Improving Model-Based Control of a Soft Robot via Gaussian Process Regression

Emilio Tavio y Cabrera

Department of Cognitive Robotics
Delft University of Technology
The Netherlands
Date of Defense: March 15, 2023
Supervisor: C. Della Santina

Improving Model-Based Control of a Soft Robot via Gaussian Process Regression

Emilio Tavio y Cabrera

***Abstract*—Soft robots have the potential to accelerate robotization in areas that are complex and impractical for hard robots. The use of soft materials results in a safe and flexible design that is unattainable for hard robots. However, this attribute results in the need for new control approaches and strategies. Hybrid controllers are a relative unexplored type of controllers that consist of a model-based controller part and a learning part to correct the model-based controller. A hybrid controller benefit by the unrequired need for accurate system identification. Simultaneously, the learning effort is reduced by the preliminary work of the model-based component.**

In this project, a model-based feedforward controller is proposed and compared with a hybrid controller consisting of the same model-based controller enhanced with a Gaussian process to reduce the end-point error in the bending angle. The controllers are tested using a crafted 2-segment pneumatic silicone soft robot, following a circular trajectory with different radii.

The results of this new control strategy highlights the potential benefits of adding a learning approach to a model-based controller to reduce model errors. Using a relative small dataset preserves a computational usable Gaussian process. The small dataset remains effective by reducing the range of the training data.

***Index Terms*—Soft robot, continuum robot, model-based control, hybrid control, supervised learning, Gaussian process**

I. Introduction

Soft robots are a new type of robots that can revolutionize the integration of robots into society due to their flexibility and safe materials. Soft or continuum robots are, inspired by nature, working in a more natural way. These robots are made of flexible and soft materials such as silicone and rubber and are often oblong shaped to mimic the trunk of an elephant, the body of a snake or the arm of an octopus. Unlike rigid robots, soft robots are flexible, compliant, and organically shaped. The flexibility and softness makes a soft robot more versatile and adaptable to unstructured environments compared to its rigid counterpart. A soft robot is also by design safer to operate among humans due to the compliant nature of the soft materials. Therefore, soft robot have the potential to excel in areas where rigid robots exhibit poor performance, including unstructured and changing environments and human robot interactions. However, because of their compliant nature, the behaviour of these systems is described by infinite-dimensional and highly nonlinear equations. Therefore, the design of controllers for these systems can be significantly more complex than the strategies used for rigid robots, representing one of the current bottlenecks in this area.

The Department of Cognitive Robotics (CoR), Delft Univeristy of Technology, the Netherlands.

A. Mathematical Description of Soft Robots

For classic rigid robots, the amount of links and joints are finite and thus limiting the degrees of freedom by design. On the contrary, the flexible body of a soft robot enables the robot an infinite amount of postures that would lead to infinite degrees of freedom. However, multiple resembling solutions to limit the amount of parameters are listed below.

*1) **Cosserat Rod Theory**:* Considering that most soft robots are round and linear shaped, mechanical rod models describing continuous bending are a logical choice. In [1] and [2] Cosserat rod is applied to model soft robots. In [3] an exact steady state solution is proposed via Cosserat rod models. A recent study describing and validating energy-based control based on Cosserat rod models can be found in [4].

The advantage of Cosserat Rod models is that an exact solution can be achieved for steady state [5]. Still, there is an infinite amount of states possible when using Cosserat rod theory, which makes it complex to use for control.

*2) **Finite Element Method**:* A common technique for solving the deformation of complex shaped solids is called Finite Element Method (FEM). FEM provides a numerical solution by meshing the complex shape into point masses and connecting them via springs. Using FEM in soft robots discretizes their shape, solving the issue of the infinite-dimensional models in soft robots. The use of more nodes leads to a more accurate model of the soft robot at the expense of a bigger state space and thus higher computational load. In [6], the external force acting on the soft robot is calculated by measuring the difference between the FEM calculation and the vision observed position of the soft robot. In [7], FEM is tested by using different shapes of soft robots under external load. A control framework is proposed in [8] based on hyper-elastic FEM. FEM results are accurate, however the computational bottleneck is emphasized, which complicates online control using FEM.

*3) **Piecewise Constant Curvature**:* The most prominent technique describing the state of a soft robot is Piecewise Constant Curvature (PCC). This framework defines the state of the soft robot along a finite set of arcs in series, assuming a constant bending of each arc segment. Thereby, each arc segment consists of 3 degrees of freedom [9], [10]. PCC simplifies the posture of a soft robot by assuming constant bending over the length of each individual curvature. In most cases, the amount of independent curvatures is equal to the number of segments of the soft robot. The success of PCC can be explained by the fact that most actuator types of soft robots control a segment of the robot with limited length, which in most cases bend equal over that length. Therefore,

the simplification of PCC is accurate enough while reducing the degrees of freedom from infinite to only 3. A variation on PCC is Variable Constant Curvature, which is more useful for tapered shaped soft robots [11], [12]. A drawback of PCC approximations is the presence of singularities for some postures. Such a problem is solved by defining new coordinates in [13]. This singularity-free approach is used in [14] where 3 PCC curvatures per segment are used since the used tapered shaped soft robot does not bend constantly over the length of each segment.

B. Model-Based Control

1) ***Static Control****:* A static controller can be defined as a steady-state controller that works under the assumption of force equilibrium, thus neglecting the dynamics of a soft robot. Consequently, the state parameters of the soft robot are given by its posture. The two main types of static controllers are the task space controller [15] and the joint space controller. For the former, instability can be a problem which is often solved by lowering the control loop frequency [16]. Static controllers tend to be slow or inaccurate due to the steady-state assumption.

2) ***Dynamic Control****:* A model-based dynamic controller is much more complex compared to static controllers, but have the potential to be more accurate, faster, energy efficient and smoother in movement [9]. To obtain the dynamic equations for a soft robot, generally energy-based models such as Euler-Lagrangian or Port-Hamiltonian mechanical system are used. In [17], a general control architecture is proposed based on shaping the potential energy. In [18], two architectures designed for dynamic trajectory tracking are presented, relying on the fundamental characteristics of the Lagrangian system. The controllers are validated through simulation and experiments on planar soft robots, respectively.

C. Learning-Based Control

Considering the complexity of controlling a soft robot and the relative large amount of unknown variables influencing its behaviour, a machine learning-based approach is a logical choice. Machine learning can outperform tasks that are too complex to be identified in a model. Different learning-based approaches exist and are listed below.

1) ***Static Control****:* Learning-based static controllers work via data-driven inverse kinematics. Data to learn the translation between the task space and the actuator space is acquired via sampling. In [19], an exploration algorithms was proposed for an elephant's trunk like robotic arm. The algorithm generates samples to translate from the task space to the actuator space. In [20], the inverse kinematics of the bionic handling assistant of a mobile robot called Robotino are modelled via a learning approach. A neural network is used to directly learn the forward kinematics. In [21], a convolutional neural network is used to estimate the angle of the soft robot. In [22] a learning approach for the control of a magnetic actuated soft robot is presented.

2) ***Dynamic Control****:* The dynamic control of a soft robot via a machine learning-based approach is relatively complex. For this reason, this area remains relative unexplored [9]. A DDPG algorithm consisting of an actor-critic reinforcement learning neural network is able to work with a continuous action space. In [23], a framework for deterministic policy gradient algorithms is presented. Results show that DDPG outperform its stochastic counterpart.

D. Hybrid Control

The combination of machine learning and prior model knowledge could be a good balance between model uncertainties and learning complexity. A learning-based approach provides robustness to uncertainties concerning the soft robot's parameters, while the model-based part provides a good indication of the behaviour of the soft robot, making the learning task smaller and less complex. In [24], a hybrid trajectory tracking approach is presented and validated using a CBHA robot. In [25], a Model Predictive Controller is used to control a soft robot, using a neural network that outputs the next state of the robot based on input and current state.

E. Research Objectives

The performance of model-based controllers, are sensitive to mismatches in the system's parameters which are complex to define and can even change such as the elasticity due to wear or temperature. In contrast, Learning-based approaches do suffer from long learning times and stability issues due to the complexity of the learning task.

In this applied research, a hybrid set-up consisting of a Gaussian process regression improvement of a model-based controller is proposed and tested on pneumatic soft robot. This approach benefits from being less sensitive to system parameters due to the Gaussian process learning part that learns to correct for deviations. Another advantage is that the relative straightforward learning part remains trackable and clear, in contrast to more complex black box models. The machine learning algorithm also reduces in time and amount of iterations due to the less complex learning task compared to learning the absolute control.

The research attempts to address gaps in control approaches for soft robots, and in doing so makes important contributions. First, the research combines a model-based and model-free controller, while in most studies one controller is selected. Second, the research uses a relative simple and transparent learning part by using Gaussian process regression. A search of the relevant literature revealed no similar approach. Third, the use of IMU sensors for measuring the posture of the robot is unique. Former research often uses camera systems or flex sensors.

The objectives of the project are:

- Construct a pneumatic soft robot to verify experimentally the proposed improved controller. The soft robot consists of 2 segments in series and 4 air chambers per segment, resulting in a total of 8 individual actuators. The design of the soft robot follows the design from Robert Katzschmann.

- Develop fast and reliable sensor data for measuring the current posture of the soft robot.
- Construct a simplified model-based feedforward controller via the Euler-Lagrangian equations and verify the performance.
- Acquire learning data to correct for steady state errors and improve the controller by providing it with a Gaussian progress regressor that predicts the optimal corrections.
- Analyse the performance of the improved hybrid controller and compare it with the model-based feed forward controller.

II. Preliminaries

A. Piecewise Continuous Curvature

In this project, the elongation of the soft robot is not considered. Fig. 1 shows a schematic front view of the soft robot and the parametrization.

$$q = \begin{bmatrix} \phi_1 & \theta_1 & \phi_2 & \theta_2 \end{bmatrix} \tag{1}$$

Equation (1) defines the soft robot state parameters $q \in \mathbb{R}^4$ via PCC assumption by the rotation of the bending plane ϕ and the amount of bending θ. The soft robot consists of 2 segments and its parametrization is consequently defined by 4 parameters. The implementation of PCC reduces the degrees of freedom of the 2 segment soft robot to 4.

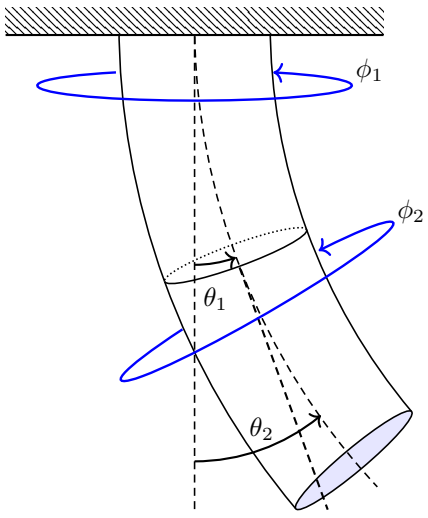


Fig. 1: Front view of the 2-segment soft robot.

B. Euler-Lagrange Dynamics

The Lagrangian of soft robots is given by

$$L \equiv T - V \tag{2}$$

where T and V represent the kinetic and potential energy, respectively. The kinetic energy can be defined by

$$T = \frac{1}{2}\dot{q}^{\mathsf{T}} M(q)\dot{q} \tag{3}$$

where $M(q) \in \mathbb{R}^{4\times4}$ represents the mass inertia matrix. The potential energy is given by

$$V = G(q) + K(q) \tag{4}$$

where $G(q) \in \mathbb{R}^4$ is the potential energy due to gravity and $K(q) \in \mathbb{R}^4$ the elastic potential energy. Moreover, the equations of motions are described as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \tau_a - D(q)\dot{q} \tag{5}$$

with $\tau_a \in \mathbb{R}^4$ the torque from the actuators and $D(q)\dot{q} \in \mathbb{R}^4$ the damping of the system. Combining equation (2) to (5) the dynamics of a soft robot can be represented as follows

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} - \left(-\frac{\partial V}{\partial q}\right) = \tau_a - D(q)\dot{q} \tag{6}$$

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D(q)\dot{q} + G(q) + K(q) = \tau_a \tag{7}$$

When the robot is stationary, the components depending on the velocity become zero and equation (7) reduces to

$$G(q) + K(q) = \tau_a \tag{8}$$

which determines the equilibrium of the system.

C. Gaussian Process Regression

A Gaussian Process Regressor (GPR) is a form of supervised learning that is able to efficient statistical predict a function based on samples or observed targets. This is achieved by updating the probabilities via observations.

The GPR involves a mean function and covariance kernel. A prediction is done by calculating the weighted mean. The covariance or kernel function returns how related observations are. A popular kernel function is the Radial Basis Function:

$$k(x,x') = e^{-\frac{1}{2\sigma^2}\|x-x'\|^2} \tag{9}$$

where σ is the length scale hyperparameter. This parameter describes how much the resulting function varies. This kernel function relates approximately similar inputs (i.e., relative small Euclidean distance in the input space) to similar outputs, while reducing the significance of distanced inputs.

III. Methodology: Soft Robot Design

The intention of this research is to investigate if a pneumatic soft robot performs accurately in endpoint position if the model-based controller is equipped with of a GPR correcting for errors in the model-based controller compared to a purely model-based controller.

A. Pneumatic Actuated Soft Robot: Design and Manufacturing

To verify and test the performance of the controllers, a pneumatic soft robot is fabricated. Fig. 2 shows a picture of the resulting 2 segment pneumatic soft robot. Each segment is made using silicone moulding technique. First, the 3D printed mould is assembled. The mould contains 4 ribs made of paraffin wax to mold the air chambers, one air chamber for each quadrant, labelled by $A1 - A4$ for segment 1 and $A5 - A8$ for segment 2 in the cross-section fig. 3. After the silicone solidified, it is removed from the mould and heated in an oven, which causes the wax inside to melt and outflow,

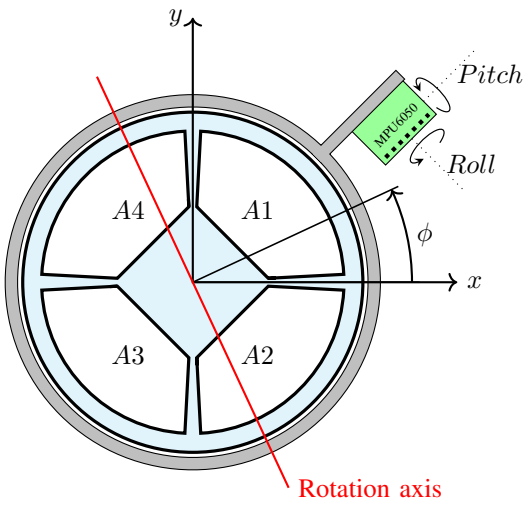
creating the cavities for the air chambers. Specification of the soft robot segments is given in Table I.

After connecting the air tubes to each segment, the segments are connected together with silicone rubber adhesive to form the 2-segment soft robot. The air tubes of the lower segment go through the centre of the upper segment. The upper segment is shifted $45°$ in the yaw-angle to improve the ability to bend in all directions more similar. Fig. 3 illustrates this convention and also shows two 3D-printed brackets to link the MPU6050 sensors to the silicone soft robot. The sensors are aligned to the centre of the chambers $A1$ and $A5$.

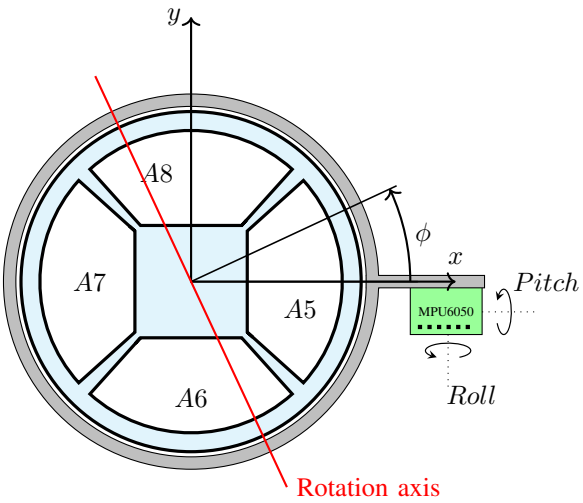
Soft Robot Properties:	
Segment length	11 cm
Diameter	45 mm
Segment weight	117g
Silicone material	DragonSkin-20
total length	22 cm

TABLE I: Soft robot specification.

Fig. 2: picture of the soft robot mounted in the experimental set-up



(a) Cross-section of the upper silicone segment of the pneumatic soft robot.



(b) Cross-section of the lower silicone segment of the pneumatic soft robot.

Fig. 3: Schematic of the pneumatic soft robot. $A1 - A8$ represents the individual air chambers. In red the rotation axis is shown which is perpendicular to the direction of ϕ.

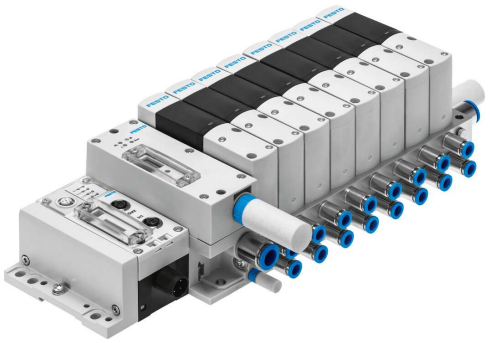


Fig. 4: Festo pressure regulator [27].

B. Pressure Regulator

For the pneumatic actuation of 8 air chambers, a pressure regulator is needed to control the pressure in each chamber independently. In this project, the Festo Motion Terminal VTEM is used. Using a LAN connection between the VTEM and a computer, communication is established according to the Modbus protocol. The VTEM pressure regulator contains a low level PID controller to match the set pressure. Further information can be found in the datasheet [26].

C. Soft Robot Sensing

To measure the current posture of the soft robot, fast and precise sensors are needed that can sense the rotation of

the soft robot. Inertial Measurement Unit (IMU) sensors are small, light, fast, reliable, cheap and widely available. These neat sensors can play an important role into the future of soft robots because the sensor chip is small enough to fit well inside macro-scale soft robots, enhancing their flexibility and mobility. Besides, the energy-efficiency of these type of sensors also contributes to the mobility of soft robots.

$$q_g(n) = \underbrace{q_{(n-1)}}_{\text{previous position}} + \underbrace{\dot{q_g}\Delta t}_{\text{time integration}} \tag{10}$$

$$q(n) = 0.96q_g(n) + 0.04q_a(n) \tag{11}$$

In this project, two IMUs are used to sense the posture of the two segments. Details about the MPU6050 IMU sensor can be found in Table II. The sensors are connected to an Arduino Nano via a I^2C communication protocol. The main program of the Arduino comprises continuous tracking the rotations of the IMU sensors by reading the data of the gyroscope and accelerometer. The complementary filter combines the gyroscope and accelerometer data to compensate the drift caused by dead reckoning due to integrating the angular velocities of the gyroscope to obtain the angular position. Equation 10 shows how the angular position follows from the gyroscopic data. In equation 11 the compensatory filter, used by the Arduino, is shown.

The complementary filter is only capable of compensating drift for the pitch and roll angles. The filter is not able to compensate the drift in the yaw angle because the gravitational acceleration always points along the yaw angle. However, the yaw angle is not used in this research because the soft robot does not exhibit torsional rotation.

MPU6050:	
Serial Interface	I^2C
Operating power consumption	12 mW
Chip size	4 x 4 x 0.9 mm
Gyroscope:	
Available angular range	±250, ±500, ±1000, and ±2000°/sec
Output data rate	8000 Hz
Analogue to Digital converter	16-bit
Sensitivity scale	131 LSB/(°/sec)
Accelerometer:	
Available acceleration range	±2g, ±4g, ±8g and ±16g
Output data rate	1000 Hz
Analogue to Digital converter	16-bit
Sensitivity scale	16384 LSB/g (at range ±2g)

TABLE II: Specification of the MPU6050 IMU sensor. Further details can be found in the datasheet [28].

IV. Methodology: Control Design

In this part, the control design of the soft robot is explained. The control of a soft robot consists of multiple parts: A starting point is needed to conduct the robot desired positions over time, the controller should proceed these commands and calculate needed pressures, communication with the actuator is needed and the data from the sensors needs to be read and evaluated.

A. Robot Operating System 2: Foxy

The communication interface with the Festo pressure regulator is written in C++, while other parts of the application requires Python. Hence, ROS 2 is used, which allows to integrate both programming languages. Fig. 5 shows the overview of different ROS nodes.

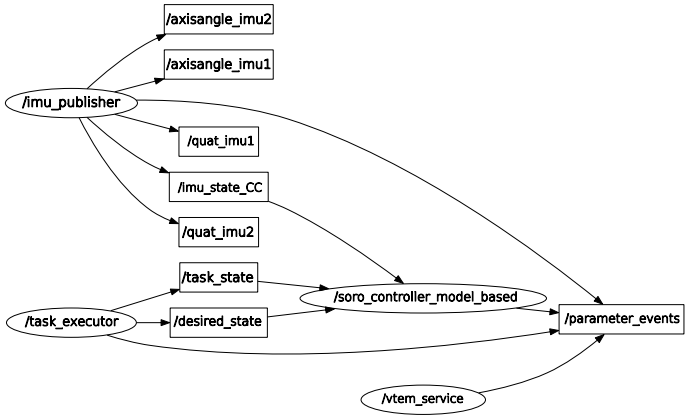


Fig. 5: Rqt graph of ROS2 nodes and topics.

Details about the ROS nodes can be found in appendix A.

B. Model-Based Feedforward Control

The model-based controller in this project is restricted to a static feedforward controller. Considering only the static components is valid if the robot's velocity is small ($\dot{q} \approx 0$) to such an extent that the inertia, damping and Coriolis torque in equation (7) does not significant influence the behaviour of the robot. The soft robot does move in a circular path, however the time for one rotation is set to 10 sec resulting in slow movement, thus a static controller is assumed a valid choice.

A model-based controller can be obtained by solving equation (7) and finding the external torque needed so that the desired configuration is the new equilibrium. When the soft robot is at the equilibrium ($\dot{q} = \ddot{q} = 0$)

$$\left(\frac{\partial V(q)}{\partial q}\right)_{q=q^*} = \tau_a \tag{12}$$

$$G(q^*) + K(q^*) = \tau_a \tag{13}$$

Where $\tau_a \in \mathbb{R}^4$ is the applied torque from the actuator and $q^* \in \mathbb{R}^4$ is the desired robot configuration.

*1) **Gravitational Torque**:* Finding an accurate expression for the gravitational component is more complex compared to rigid link robots because of the highly nonlinear behaviour of the soft robot. However, limited amounts of pressures are used to protect the soft robot from damage due to overstressing the fragile silicone. Therefore, the resulting maximum bending angle is small.

Tests show that the robot is able to safely bend the endpoint no more than 15 degrees. A small bending angle leads to a more dominant elastic torque compared to the gravitational

θ [Degrees]	Measured Weight [gram]	Elastic Force [N]
15	250	2.5
30	500	4.9
45	770	7.6
60	900	8.8
75	950	9.3
90	1150	11.3

TABLE III: Resulting elastic forces when bending the soft robot.

torque because the direction of the gravitational force is almost aligned with the longitudinal axis of the soft robot (recall that the elongation of the soft robot is not considered). Notably, when holding the soft robot horizontally (without actuation) the soft robot slightly bends downwards for approximately 15 degrees (see Fig.(6)) and so even when the gravitational component exerts its maximum possible torque due to its own weight, still the elastic force is more dominant. The gravitational component is relatively small compared to the elastic component. Hence, given the complexity of calculating the gravitational torque and considering its minor contribution to the steady-state configuration of the robot, we decide to neglect the gravitational component in the model (13).

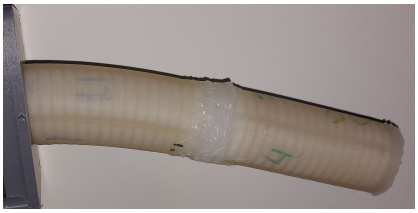


Fig. 6: Maximum gravitational torque is applied in the θ direction by holding the robot horizontally. This results in a bending angle of approximately 15 degrees.

*2) **Elastic Torque**:* To estimate the elastic torque, a weighing scale is placed vertical such that the endpoint of the soft robot is touching the scale. Then, the fixed endpoint of the robot is bent at different angles and the resulting elastic force (via: $F = mg$) of the soft robot is measured. The results are in Table III. To calculate the stiffness of the robot, the slope between the bending angle and elastic force is calculated using the line of best fit formula, i.e.,

$$k = \frac{\sum(\theta - \bar{\theta})(F - \bar{F})}{\sum(\theta - \bar{\theta})^2} \tag{14}$$

*3) **Actuation Matrix** $A(q)$:* The actuation matrix maps the calculated torque to the actuator pressure vector. The mapping involves a translation of coordinates and a ratio scaling between pressure and torque. The actuation matrix $A(q)$ is defined as follows

$$P = A(q)\tau_a \tag{15}$$

In equation (15), $P \in \mathbb{R}^8$ is the pressure vector, equal in size to the amount of actuators. Consequently, $A(q) \in \mathbb{R}^{8\times4}$. To map the translation, we calculate the corresponding Jacobian, whose details can be found in appendix B.

The ratio between pressure and torque is experimentally estimated by measuring, for different pressure levels, the amount of bending and using the previous measured stiffness to estimate the amount of millibars per Newton meter.

Fig. 7 shows the resulting actuation graph that is obtained from the Jacobian. The upper graph corresponds to the upper segment Fig. 3a and the lower graph to the second segment Fig. 3b.

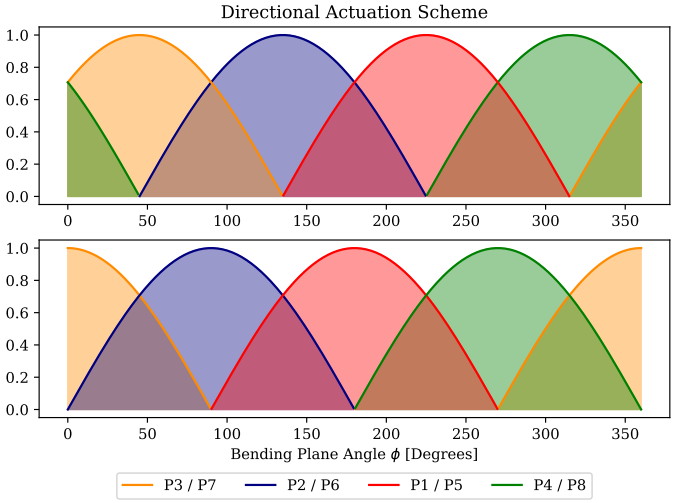


Fig. 7: Directional actuation scheme shows the actuator dependency on ϕ.

*4) **Block diagram Feedforward Controller**:* The block diagram of the feedforward controller is shown in Fig. 8. The actuation matrix block $A(q^*)$ depends on the desired configuration due to the Jacobian. The actuation graph in Fig. 7 is normalized and therefore shows the fraction of actuation of each air chamber compared to the amount of actuation when ϕ is aligned with the actuation angle of the air chamber.

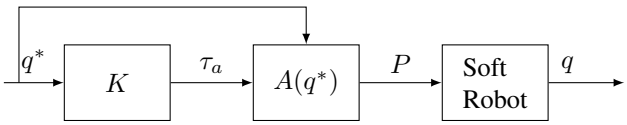


Fig. 8: Block diagram of the model-based feedforward controller. Here, q^* corresponds to the desired robot configuration.

C. Supervised Learning: Gaussian Process

To improve the feedforward controller, a Gaussian process regressor (GPR) can learn the relation between a correction torque τ_l and the resulting bending angle θ for different desired bending angles θ^*. Then, the GPR is used to predict the bending angle θ_p for different torques τ_{try} and selecting the best correction torque τ_l (i.e., the torque for which the error ϵ is the closest to zero.).

The correction torque τ_l is expected to be different for each desired bending angle θ^* of the soft robot, therefore it would be necessary to include the desired bending angle of the soft robot to the input space of the Gaussian Process. However, a Gaussian process suffers from scaling and so the dimensions

of the input space should be limited to avoid that the learning process becomes too slow. It is consequently assumed that τ_l does not significantly depend on the angle ϕ. Thus, the selected input for the Gaussian process is θ^* and τ_{try}, which are the desired bending angle and the applied torque and the output is the predicted angle as shown in Fig. 9.

Training data points are equally spread over the input space to acquire knowledge over the full input space. The process of acquiring training data consists of using the feedforward model-based controller and on top of that apply a varying τ_{try} to measure the resulting bending angle θ. Algorithm 1 shows the pseudocode of acquiring learning data. Fig. 11 visualizes the procedure, where θ^* is set to different levels, and the feedforward controller applies the corresponding torque. Then τ_{try} is applied, varying from -0.03 to 0.03. The black line represents the sum of the feedforward torque and τ_{try}.

$$\epsilon = \theta_p - \theta^* \tag{16}$$

$$\tau_l = \tau_{try}(\min|\epsilon|) \tag{17}$$

Fig. 9 shows a block diagram of the learning part. After training the Gaussian process, the GPR is used online to estimate the bending angle for different torquesτ_{try} for the desired bending angle. τ_{try} is a linear-spaced array with size 100. Thus, 100 predictions are done. Then for all predictions, the error is calculated using equation 16. Finally, the torque corresponding to the smallest error is selected using equation 17.

Fig. 10 shows the block diagram of the hybrid controller. In this diagram, the learning part from Fig. 9 is summarized into one block.

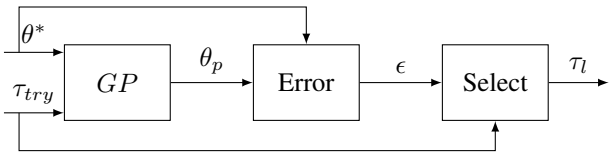


Fig. 9: Block diagram of the learning part: the 2D-input of the Gaussian process (GP) is the desired angle θ^* and τ_{try} . The output is the predicted bending angle θ_p.

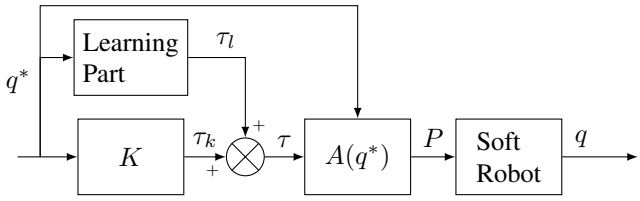


Fig. 10: Block diagram of the proposed improved controller.

Algorithm 1 Algorithm to acquire training data for the Gaussian Process

```
begin
  for φ:=0  to 360 step 45 do
    for θ*:=0  to 15 step 3 do
      for τ_try:=-0.03  to 0.03 step 0.003 do
        τ = τ_ff(q) + τ_try;
        P = A(q*)τ;
        apply pressure P;
        Delay (1s);
        θ(θ*, τ_try) := θ_current;  od
    od
  od
end
```

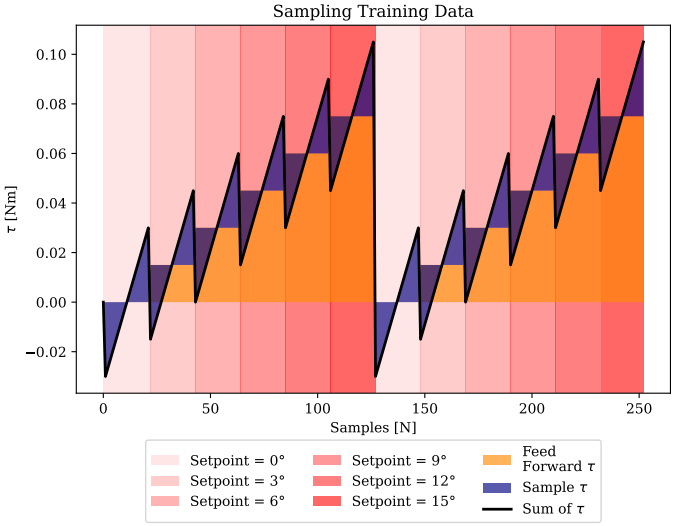


Fig. 11: Visual representation of sampling the data for the Gaussian process. For different desired bending angles, a varying τ_{try} is applied on top of the feedforward controller.

*1) **Gaussian Process Kernel**:* The kernel function of a Gaussian process is an essential function and defines the similarities between different points. Consequently, the kernel function defines assumption on the system. The RBF kernel function is a logical choice because approximately similar values for τ_{try} should lead to similar bending angles θ, while not being affected by points further away. This is the (valid) assumption that is defined in the kernel function. After manually tuning the hyperparameter, the resulting kernel function is given by

$$k(x, x') = e^{-50\|x-x'\|^2} \tag{18}$$

The kernel function is plotted in Fig. 12.

*2) **Gaussian Process Prediction Time**:* Since the Gaussian process is fitted only once (before starting the movement), the time to predict τ_l for the controller using the Gaussian Process is reduced compared to fitting every iteration with new data. Fig. 20 shows the Gaussian process prediction time for each controller during control iterations.

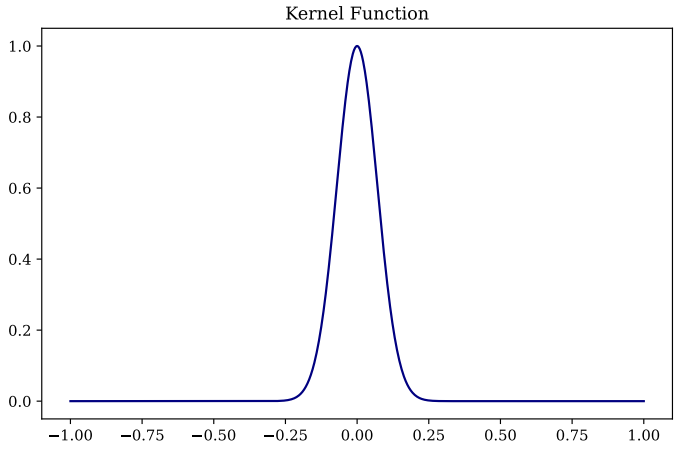


Fig. 12: Kernel function for $\sigma = 0.1$

V. Results

In this section, the experimental results are presented in three parts. In the first part, the results of the movement of the soft robot is presented for different controllers, then the outcomes of the Gaussian process are shown and in the last part performances are compared for different controllers.

A. Soft Robot Movement

During the experiment, the soft robot moves in a circular path. To this end θ^* is set to a constant radius, and the angle ϕ^* rotates in 10 seconds one revolution.

Fig. 13 shows the measured endpoint position for 4 types of controllers. The experiments last one minute of movement, which is equivalent to 6 rotations. The first controller is the model-based feedforward controller. This controller shows the results without the correction torque τ_l. The following polar plot shows the results where both segments use an improved controller. For each segment, a separate Gaussian process optimization torque is calculated. The last two polar plots show the improved control result, for which one of the segments uses the improved controller and the other segment is purely controlled by the model-based controller.

Analyzing the results of the three types of improved control reveals that when only segment 2 is corrected, the results are better. Fig. 14 shows the result of the model-based controller and the result of the segment 2 improved controller in a cartesian plot for different desired bending angles.

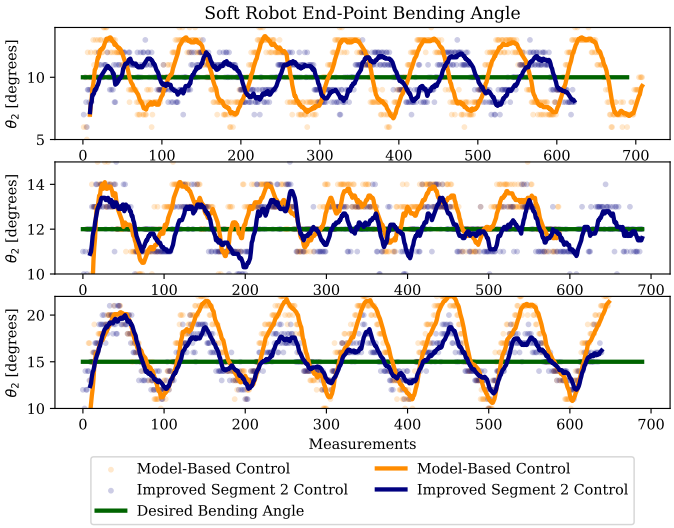


Fig. 14: Soft robot end-point bending angle θ_2 for different radius circular movement. Comparison between model-based control and segment 2 improved control.

B. Gaussian Process

Fig. 15 and 16 show a scattered part of the learning data for 3 desired angles for segment 1 and 2, respectively. At each desired angle τ_{ff} is constant since the desired angle is constant and τ_{try} varies. Then the resulting angle θ is measured. The figures also show the fitted Gaussian process, where the intersection between the fitted Gaussian process and the corresponding desired angle line corresponds to the best τ_{try} following from the training data. When using the improved controller, the amount of torque at the intersection is calculated and used in the controller as τ_l. If there is no intersection, the torque with the smallest difference between expected and desired angle is selected.

C. Soft Robot Performance

To compare the performances of different controllers, the root-mean-square (RMS) error is a suitable measure because smaller errors in the movement of the robot are less relevant and could even be caused by sensor noise. Instead, larger errors are of more relevant and should penalize the performance more. Fig. 17 shows the resulting RMS errors for the model-based controller and the S2 improved controller for different angles.

A box plot involves not only information about the median, but also about how the data is varying. This is useful because if the soft robot oscillates more around the target angle, the median, and average angle value could still be accurate, however this would result in a box plot to a wider box showing that the angle varies more. Therefore, comparing the performances of the controllers in a box plot enables to also compare the variance of each controller. Fig. 18 shows three pairs of box plots analyzing the two controllers for different desired angles.

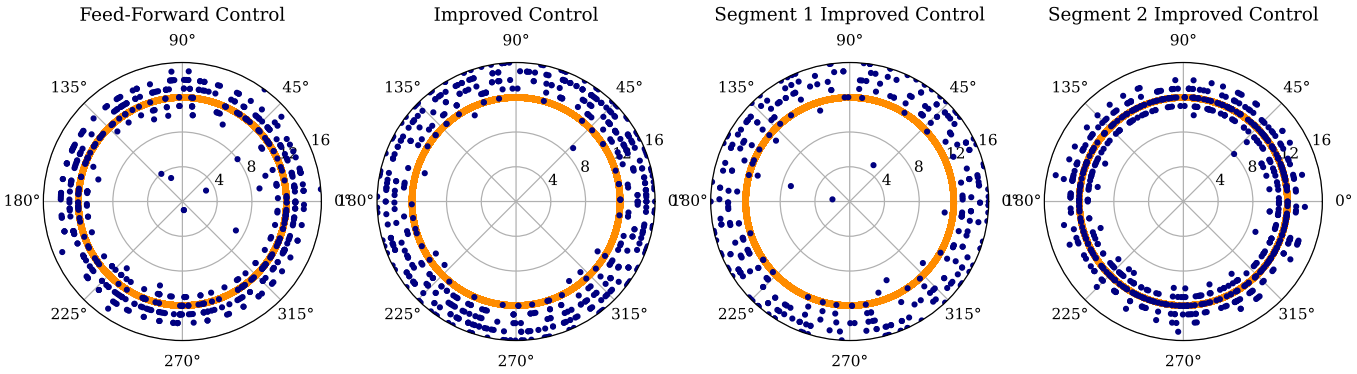


Fig. 13: Polar plot showing the circular movement of the soft robot for different controllers. The target angle of $\theta = 12°$ shown in orange.

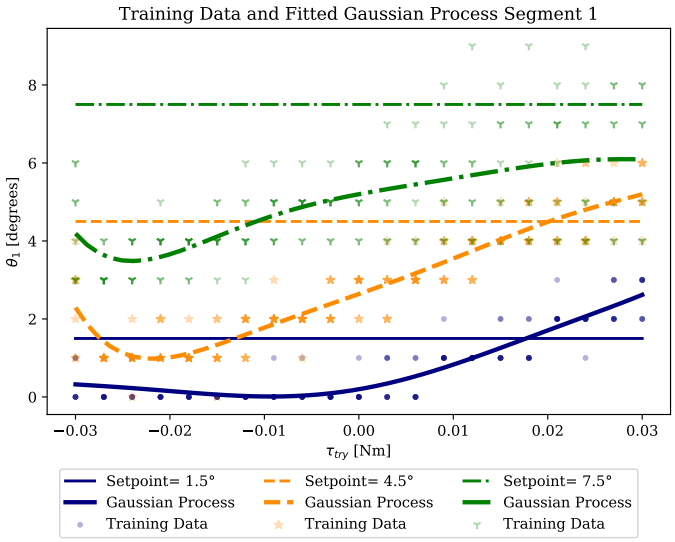


Fig. 15: A snippet of the training data for segment 1: The resulting bending angle θ for varying τ_{try}. For readability, only 3 different desired angles θ^* are shown.

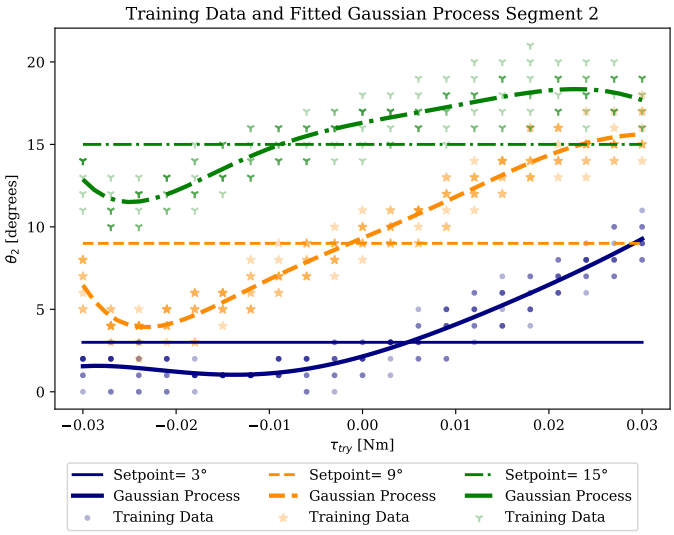


Fig. 16: A snippet of the training data for segment 2: The resulting bending angle θ for varying τ_{try}. For readability, only 3 different desired angles θ^* are shown.

VI. Discussion

A. Soft Robot Control Behaviour

The experimental results of the feedforward controller does already show that results are close to the desired bending angle. Considering the simplicity of the controller, the results are better than expected. Some errors were expected since we omitted the gravitational terms. Therefore, depending on the application demands, a soft robot controlled by a feedforward controller might be a valid, working, and relative easy to implement choice.

However, the improved S2 controller shows the best results. Comparing the RMS error of the 2 controllers in Fig. 17 shows a significant decrease of approximately 40% for the three different desired bending angles. A noticeable difference in accuracy is also visible in the polar plot. Data points are more accurate and more precise, also shown in the box plot.

Fig. 14 shows an oscillating behaviour of the soft robot, which is not clearly visible in the polar plot. A possible

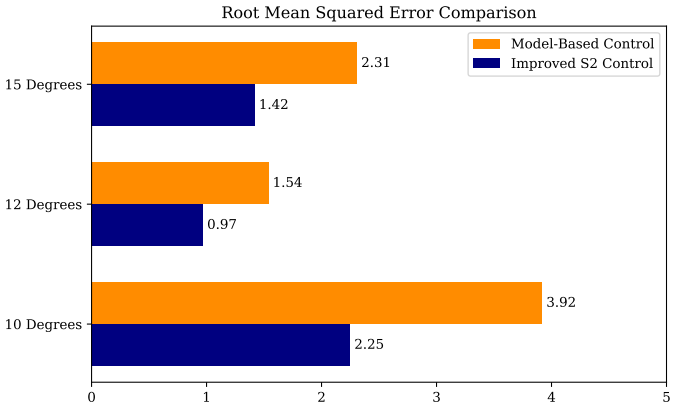


Fig. 17: RMS error graph for the model based controller and the S2 improved controller.

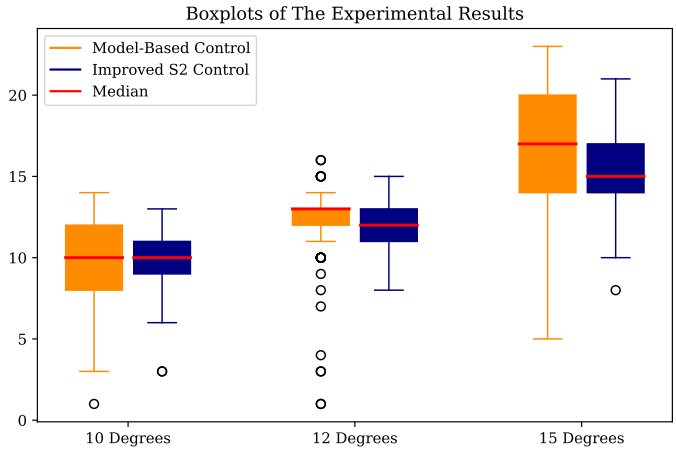


Fig. 18: Box plots visualizing the accuracy and precision performance of the model-based controller and the improved S2 controller. A smaller box indicates more precision, while the median indicates the accuracy.

cause could be the 'dead-zones' of the actuators located at the direction of the walls. However, this is captured by the actuation matrix by applying $\sqrt{0.5}\tau_{max}$ to the two air chambers (instead of $0.5\tau_{max}$) since the chambers partially counteract each other's torque. Another cause could be the increase in stiffness due to the same silicone walls between the air chambers resulting in oscillating behaviour for when a circular trajectory is followed. A manual test by bending a segment and estimate the stiffness reveals this difference in stiffness between walls and air chambers. The segments are connected with a $45°$ shift to reduce this effect, however since the lower segment does bend more than the upper segment due to the weight of the lower segment, this effect is not completely removed. Another cause for this fluctuating behaviour could be that not all actuators deliver the same torque when similar pressure is applied.

*1) **Gaussian Process Improvements**:* The learning data shows that the desired torque (i.e., the intersection) lies in the range of τ_{try} in most cases for segment 1 and in all cases for segment 2. This means that for segment 2 this range is a sufficient choice to find the optimal τ_l. Moreover, all intersection values lie between $-0.01 < \tau_{try} < 0.01$ therefore there is even room to narrow the range further. For segment 1 when the desired angle is $7.5°$ no intersection is found. Despite that some data points do exceed the desired angle, the weighted mean of the Gaussian does not. A solution would be to increase or shift the range of τ_{try} upwards to capture the intersection points.

*2) **The Success of Improved Control**:* Every extra data point τ_{try} that is added to find the intersection leads in the current set-up to 48 new data points (8 ϕ directions and 6 levels of θ) and causes slower performance of the Gaussian process. Because the final τ consists of the summed model-based τ_k and τ_l where the larger contribution is given by τ_k (the coarse part). Hence, the Gaussian process can use a smaller range for τ_{try} in the training data, leading to fewer data points when training with the same τ_{try} resolution (the step size of τ_{try} during learning). Therefore, this set-up excels the performance of a comparable purely Gaussian process set-up (learning based controller) because of the smaller learning task.

B. Further Research

*1) **Design of the soft robot**:* The design of the robot could be further improved. The used material is relative inelastic. Softer material would lead to more stretching and bending for the same pressure. Consequently, the soft robot could bend more. However, the gravitational component would have a greater impact on the dynamics. Therefore, it must be investigated if the Gaussian process can compensate this or if a model of the gravitational torque is needed.

*2) **Adding load to the soft robot**:* Something that is not considered in this project is how the robot would behave when a gripper or load is added. This requires an addition to the control framework, since the needed torque varies for different loads. Different Learning data sets for different loads and a system that select the right learning data could be investigated.

*3) **Soft robot posture control**:* This research focuses on the endpoint of the soft robot instead of the entire posture of the robot. To control the posture, it is required to improve both segments. Controlling the posture, allows verifying what happens if the segments bend in different directions.

*4) **Gaussian process modification**:* Another interesting possible improvement would be to investigate an iterative Gaussian optimization framework in which the fitted optimal τ_l is added as an angle dependent constant to the controller, releasing computational load while maintaining the benefit of the correction. Then a new Gaussian Process with a smaller range for τ_{try} can be learned to find the error of the already improved controller. Using a smaller range also enables to make the step size in τ_{try} smaller. A smaller step size zooms in to find a more accurate τ_l. Even if for any reason the previous Gaussian process did make a mistake, that would result in an error which will be captured and corrected by the new Gaussian process. This could also be an interesting approach, considering wear and tear of the silicone, which leads to different behaviour.

VII. Conclusion

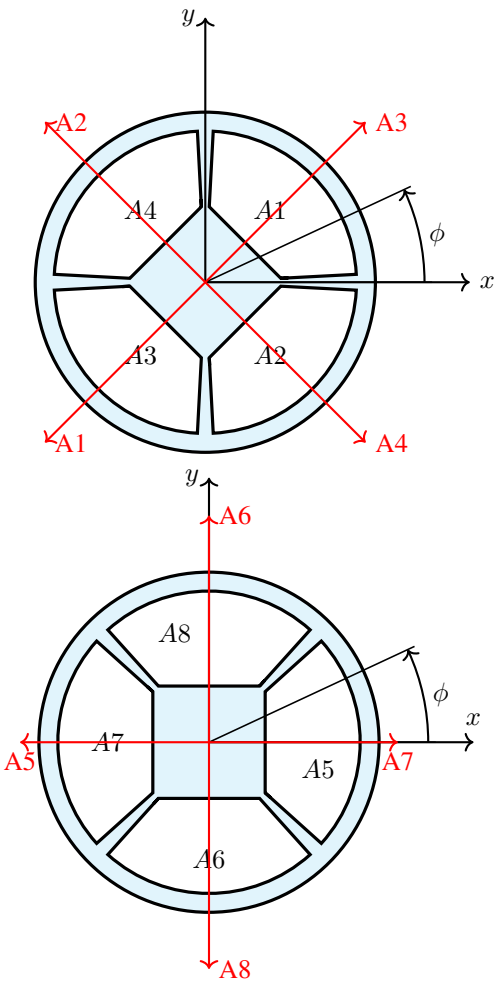
Soft robots are useful for workspaces that are unpredictable and involve humans due to the safe and flexible nature of the materials. However, this also induces a control challenge that requires new approaches. In this research, a hybrid controller is designed and validated in a two-segment pneumatic soft robot. Then, the performance of the hybrid controller is compared with the model-based one. The control goal was to follow a circular trajectory, with the end point of the robot. Experimental results show that a hybrid controller consisting of a model-based controller with estimated parameters and a learning based correction part leads to better performance compared to only using a model-based controller. This improvement is achieved when the second segment only is corrected with the Gaussian process. The findings show that closed-loop performance can be significantly improved with a relatively simple learning approach.

APPENDIX A
ROS 2 NODES

- ***Imu_publisher:*** This node establishes a USB serial connection with the Arduino Nano. When this node sends a request for data to the Arduino, the Arduino replies by sending the quaternion and the Tait-Bryant angles (yaw, pitch, roll) for each IMU sensors. The data is processed and published into 3 parametrizations: axis-angle, quaternion and PCC notation using conversion equations.
- ***vtem_service:*** The VTEM service node manages the communication with the pressure regulator over Ethernet via a Modbus protocol. It activates the pressure regulator and when ready offers two ROS services named `set_pressure` and `read_pressure`. A client node can request the `set_pressure` service via a custom service message containing 8 pressure values in mbar for air chambers $A1 - A8$ respectively. The service will reply with a boolean that is true if the requested pressure is successful send to the VTEM. The `read_pressure` service works by sending an empty request, which will be replied with the 8 actual pressure's.
- ***task_executor:*** The task executor node is the starting point of the system. This node publishes the desired configuration of the robot with a frequency of 10 Hz. The node selects which shape the robot must follow. Possible shapes are a circle, lemniscate, turbine, or an Archimedes screw. Besides shape trajectories, the node too features holding a fixed configuration. More parameters of this node are the time to complete the shape and the maximum theta bending angle. This node besides publishes the actual shape and percentage of completion for logging purposes.
- ***soro_controller_model_based:*** The model based controller node is the main node and consists of subscribing to `desired_state`, `current_state` and `task_state` and then based on the information calculating the updated pressure and requesting these pressures via the service `set_pressure`. This node also logs the data to a CSV-file. Further details on the controller can be found in IV-B.
- ***soro_controller_improved_control:*** The improved controller is not visible in Fig. 5 because it replaces the model-based controller after sampling the behaviour of the soft robot. The node has the same functionality, however the control performance is enhanced via the Gaussian Process. This is further explained in section IV-C.

APPENDIX B
JACOBIAN MATRIX: ACTUATOR SPACE TO ROBOT CONFIGURATION SPACE

Fig. 19 shows in red the direction of bending for each actuator. Each actuator works only in the shown direction and not in the opposite direction. To map this to the configuration space of the soft robot, the Jacobian is needed:



(a) Cross-section of the lower silicone segment of the pneumatic soft robot.

Fig. 19: Cross-section schematic direction of pneumatic actuation.

$$J = \frac{\partial P}{\partial q} = \begin{bmatrix} -\theta_1 c(\phi_1 + \frac{\pi}{4}) & -s(\phi_1 + \frac{\pi}{4}) & 0 & 0 \\ +\theta_1 c(\phi_1 - \frac{\pi}{4}) & +s(\phi_1 - \frac{\pi}{4}) & 0 & 0 \\ +\theta_1 c(\phi_1 + \frac{\pi}{4}) & +s(\phi_1 + \frac{\pi}{4}) & 0 & 0 \\ -\theta_1 c(\phi_1 - \frac{\pi}{4}) & -s(\phi_1 - \frac{\pi}{4}) & 0 & 0 \\ 0 & 0 & +\theta_2 s(\phi_2) & -c(\phi_2) \\ 0 & 0 & +\theta_2 c(\phi_2) & +s(\phi_2) \\ 0 & 0 & -\theta_2 s(\phi_2) & +c(\phi_2) \\ 0 & 0 & -\theta_2 c(\phi_2) & -s(\phi_2) \end{bmatrix} \tag{19}$$

APPENDIX C
GAUSSIAN PROCESS: COMPUTATIONAL RESULTS

A Gaussian process does not scale well due to the curse of dimensionality. The training data consists of 1008 samples (8 directions, 6 bending angles and 21 torques). Before movement starts, at start-up the Gaussian process is fitted based on the training data. During movement, there is no new data added and so no new fitting is required, which helps to

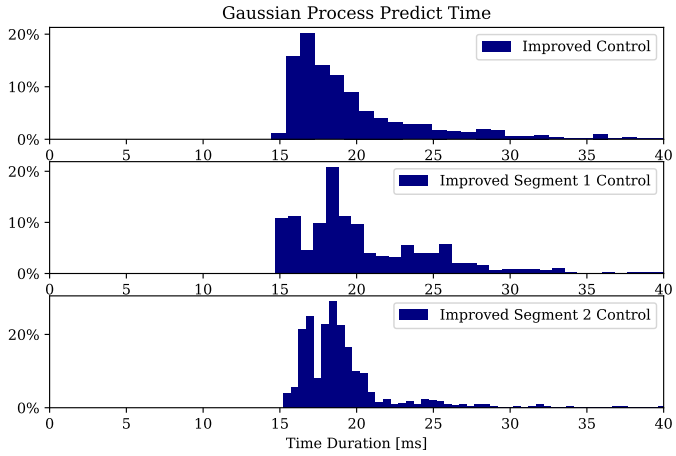


Fig. 20: Overview of the time for the controller to calculate the new τ_{gp} during the experiments.

reduce the computational. Fig. 20 shows a histogram of the time the computer needed to predict a value (i.e., calculating the weighted mean for a giving input). All experiments are performed using an Intel®i7-9750H 12-core CPU.

ACKNOWLEDGMENT

I would like to thank Pablo Borja Rosales and Cosimo Della Santina for their supervision and help. Also thanks to Maximilian Stölzle for explaining the experimental equipment. Finally, thanks to my family for supporting me during my study.

REFERENCES

[1] L. Niu, L. Ding, H. Gao, Y. Su, Z. Deng, and Z. Liu, "Closed-form equations and experimental verification for soft robot arm based on cosserat theory," pp. 6630–6635, 2019.

[2] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[3] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, "A 3d steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm," *Bioinspiration Biomimetics*, vol. 7, no. 2, p. 025006, may 2012. [Online]. Available: https://dx.doi.org/10.1088/1748-3182/7/2/025006

[4] B. Caasenbrood, A. Pogromskiy, and H. Nijmeijer, "Energy-based control for soft manipulators using cosserat-beam models," in *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2021*, O. Gusikhin, H. Nijmeijer, and K. Madani, Eds., 2021, pp. 311–319, 18th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2021), ICINCO 2021 ; Conference date: 06-07-2021 Through 08-07-2021. [Online]. Available: http://www.icinco.org/?y=2021

[5] C. Della Santina, C. Duriez, and D. Rus, "Model based control of soft robots: A survey of the state of the art and open challenges," 2021.

[6] Z. Zhang, J. Dequidt, and C. Duriez, "Vision-based sensing of external forces acting on soft robots using finite element method," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1529–1536, 2018.

[7] K.-H. Lee, M. C. W. Leong, M. C. K. Chow, H.-C. Fu, W. Luk, K.-Y. Sze, C.-K. Yeung, and K.-W. Kwok, "Fem-based soft robotic control framework for intracavitary navigation," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2017, pp. 11–16.

[8] M. Dubied, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-real for soft robots using differentiable fem: Recipes for meshing, damping, and actuation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5015–5022, 2022.

[9] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," 2018.

[10] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," 2010.

[11] M. T., H. A., and S. O., "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," 2014.

[12] M. T., H. A., and S. O. M. A., "A variable curvature modeling approach for kinematic control of continuum manipulators," 2013.

[13] C. Della Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," 2020.

[14] Y. Toshimitsu, K. W. Wong, T. Buchner, and R. Katzschmann, "Sopra: Fabrication dynamical modeling of a scalable soft continuum robotic arm with integrated proprioceptive sensing," 2021.

[15] A. Kapadia and I. D. Walker, "Task-space control of extensible continuum manipulators," 2011.

[16] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Task-space control of continuum manipulators with coupled tendon drive," in *Experimental Robotics*, O. Khatib, V. Kumar, and G. J. Pappas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 271–280.

[17] P. Borja, A. Dabiri, and C. D. Santina, "Energy-based shape regulation of soft robots with unactuated dynamics dominated by elasticity," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, 2022, pp. 396–402.

[18] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.

[19] M. Rolf and J. J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1147–1160, 2014.

[20] A. Melingui, R. Merzouki, J. B. Mbede, C. Escande, B. Daachi, and N. Benoudjit, "Qualitative approach for inverse kinematic modeling of a compact bionic handling assistant trunk," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 754–761.

[21] M. Hofer, C. Sferrazza, and R. D'Andrea, "A vision-based sensing approach for a spherical soft robotic arm," *Frontiers in Robotics and AI*, vol. 8, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2021.630935

[22] P. Lloyd, A. K. Hoshiar, T. da Veiga, A. Attanasio, N. Marahrens, J. H. Chandler, and P. Valdastri, "A learnt approach for the design of magnetically actuated shape forming soft tentacle robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3937–3944, 2020.

[23] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *31st International Conference on Machine Learning, ICML 2014*, vol. 1, 06 2014.

[24] O. Lakhal, A. Melingui, and R. Merzouki, "Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1326–1335, 2016.

[25] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, 2018, pp. 39–45.

[26] *Motion Terminal VTEM*, Festo, 12 2012.

[27] Festo, "Buy motion terminal vtem online festo ee," https://www.festo.com/ee/en/p/motion-terminal-id_VTEM/, 2023.

[28] *MPU-6000 and MPU-6050 Product Specification*, InvenSense Inc., 8 2013, revision 3.4.