A Hybrid Control Approach for a Pneumatic-Actuated Soft Robot

Tavio y Cabrera, Emilio; Santina, Cosimo Della; Borja, Pablo

DOI
10.1007/978-3-031-55000-3_2

Publication date
2024

Document Version
Final published version

Published in
Human-Friendly Robotics 2023 - HFR

A Hybrid Control Approach for a Pneumatic-Actuated Soft Robot

Emilio Tavio y Cabrera[1], Cosimo Della Santina[1,2], and Pablo Borja[3(✉)]

[1] Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands
emilio@hotmail.nl, c.dellasantina@tudelft.nl
[2] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany
[3] School of Engineering, Computing and Mathematics, University of Plymouth, Plymouth, UK
pablo.borjarosales@plymouth.ac.uk

Abstract. The compliant nature of soft robots is appealing to a wide range of applications. However, this compliant property also poses several control challenges, e.g., how to deal with infinite degrees of freedom and highly nonlinear behaviors.

This paper proposes a hybrid controller for a pneumatic-actuated soft robot. To this end, a model-based feedforward controller is designed and combined with a correction torque calculated via Gaussian process regression. Then, the proposed model-based and hybrid controllers are experimentally validated, and a detailed comparison between controllers is presented. Notably, the experimental results highlight the potential benefits of adding a learning approach to a model-based controller to enhance the closed-loop performance while reducing the computational load exhibited by purely learning strategies.

1 Introduction

Soft robotics has become an increasingly studied field due to its potential application in high-impact areas such as medicine, the food industry, and human-robot interaction. However, because of their compliant nature, the behavior of soft robots 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.

Several approximation approaches are adopted to deal with the challenge posed by the infinite-dimensional nature of soft robots. Below, we list the three more common approximation methods:

- Cosserat rod theory. This approach is suitable for modeling soft robots consisting of slender rods while accounting for bend, twist, stretch, and shear deformations. Some examples of studies using this framework to model soft robots are given in [2,18,20,24]. The advantage of Cosserat rod models is that an exact steady-state solution can be achieved [6]. Nevertheless, an infinite

C. Piazza et al. (Eds.): HFR 2023, SPAR 29, pp. 19–35, 2024.
https://doi.org/10.1007/978-3-031-55000-3_2

number of states are possible when using the Cosserat rod theory, making it complex to use for control.

- Finite element method (FEM). This approach provides a numerical solution to the partial differential equations describing the dynamics of the soft robot. FEM consists in decomposing complex systems into simpler ones, which involves a discretization process. In particular, this approach often decomposes the complex shape of soft robots into a mesh of point masses (nodes) interconnected via springs. Hence, using more nodes leads to a more accurate model of the soft robot at the expense of having a larger state space—thus, a higher computational load. Examples of the application of FEM in soft robotics are [8,15,27]. FEM results are accurate. However, the computational bottleneck is emphasized, complicating online control using this approach.
- Piecewise constant curvature (PCC). This framework defines the state of the soft robot along a finite set of arcs (segments) in series, where each segment is assumed to have constant curvature (bending). In particular, each segment has three degrees of freedom given by the bending, the rotation orthogonal to the bending plane, and the elongation [26]. In most cases, the amount of independent curvatures is equal to the number of links (physical segments) that compose 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, bends equally over that length. Therefore, the simplification of PCC is accurate enough to reduce infinite degrees of freedom to only three configuration variables. However, a drawback of PCC approximations is the presence of singularities for some postures. Such a problem is solved by defining new coordinates in [5]. This singularity-free approach has been adopted in [23,25].

Control approaches for soft robots can be classified into model-based and learning-based strategies. Model-based control approaches are suitable for exploiting physical properties and prior system knowledge. This often yields controllers that are more energy-efficient than learning-based controllers. Additionally, these model-based strategies are ideal for simulation purposes. Nonetheless, these approaches are sensitive to model mismatches, and they are often analytically complex due to the highly nonlinear nature of soft robots. Some examples of model-based strategies for soft robots are provided in [1,3,4,7,9,13,19]. On the other hand, learning-based control approaches are robust concerning model mismatches, disturbances, and sometimes changing environments. Moreover, the analytical complexity is significantly reduced as most learning strategies are model-free. However, these approaches may suffer from long learning times, a large amount of data required, greater computational load than model-based methods, and stability problems. Additionally, the learning stage may be destructive for the soft robot. Examples of learning-based control approaches for soft robots can be found in [12,16,17,21,22]. A third classification for control strategies is given by the so-called hybrid techniques, which combine model-based and learning-based approaches. Hybrid control methods may exhibit the most appealing properties of model-based and learning-based approaches, i.e., they may be robust to uncertainties concerning the soft robot's parameters and model mismatches while exploiting

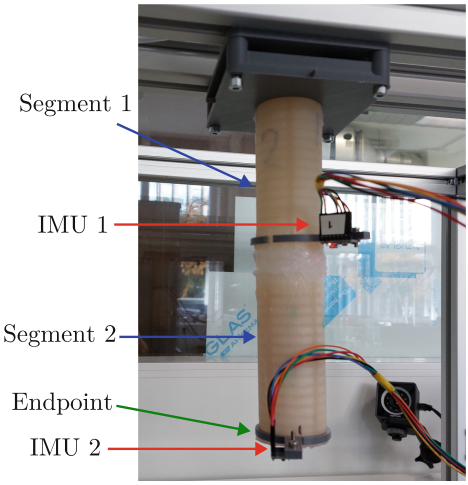


Fig. 1. Pneumatic-driven soft manipulator

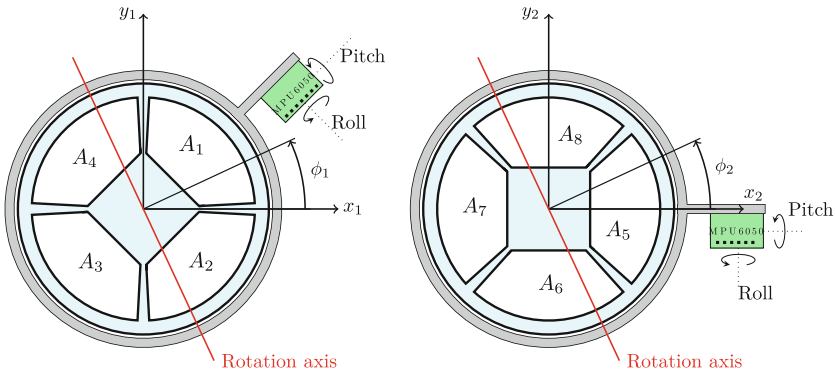
prior system knowledge to reduce the learning process' complexity and the associated computational burden. Examples of hybrid control strategies for soft robots are found in [11,14]. For a more extensive discussion on control design strategies for soft robots, we refer the reader to [6,10].

This paper proposes a simple hybrid control strategy for a pneumatic-driven soft robot made of silicone. To this end, we design a feedforward controller based on a simplified PCC model. Then, we design a correction learning-based controller using Gaussian process regression (GPR). Notably, the proposed model-based feedforward controller is one of the most straightforward model-based control strategies that can be formulated. Nevertheless, it is suitable to reduce significantly the computational burden associated with the learning strategy while ensuring the stability of the closed-loop equilibrium. Moreover, GPR permits constructively limiting the range of learning.

2 Physical System and Problem Formulation

Consider the pneumatic-driven soft manipulator depicted in Fig. 1. The body of the manipulator consists of two segments—denoted as Segment 1 and Segment 2, respectively—made of DragonSkin-20 silicone, where each segment has a length, diameter, and weight of 11[cm], 45[mm], and 117[g], respectively. Moreover, each segment has four internal air chambers that can be pressurized to bend up to 8° the corresponding segment without damaging the system. The pressure regulator Festo Motion Terminal VTEM is used to actuate each chamber independently.

We use the inertial measurement units (IMUs) MPU6050—shown in Figs. 1 and 2—together with a complementary filter to measure the position of each segment's tip. This sensor choice is more portable and cheaper than a motion



(a) Cross-section of Segment 1. (b) Cross-section of Segment 2.

Fig. 2. Schematic of the pneumatic soft robot. $A1$–$A8$ represent the individual air chambers. In red the rotation axis is shown which is perpendicular to the direction of ϕ_i.

capture system. However, the IMUs only measure the position of a specific part of each segment, in this case, the tip. In addition, the segments are shifted 45° degrees in the yaw angle, as illustrated in Fig. 2, where A_1–A_4 represent the air chambers in Segment 1, and A_5–A_8 represent the air chambers in Segment 2.

The control objective is to describe a desired trajectory with the robot's endpoint (tip of Segment 2). To this end, the desired trajectory is expressed by N points, i.e., the trajectory is discretized. Hence, the problem reduces to stabilizing the robot sequentially at the N desired configurations. Moreover, because of the limited bending capacity of the system, we consider trajectories where the desired bending angles have the same magnitude. We stress that the dynamics of this system are highly nonlinear. Therefore, the proposed approach is split into two parts: (i) designing a controller based on a simplified model of the system and (ii) implementing a learning method to correct the errors caused by the model simplifications and unmodeled phenomena. Consequently, the resulting controller has a hybrid nature, which is expected to require fewer data than a pure learning method and exhibit better performance than a model-based controller.

3 Modeling

The compliant nature of the manipulator poses a challenge from a control perspective, as the controller needs to deal with an infinite number of degrees of freedom. Moreover, as mentioned in Sect. 2, the sensors only read the position of each segment's endpoint. To overcome these problems, we consider a PCC approximation and neglect the elongation of the segments. Hence, the configuration variables of the robot are given by

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

where $\phi_i \in \mathbb{R}$ and $\theta_i \in \mathbb{R}$ denote the rotation of the bending plane and the bending angle of the corresponding segment, respectively. Figure 3 illustrates the configuration variables of the ith segment, where $i \in \{1, 2\}$.

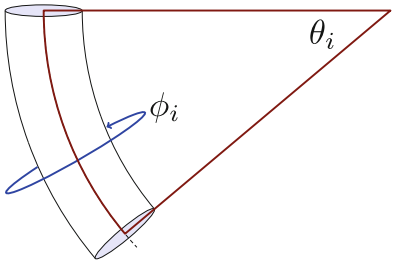


Fig. 3. Front view of the ith segment of the soft robot.

The dynamics of the robot can be obtained via the Euler-Lagrange formalism. To this end, we consider the Lagrangian

$$\mathcal{L}(q,\dot{q}) = T(q,\dot{q}) - V(q); \qquad \begin{array}{l} T(q,\dot{q}) = \dot{q}^\top M(q)\dot{q} \\ V(q) = V_{\mathtt{k}}(q) + V_{\mathtt{g}}(q), \end{array}$$

where $T : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}$ denotes the (co-)kinetic energy; $M : \mathbb{R}^4 \to \mathbb{R}^{4\times4}$ is the mass inertia matrix, which is positive definite; $V : \mathbb{R}^4 \to \mathbb{R}$ represents the potential energy of the system, which is split into the elastic potential $V_{\mathtt{k}} : \mathbb{R}^4 \to \mathbb{R}$ and the potential energy due to gravity $V_{\mathtt{g}} : \mathbb{R}^4 \to \mathbb{R}$. Hence, the dynamics of the system are given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} + D(q,\dot{q})\dot{q} = \tau, \tag{2}$$

where $D : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^{4\times4}$ is the damping matrix and $\tau \in \mathbb{R}^4$ are the input torques. Customarily, (2) is rewritten as

$$M(q)\ddot{q} + \left(C(q,\dot{q}) + D(q,\dot{q})\right)\dot{q} + \frac{\partial V_{\mathtt{k}}(q)}{\partial q} + \frac{\partial V_{\mathtt{g}}(q)}{\partial q} = \tau, \tag{3}$$

where $C : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^{4\times4}$ contains the Coriolis and centrifugal terms. Given the control problem explained in Sect. 2, at the desired equilibria $\ddot{q} = \dot{q} = \mathbf{0}$. Therefore, at the desired configuration, (3) reduces to

$$\frac{\partial V_{\mathtt{k}}(q)}{\partial q} + \frac{\partial V_{\mathtt{g}}(q)}{\partial q} = \tau. \tag{4}$$

3.1 Simplified Model

Even if PCC provides an approximation of the actual behavior of the system, the dynamics (3) are still highly nonlinear and complex. However, we adopt the following simplifications based on the fact that the maximum expected bending angle is 16°:

(A1) The elastic potential is given by

$$V_{\mathtt{k}}(q) = \frac{1}{2}k\theta_1^2 + \frac{1}{2}k\theta_2^2, \tag{5}$$

yielding a linear torque given by

$$\frac{\partial V_{\mathrm{k}}(q)}{\partial q} = K_\theta q; \quad K_\theta := k \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

with $k > 0$.

(A2) We neglect the torque due to gravity, i.e.,

$$\frac{\partial V_{\mathrm{g}}(q)}{\partial q} = \mathbf{0}.$$

We remark that, because of the relatively small bending angle, the effect of gravity on the dynamics is negligible compared to the impact of damping or the torque described in (6).

Under assumptions **(A1)** and **(A2)**, the dynamics (3) take the form

$$M(q)\ddot{q} + (C(q,\dot{q}) + D(q,\dot{q}))\,\dot{q} + K_\theta q = \tau, \tag{7}$$

and the set of equilibria is characterized by

$$K_\theta q = \tau. \tag{8}$$

4 Control Design

In this section we present a hybrid controller consisting of a simple model-based feedforward controller and a correction torque obtained via GPR. Furthermore, given the physical limitations of the system mentioned in Sect. 2, we restrict our attention to desired configurations satifying the following:

$$\theta_\star = \theta_{1_\star} + \theta_{2_\star} \qquad \phi_\star = \phi_{1_\star} = \phi_{2_\star},$$

where $\theta_{1_\star}, \theta_{2_\star}, \phi_{1_\star}$, and $\phi_{2_\star}$ denote the desired angles.

4.1 Feedforward Controller

Given the system's configuration, which is hanging similar to a pendulum, any constant torque—producing a bending within the physical limitations of the system—modifies the equilibrium of the closed-loop system without destabilizing it, i.e., a constant torque τ_{F} shifts the (stable) open-loop equilibrium $q = \mathbf{0}$ to a new stable closed-loop equilibrium $\bar{q} \in \mathbb{R}^4$. To prove this, we can consider the Lyapunov function

$$H(q,\dot{q}) = T(q,\dot{q}) + V(q) - q^\top \tau_{\mathrm{F}},$$

which guarantees the stability of any configuration $\bar{q} = [\phi_1 \ \bar{\theta}_1 \ \phi_2 \ \bar{\theta}_2]^\top$, where $\bar{\theta}_1, \bar{\theta}_2$ are determined by τ_{F}, and ϕ_1, ϕ_2 can take any value. We remark that the

specific values of ϕ_1 and ϕ_2 are determined during the implementation of the controller via the mapping from pressures to torques, as explained in Sect. 5. For simplicity, we consider a constant feedforward controller $\tau_{\mathtt{F}}$ to drive the manipulator's endpoint to the N desired configurations that describe the desired trajectory. Moreover, considering the assumptions **(A1)** and **(A2)**, $\tau_{\mathtt{F}}$ can be computed as follows

$$\tau_{\mathtt{F}} = K_\theta q_\star, \tag{9}$$

where $q_\star \in \mathbb{R}^4$ denotes the desired configuration. Because of the arguments given in Subsect. 3.1, (9) provides an approximation of the torque required to steer the endpoint of the manipulator to the desired configuration. However, steady-state errors are expected because we neglect the effect of gravity and other phenomena, such as nonlinear damping and dead zones in the actuator. Such error can be compensated with a learning approach, which is more straightforward than obtaining the exact value analytically. Moreover, we stress that, in this formulation, the model-based feedforward controller does not depend on the matrices $M(q)$, $C(q, \dot{q})$, $D(q, \dot{q})$.

4.2 Supervised Learning: GPR

GPR is suitable for efficient statistical prediction based on samples or observed targets. This is achieved by updating the probabilities via observations. GPR involves a mean function and covariance kernel. In particular, a prediction is made by calculating the weighted mean. Then, the covariance or kernel function returns how related observations are. A popular kernel function is the radial basis function

$$\kappa(x, x') = e^{-\frac{1}{2\sigma^2}\|x - x'\|^2} \tag{10}$$

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.

To improve the feedforward controller (9), a Gaussian process regressor can learn the relation between a correction torque $\tau_{\mathtt{L}}$ and the resulting bending angle θ for different desired bending angles $\theta_\star$. Hence, GPR is used to predict the bending angle $\theta_{\mathtt{p}}$ for different torques $\tau_{\mathtt{T}}$. Then, the best correction torque $\tau_{\mathtt{L}}$ is selected—i.e., the torque for which the error

$$\epsilon = \theta_{\mathtt{p}} - \theta_\star \tag{11}$$

is the closest to zero. The correction torque $\tau_{\mathtt{L}}$ is expected to differ for each desired bending angle $\theta_\star$. Therefore, it is necessary to include $\theta_\star$ in the input space of the Gaussian process. However, because GPR suffers from scaling, the input space dimensions should be limited to avoid the learning process becoming too slow. Consequently, we assume that $\tau_{\mathtt{L}}$ does not significantly depend on the angle ϕ. Thus, the selected input to the Gaussian process is $\theta_\star$ and $\tau_{\mathtt{T}}$, which are the desired bending angle and the applied torque. Moreover, the output is the predicted angle $\theta_{\mathtt{p}}$ as shown in Fig. 4. Once the correction torque $\tau_{\mathtt{L}}$ is selected, the hybrid controller is given by $\tau = \tau_{\mathtt{F}} + \tau_{\mathtt{L}}$.

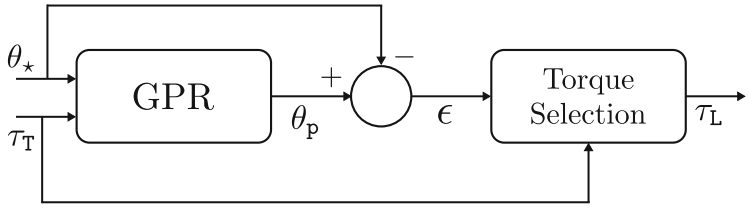


Fig. 4. Block diagram of the learning part: the inputs to the Gaussian process regressor are $\theta_\star$ and τ_T, while the output is the predicted bending angle θ_p.

5 Implementation Process

5.1 Identifying K_θ

To compute τ_F in (9), it is necessary to know K_θ. To this end, a weighing scale is placed vertically such that the endpoint of the soft robot is touching it. Then, the other endpoint of the robot is bent at different angles without applying any input (pressure), and the resulting elastic force of the soft robot is computed via the formula $F = mg$. The results are shown in Table 1. To obtain the stiffness coefficient k, the slope between the bending angle and elastic force is calculated using the line of best fit formula, i.e.,

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

Finally, K_θ is obtained from (6).

5.2 From Pressures to Torques

The dynamics (3), alternatively (7), consider torques as the control input. However, as explained in Sect. 2, the system is actuated via pressures. The relationship between pressures and torques has the structure $\tau = A(q)P$, where $P \in \mathbb{R}^8$ denotes the (positive) pressures injected into the air chambers and

Table 1. Resulting elastic forces when bending the soft robot.

θ [°]	Measured Weight [g]	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

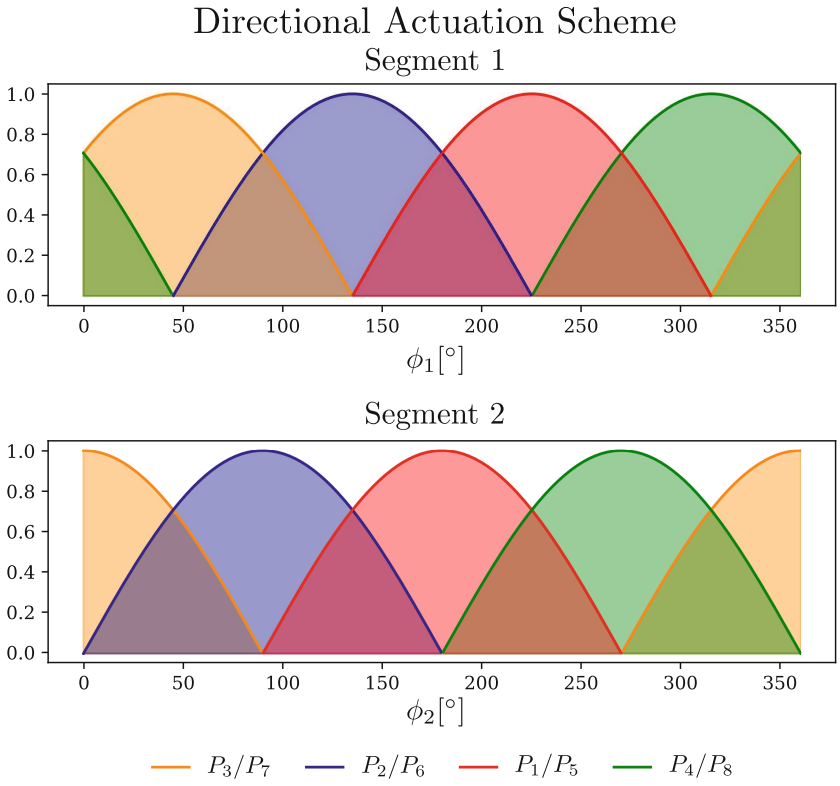


Fig. 5. Directional actuation scheme showing the actuator dependency on ϕ_1 and ϕ_2.

$A : \mathbb{R}^4 \to \mathbb{R}^{4\times 8}$ is the actuation matrix, which has full rank. Therefore, (8) becomes $K_\theta q = A(q)P$.

The actuation matrix is obtained 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. The resulting matrix is given by $A(q) = \begin{bmatrix} A_1(q) & A_2(q) \end{bmatrix}$, where

$$A_1(q) := \begin{bmatrix} -\theta_1 \cos(\phi_1 + \frac{\pi}{4}) & -\sin(\phi_1 + \frac{\pi}{4}) \\ \theta_1 \cos(\phi_1 - \frac{\pi}{4}) & \sin(\phi_1 - \frac{\pi}{4}) \\ \theta_1 \cos(\phi_1 + \frac{\pi}{4}) & \sin(\phi_1 + \frac{\pi}{4}) \\ -\theta_1 \cos(\phi_1 - \frac{\pi}{4}) & -\sin(\phi_1 - \frac{\pi}{4}) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, A_2(q) := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \theta_2 \sin(\phi_2) & -\cos(\phi_2) \\ \theta_2 \cos(\phi_2) & \sin(\phi_2) \\ -\theta_2 \sin(\phi_2) & \cos(\phi_2) \\ -\theta_2 \cos(\phi_2) & -\sin(\phi_2) \end{bmatrix}.$$

Figure 5 shows the relation between the angles ϕ_1 and ϕ_2 and the pressures. Note that $A_1(q)$ is associated with Segment 1, and $A_2(q)$ with Segment 2. Accordingly, the control input corresponding to (9) must satisfy

$$A(q_\star)P = K_\theta q_\star. \tag{12}$$

Because $A(q_\star)$ depends on the desired values for ϕ_1 and ϕ_2, the expression (12) ensures that the robot moves in the desired direction for these angles.

5.3 Implementation of the GPR

The 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 in using the feedforward model-based controller and, on top of that, applying a varying τ_T to measure the resulting bending angle θ. Algorithm 1 shows the process adopted for acquiring the learning data, where $A^{-1}(q_\star)$ denotes the pseudo inverse of $A(q_\star)$. Figure 6 illustrates the procedure, where $\theta_\star$ is set to different values (under the label "Setpoint"), and the feedforward controller applies the corresponding torque. Then, τ_T is applied varying from -0.03 to 0.03. The black line represents the sum of the feedforward torque and τ_T.

Concerning the Gaussian process kernel, (10) ensures that similar values for τ_T yield similar bending angles, while not being affected by points further away. Hence, it only remains to select the value for the hyperparameter σ, which, based on a manually tuning process, is selected as $\sigma = 0.1$.

After training the Gaussian process, GPR is used online to estimate the bending angle for different torques $\tau_\mathrm{T} \in \mathcal{T}_\mathrm{T}$ for the desired bending angle, where $\mathcal{T}_\mathrm{T}$ is a linear-spaced array with size 100. Thus, 100 predictions are considered. Then, for all the predictions, the error is calculated as in (11). Finally, the torque corresponding to the smallest error is selected as follows:

$$\tau_\mathrm{L} = \min_{\tau_\mathrm{T} \in \mathcal{T}_\mathrm{T}} |\theta_\mathrm{p}(\tau_\mathrm{T}, \theta_*) - \theta_*| \tag{13}$$

The Gaussian process is fitted only once (before starting the movement). Thus, the time to predict τ_L for the controller using the Gaussian Process is reduced compared to fitting every iteration with new data. Figure 7 shows the

Algorithm 1. Algorithm to acquire training data for the Gaussian Process

```
1: begin
2: for φ := 0 to 360 step 45 do
3:     for θ⋆ := 0 to 15 step 3 do
4:         for τT := −0.03 to 0.03 step 0.003 do
5:             τ = τF + τT;
6:             P = A^{-1}(q⋆)τ;
7:             apply pressure P;
8:             Delay (1s);
9:             θ(θ⋆, τT) := θcurrent;
10: end
```

Gaussian process prediction time for each controller during control iterations. Finally, the torque τ_{L} needs to be transformed to pressures following the approach explained in Subsect. 5.2.

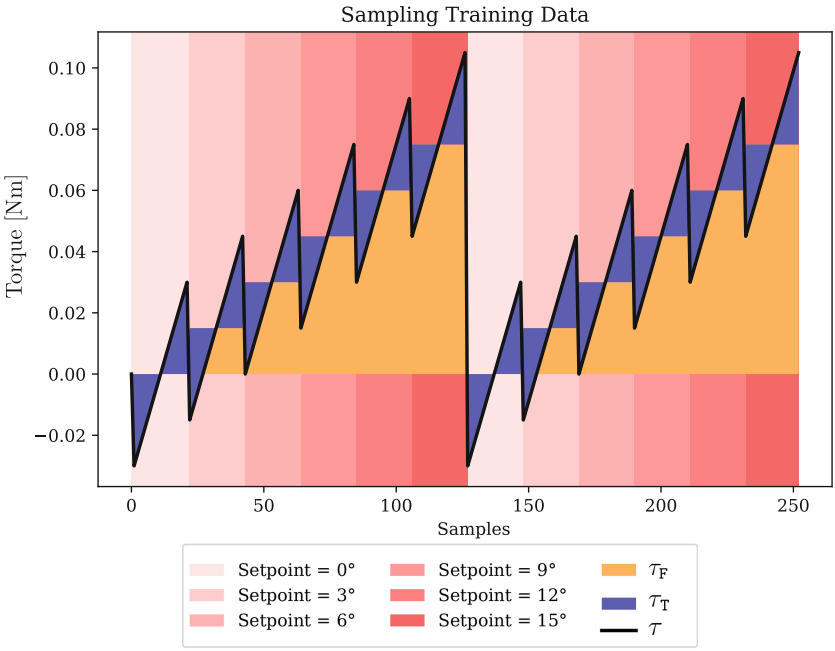


Fig. 6. Visual representation of how the data for the Gaussian process is sampled. For different desired bending angles, a varying τ_{T} is applied on top of the feedforward controller.

6 Experimental Results

For the experiments, we consider circular trajectories of the manipulator's endpoint. To this end, $\theta_\star$—which determines the radius of the circle—is set to a constant value for the N points describing the trajectory. Then, the angle $\phi_\star$ rotates one revolution in 10 s.

Figure 8 shows a scattered part of the learning data for three desired bending angles. At each desired angle, τ_{F} is constant because τ_{T} varies. Then, the resulting angle θ is measured. The figure also shows the fitted Gaussian process, where its intersection with the corresponding desired angle line determines the best τ_{T} obtained from the training data. When using the improved controller, τ_{L} takes the value corresponding to the intersection between the fitted Gaussian process and the desired bending angle.

Figure 9 depicts the measured endpoint position for two different controllers: (i) the model-based feedforward τ_{F} and (ii) the hybrid controller, i.e., the feedforward controller in combination with the learning approach explained in

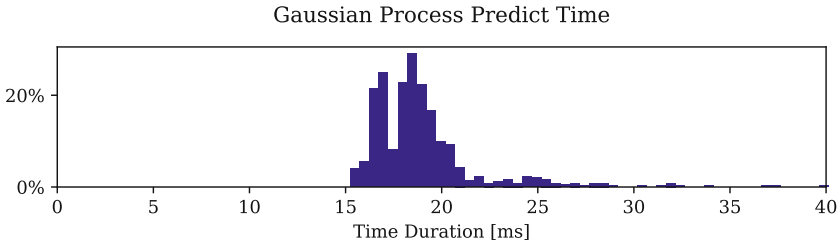


Fig. 7. Overview of the time for the controller to calculate the new τ_L during the experiments.

Subsect. 4.2. The experiments last one minute, which is equivalent to 6 rotations. The left polar plot in Fig. 9 shows the experimental results of the feedforward without correction torque, while the right polar plot in Fig. 9 depicts the experimental results considering the correction torque τ_L obtained via GPR. The correction torque is implemented only for Segment 2. The reason for this is that the learning approach considers the total bending of the manipulator. Hence, implementing a correction torque in both segments results in a larger error as the GPR tries to correct the error twice.

Figure 10 provides a comparison between the performance of the model-based controller and the hybrid one for different radii of the circular trajectory. In this figure, the improvement using the hybrid controller is particularly notorious for 15°.

6.1 Performance Assessment

Small errors in the desired bending angle can be caused by sensor noise or the internal PID controllers in the VTEM regulator. In contrast, large errors can be associated with the controller's limitations. Hence, the root-mean-square (RMS) error can be used as a performance index to compare the controllers. Figure 11 shows the resulting RMS errors for the model-based and improved controller for different desired bending angles. The figure shows that the RMS error is considerably reduced—approximately 40%—for a desired bending angle of 10°, highlighting a performance improvement that is not evident in Fig. 10. To further assess the performance of the proposed controllers, we compare the experimental results for three different desired bending angles—10°, 12°, and 15°—via the box plots provided in Fig. 12. The box plots show that the median corresponding to the improved controller is very close to the desired value for the three experiments. In contrast, the median corresponding to the model-based feedforward controller diverges from the desired value as this increases, as observed in the experiments where the desired angle is 12° and 15°. Moreover, the improved controller outperforms the model-based one in terms of precision (determined by the box size) in the experiments where the desired angle is 10° and 15°. Hence, we conclude that for the particular experiments where the desired angle is 15°, the improved controller performs considerably better than the model-based one.

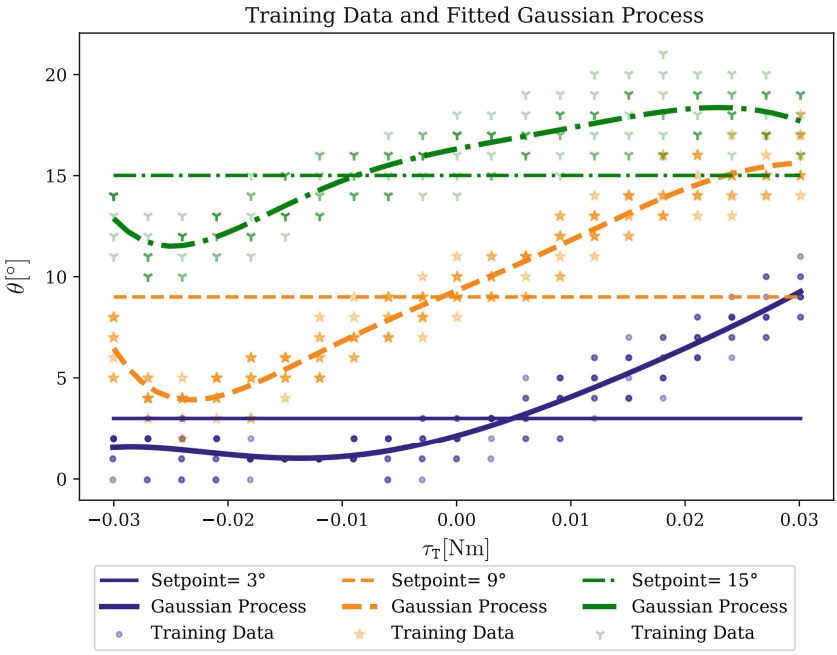


Fig. 8. A snippet of the training data. The resulting bending angle θ for varying τ_T. For readability, only three different desired angles $\theta_\star$ are shown.

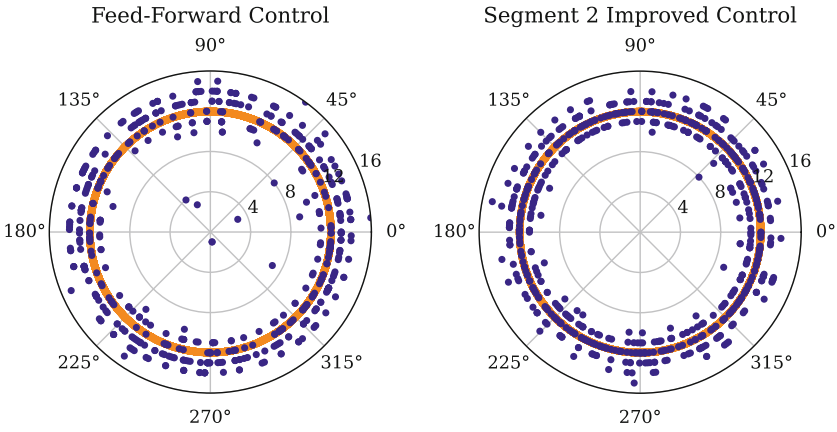


Fig. 9. Polar plot showing the circular trajectory of the soft robot for different controllers. (Left) Model-based feedforward controller. (Right) Feedforward controller improved with the learning approach. The target angle $\theta_\star = 12°$ is shown in orange. (Color figure online)

This result is unsurprising because gravity has a greater influence on the system dynamics as the bending angle increases.

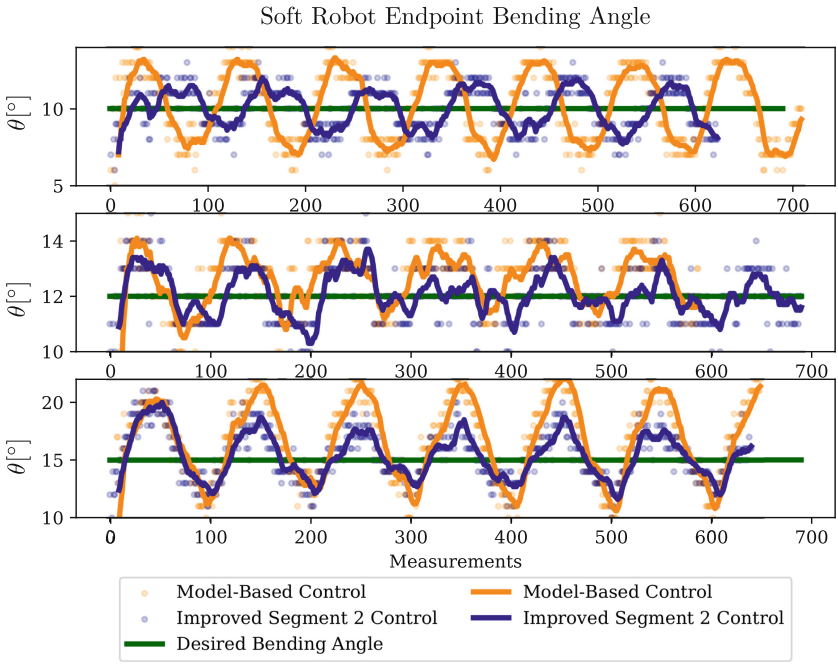


Fig. 10. Comparison between the feedforward and the hybrid controller. The task consists in describing circular trajectories with different radii. The soft robot bending angle is denoted as θ.

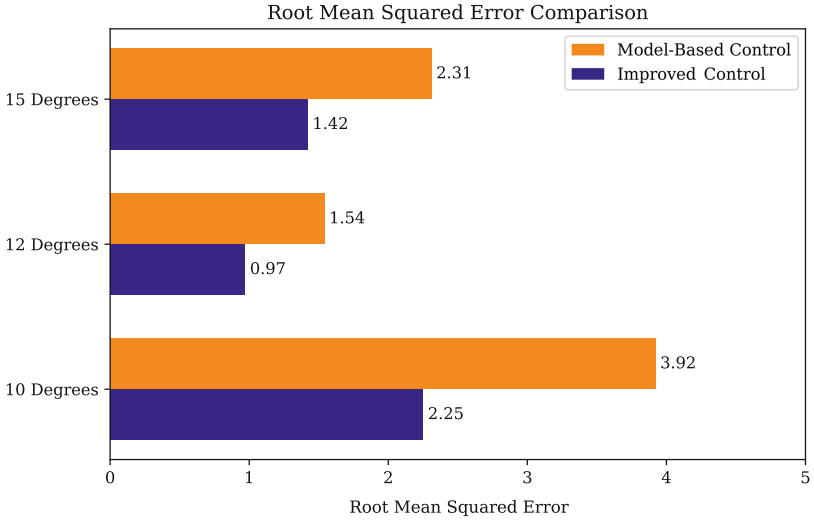


Fig. 11. RMS error graph for the model-based controller and the hybrid (improved) controller.

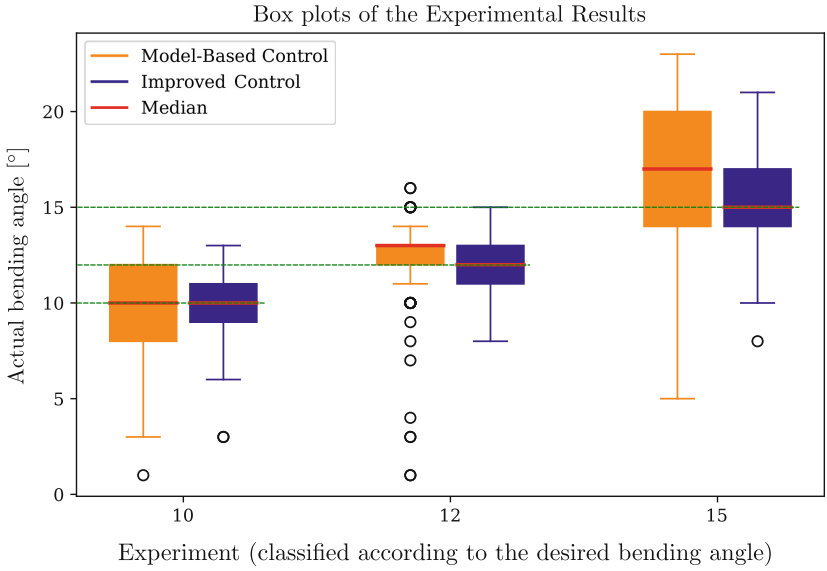


Fig. 12. Box plots that illustrate the accuracy and precision of the model-based and improved controllers. A smaller box indicates more precision, while the median indicates the task's accuracy. The desired angles are plotted with dashed green lines for reference. (Color figure online)

7 Concluding Remarks and Future Work

This paper proposed a simple yet effective hybrid controller for a pneumatic-driven soft manipulator. The proposed hybrid controller has been validated experimentally in a soft manipulator consisting of two segments made of silicone. Moreover, the benefits of combining learning and model-based strategies have been thoroughly analyzed by comparing the performance of the closed-loop system with and without the correction torque obtained via the learning approach. Unsurprisingly, the hybrid controller performs better in scenarios where the nonlinear phenomena neglected in the mathematical model significantly influence the system's behavior. However, it is remarkable that the proposed simplifications greatly reduced the analytical complexity of the model-based controller and guaranteed a learning process fast enough for implementation purposes.

The control design strategy is expected to be suitable for implementation in other systems with similar properties, e.g., tendon-driven continuum soft robots and different designs of pneumatic-actuated soft manipulators. Furthermore, the following items are proposed for future research:

- To include feedback terms in the model-based part of the hybrid control strategy. Feedback may be essential in control tasks involving disturbances.
- To analyze the performance of the proposed controller in scenarios with larger bending angles or load manipulation. Nonlinearities a more important as the

bending angle increases. The soft robot used in this paper is unsuitable for performing such control tasks because of its physical limitations.
- To adapt the learning part for control tasks focusing on the robot's posture instead of its ending point.

Acknowledgments. Cosimo Della Santina has been supported by the Horizon Europe Program from Project EMERGE under Grant 101070918.

References

1. Borja, P., Dabiri, A., Della Santina, C.: Energy-based shape regulation of soft robots with unactuated dynamics dominated by elasticity. In: 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft), pp. 396–402 (2022). https://doi.org/10.1109/RoboSoft54090.2022.9762071
2. Caasenbrood, B., Pogromskiy, A., Nijmeijer, H.: Energy-based control for soft manipulators using Cosserat-beam models. In: O. Gusikhin, H. Nijmeijer, K. Madani (eds.) Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2021, pp. 311–319 (2021)
3. Caasenbrood, B., Pogromsky, A., Nijmeijer, H.: Control-oriented models for hyper-elastic soft robots through differential geometry of curves. Soft Robot. **10**(1), 129–148 (2022)
4. Camarillo, D.B., Carlson, C.R., Salisbury, J.K.: Task-space control of continuum manipulators with coupled tendon drive. In: Khatib, O., Kumar, V., Pappas, G.J. (eds.) Experimental Robotics, pp. 271–280. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-00196-3_32
5. Della Santina, C., Bicchi, A., Rus, D.: On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. IEEE Robot. Autom. Lett. **5**(2), 1001–1008 (2020)
6. Della Santina, C., Duriez, C., Rus, D.: Model-based control of soft robots: a survey of the state of the art and open challenges. IEEE Control Syst. Mag. **43**(3), 30–65 (2023)
7. Della Santina, C., Katzschmann, R.K., Bicchi, A., Rus, D.: Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. Int. J. Robot. Res. **39**(4), 490–513 (2020). https://doi.org/10.1177/0278364919897292
8. Dubied, M., Michelis, M.Y., Spielberg, A., Katzschmann, R.K.: Sim-to-real for soft robots using differentiable FEM: recipes for meshing, damping, and actuation. IEEE Robot. Autom. Lett. **7**(2), 5015–5022 (2022)
9. Franco, E., Ayatullah, T., Sugiharto, A., Garriga-Casanovas, A., Virdyawan, V.: Nonlinear energy-based control of soft continuum pneumatic manipulators. Nonlinear Dyn. **106**(1), 229–253 (2021)
10. George Thuruthel, T., Ansari, Y., Falotico, E., Laschi, C.: Control strategies for soft robotic manipulators: a survey. Soft Rob. **5**(2), 149–163 (2018)
11. Gillespie, M.T., Best, C.M., Townsend, E.C., Wingate, D., Killpack, M.D.: Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. In: 2018 IEEE International Conference on Soft Robotics (RoboSoft), pp. 39–45 (2018). https://doi.org/10.1109/ROBOSOFT.2018.8404894
12. Hofer, M., Sferrazza, C., D'Andrea, R.: A vision-based sensing approach for a spherical soft robotic arm. Front. Robot. AI **8**, 630935 (2021)

13. Kapadia, A., Walker, I.D.: Task-space control of extensible continuum manipulators. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1087–1092. IEEE (2011)
14. Lakhal, O., Melingui, A., Merzouki, R.: Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator. IEEE/ASME Trans. Mechatron. **21**(3), 1326–1335 (2016). https://doi.org/10.1109/TMECH.2015.2490180
15. Lee, K.H., et al.: FEM-based soft robotic control framework for intracavitary navigation. In: 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 11–16 (2017)
16. Lloyd, P., et al.: A learnt approach for the design of magnetically actuated shape forming soft tentacle robots. IEEE Robot. Autom. Lett. **5**(3), 3937–3944 (2020). https://doi.org/10.1109/LRA.2020.2983704
17. Melingui, A., Merzouki, R., Mbede, J.B., Escande, C., Daachi, B., Benoudjit, N.: Qualitative approach for inverse kinematic modeling of a compact bionic handling assistant trunk. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 754–761 (2014). https://doi.org/10.1109/IJCNN.2014.6889947
18. Niu, L., Ding, L., Gao, H., Su, Y., Deng, Z., Liu, Z.: Closed-form equations and experimental verification for soft robot arm based on Cosserat theory. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6630–6635 (2019). https://doi.org/10.1109/IROS40897.2019.8968477
19. ch1Pustina, P., Borja, P., Della Santina, C., De Luca, A.: P-satI-D shape regulation of soft robots. IEEE Robot. Autom. Lett. **8**(1), 1–8 (2023)
20. Renda, F., Boyer, F., Dias, J., Seneviratne, L.: Discrete Cosserat approach for multisection soft manipulator dynamics. IEEE Trans. Rob. **34**(6), 1518–1533 (2018)
21. Rolf, M., Steil, J.J.: Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. IEEE Trans. Neural Netw. Learn. Syst. **25**(6), 1147–1160 (2014). https://doi.org/10.1109/TNNLS.2013.2287890
22. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: 31st International Conference on Machine Learning, ICML 2014, vol. 1 (2014)
23. Toshimitsu, Y., Wong, K.W., Buchner, T., Katzschmann, R.: Sopra: fabrication & dynamical modeling of a scalable soft continuum robotic arm with integrated proprioceptive sensing. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 653–660. IEEE (2021)
24. Trivedi, D., Lotfi, A., Rahn, C.D.: Geometrically exact models for soft robotic manipulators. IEEE Trans. Rob. **24**(4), 773–780 (2008). https://doi.org/10.1109/TRO.2008.924923
25. Trumic, M., Della Santina, C., Jovanovic, K., Fagiolini, A.: Adaptive control of soft robots based on an enhanced 3D augmented rigid robot matching. IEEE Control Syst. Lett. **5**(6) (2021)
26. Webster, R.J., III., Jones, B.A.: Design and kinematic modeling of constant curvature continuum robots: a review. Int. J. Robot. Res. **29**(13), 1661–1683 (2010)
27. Zhang, Z., Dequidt, J., Duriez, C.: Vision-based sensing of external forces acting on soft robots using finite element method. IEEE Robot. Autom. Lett. **3**(3), 1529–1536 (2018)