2);
M2 = zeros(M,N1,N2);
```

```
M3 = zeros(M,N1,N2);
M4 = zeros(M,N1,N2);
F1xt = zeros(M,N1,N2);
F1yt = zeros(M,N1,N2);
M2lt = zeros(M,N1,N2);
M3lt = zeros(M,N1,N2);
V2 = zeros(M,N1,N2);
V3 = zeros(M,N1,N2);
V4 = zeros(M,N1,N2);
V = zeros(M,N1,N2);
x2 = zeros(M,N1,N2);
y2 = zeros(M,N1,N2);
x3 = zeros(M,N1,N2);
y3 = zeros(M,N1,N2);
x4 = zeros(M,N1,N2);
y4 = zeros(M,N1,N2);
Mtheta12 = zeros(M,N1,N2);
theta12P = zeros(M,N1,N2);
phi12 = zeros(M,N1,N2);
phi12v = zeros(M,N1,N2);
l12 = zeros(M,N1,N2);
d = zeros(M,N1,N2);

%-----------------------------------------------------------------------------
% figure(1)
% hold on
% axis equal

%start a loop throughout all precision points
for j = 1:1:M
%divide the 90 deg range of motion into equally sized segments
alpha(j) = (pi/2)*(j/M);

%lowerbound of theta1 such that precision point (j) is still reached
theta100(j) = alpha(j) - acos((r^2 + l1^2 - (l2 + l3 + l4)^2)/(2*r*l1));
%upperbound of theta1 such that precision point (j) is still reached
theta1ff(j) = alpha(j) + acos((r^2 + l1^2 - (l2 + l3 + l4)^2)/(2*r*l1));

%check whether segment 2,3 and 4 can be aligned (stretched arm)
if (l1+r) <= (l2+l3+l4)
    %alternative formulation lowerbound of theta1 if arm cannot be
    %stretched
    theta100(j) = alpha(j) - pi;
    %alternative formulation upperbound of theta1 if arm cannot be
    %stretched
    theta1ff(j) = alpha(j) + pi;
end

BEGIN1(j) = theta100(j);                                %begin interval
END1(j) = theta1ff(j);                                  %end interval
STEP1(j) = (END1(j)-BEGIN1(j))/N1;                      %stepsize

%inserted code needed for prestress here
if prestress == 1
    %lowerbound of theta1
    theta100pa(j) = alpha(j) - acos((r^2+(l1+l2)^2 -...
        (l3+l4)^2)/(2*r*(l1+l2)));
    %upperbound of theta1
    theta1ffpa(j) = alpha(j) + acos((r^2+(l1+l2)^2 -...
        (l3+l4)^2)/(2*r*(l1+l2)));
    %define boundaries segment 1 sweep
    BEGIN1pa(j) = theta100pa(j);
    END1pa(j) = theta1ffpa(j);
    %define stepsize segment 1 sweep
    STEP1pa(j) = (END1pa(j)-BEGIN1pa(j))/N1;

    %calculate angle between segment 2 and 3
    psi23 = pi - abs(theta3i-theta2i);
    %calculate length of imaginary connection line between begin of
    %segment 2 and end of segment 3
    l23 = sqrt(l2^2 + l3^2 - 2*l2*l3*cos(psi23));

    %lower - and upperbound of segment 1 when spring 2 is engaged and
    %spring 3 is still making contact with the environment
    theta100pa2(j) = alpha(j) - acos((l1^2+r^2 - (l23+l4)^2)/(2*r*l1));
    theta1ffpa2(j) = alpha(j) + acos((l1^2+r^2 - (l23+l4)^2)/(2*r*l1));

    %check whether the imaginary connection line "l23" and segment 4
    %can be aligned (stretched arm)
    if (l1+r) <= (l23+l4)
        %alternative formulation lowerbound of theta1 if arm cannot be
        %stretched
        theta100pa2(j) = alpha(j) - pi;
        %alternative formulation upperbound of theta1 if arm cannot be
```

```
        %stretched
        theta1ffpa2(j) = alpha(j) + pi;
    end

    %define boundaries segment 1 sweep
    BEGIN1pa2(j) = theta100pa2(j);
    END1pa2(j) = theta1ffpa2(j);
    %define stepsize segment 1 sweep
    STEP1pa2(j) = (END1pa2(j)-BEGIN1pa2(j))/N1;
end
%

if prestress == 0

%loop for segment 1 angle sweep for N1 different angles of segment 1
for i = 1:1:N1

%loop for segment 2 angle sweep for N2 different angles of segment 2
for k = 1:1:N2

%increase angle with steps equal to the stepsize STEP(j)
theta1(j,i) = BEGIN1(j) + STEP1(j)*i;

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
    theta1n(j,i) = - theta1(j,i);

    %lowerbound and upperbound of segment 2, respectively
    %for given precision point and angle of segment 1
    theta20(j,i) = real(-log(-(l1*r - ((l1*r - l1^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) + l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))*...
        (l1*r - l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i)))^(1/2) -...
        l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))/...
        (2*(l2*r*exp(theta1n(j,i)*1i) -...
        l1*l2*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))))*1i);

    theta2f(j,i) = real(-log(-(((l1*r - l1^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) + l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - 2*l2*l3*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) - 2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i)+...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l1*r*exp(alpha(j)*2i)*...
        exp(theta1n(j,i)*2i))*(l1*r - l1^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) + l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i)))^(1/2) +...
        l1*r - l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l1*r*exp(alpha(j)*2i)*...
        exp(theta1n(j,i)*2i))/(2*(l2*r*exp(theta1n(j,i)*1i) -...
        l1*l2*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
```

```
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %prevent the upperbound of segment 2 from being smaller
    %than the lowerbound
    if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %define boundaries segment 2 sweep
    BEGIN2(j,i) = theta20(j,i);
    END2(j,i) = theta2f(j,i);
    %define stepsize segment 2 sweep
    STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;

    %start angle of segment 2 equal to lowerbound,increase with stepsize
    theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;

    %angle connection line origin and endpoint segment 2
    Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if endpoint of second segment is in Q3
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/...
            abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi/2;
    end

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle of segment 3 and segment 4
    %for given precision point & angle segment 1 & angle segment 2
    theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r-...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
        l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
        l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

    theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
        l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
```

```
    if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
        theta4(j,i,k) =  2*pi + real(-log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(l4*r*...
            exp(Mtheta12(j,i,k)*1i) -...
            l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
    end

    %calculate the deviations in x and y of the coordinates
    %of the compensator,respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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^-12
        theta3(j,i,k) = pi + real( - asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
            l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));
    end

    %if endpoint of second segment is in Q1
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) > 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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);
        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) +...
            l2*sin(A2(j,i,k)))/(l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));


        %angle of segment 3 and segment 4
        %for given precision point & angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
            r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
```

```
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));


        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator,
             respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

            %if the absolute value of any of these deviations transcends a
            %certain threshold, then use alternative formulations for
            %theta3
            if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
            theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
                l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

                if theta3(j,i,k) < - pi
                    theta3(j,i,k) = 2*pi + pi/2 -...
                        real(pi + acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
                        r*cos(Ar(j)) +...
                        l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
                        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
                        (l12(j,i,k)*r*exp(Ar(j)*2i) +...
                        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
                        l12(j,i,k)*r*exp(Ar(j)*2i) -...
                        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
                        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
```

```
                        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
                        (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                        l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
                end
            end
    end

    %if endpoint of second segment is in Q2
    if ((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0)...
            || (passedX(j,i) == 1)

        %indicate that the endpoint of second segment passed x-axis
        passedX(j,i) = 1;
        %angle pendulum w.r.t. positive x-axis, (CCW positive)
        Ar(j) = (pi/2) - alpha(j);
        %angle of segment 1 with respect to positive x-axis (CW positive)
        theta1p(j,i) = theta1(j,i) - (pi/2);
        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to
        %positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);

        if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
            theta12P(j,i,k) = theta12P(j,i,k) + pi;
        end

        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = -theta12P(j,i,k);

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

        theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) +...
            l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r-...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
            l12(j,i,k)*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i)))^(1/2) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) +...
            l12(j,i,k)*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator,
             respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));
```

```

        %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((l4*sin(log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
                l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
        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((((- l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) +...
        l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) +...
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) +...
        l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) -...
        2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        2*l2*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i))*(- l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) +...
        l1*r*exp(Ar(j)*2i) + 2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l2*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 2*l3*l4*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i)))^(1/2) - l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) +...
        l1*r*exp(Ar(j)*2i) + 2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i))/...
        (2*(l1*l2*exp(Ar(j)*1i) - l2*r*exp(A1(j,i)*1i))))*1i);


    theta2f(j,i) = (pi/2) - real(-log((-((-l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i)+...
        l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) +...
        l1*r*exp(Ar(j)*2i) - 2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        2*l2*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i))*(- l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) +...
        l1*r*exp(Ar(j)*2i) + 2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l2*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i)))^(1/2) -...
        l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
```

```
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) -...
        r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) +...
        l1*r*exp(Ar(j)*2i) + 2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i))/...
        (2*(l1*l2*exp(Ar(j)*1i) - l2*r*exp(A1(j,i)*1i))))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta20(j,i) - theta20(j,i-1)) > pi
        theta20(j,i) = theta20(j,i) - 2*pi;
    end

    %prevent the upperbound of segment 2 from being smaller
    %than the lowerbound
    if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %define boundaries segment 2 sweep
    BEGIN2(j,i) = theta20(j,i);
    END2(j,i) = theta2f(j,i);
    %define stepsize segment 2 sweep
    STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;

    %start angle of segment 2 equal to lowerbound,increase with stepsize
    theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;

    %angle segment 2 w.r.t. positive x-axis, (CCW positive)
    A2(j,i,k) = (pi/2) - theta2(j,i,k);

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle imaginary connection line origin and endpoint segment 2
    phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
        (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

    %...and the same angle calculated by using other variables
    phi12v(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if the node at the end of the second segment is located beneath the
    %positive x-axis
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0
        phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
    end

    %if endpoint of second segment is in Q3
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))/...
            abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi;
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (phi12(j,i,k)-phi12(j,i,k-1)) > pi
        phi12(j,i,k) = phi12(j,i,k) - 2*pi;
    end

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
        r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
        (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
```

```
        exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
        l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));


    theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
        (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
        l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
        (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);

    if phi12(j,i,k) > pi/2
        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) +...
            l2*sin(theta2(j,i,k)))/...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

        %if endpoint of second segment is in Q4
        if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
                (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

            %angle connection line origin and endpoint segment 2
            Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
                l2*cos(theta2(j,i,k)))/abs(l1*sin(theta1(j,i)) +...
                l2*sin(theta2(j,i,k)))) + pi/2;
        end

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
            (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
            l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

        theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
            (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
            l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);

        %compensate for erroneous results due to periodicity of the loop
        %closure equations
        if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
```

```
                theta4(j,i,k) =  2*pi + real(-log(-(l12(j,i,k)*r +...
                    ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
                    exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
                    exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
                    exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
                    exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
                    exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
                    exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                    l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                    l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
                    r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                    2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                    l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
                    exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
                    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                    l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
                    l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
                    r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                    l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
                    (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
                    l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
            end

            %calculate the deviations in x and y of the coordinates of the compensator, respectively
            DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
                l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
            DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
                l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

            %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((l4*sin(log(-(l12(j,i,k)*r +...
                    ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                    exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                    exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                    exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                    exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                    l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
                    l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
            end

        end

        if phi12(j,i,k) < 0
            %angle pendulum w.r.t. positive x-axis, (CCW positive)
            Ar(j) = (pi/2) - alpha(j);
            %angle of pendulum with respect to positive x-axis (CW positive)
            theta1p(j,i) = theta1(j,i) - (pi/2);

            %angle of imaginary connection (between the origin and the
            %node at the end of the second segment) with respect to
            %positive x-axis
            %(clockwise positive)
            theta12P(j,i,k) = - phi12(j,i,k);

            %angle of segment 3 and segment 4, for given precision point &
            %angle segment 1 & angle segment 2
            theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
```

```
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) - l12(j,i,k)*...
                cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

            theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

            %calculate the deviations in x and y of the coordinates of the compensator,
                 respectively
            DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
                l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
            DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
                l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

            %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((l4*sin(log(-(l12(j,i,k)*r +...
                    ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                    exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                    exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                    exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                    exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                    l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
                    l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
            end

        end

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
                (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
```

```
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        if theta3(j,i,k) > pi
            theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - ...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
                l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
                (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
        end
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
        theta4(j,i,k) = 2*pi + theta4(j,i,k);
    end

end

%in the case of a horizontally positioned segment 1, MATLAB solve() has
%troubles finding a solution... Therefore, perturb by small amount to solve
if theta1(j,i) == pi/2
   theta1(j,i) = pi/2 + STEP1(j);
end

%the expressions within this loop are valid for theta1 > pi/2
if theta1(j,i) > pi/2
    %angle pendulum w.r.t. positive x-axis, (CCW positive)
    Ar(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (CW positive)
    theta1p(j,i) = theta1(j,i) - (pi/2);

    %lowerbound and upperbound of segment 2, respectively,
    %for given precision point and angle of segment 1
    theta20(j,i) = real(-log(-(l1*r - ((l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*2i))/(2*(l1*l2*exp(Ar(j)*1i)*1i - l2*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*1i)*1i)))*1i);

    theta2f(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
```

```
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*2i))/(2*(l1*l2*exp(Ar(j)*1i)*1i - l2*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*1i)*1i)))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta20(j,i) - theta20(j,i-1)) > pi
        theta20(j,i) = theta20(j,i) - 2*pi;
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %prevent the upperbound of segment 2 from being smaller than the lowerbound
    if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %define boundaries segment 2 sweep
    BEGIN2(j,i) = theta20(j,i);
    END2(j,i) = theta2f(j,i);
    %define stepsize segment 2 sweep
    STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;

    %start angle of segment 2 equal to lowerbound,increase with stepsize
    theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle of imaginary connection (between the origin and the
    %node at the end of the second segment) with respect to positive x-axis
    %(clockwise positive)
    theta12P(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);

    if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
        theta12P(j,i,k) = theta12P(j,i,k) + pi;
    end

    %if endpoint of second segment is in Q4
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = -atan(abs(l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/abs(l1*sin(theta1(j,i)) +...
            l2*sin(theta2(j,i,k)))) - pi;
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && abs(theta12P(j,i,k)-theta12P(j,i,k-1)) > pi
        theta12P(j,i,k) = theta12P(j,i,k) + 2*pi;
    end

    %angle imaginary connection line origin and endpoint segment 2
    phi12(j,i,k) = -theta12P(j,i,k);

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
        exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
```

```
        r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) - l12(j,i,k)*...
        cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

    theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
        l12(j,i,k)*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i)))^(1/2) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

    %if endpoint segment 2 is in Q3
    if theta12P(j,i,k) <= 0 && theta12P(j,i,k) > -pi/2
        %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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);
        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
            (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &...
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
            r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i);

        %compensate for erroneous results due to periodicity of the loop
        %closure equations
        if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
            theta4(j,i,k) = 2*pi + pi/2 - real(-log(-(((l12(j,i,k)*r*...
                exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
```

```
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                2*l3*l4*exp(Ar(j)*1i)*  exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
                exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*...
                exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
                l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
                (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);
        end

        %calculate the deviations in x and y of the coordinates of the compensator,
            respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
                exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
                l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i)+...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
                (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

            if theta3(j,i,k) > pi
                theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
                    cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                    l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
                    l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                    l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                    l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                    r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                    2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*...
                    r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                    l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                    l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                    r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*...
                    exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*...
                    r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
                    l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                    l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                    r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
                    (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*...
                    exp(phi12(j,i,k)*1i))))*1i))/l3));
            end
        end

    end

    %if endpoint of second segment is in Q4
    if ((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0)

        %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
        theta1n(j,i) = - theta1(j,i);

        %length of imaginary connection line between origin and end of segment 2
        l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);
```

```
%angle connection line origin and endpoint segment 2
Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
    (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

%if endpoint of second segment is still in Q4
if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

    %angle connection line origin and endpoint segment 2
    Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
        l2*cos(theta2(j,i,k)))/abs(l1*sin(theta1(j,i)) +...
        l2*sin(theta2(j,i,k)))) + pi/2;
end

%angle of segment 3 and segment 4
%for given precision point & angle segment 1 & angle segment 2
theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
    ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
    exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i)...
    + l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
    r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
    exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r -...
    l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
    exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) - l12(j,i,k)^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
    exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
    (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
    exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
    l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
    l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
    r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
    2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
    (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
    exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
    exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i)...
    - r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2)...
    - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
    l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
    l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
    exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
    exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(l4*r*...
    exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*...
    exp(alpha(j)*1i))))*1i);

%calculate the deviations in x and y of the coordinates of the compensator,
%respectively
DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
    l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
    l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

%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((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
        exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
```

```
                l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
                l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
                r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
                (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
                l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));
        end

    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
        theta4(j,i,k) = 2*pi + theta4(j,i,k);
    end

end

%initial relative angle of segment 1
alpha10 = theta1i;
%initial relative angle of segment 2
alpha20 = theta2i - theta1i;
%initial relative angle of segment 3
alpha30 = theta3i - theta2i;
%initial relative angle of segment 4
alpha40 = theta4i - theta3i;

%angle of rotation torsion spring 1
alpha1(j,i) = theta1(j,i) - alpha10;
%angle of rotation torsion spring 2
alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
%angle of rotation torsion spring 3
alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
%angle of rotation torsion spring 4
alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;

if  nonlinearity == 0
    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2);
    %potential energy spring 3
    V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2);
    %potential energy spring 4
    V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

if  nonlinearity == 1
```

```
    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
    V2(j,i,k) = (A/3)*alpha2(j,i,k)^3 + (B/2)*alpha2(j,i,k)^2;
    %potential energy spring 3
    V3(j,i,k) = (A/3)*alpha3(j,i,k)^3 + (B/2)*alpha3(j,i,k)^2;
    %potential energy spring 4
    V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

%calculate the deviations in x and y of the coordinates of the compensator, respectively
DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
    l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
    l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

%calculate the distance from the endpoint of the second segment to the end
%effector of the inverted pendulum
d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
    (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);

%check condition upper loop closure
if (l4-l3-d(j,i,k)) > 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture does not exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count2" instead
    Count2 = Count2 + 1;
end

%if segment 1 and segment 2 are not at their lowerbound
if i>1 && k>1
    %if the angle of the third segment was previously - for the same angle
    %of the pendulum - NaN, then it will remain NaN for this angle of the
    %pendulum (infeasible solution space)
    if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)      %#ok<COMPNOP>
        theta3(j,i,k) = NaN;

        %the potential energy and the angle of segment 4 should
        %consequently be NaN as well
        V(j,i,k) = NaN;
        theta4(j,i,k) = NaN;
    end
end

%check condition upper loop closure
if l4-l3+d(j,i,k) < 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count3" instead
    Count3 = Count3 + 1;
end

%if the absolute value of any of these deviations transcends a
%certain threshold, then increase the variable "Count" by one
if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
    Count = Count + 1;
end

%x - coordinate origin (and first spring)
x0      = 0;
%y - coordinate origin (and first spring)
y0      = 0;
```

```
%x - coordinate 2nd spring
x1(j,i) = l1*sin(theta1(j,i));
%y - coordinate 2nd spring
y1(j,i) = l1*cos(theta1(j,i));
%x - coordinate 3rd spring
x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
%y - coordinate 3rd spring
y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
%x - coordinate 4th spring
x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
%y - coordinate 4th spring
y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
%x - coordinate end effector
x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
%y - coordinate end effector
y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));

end
end
end

if prestress == 1

%loop for segment 1 angle sweep for N1 different angles of segment 1
for i = 1:1:N1

%loop for segment 2 angle sweep for N2 different angles of segment 2
for k = 1:1:N2

%increase angle with steps equal to the stepsize STEP1pa(j)
%formulation for balancer with spring 3 enabled, spring 2 locked
theta1sw(j,i) = BEGIN1pa(j) + STEP1pa(j)*i;

%increase angle with steps equal to the stepsize STEP1pa2(j)
%formulation for balancer with spring 2 enabled, spring 3 locked
theta1sw2(j,i) = BEGIN1pa2(j) + STEP1pa2(j)*i;

%increase angle with steps equal to the stepsize STEP1(j)
%formulation for balancer with all springs enabled
theta1fa(j,i) = BEGIN1(j) + STEP1(j)*i;

%if both spring 2 and spring 3 are locked OR
%if both segment 1 and 2 are at their lowerbound and the inverted pendulum
%has made just 1 step
if (M31t(j,i,k) < M03 && M21t(j,i,k) < M02) || (j == 1 && i == 1 && k == 1)
    %the angle of the first segment increases linearly with the angle of
    %the inverted pendulum
    theta1(j,i) = alpha(j)+theta1i;
    %the angle of the second segment increases linearly with the angle of
    %segment 1
    theta2(j,i,k) = theta1(j,i) + (theta2i-theta1i);

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
    theta1n(j,i) = - theta1(j,i);

    %angle connection line origin and endpoint segment 2
    Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
```

```
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
        l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

    theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);

    %if the endpoint of segment 2 is located above the x-axis
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) > 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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);
        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
            (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

        %angle of segment 3 and segment 4
        %for given precision point & angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
            r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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^-12
        theta3(j,i,k) = pi + real( - asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
```

```
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));
    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);
    %angle segment 2 w.r.t. positive x-axis, (CCW positive)
    A2(j,i,k) = (pi/2) - theta2(j,i,k);

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle imaginary connection line origin and endpoint segment 2
    phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
        (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));


    %...and the same angle calculated by using other variables
    phi12v(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if the node at the end of the second segment is located left to the
    %positive y-axis
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0
        phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
    end

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
        r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
        (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
        l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

    theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*  exp(phi12(j,i,k)*1i))*...
        (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
```

```
        exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
        l4*r*exp(phi12(j,i,k)*1i))))*1i);

    % if angle of imaginary connection with respect to positive x-axis
    %(clockwise positive) is larger than 90 deg
    if phi12(j,i,k) > pi/2
        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) +...
            l2*sin(theta2(j,i,k)))/(l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

        theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - ...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
    end

    % if angle of imaginary connection with respect to positive x-axis
    %(clockwise positive) is negative
    if phi12(j,i,k) < 0
        %angle pendulum w.r.t. positive x-axis, (CCW positive)
        Ar(j) = (pi/2) - alpha(j);
        %angle of segment 1 with respect to positive x-axis (CW positive)
        theta1p(j,i) = theta1(j,i) - (pi/2);

        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
```

```
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) - l12(j,i,k)*...
                cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

            theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);
        end

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*...
                exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
                exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
        end

    end

    %in the case of a horizontally positioned segment 1, MATLAB solve() has
    %troubles finding a solution... Therefore, perturb by small amount to solve
    if theta1(j,i) == pi/2
       theta1(j,i) = pi/2 + STEP1(j);
    end

    %the expressions within this loop are valid for theta1 > pi/2
    if theta1(j,i) > pi/2
        %angle pendulum w.r.t. positive x-axis, (CCW positive)
        Ar(j) = (pi/2) - alpha(j);
        %angle of segment 1 with respect to positive x-axis (CW positive)
        theta1p(j,i) = theta1(j,i) - (pi/2);

        %length of imaginary connection line between origin and end of segment 2
        l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to positive x-axis
        %(clockwise positive)
```

```
    theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) +...
        l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
        exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
        l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

    theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
        exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
        l4^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

    if theta12P(j,i,k) < 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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);

        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
            (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
            r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
            exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
```

```
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator,
            respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)...
                *r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
        end

    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
    end

end

%calculate the deviations in x and y of the coordinates of the compensator, respectively
DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
    l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
    l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

%calculate the distance from the endpoint of the second segment to the end
%effector of the inverted pendulum
d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
```

```
    (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);

%check condition upper loop closure
if (l4-l3-d(j,i,k)) > 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count2" instead
    Count2 = Count2 + 1;
end

%if segment 1 and segment 2 are not at their lowerbound
if i>1 && k>1
    %if the angle of the third segment was previously - for the same angle
    %of the pendulum - NaN, then it will remain NaN for this angle of the
    %pendulum (infeasible solution space)
    if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)      %#ok<COMPNOP>
        theta3(j,i,k) = NaN;

        %the potential energy and the angle of segment 4 should
        %consequently be NaN as well
        V(j,i,k) = NaN;
        theta4(j,i,k) = NaN;
    end
end

%check condition upper loop closure
if l4-l3+d(j,i,k) < 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count3" instead
    Count3 = Count3 + 1;
end

%if the absolute value of any of these deviations transcends a
%certain threshold, then increase the variable "Count" by one
if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
    Count = Count + 1;
end

%initial relative angle of segment 1
alpha10 = theta1i;
%initial relative angle of segment 2
alpha20 = theta2i - theta1i;
%initial relative angle of segment 3
alpha30 = theta3i - theta2i;
%initial relative angle of segment 4
alpha40 = theta4i - theta3i;

%angle of rotation torsion spring 1
alpha1(j,i) = theta1(j,i) - alpha10;
%angle of rotation torsion spring 2
alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
%angle of rotation torsion spring 3
alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
%angle of rotation torsion spring 4
alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;

if  nonlinearity == 0
    %internal moment spring 1
    M1(j,i) = k1*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = k2*alpha2(j,i,k) + M02;
    %internal moment spring 3
    M3(j,i,k) = k3*alpha3(j,i,k) + M03;
```

```
    %internal moment spring 4
    M4(j,i,k) = k4*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) +...
        ((k2/2)*(M02/k2)^2);
    %potential energy spring 3
    V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) +...
        ((k3/2)*(M03/k3)^2);
    %potential energy spring 4
    V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

if  nonlinearity == 1
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
    %second solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M2 < 0
        alphastar1M2 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M2 < 0
        alphastar2M2 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM2"
    alphastarsM2 = [alphastar1M2,alphastar2M2];

    %store the smallest solution for the prestress angle
    alphastarM2 = min(abs(alphastarsM2));

    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M3 < 0
        alphastar1M3 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M3 < 0
        alphastar2M3 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM3"
    alphastarsM3 = [alphastar1M3,alphastar2M3];

    %store the smallest solution for the prestress angle
    alphastarM3 = min(abs(alphastarsM3));

    %internal moment spring 1
    M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 +...
        B*(alpha2(j,i,k)+alphastarM2);
    %internal moment spring 3
    M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 +...
        B*(alpha3(j,i,k)+alphastarM3);
    %internal moment spring 4
    M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
    V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 +...
        (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
    %potential energy spring 3
    V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 +...
        (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
    %potential energy spring 4
```

```
    V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

%x - coordinate origin (and first spring)
x0      = 0;
%y - coordinate origin (and first spring)
y0      = 0;
%x - coordinate 2nd spring
x1(j,i) = l1*sin(theta1(j,i));
%y - coordinate 2nd spring
y1(j,i) = l1*cos(theta1(j,i));
%x - coordinate 3rd spring
x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
%y - coordinate 3rd spring
y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
%x - coordinate 4th spring
x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
%y - coordinate 4th spring
y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
%x - coordinate end effector
x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
%y - coordinate end effector
y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));

%magnitude reaction force y-direction
F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))/...
    (-tan(theta4(j,i,k))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))...
    + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));

%magnitude reaction force x-direction
F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
    (l4*cos(theta4(j,i,k)));

%external moment on second spring (node 2)
M21t(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
    F1yt(j,i,k)*l1*sin(theta1(j,i));

%external moment on third spring (node 3)
M31t(j,i,k) = M1(j,i) +...
    F1xt(j,i,k)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))) -...
    F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
end

%if spring 3 is activated and spring 2 is still locked
if M31t(j,i,k) >= M03 && M21t(j,i,k) < M02

    %formulation for angle segment 1 with spring 3 enabled, spring 2 locked
    theta1(j,i) = theta1sw(j,i);

    %the angle of the second segment increases linearly with the angle of
    %the first segment
    theta2(j,i,k) = theta1(j,i) + (theta2i-theta1i);

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
    theta1n(j,i) = - theta1(j,i);

    %angle connection line origin and endpoint segment 2
    Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))...
        /(l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
        (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
```

```
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
        l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
        l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

    theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
        (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
        l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);

    %if the endpoint of segment 2 is located above the x-axis
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) > 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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);
        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
            (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*...
            cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
            l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
```

```
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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^-12
        theta3(j,i,k) = pi + real( - asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
            l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));
    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);

    %angle segment 2 w.r.t. positive x-axis, (CCW positive)
    A2(j,i,k) = (pi/2) - theta2(j,i,k);

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle imaginary connection line origin and endpoint segment 2
    phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
        (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

    %...and the same angle calculated by using other variables
    phi12v(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if the node at the end of the second segment is located beneath the
    %positive x-axis
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0
        phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
    end

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
        r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
        (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*...
        r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
        (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
        l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

    theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
```

```
        l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
        (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*...
        r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
        (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);

    if phi12(j,i,k) > pi/2
        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) +...
            l2*sin(theta2(j,i,k)))/...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));


        theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) -...
            l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
    end

    if phi12(j,i,k) < 0
        %angle pendulum w.r.t. positive x-axis, (CCW positive)
        Ar(j) = (pi/2) - alpha(j);
        %angle of segment 1 with respect to positive x-axis (CW positive)
        theta1p(j,i) = theta1(j,i) - (pi/2);
        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to
        %positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
```

```
        theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

        theta4(j,i,k) = real(-log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*...
            exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*1i)*1i)))*1i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
            cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*...
            r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
            exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
            l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
            (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
    end

end

%in the case of a horizontally positioned segment 1, MATLAB solve() has
%troubles finding a solution... Therefore, perturb by small amount to solve
if theta1(j,i) == pi/2
   theta1(j,i) = pi/2 + STEP1(j);
end

%the expressions within this loop are valid for theta1 > pi/2
```

```
if theta1(j,i) > pi/2
    %angle pendulum w.r.t. positive x-axis, (CCW positive)
    Ar(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (CW positive)
    theta1p(j,i) = theta1(j,i) - (pi/2);

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle of imaginary connection (between the origin and the
    %node at the end of the second segment) with respect to positive x-axis
    %(clockwise positive)
    theta12P(j,i,k) = acos((l1*cos(theta1(j,i)) +...
        l2*cos(theta2(j,i,k)))/l12(j,i,k)) - (pi/2);

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
        l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

    theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
        exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

    if theta12P(j,i,k) < 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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);
        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
            (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &...
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*...
            cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
            l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
```

```
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
                exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
        end

    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
```

```
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
                l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)))*1i) -...
                l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
        end

    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

    DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %calculate the distance from the endpoint of the second segment to the end
    %effector of the inverted pendulum
    d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
        (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);

    %check condition upper loop closure
    if (l4-l3-d(j,i,k)) > 0
        %set the deviation in x...
        DEV11(j,i,k) = 0;
        %... and y to zero such that this scenario won't be flagged
        DEV22(j,i,k) = 0;
        %posture doesn't exist, so potential energy not a number
        V(j,i,k) = NaN;

        %define the angles of the third and fourth segment to be no value;
        %the surface plots of these tensors (used for debugging) would
        %otherwise be nonsmooth
        theta3(j,i,k) = NaN;
        theta4(j,i,k) = NaN;
        %flag this event with variable "Count2" instead
        Count2 = Count2 + 1;
    end

    %if segment 1 and segment 2 are not at their lowerbound
    if i>1 && k>1
        %if the angle of the third segment was previously - for the same angle
        %of the pendulum - NaN, then it will remain NaN for this angle of the
        %pendulum (infeasible solution space)
        if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)      %#ok<COMPNOP>
            theta3(j,i,k) = NaN;

            %the potential energy and the angle of segment 4 should
            %consequently be NaN as well
            V(j,i,k) = NaN;
            theta4(j,i,k) = NaN;
        end
    end

    %check condition upper loop closure
    if l4-l3+d(j,i,k) < 0
        %set the deviation in x...
        DEV11(j,i,k) = 0;
        %... and y to zero such that this scenario won't be flagged
        DEV22(j,i,k) = 0;
        %posture doesn't exist, so potential energy not a number
        V(j,i,k) = NaN;

        %define the angles of the third and fourth segment to be no value;
        %the surface plots of these tensors (used for debugging) would
        %otherwise be nonsmooth
        theta3(j,i,k) = NaN;
        theta4(j,i,k) = NaN;
        %flag this event with variable "Count3" instead
        Count3 = Count3 + 1;
    end

    %if the absolute value of any of these deviations transcends a
    %certain threshold, then increase the variable "Count" by one
    if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
        Count = Count + 1;
    end

    %initial relative angle of segment 1
    alpha10 = theta1i;
    %initial relative angle of segment 2
    alpha20 = theta2i - theta1i;
```

```
%initial relative angle of segment 3
alpha30 = theta3i - theta2i;
%initial relative angle of segment 4
alpha40 = theta4i - theta3i;

%angle of rotation torsion spring 1
alpha1(j,i) = theta1(j,i) - alpha10;
%angle of rotation torsion spring 2
alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
%angle of rotation torsion spring 3
alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
%angle of rotation torsion spring 4
alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;

if nonlinearity == 0
    %internal moment spring 1
    M1(j,i) = k1*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = k2*alpha2(j,i,k) + M02;
    %internal moment spring 3
    M3(j,i,k) = k3*alpha3(j,i,k) + M03;
    %internal moment spring 4
    M4(j,i,k) = k4*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) +...
        ((k2/2)*(M02/k2)^2);
    %potential energy spring 3
    V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) +...
        ((k3/2)*(M03/k3)^2);
    %potential energy spring 4
    V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

if nonlinearity == 1
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
    %second solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M2 < 0
        alphastar1M2 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M2 < 0
        alphastar2M2 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM2"
    alphastarsM2 = [alphastar1M2,alphastar2M2];

    %store the smallest solution for the prestress angle
    alphastarM2 = min(abs(alphastarsM2));

    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);

    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M3 < 0
        alphastar1M3 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M3 < 0
        alphastar2M3 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM3"
    alphastarsM3 = [alphastar1M3,alphastar2M3];

    %store the smallest solution for the prestress angle
```

```
    alphastarM3 = min(abs(alphastarsM3));

    %internal moment spring 1
    M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 +...
        B*(alpha2(j,i,k)+alphastarM2);
    %internal moment spring 3
    M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 +...
        B*(alpha3(j,i,k)+alphastarM3);
    %internal moment spring 4
    M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
    V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 +...
        (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
    %potential energy spring 3
    V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 +...
        (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
    %potential energy spring 4
    V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

%allow for nonnegative rotation of spring 2 only
if alpha2(j,i,k) < 0
    V(j,i,k) = NaN;
end

%allow for nonnegative rotation of spring 3 only
if alpha3(j,i,k) < 0
    V(j,i,k) = NaN;
end

%x - coordinate origin (and first spring)
x0      = 0;
%y - coordinate origin (and first spring)
y0      = 0;
%x - coordinate 2nd spring
x1(j,i) = l1*sin(theta1(j,i));
%y - coordinate 2nd spring
y1(j,i) = l1*cos(theta1(j,i));
%x - coordinate 3rd spring
x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
%y - coordinate 3rd spring
y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
%x - coordinate 4th spring
x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
%y - coordinate 4th spring
y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
%x - coordinate end effector
x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
%y - coordinate end effector
y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));

%magnitude reaction force y-direction
F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))/...
    (-tan(theta4(j,i,k))*(l1*cos(theta1(j,i))+...
    l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))...
    + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));

%magnitude reaction force x-direction
F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
    (l4*cos(theta4(j,i,k)));

%external moment on second spring (node 2)
M21t(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
    F1yt(j,i,k)*l1*sin(theta1(j,i));

%external moment on third spring (node 3)
M31t(j,i,k) = M1(j,i) +...
    F1xt(j,i,k)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))) -...
    F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
end

%if spring 2 is activated and spring 3 is still locked
if M31t(j,i,k) < M03 && M21t(j,i,k) >= M02

    %formulation for angle segment 1 with spring 2 enabled, spring 3 locked
    theta1(j,i) = theta1sw2(j,i);
```

```
    %angle of imaginary connection (between the node 2 and node 4)
    %with respect to the the second segment
    phi232 = acos((l2^2 + l23^2 - l3^2)/(2*l2*l23));

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
    theta1n(j,i) = - theta1(j,i);

    %formulation for theta4: elbow up
    theta4(j,i,k) = real(-log(-(l1*r + ((l1*r - l1^2*exp(theta1n(j,i)*1i)*...
        exp(alpha(j)*1i) + l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))*...
        (l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
        l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(theta1n(j,i)*1i) -...
        l1*l4*exp(theta1n(j,i)*2i)*exp(alpha(j)*1i))))*1i);

    %formulation for theta2: elbow up
    theta23(j,i) = real(asin((l4*sin(log(-(l1*r +...
        ((l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))*(l1*r -...
        l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) + 2*l23*l4*...
        exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) + l1*r*exp(theta1n(j,i)*2i)*...
        exp(alpha(j)*2i))).^(1/2) - l1^2*exp(theta1n(j,i)*1i)*...
        exp(alpha(j)*1i) + l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) + l1*r*exp(theta1n(j,i)*2i)*...
        exp(alpha(j)*2i))/(2*(l4*r*exp(theta1n(j,i)*1i) -...
        l1*l4*exp(theta1n(j,i)*2i)*exp(alpha(j)*1i))))*1i) +...
        l1*sin(theta1n(j,i)) + r*sin(alpha(j)))/l23));

    theta2(j,i,k) = theta23(j,i) + phi232;
    theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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^-12
        theta23(j,i) = pi + real( - asin((l4*sin(log(-(l1*r +...
            ((l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
            2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))*...
            (l1*r - l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            2*l23*l4*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i)))^(1/2) -...
            l1^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l23^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
            l4^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(theta1n(j,i)*1i)*exp(alpha(j)*1i) +...
            l1*r*exp(theta1n(j,i)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(theta1n(j,i)*1i) - l1*l4*exp(theta1n(j,i)*2i)*...
            exp(alpha(j)*1i))))*1i) +...
```

```
            l1*sin(theta1n(j,i)) + r*sin(alpha(j)))/l23));

        theta2(j,i,k) = theta23(j,i) + phi232;
        theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
    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);

    %formulation for theta4: elbow up
    theta4(j,i,k) = pi/2 - real(-log(-(((l1*r*exp(Ar(j)*2i) +...
        l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
        l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
        2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i))*(l1*r*exp(Ar(j)*2i) +...
        l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
        l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
        2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i)))^(1/2) -...
        l1*r*exp(Ar(j)*2i) - l1*r*exp(A1(j,i)*2i) + l1^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i) - l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + r^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i))/...
        (2*(l1*l4*exp(Ar(j)*1i) - l4*r*exp(A1(j,i)*1i))))*1i);

    %formulation for theta2 and theta3: elbow up
    theta23(j,i) = pi/2 - real(pi - acos((l1*cos(A1(j,i)) - r*cos(Ar(j)) +...
        l4*cos(log(-(((l1*r*exp(Ar(j)*2i) + l1*r*exp(A1(j,i)*2i) -...
        l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l23^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) - 2*l23*l4*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i))*(l1*r*exp(Ar(j)*2i) + l1*r*exp(A1(j,i)*2i) -...
        l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l23^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
        r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i)))^(1/2) - l1*r*exp(Ar(j)*2i) -...
        l1*r*exp(A1(j,i)*2i) + l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
        l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(A1(j,i)*1i) + r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i))/...
        (2*(l1*l4*exp(Ar(j)*1i) - l4*r*exp(A1(j,i)*1i))))*1i))/l23));

    theta2(j,i,k) = theta23(j,i) + phi232;
    theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %if the absolute value of any of these deviations transcends a
    %certain threshold, then use alternative formulation for theta2 and
    %theta3
    if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
        theta23(j,i) = 2*pi + pi/2 - real(pi + acos((l1*cos(A1(j,i)) -...
            r*cos(Ar(j)) + l4*cos(log(-(((l1*r*exp(Ar(j)*2i) +...
            l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
            l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) -...
            2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i))*(l1*r*exp(Ar(j)*2i) +...
            l1*r*exp(A1(j,i)*2i) - l1^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
            l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(A1(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
            2*l23*l4*exp(Ar(j)*1i)*exp(A1(j,i)*1i)))^(1/2) -...
            l1*r*exp(Ar(j)*2i) - l1*r*exp(A1(j,i)*2i) + l1^2*exp(Ar(j)*1i)*...
            exp(A1(j,i)*1i) - l23^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(A1(j,i)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(A1(j,i)*1i))/...
            (2*(l1*l4*exp(Ar(j)*1i) - l4*r*exp(A1(j,i)*1i))))*1i))/l23));

        theta2(j,i,k) = theta23(j,i) + phi232;
        theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
    end

end

%in the case of a horizontally positioned segment 1, MATLAB solve() has
```

```
%troubles finding a solution... Therefore, perturb by small amount to solve
if theta1(j,i) == pi/2
   theta1(j,i) = pi/2 + STEP1(j);
end

%the expressions within this loop are valid for theta1 > pi/2
if theta1(j,i) > pi/2
    %angle pendulum w.r.t. positive x-axis, (CCW positive)
    Ar(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (CW positive)
    theta1p(j,i) = theta1(j,i) - (pi/2);

    %formulation for theta4: elbow up
    theta4(j,i,k) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
        l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))/...
        (2*(l1*l4*exp(Ar(j)*1i)*1i -...
        l4*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*1i)*1i)))*1i);

    %formulation for theta2 and theta3: elbow up
    theta23(j,i) = real(asin((l4*sin(log(-(l1*r+((l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
        l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))/...
        (2*(l1*l4*exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*1i)*1i)))*1i) -...
        l1*cos(theta1p(j,i)) + r*cos(Ar(j)))/l23));

    theta2(j,i,k) = theta23(j,i) + phi232;
    theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %if the absolute value of any of these deviations transcends a
    %certain threshold, then use alternative formulation for theta2 and
    %theta3
    if abs(DEV1(j,i,k)) > 10^-12 || abs(DEV2(j,i,k)) > 10^-8
        theta23(j,i) = pi + real( - asin((l4*sin(log(-(l1*r +...
            ((l1*r - l1^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
            l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
            2*l23*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
            exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
            exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta1p(j,i)*1i) + 2*l23*l4*exp(Ar(j)*1i)*...
            exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
            exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
            exp(theta1p(j,i)*1i) + l23^2*exp(Ar(j)*1i)*...
            exp(theta1p(j,i)*1i) - l4^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
            r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
            exp(theta1p(j,i)*2i))/(2*(l1*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*1i)*1i)))*1i) -...
            l1*cos(theta1p(j,i)) + r*cos(Ar(j)))/l23));

        theta2(j,i,k) = theta23(j,i) + phi232;
        theta3(j,i,k) = theta23(j,i) + phi232 + (theta3i-theta2i);
    end

end

%calculate the deviations in x and y of the coordinates of the compensator, respectively
```

```
DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
    l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
    l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

%calculate the distance from the endpoint of the second segment to the end
%effector of the inverted pendulum
d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
    (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);

%check condition upper loop closure
if (l4-l3-d(j,i,k)) > 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count2" instead
    Count2 = Count2 + 1;
end

%if segment 1 and segment 2 are not at their lowerbound
if i>1 && k>1
    %if the angle of the third segment was previously - for the same angle
    %of the pendulum - NaN, then it will remain NaN for this angle of the
    %pendulum (infeasible solution space)
    if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)       %#ok<COMPNOP>
        theta3(j,i,k) = NaN;

        %the potential energy and the angle of segment 4 should
        %consequently be NaN as well
        V(j,i,k) = NaN;
        theta4(j,i,k) = NaN;
    end
end

%check condition upper loop closure
if l4-l3+d(j,i,k) < 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count3" instead
    Count3 = Count3 + 1;
end

%if the absolute value of any of these deviations transcends a
%certain threshold, then increase the variable "Count" by one
if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
    Count = Count + 1;
end

%initial relative angle of segment 1
alpha10 = theta1i;
%initial relative angle of segment 2
alpha20 = theta2i - theta1i;
%initial relative angle of segment 3
alpha30 = theta3i - theta2i;
%initial relative angle of segment 4
alpha40 = theta4i - theta3i;

%angle of rotation torsion spring 1
alpha1(j,i) = theta1(j,i) - alpha10;
%angle of rotation torsion spring 2
alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
%angle of rotation torsion spring 3
alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
%angle of rotation torsion spring 4
```

```
alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;

if nonlinearity == 0
    %internal moment spring 1
    M1(j,i) = k1*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = k2*alpha2(j,i,k) + M02;
    %internal moment spring 3
    M3(j,i,k) = k3*alpha3(j,i,k) + M03;
    %internal moment spring 4
    M4(j,i,k) = k4*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) +...
        ((k2/2)*(M02/k2)^2);
    %potential energy spring 3
    V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) +...
        ((k3/2)*(M03/k3)^2);
    %potential energy spring 4
    V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

if nonlinearity == 1
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
    %second solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M2 < 0
        alphastar1M2 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M2 < 0
        alphastar2M2 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM2"
    alphastarsM2 = [alphastar1M2,alphastar2M2];

    %store the smallest solution for the prestress angle
    alphastarM2 = min(abs(alphastarsM2));

    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M3 < 0
        alphastar1M3 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M3 < 0
        alphastar2M3 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM3"
    alphastarsM3 = [alphastar1M3,alphastar2M3];

    %store the smallest solution for the prestress angle
    alphastarM3 = min(abs(alphastarsM3));

    %internal moment spring 1
    M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 +...
        B*(alpha2(j,i,k)+alphastarM2);
    %internal moment spring 3
    M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 +...
        B*(alpha3(j,i,k)+alphastarM3);
    %internal moment spring 4
    M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);
```

```
    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
    V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 +...
        (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
    %potential energy spring 3
    V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 +...
        (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
    %potential energy spring 4
    V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

%x - coordinate origin (and first spring)
x0      = 0;
%y - coordinate origin (and first spring)
y0      = 0;
%x - coordinate 2nd spring
x1(j,i) = l1*sin(theta1(j,i));
%y - coordinate 2nd spring
y1(j,i) = l1*cos(theta1(j,i));
%x - coordinate 3rd spring
x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
%y - coordinate 3rd spring
y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
%x - coordinate 4th spring
x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
%y - coordinate 4th spring
y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
%x - coordinate end effector
x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
%y - coordinate end effector
y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));

%magnitude reaction force y-direction
F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))))/...
    (-tan(theta4(j,i,k))*(l1*cos(theta1(j,i))+...
    l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))...
    + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));

%magnitude reaction force x-direction
F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
    (l4*cos(theta4(j,i,k)));

%external moment on second spring (node 2)
M21t(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
    F1yt(j,i,k)*l1*sin(theta1(j,i));

%external moment on third spring (node 3)
M31t(j,i,k) = M1(j,i) +...
    F1xt(j,i,k)*(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))) -...
    F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));

end

%if spring 2 and 3 are both activated
if M31t(j,i,k) >= M03 && M21t(j,i,k) >= M02

    %formulation for angle segment 1 with spring 2 and 3 both enabled
    theta1(j,i) = theta1fa(j,i);

%the expressions within this loop are valid for theta1 < 0
if theta1(j,i) < 0
    %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
    theta1n(j,i) = - theta1(j,i);

    %lowerbound and upperbound of segment 2, respectively,
    %for given precision point and angle of segment 1
    theta20(j,i) = real(-log(-(l1*r - ((l1*r - l1^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) + l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))*...
        (l1*r - l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l3^2*exp(alpha(j)*1i)*...
        exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
        r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
        2*l2*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
```

```
            2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i)))^(1/2) -...
            l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - l2^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + 2*l3*l4*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))/...
            (2*(l2*r*exp(theta1n(j,i)*1i) -...
            l1*l2*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))))*1i);

        theta2f(j,i) = real(-log(-(((l1*r - l1^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
            2*l2*l3*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) - 2*l2*l4*...
            exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 2*l3*l4*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))*...
            (l1*r - l1^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l3^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) -...
            r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + 2*l2*l3*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + 2*l2*l4*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + 2*l3*l4*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) + l1*r*exp(alpha(j)*2i)*...
            exp(theta1n(j,i)*2i)))^(1/2) + l1*r - l1^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) - l2^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            l3^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) + l4^2*exp(alpha(j)*1i)*...
            exp(theta1n(j,i)*1i) - r^2*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            2*l3*l4*exp(alpha(j)*1i)*exp(theta1n(j,i)*1i) +...
            l1*r*exp(alpha(j)*2i)*exp(theta1n(j,i)*2i))/...
            (2*(l2*r*exp(theta1n(j,i)*1i) -...
            l1*l2*exp(alpha(j)*1i)*exp(theta1n(j,i)*2i))))*1i);

        %compensate for erroneous results due to periodicity of the loop
        %closure equations
        if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
            theta2f(j,i) = theta2f(j,i) + 2*pi;
        end

        %prevent the upperbound of segment 2 from being smaller than
        %the lowerbound
        if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
            theta2f(j,i) = theta2f(j,i) + 2*pi;
        end

        %define boundaries segment 2 sweep
        BEGIN2(j,i) = theta20(j,i);
        END2(j,i) = theta2f(j,i);
        %define stepsize segment 2 sweep
        STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;

        %start angle of segment 2 equal to lowerbound,increase with stepsize
        theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;

        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

        %if endpoint of second segment is still in Q4
        if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
                (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

            %angle connection line origin and endpoint segment 2
            Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
                l2*cos(theta2(j,i,k)))/...
                abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi/2;
        end

        %length of imaginary connection line between origin and end of segment 2
        l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
            (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
```

```
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(l4*r*...
        exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*...
        exp(alpha(j)*1i))))*1i) +...
        l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));

    theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*...
        (l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/(2*(l4*r*...
        exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*exp(Mtheta12(j,i,k)*2i)*...
        exp(alpha(j)*1i))))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
        theta4(j,i,k) =  2*pi + real(-log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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^-12
        theta3(j,i,k) = pi + real( - asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
            exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
            exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
```

```
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
            l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));
    end

    %if endpoint of second segment is still in Q1
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) > 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);
        %angle segment 2 w.r.t. positive x-axis, (CCW positive)
        A2(j,i,k) = (pi/2) - theta2(j,i,k);
        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
            (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = pi/2 -...
            real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
            l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
            exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*...
            r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

            %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/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*...
                r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
```

```
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

                if theta3(j,i,k) < - pi
                    theta3(j,i,k) = 2*pi + pi/2 -...
                        real(pi + acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
                        r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*...
                        exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*...
                        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*...
                        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - 2*l3*l4*...
                        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
                        exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                        l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*...
                        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*...
                        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*...
                        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
                        l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*...
                        exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
                        exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                        exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                        exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                        exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*...
                        exp(Ar(j)*1i) -...
                        l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
                end
            end
    end

    %if endpoint of second segment is still in Q2
    if ((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) >= 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0) ||...
            (passedX(j,i) == 1)

        %indicate that the endpoint of second segment passed x-axis
        passedX(j,i) = 1;
        %angle pendulum w.r.t. positive x-axis, (CCW positive)
        Ar(j) = (pi/2) - alpha(j);
        %angle of segment 1 with respect to positive x-axis (CW positive)
        theta1p(j,i) = theta1(j,i) - (pi/2);

        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to
        %positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = atan((l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)))/...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);

        if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
            theta12P(j,i,k) = theta12P(j,i,k) + pi;
        end

        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = -theta12P(j,i,k);

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
```

```
        theta4(j,i,k) = real(-log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator, respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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((l4*sin(log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*...
                exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*1i)*1i)))*1i) - l12(j,i,k)*...
                cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
        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((((- l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*...
        exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) + l1*r*...
        exp(Ar(j)*2i) - 2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - 2*l2*l4*...
        exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 2*l3*l4*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i))*(- l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l2^2*...
        exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l4^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) +...
        2*l2*l3*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + 2*l2*l4*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + 2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i)))^(1/2) -...
        l1^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) - l2^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l4^2*...
```

```
        exp(A1(j,i)*1i)*exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) + 2*l3*l4*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i))/(2*(l1*l2*exp(Ar(j)*1i) - l2*r*exp(A1(j,i)*1i))))*1i);

    theta2f(j,i) = (pi/2) - real(-log((- ((- l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l4^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) - 2*l2*l3*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) - 2*l2*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i))*(- l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l4^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) + 2*l2*l3*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) + 2*l2*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        2*l3*l4*exp(A1(j,i)*1i)*exp(Ar(j)*1i)))^(1/2) - l1^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) - l2^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l3^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) + l4^2*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i) - r^2*exp(A1(j,i)*1i)*exp(Ar(j)*1i) +...
        l1*r*exp(A1(j,i)*2i) + l1*r*exp(Ar(j)*2i) + 2*l3*l4*exp(A1(j,i)*1i)*...
        exp(Ar(j)*1i))/(2*(l1*l2*exp(Ar(j)*1i) - l2*r*exp(A1(j,i)*1i))))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta20(j,i) - theta20(j,i-1)) > pi
        theta20(j,i) = theta20(j,i) - 2*pi;
    end

    %prevent the upperbound of segment 2 from being smaller than
    %the lowerbound
    if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %define boundaries segment 2 sweep
    BEGIN2(j,i) = theta20(j,i);
    END2(j,i) = theta2f(j,i);
    %define stepsize segment 2 sweep
    STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;

    %start angle of segment 2 equal to lowerbound,increase with stepsize
    theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;

    %angle segment 2 w.r.t. positive x-axis, (CCW positive)
    A2(j,i,k) = (pi/2) - theta2(j,i,k);

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle imaginary connection line origin and endpoint segment 2
    phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
        (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

    %...and the same angle calculated by using other variables
    phi12v(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if the node at the end of the second segment is located left to the
    %positive y-axis
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0
        phi12(j,i,k) = (pi/2) - phi12v(j,i,k);
    end

    %if endpoint of second segment is in Q4
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle imaginary connection line origin and endpoint segment 2
        phi12(j,i,k) = atan(abs(l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k)))/...
            abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi;
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (phi12(j,i,k)-phi12(j,i,k-1)) > pi
        phi12(j,i,k) = phi12(j,i,k) - 2*pi;
```

```matlab
end

%angle of segment 3 and segment 4, for given precision point &
%angle segment 1 & angle segment 2
theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*cos(phi12(j,i,k)) -...
    r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
    l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
    (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
    l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
    r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
    l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
    l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
    l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*...
    (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
    l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*...
    exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
    l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
    l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
    exp(phi12(j,i,k)*1i))/...
    (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);

%angle imaginary connection line origin and endpoint segment 2
if phi12(j,i,k) > pi/2
    %angle connection line origin and endpoint segment 2
    Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) +...
        l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if endpoint of second segment is in Q4
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/...
            abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi/2;
    end

    %angle of segment 3 and segment 4, for given precision point &
    %angle segment 1 & angle segment 2
    theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
        exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
        l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));


    theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
```

```
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - 2*l3*l4*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))*(l12(j,i,k)*r -...
            l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
            r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
            2*l3*l4*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
            exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);

        if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
            theta4(j,i,k) =  2*pi + real(-log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
                exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
                l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
                (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);
        end

        %calculate the deviations in x and y of the coordinates of the compensator,
             respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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((l4*sin(log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*...
                exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
                l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
        end

    end

    if phi12(j,i,k) < 0
        %angle pendulum w.r.t. positive x-axis, (CCW positive)
        Ar(j) = (pi/2) - alpha(j);
        %angle of segment 1 with respect to positive x-axis (CW positive)
        theta1p(j,i) = theta1(j,i) - (pi/2);

        %angle of imaginary connection (between the origin and the
```

```
        %node at the end of the second segment) with respect to
        %positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = - phi12(j,i,k);

        %angle of segment 3 and segment 4, for given precision point &
        %angle segment 1 & angle segment 2
        theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));


        theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

        %calculate the deviations in x and y of the coordinates of the compensator,
             respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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((l4*sin(log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*...
                exp(Ar(j)*1i)*1i - l4*r*exp(Ar(j)*2i)*...
                exp(theta12P(j,i,k)*1i)*1i)))*1i) - ...
                l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
        end
```

```
    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
            cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
            l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*...
            exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*...
            (l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
            l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l3^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
            r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

        if theta3(j,i,k) > pi
            theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*...
                r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
                exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
        end
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
        theta4(j,i,k) = 2*pi + theta4(j,i,k);
    end

end

%in the case of a horizontally positioned segment 1, MATLAB solve() has
%troubles finding a solution... Therefore, perturb by small amount to solve
if theta1(j,i) == pi/2
   theta1(j,i) = pi/2 + STEP1(j);
end

%the expressions within this loop are valid for theta1 > pi/2
if theta1(j,i) > pi/2
    %angle pendulum w.r.t. positive x-axis, (CCW positive)
    Ar(j) = (pi/2) - alpha(j);
    %angle of segment 1 with respect to positive x-axis (CW positive)
    theta1p(j,i) = theta1(j,i) - (pi/2);

    %lowerbound and upperbound of segment 2, respectively,
    %for given precision point and angle of segment 1
    theta20(j,i) = real(-log(-(l1*r - ((l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
```

```
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*2i))/(2*(l1*l2*exp(Ar(j)*1i)*1i -...
        l2*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*1i)*1i)))*1i);


    theta2f(j,i) = real(-log(-(l1*r + ((l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) -...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) - 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i))*(l1*r - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l2*l3*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + 2*l2*l4*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l1*r*exp(Ar(j)*2i)*exp(theta1p(j,i)*2i)))^(1/2) - l1^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - l2^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        l3^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(theta1p(j,i)*1i) - r^2*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(theta1p(j,i)*1i) + l1*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*2i))/...
        (2*(l1*l2*exp(Ar(j)*1i)*1i - l2*r*exp(Ar(j)*2i)*...
        exp(theta1p(j,i)*1i)*1i)))*1i);

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if (i>1) && (theta2f(j,i) - theta2f(j,i-1)) < -pi
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %prevent the upperbound of segment 2 from being
    %smaller than the lowerbound
    if theta2f(j,i) < (theta20(j,i) - 0.1*pi/180)
        theta2f(j,i) = theta2f(j,i) + 2*pi;
    end

    %define boundaries segment 2 sweep
    BEGIN2(j,i) = theta20(j,i);
    END2(j,i) = theta2f(j,i);
    %define stepsize segment 2 sweep
    STEP2(j,i) = (END2(j,i)-BEGIN2(j,i))/N2;

    %start angle of segment 2 equal to lowerbound,increase with stepsize
    theta2(j,i,k) = BEGIN2(j,i) + STEP2(j,i)*k;

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle of imaginary connection (between the origin and the
    %node at the end of the second segment) with respect to positive x-axis
    %(clockwise positive)
    theta12P(j,i,k) = atan((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))) - (pi/2);

    if (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0
        theta12P(j,i,k) = theta12P(j,i,k) + pi;
    end

    %if endpoint of second segment is in Q4
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle of imaginary connection (between the origin and the
        %node at the end of the second segment) with respect to
        %positive x-axis
        %(clockwise positive)
        theta12P(j,i,k) = -atan(abs(l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/...
            abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) - pi;
    end

```

```
%compensate for erroneous results due to periodicity of the loop
%closure equations
if k>1 && abs(theta12P(j,i,k)-theta12P(j,i,k-1)) > pi
    theta12P(j,i,k) = theta12P(j,i,k) + 2*pi;
end

%angle imaginary connection line origin and endpoint segment 2
phi12(j,i,k) = -theta12P(j,i,k);

%angle of segment 3 and segment 4, for given precision point &
%angle segment 1 & angle segment 2
theta3(j,i,k) = real(asin((l4*sin(log(-(l12(j,i,k)*r +...
    ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) +...
    l3^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) -...
    2*l3*l4*exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*...
    exp(Ar(j)*2i)*exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
    exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
    exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
    exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
    l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) - l12(j,i,k)*...
    cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));

theta4(j,i,k) = real(-log(-(l12(j,i,k)*r + ((l12(j,i,k)*r - l12(j,i,k)^2*...
    exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
    exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
    exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
    exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
    exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
    exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
    l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i);

%if endpoint segment 2 is in Q3
if theta12P(j,i,k) <= 0 && theta12P(j,i,k) > -pi/2
    %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);
    %angle segment 2 w.r.t. positive x-axis, (CCW positive)
    A2(j,i,k) = (pi/2) - theta2(j,i,k);
    %angle imaginary connection line origin and endpoint segment 2
    phi12(j,i,k) = atan((l1*sin(A1(j,i)) + l2*sin(A2(j,i,k)))/...
        (l1*cos(A1(j,i)) + l2*cos(A2(j,i,k))));

    %angle of segment 3 and segment 4, for given precision point &...
    %angle segment 1 & angle segment 2
    theta3(j,i,k) = pi/2 - real(pi - acos((l12(j,i,k)*...
        cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*...
        r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*...
        exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
        2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*...
        r*exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
        l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*...
        exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
        exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/...
        (2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
        l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
```

```
        theta4(j,i,k) = pi/2 - real(-log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) +...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*...
            r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
            l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) +...
            l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
            exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
            l4*r*exp(phi12(j,i,k)*1i))))*1i);

        %compensate for erroneous results due to periodicity of the loop
        %closure equations
        if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
            theta4(j,i,k) = 2*pi + pi/2 - real(-log(-(((l12(j,i,k)*r*...
                exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*...
                1i) + 2*l3*l4*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i)))^(1/2) -...
                l12(j,i,k)*r*exp(Ar(j)*2i) - l12(j,i,k)*r*...
                exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*...
                1i) + l4^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*...
                exp(Ar(j)*1i) - l4*r*exp(phi12(j,i,k)*1i))))*1i);
        end

        %calculate the deviations in x and y of the coordinates of the compensator,
         respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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) = 2*pi + pi/2 - real(pi + acos((l12(j,i,k)*...
                cos(phi12(j,i,k)) - r*cos(Ar(j)) + l4*cos(log(-(((l12(j,i,k)*...
                r*exp(Ar(j)*2i) + l12(j,i,k)*r*exp(phi12(j,i,k)*2i) -...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*exp(Ar(j)*2i) -...
                l12(j,i,k)*r*exp(phi12(j,i,k)*2i) + l12(j,i,k)^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) - l3^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));

            if theta3(j,i,k) > pi
                theta3(j,i,k) = pi/2 - real(pi + acos((l12(j,i,k)*...
                    cos(phi12(j,i,k)) - r*cos(Ar(j)) +...
                    l4*cos(log(-(((l12(j,i,k)*r*exp(Ar(j)*2i) + l12(j,i,k)*r...
                    *exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
                    exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
                    exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                    exp(phi12(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
                    exp(phi12(j,i,k)*1i))*(l12(j,i,k)*r*exp(Ar(j)*2i) +...
                    l12(j,i,k)*r*exp(phi12(j,i,k)*2i) - l12(j,i,k)^2*...
                    exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
                    exp(phi12(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
```

```
                exp(phi12(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i)))^(1/2) - l12(j,i,k)*r*...
                exp(Ar(j)*2i) - l12(j,i,k)*r*exp(phi12(j,i,k)*2i) +...
                l12(j,i,k)^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) -...
                l3^2*exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + l4^2*...
                exp(Ar(j)*1i)*exp(phi12(j,i,k)*1i) + r^2*exp(Ar(j)*1i)*...
                exp(phi12(j,i,k)*1i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i) -...
                l4*r*exp(phi12(j,i,k)*1i))))*1i))/l3));
        end
    end

end


if theta12P(j,i,k) < 0 && theta12P(j,i,k) < -pi/2

    %theta1n(j,i) is used instead of theta1(j,i) for practical reasons
    theta1n(j,i) = - theta1(j,i);

    %length of imaginary connection line between origin and end of segment 2
    l12(j,i,k) = sqrt((l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))^2 +...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)))^2);

    %angle connection line origin and endpoint segment 2
    Mtheta12(j,i,k) = - atan((l1*sin(theta1(j,i)) +...
        l2*sin(theta2(j,i,k)))/...
        (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))));

    %if endpoint of second segment is still in Q4
    if (l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k))) < 0 &&...
            (l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k))) < 0

        %angle connection line origin and endpoint segment 2
        Mtheta12(j,i,k) = atan(abs(l1*cos(theta1(j,i)) +...
            l2*cos(theta2(j,i,k)))/...
            abs(l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)))) + pi/2;
    end

    %angle of segment 3 and segment 4
    %for given precision point & angle segment 1 & angle segment 2
    theta3(j,i,k) =  real(asin((l4*sin(log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
        exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))).^(1/2) -...
        l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
        l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        l4^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) -...
        r^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
        (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
        l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));


    theta4(j,i,k) =  real(-log(-(l12(j,i,k)*r +...
        ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
        exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
        exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) - l12(j,i,k)^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
        exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
        exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
```

```
            (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
            exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i);

        %calculate the deviations in x and y of the coordinates of
        %the compensator,respectively
        DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
            l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));

        DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
            l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

        %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((l4*sin(log(-(l12(j,i,k)*r +...
                ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) + l3^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) + l4^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) - r^2*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) - 2*l3*l4*exp(Mtheta12(j,i,k)*1i)*...
                exp(alpha(j)*1i) + l12(j,i,k)*r*exp(Mtheta12(j,i,k)*2i)*...
                exp(alpha(j)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l3^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l4^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + 2*l3*l4*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i)))^(1/2) -...
                l12(j,i,k)^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) +...
                l3^2*exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - l4^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) - r^2*...
                exp(Mtheta12(j,i,k)*1i)*exp(alpha(j)*1i) + l12(j,i,k)*r*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*2i))/...
                (2*(l4*r*exp(Mtheta12(j,i,k)*1i) - l12(j,i,k)*l4*...
                exp(Mtheta12(j,i,k)*2i)*exp(alpha(j)*1i))))*1i) +...
                l12(j,i,k)*sin(Mtheta12(j,i,k)) + r*sin(alpha(j)))/l3));
        end

    end

    %calculate the deviations in x and y of the coordinates of the compensator, respectively
    DEV1(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
        l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
    DEV2(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
        l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

    %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((l4*sin(log(-(l12(j,i,k)*r +...
            ((l12(j,i,k)*r - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))*(l12(j,i,k)*r - l12(j,i,k)^2*...
            exp(Ar(j)*1i)*exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + 2*l3*l4*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i)))^(1/2) - l12(j,i,k)^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l3^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - l4^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) - r^2*exp(Ar(j)*1i)*...
            exp(theta12P(j,i,k)*1i) + l12(j,i,k)*r*exp(Ar(j)*2i)*...
            exp(theta12P(j,i,k)*2i))/(2*(l12(j,i,k)*l4*exp(Ar(j)*1i)*1i -...
            l4*r*exp(Ar(j)*2i)*exp(theta12P(j,i,k)*1i)*1i)))*1i) -...
            l12(j,i,k)*cos(theta12P(j,i,k)) + r*cos(Ar(j)))/l3));
    end

    %compensate for erroneous results due to periodicity of the loop
    %closure equations
    if k>1 && (abs(theta4(j,i,k)-theta4(j,i,k-1)) > pi)  %#ok<*COMPNOT>
        theta4(j,i,k) = 2*pi + theta4(j,i,k);
    end

end

%calculate the deviations in x and y of the coordinates of the compensator, respectively
DEV11(j,i,k) = l1*sin(theta1(j,i)) + l2*sin(theta2(j,i,k)) +...
    l3*sin(theta3(j,i,k)) + l4*sin(theta4(j,i,k)) - r*sin(alpha(j));
```

```
DEV22(j,i,k) = l1*cos(theta1(j,i)) + l2*cos(theta2(j,i,k)) +...
    l3*cos(theta3(j,i,k)) + l4*cos(theta4(j,i,k)) - r*cos(alpha(j));

%calculate the distance from the endpoint of the second segment to the end
%effector of the inverted pendulum
d(j,i,k) = sqrt((r*sin(alpha(j))-l12(j,i,k)*cos(phi12(j,i,k)))^2 +...
    (r*cos(alpha(j))-l12(j,i,k)*sin(phi12(j,i,k)))^2);

%check condition upper loop closure
if (l4-l3-d(j,i,k)) > 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count2" instead
    Count2 = Count2 + 1;
end

%if segment 1 and segment 2 are not at their lowerbound
if i>1 && k>1
    %if the angle of the third segment was previously - for the same angle
    %of the pendulum - NaN, then it will remain NaN for this angle of the
    %pendulum (infeasible solution space)
    if (isnan(theta3(j,i,k-1)) == 1) || (isnan(theta3(j,i-1,k)) == 1)       %#ok<COMPNOP>
        theta3(j,i,k) = NaN;

        %the potential energy and the angle of segment 4 should
        %consequently be NaN as well
        V(j,i,k) = NaN;
        theta4(j,i,k) = NaN;
    end
end

%check condition upper loop closure
if l4-l3+d(j,i,k) < 0
    %set the deviation in x...
    DEV11(j,i,k) = 0;
    %... and y to zero such that this scenario won't be flagged
    DEV22(j,i,k) = 0;
    %posture doesn't exist, so potential energy not a number
    V(j,i,k) = NaN;

    %define the angles of the third and fourth segment to be no value;
    %the surface plots of these tensors (used for debugging) would
    %otherwise be nonsmooth
    theta3(j,i,k) = NaN;
    theta4(j,i,k) = NaN;
    %flag this event with variable "Count3" instead
    Count3 = Count3 + 1;
end

%if the absolute value of any of these deviations transcends a
%certain threshold, then increase the variable "Count" by one
if abs(DEV11(j,i,k)) > 10^-10 || abs(DEV22(j,i,k)) > 10^-10
    Count = Count + 1;
end

%initial relative angle of segment 1
alpha10 = theta1i;
%initial relative angle of segment 2
alpha20 = theta2i - theta1i;
%initial relative angle of segment 3
alpha30 = theta3i - theta2i;
%initial relative angle of segment 4
alpha40 = theta4i - theta3i;

%angle of rotation torsion spring 1
alpha1(j,i) = theta1(j,i) - alpha10;
%angle of rotation torsion spring 2
alpha2(j,i,k) = theta2(j,i,k) - theta1(j,i) - alpha20;
%angle of rotation torsion spring 3
alpha3(j,i,k) = theta3(j,i,k) - theta2(j,i,k) - alpha30;
%angle of rotation torsion spring 4
alpha4(j,i,k) = theta4(j,i,k) - theta3(j,i,k) - alpha40;

if nonlinearity == 0
```

```
    %internal moment spring 1
    M1(j,i) = k1*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = k2*alpha2(j,i,k) + M02;
    %internal moment spring 3
    M3(j,i,k) = k3*alpha3(j,i,k) + M03;
    %internal moment spring 4
    M4(j,i,k) = k4*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = ((k1/2)*alpha1(j,i)^2);
    %potential energy spring 2
    V2(j,i,k) = ((k2/2)*alpha2(j,i,k)^2) + M02*alpha2(j,i,k) +...
        ((k2/2)*(M02/k2)^2);
    %potential energy spring 3
    V3(j,i,k) = ((k3/2)*alpha3(j,i,k)^2) + M03*alpha3(j,i,k) +...
        ((k3/2)*(M03/k3)^2);
    %potential energy spring 4
    V4(j,i,k) = ((k4/2)*alpha4(j,i,k)^2);
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

if nonlinearity == 1
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar1M2 = (-B + sqrt(B^2 + 4*M02*A))/(2*A);
    %second solution prestress angle: angle of rotation corresponding to
    %prestress spring 2
    alphastar2M2 = (-B - sqrt(B^2 + 4*M02*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M2 < 0
        alphastar1M2 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M2 < 0
        alphastar2M2 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM2"
    alphastarsM2 = [alphastar1M2,alphastar2M2];

    %store the smallest solution for the prestress angle
    alphastarM2 = min(abs(alphastarsM2));

    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar1M3 = (-B + sqrt(B^2 + 4*M03*A))/(2*A);
    %first solution prestress angle: angle of rotation corresponding to
    %prestress spring 3
    alphastar2M3 = (-B - sqrt(B^2 + 4*M03*A))/(2*A);

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar1M3 < 0
        alphastar1M3 = NaN;
    end

    %allow only for nonnegative solutions; set to NaN if negative
    if alphastar2M3 < 0
        alphastar2M3 = NaN;
    end

    %store solutions prestress angle in array called "alphastarsM3"
    alphastarsM3 = [alphastar1M3,alphastar2M3];

    %store the smallest solution for the prestress angle
    alphastarM3 = min(abs(alphastarsM3));

    %internal moment spring 1
    M1(j,i) = A*alpha1(j,i)^2 + B*alpha1(j,i);
    %internal moment spring 2
    M2(j,i,k) = A*(alpha2(j,i,k)+alphastarM2)^2 +...
        B*(alpha2(j,i,k)+alphastarM2);
    %internal moment spring 3
    M3(j,i,k) = A*(alpha3(j,i,k)+alphastarM3)^2 +...
        B*(alpha3(j,i,k)+alphastarM3);
    %internal moment spring 4
    M4(j,i,k) = A*alpha4(j,i,k)^2 + B*alpha4(j,i,k);

    %potential energy spring 1
    V1(j,i) = (A/3)*alpha1(j,i)^3 + (B/2)*alpha1(j,i)^2;
    %potential energy spring 2
```

```
    V2(j,i,k) = (A/3)*(alpha2(j,i,k)+alphastarM2)^3 +...
        (B/2)*(alpha2(j,i,k)+alphastarM2)^2;
    %potential energy spring 3
    V3(j,i,k) = (A/3)*(alpha3(j,i,k)+alphastarM3)^3 +...
        (B/2)*(alpha3(j,i,k)+alphastarM3)^2;
    %potential energy spring 4
    V4(j,i,k) = (A/3)*alpha4(j,i,k)^3 + (B/2)*alpha4(j,i,k)^2;
    %total potential energy
    V(j,i,k) = V1(j,i) + V2(j,i,k) + V3(j,i,k) + V4(j,i,k);
end

%allow only for nonnegative solutions; set to NaN if negative
if alpha2(j,i,k) < 0
    V(j,i,k) = NaN;
end

%allow only for nonnegative solutions; set to NaN if negative
if alpha3(j,i,k) < 0
    V(j,i,k) = NaN;
end

%x - coordinate origin (and first spring)
x0      = 0;
%y - coordinate origin (and first spring)
y0      = 0;
%x - coordinate 2nd spring
x1(j,i) = l1*sin(theta1(j,i));
%y - coordinate 2nd spring
y1(j,i) = l1*cos(theta1(j,i));
%x - coordinate 3rd spring
x2(j,i,k) = x1(j,i) + l2*sin(theta2(j,i,k));
%y - coordinate 3rd spring
y2(j,i,k) = y1(j,i) + l2*cos(theta2(j,i,k));
%x - coordinate 4th spring
x3(j,i,k) = x2(j,i,k) + l3*sin(theta3(j,i,k));
%y - coordinate 4th spring
y3(j,i,k) = y2(j,i,k) + l3*cos(theta3(j,i,k));
%x - coordinate end effector
x4(j,i,k) = x3(j,i,k) + l4*sin(theta4(j,i,k));
%y - coordinate end effector
y4(j,i,k) = y3(j,i,k) + l4*cos(theta4(j,i,k));

%magnitude reaction force y-direction
F1yt(j,i,k) = (M1(j,i) - M4(j,i,k) + (-M4(j,i,k)/(l4*cos(theta4(j,i,k))))*...
    (l1*cos(theta1(j,i))+l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k))))/...
    (-tan(theta4(j,i,k))*(l1*cos(theta1(j,i))+...
    l2*cos(theta2(j,i,k))+l3*cos(theta3(j,i,k)))...
    + (l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k))+l3*sin(theta3(j,i,k))));

%magnitude reaction force x-direction
F1xt(j,i,k) = (-M4(j,i,k) + F1yt(j,i,k)*l4*sin(theta4(j,i,k)))/...
    (l4*cos(theta4(j,i,k)));

%external moment on second spring (node 2)
M2lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*l1*cos(theta1(j,i)) -...
    F1yt(j,i,k)*l1*sin(theta1(j,i));

%external moment on third spring (node 3)
M3lt(j,i,k) = M1(j,i) + F1xt(j,i,k)*(l1*cos(theta1(j,i))+...
    l2*cos(theta2(j,i,k))) -...
    F1yt(j,i,k)*(l1*sin(theta1(j,i))+l2*sin(theta2(j,i,k)));
end


end
end
end
end


%find the minimum value of the potential energy for each precision point
%and store the linear index
[Vmin,I] = min(V,[],[2 3],"linear");

%convert linear index to j,i,k indices
ind = I;                                                %linear index
sz = [M N1 N2];                                         %size of the V tensor
%convert the linear index into 3 indices for j, i & k
[I1,I2,I3] = ind2sub(sz,ind);

%coordinates nodes in initial (relaxed) configuration

%x - coordinate origin (and first spring)
x00 = 0;
```

```
%x - coordinate origin (and first spring)
y00 = 0;
%x - coordinate 2nd spring
x10 = l1*sin(theta1i);
%y - coordinate 2nd spring
y10 = l1*cos(theta1i);
%x - coordinate 3rd spring
x20 = x10 + l2*sin(theta2i);
%y - coordinate 3rd spring
y20 = y10 + l2*cos(theta2i);
%x - coordinate 4th spring
x30 = x20 + l3*sin(theta3i);
%y - coordinate 4th spring
y30 = y20 + l3*cos(theta3i);
%x - coordinate end effector
x40 = x30 + l4*sin(theta4i);
%y - coordinate end effector
y40 = y30 + l4*cos(theta4i);


%create new figure to plot the lowest energy configurations
figure(2);                                                    %create figure
%plot following plot commands in that same figure
hold on
axis equal
title("Lowest energy configurations")

%start a loop throughout all precision points
for j = 1:1:M %divide the 90 deg range of motion into equally sized segments
alpha(j) = (pi/2)*(j/M);

%plot connection line between spring 1 and 2 in black
plot([x0 x1(j,I2(j))],[y0 y1(j,I2(j))],'k')
%plot connection line between spring 2 and 3 in black
plot([x1(j,I2(j)) x2(j,I2(j),I3(j))],[y1(j,I2(j)) y2(j,I2(j),I3(j))],'k')
%plot connection line between spring 3 and 4 in black
plot([x2(j,I2(j),I3(j)) x3(j,I2(j),I3(j))],...
    [y2(j,I2(j),I3(j)) y3(j,I2(j),I3(j))],'k')
%plot connection line between spring 4 and end - effector in black
plot([x3(j,I2(j),I3(j)) x4(j,I2(j),I3(j))],...
    [y3(j,I2(j),I3(j)) y4(j,I2(j),I3(j))],'k')
%plot the location of the end effector of the pendulum with a circle
plot(r*sin(alpha(j)),r*cos(alpha(j)),"b--o")

%coordinates nodes in initial (relaxed) configuration

%plot connection line between spring 1 and 2 in black
plot([x00 x10],[y00 y10],'r')
%plot connection line between spring 2 and 3 in black
plot([x10 x20],[y10 y20],'r')
%plot connection line between spring 3 and 4 in black
plot([x20 x30],[y20 y30],'r')
%plot connection line between spring 4 and end - effector in black
plot([x30 x40],[y30 y40],'r')
%plot the location of the end effector of the pendulum with a circle
plot(r*sin(0),r*cos(0),"r--o")

%minimum value of alpha1 per precision point
alpha1m(j) = alpha1(j,I2(j));
%minimum value of alpha2 per precision point
alpha2m(j) = alpha2(j,I2(j),I3(j));
%minimum value of alpha3 per precision point
alpha3m(j) = alpha3(j,I2(j),I3(j));
%minimum value of alpha4 per precision point
alpha4m(j) = alpha4(j,I2(j),I3(j));

if prestress == 0 && nonlinearity == 0
    %minimum moment in torsion spring 1 per precision point
    M1m(j) = k1*alpha1(j,I2(j));
    %minimum moment in torsion spring 2 per precision point
    M2m(j) = k2*alpha2(j,I2(j),I3(j));
    %minimum moment in torsion spring 3 per precision point
    M3m(j) = k3*alpha3(j,I2(j),I3(j));
    %minimum moment in torsion spring 4 per precision point
    M4m(j) = k4*alpha4(j,I2(j),I3(j));
end

%if springs are nonlinear
if prestress == 0 && nonlinearity == 1
    %minimum moment in torsion spring 1 per precision point
    M1m(j) = A*alpha1(j,I2(j))^2 + B*alpha1(j,I2(j));
    %minimum moment in torsion spring 2 per precision point
    M2m(j) = A*alpha2(j,I2(j),I3(j))^2 + B*alpha2(j,I2(j),I3(j));
    %minimum moment in torsion spring 3 per precision point
```

```
    M3m(j) = A*alpha3(j,I2(j),I3(j))^2 + B*alpha3(j,I2(j),I3(j));
    %minimum moment in torsion spring 4 per precision point
    M4m(j) = A*alpha4(j,I2(j),I3(j))^2 + B*alpha4(j,I2(j),I3(j));
end

%if springs are prestressed
if prestress == 1 && nonlinearity == 0
    %minimum moment in torsion spring 1 per precision point
    M1m(j) = k1*alpha1(j,I2(j));
    %minimum moment in torsion spring 2 per precision point
    M2m(j) = k2*alpha2(j,I2(j),I3(j)) + M02;
    %minimum moment in torsion spring 3 per precision point
    M3m(j) = k3*alpha3(j,I2(j),I3(j)) + M03;
    %minimum moment in torsion spring 4 per precision point
    M4m(j) = k4*alpha4(j,I2(j),I3(j));
end

%if springs are prestressed and nonlinear
if prestress == 1 && nonlinearity == 1
    %minimum moment in torsion spring 1 per precision point
    M1m(j) = A*alpha1(j,I2(j))^2 + B*alpha1(j,I2(j));
    %minimum moment in torsion spring 2 per precision point
    M2m(j) = A*(alpha2(j,I2(j),I3(j))+alphastarM2)^2 +...
        B*(alpha2(j,I2(j),I3(j))+alphastarM2);
    %minimum moment in torsion spring 3 per precision point
    M3m(j) = A*(alpha3(j,I2(j),I3(j))+alphastarM3)^2 +...
        B*(alpha3(j,I2(j),I3(j))+alphastarM3);
    %minimum moment in torsion spring 4 per precision point
    M4m(j) = A*alpha4(j,I2(j),I3(j))^2 + B*alpha4(j,I2(j),I3(j));
end

%minimum value of theta1 per precision point
theta1m(j) = theta1(j,I2(j));
%minimum value of theta2 per precision point
theta2m(j) = theta2(j,I2(j),I3(j));
%minimum value of theta3 per precision point
theta3m(j) = theta3(j,I2(j),I3(j));
%minimum value of theta4 per precision point
theta4m(j) = theta4(j,I2(j),I3(j));

if objective == "sinus"
    Vm(j) = mg*r*cos(alpha(j));          %potential energy : height energy
    Mobj(j) = mg*r*sin(alpha(j));%the moment around the origin caused by mass
end

if objective == "Laevo"
    Vm(j) = 0.05022*alpha(j)^5 - 0.33575*alpha(j)^4 +...
        0.97*alpha(j)^3 - 1.412*alpha(j)^2 + 0.006501*alpha(j) + 1;
    Mobj(j) = -0.2511*alpha(j)^4 + 1.343*alpha(j)^3 -...
        2.91*alpha(j)^2 + 2.824*alpha(j) - 0.006501;
end

if objective == "stiffening"
    Vm(j) = sin(alpha(j)) - alpha(j);
    Mobj(j) = -cos(alpha(j))+1;
end

if objective == "sqrt"
    Vm(j) = - (2/3)*alpha(j)^(3/2);
    Mobj(j) = sqrt(alpha(j));
end

if objective == "quadratic"
    Vm(j) = - (1/3)*alpha(j)^(3);
    Mobj(j) = alpha(j)^2;
end

if objective == "hardening-softening"
    Vm(j) = 0.25*cos(2*alpha(j)-pi/2) - 0.5*alpha(j);
    Mobj(j) = (sin(2*alpha(j)-pi/2)+1)/2;
end

if objective == "hardening-softening2"
    Vm(j) = -0.5*alpha(j)+(-0.333333+0.424413*alpha(j))*...
        atan(2.41421-3.07387*alpha(j))+...
        0.0690356*log(9.8696-21.4521*alpha(j)+13.6569*alpha(j)^2);
    Mobj(j) = 0.5 + (4/(3*pi))*atan( tan((3*pi)/8)*((4/pi)*alpha(j)-1));
end

if objective == "softening-hardening"
    Vm(j) = 0.5*log(cos(alpha(j)-pi/4)) - 0.5*alpha(j);
    Mobj(j) = 0.5*tan(alpha(j)-pi/4) + 0.5;
end
```

```
if objective == "softening-hardening2"
    Vm(j) = -0.5*alpha(j) - 0.0690356*log(1 + tan(1.1781 - 1.5*alpha(j))^2);
    Mobj(j) = 0.5*tan(1.5*(alpha(j)-pi/4))/tan(1.5*(pi/4))+0.5;
end

if objective == "sinuspi"
    Vm(j) = 0.5*cos(2*alpha(j));
    Mobj(j) = sin(2*alpha(j));
end

%vertical reaction force at segment 1 (positive upwards)
F1y(j) = (M1m(j) - M4m(j) + (-M4m(j)/(l4*cos(theta4m(j)))))*...
    (l1*cos(theta1m(j))+l2*cos(theta2m(j))+l3*cos(theta3m(j))))/...
    (-tan(theta4m(j))*(l1*cos(theta1m(j))+...
    l2*cos(theta2m(j))+l3*cos(theta3m(j)))...
    + (l1*sin(theta1m(j))+l2*sin(theta2m(j))+l3*sin(theta3m(j))));

%horizontal reaction force at segment 1 (positive to the right)
F1x(j) = (-M4m(j) + F1y(j)*l4*sin(theta4m(j)))/(l4*cos(theta4m(j)));

%the external load (moment) on nodes 2 and 3...
%...(where springs 2 and 3 are located), respectively
M2l(j) = M1m(j) + F1x(j)*l1*cos(theta1m(j)) - F1y(j)*l1*sin(theta1m(j));
M3l(j) = M1m(j) + F1x(j)*(l1*cos(theta1m(j))+l2*cos(theta2m(j))) -...
    F1y(j)*(l1*sin(theta1m(j))+l2*sin(theta2m(j)));
end

%print the root mean square error (objective function)
e = sqrt(mean((M1m - Mobj).^2))   %#ok<NOPTS>
%integrate the residual moment - angle plot to obtain the required work
IntM = trapz(alpha,abs(M1m-Mobj));
%integrate the original moment - angle plot to obtain the required work
IntMo = trapz(alpha,Mobj);

%arrays with data exported from SAM
%moment and potential energy, respectively
MSAM = [-0.03520,-0.05665,-0.07593,-0.09171,-0.11565,-0.12151,-0.12093,...
    -0.11309,-0.09704,-0.07212,-0.03772,0.00668,0.06148,0.12702,0.20346];
VSAM = [0.99866,0.99414,0.98723,0.97789,0.96674,0.95425,0.94145,0.92896,...
    0.91733,0.90790,0.90161,0.89939,0.90228,0.91169,0.92878];

figure(3)
hold on
plot(alpha*180/pi,M1m)
plot(alpha*180/pi,Mobj)
plot(alpha*180/pi,M1m - Mobj)
xlabel("Angle of rotation pendulum (deg)")
ylabel("Moment around suspension-point 1 (Nm)")
legend("Moment in spring 1", "Moment objective", "Error in moment",...
    "location","northwest")

figure(4)
hold on
plot(alpha*180/pi,Vmin+transpose(Vm))
xlabel("Angle of rotation pendulum (deg)")
ylabel("Total potential energy in system (J)")

if prestress == 1
    figure(5)
    hold on
    plot(alpha*180/pi,F1x)
    plot(alpha*180/pi,F1y)
    xlabel("Angle of rotation pendulum (deg)")
    ylabel("Reaction force in point 1 (N)")
    legend("F1x","F1y","location","northwest")

    figure(6)
    hold on
    plot(alpha*180/pi,M2m)
    plot(alpha*180/pi,M3m)
    plot(alpha*180/pi,M2l)
    plot(alpha*180/pi,M3l)
    xlabel("Angle of rotation pendulum (deg)")
    ylabel("Reaction moment in nodes (Nm)")
    legend("M2m","M3m","M2l","M3l","location","northwest")
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 equilibrator 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 (8715492085566) | 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