2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Open loop Dynamic Trajectory Tracking Control of a Soft Robot using Learned Inverse Kinematics combined with a Dynamic Model

Malte Grube[1], Svenja Drücker[1] and Robert Seifried[1]

Abstract— **As new soft robotic applications emerge, control requirements increase. Therefore, precise control methods for soft robots are needed. The main challenge in controlling soft robots is that soft robots are often underactuated and redundantly actuated at the same time. In addition, modeling is usually difficult due to large elastic deformations, unknown material parameters, and manufacturing inaccuracies.**

In soft robotics, so-called kinematic controllers, which neglect the dynamics of the system, are mainly used. In particular, data-driven controllers are very popular. However, more advanced applications of soft robots require increasingly faster and more accurate movements. Here, kinematic controllers are not sufficient anymore. A direct extension of existing data-driven kinematic controllers to dynamic control is usually not practical due to the huge amount of training data required.

This paper presents a new open-loop dynamic trajectory tracking control of a redundantly actuated soft robot. A combination of a kinematic data-driven controller based on neural networks and a dynamic model-based control approach based on model inversion with the servo-constraints approach is used. This combined approach preserves the advantages of learning-based kinematic controllers for the dynamic control of soft robots while keeping the amount of training data required low. Experimental results show the strength of this approach.

I. Introduction

Soft material robots are an emerging and rapidly expanding field of research with potential applications in various fields. Unlike traditional robots made of stiff materials such as steel, soft material robots are typically made of soft materials such as silicone with a stiffness range of $10^4 \dots 10^9$ Pa. This results in large deformations that require new modeling methods and control concepts.

In the following, a short overview of the existing control approaches for soft robots is given. A more detailed overview can be found in [1], [2], [3]. In the soft robotic community, a distinction is made between kinematic and dynamic control. Controllers are considered kinematic when inertia and damping properties are neglected. Another distinction is between model-based and model-free control. Note that in soft robotics controllers using data-based models are considered model-free [2]. Furthermore, control methods can be divided into open-loop and closed-loop control [3]. Because soft robots can usually deform continuously, they often have a very large number of degrees of freedom and are therefore underactuated. They are also often redundantly actuated. This makes them much more difficult to control than most rigid robots.

Kinematic controllers are the most commonly used controllers in soft robotics. Since dynamics are neglected, they are limited to slow motions to avoid oscillations. Most model-based kinematic controllers rely on direct inversion of the kinematics [4] or, if this is not possible, differential inverse kinematics [5], [6], [7] are often used. For highly nonlinear systems, soft robots with large manufacturing inaccuracies, and soft robots that are difficult to model, model-free kinematic controllers are widely used. Here, the mapping of actuation variables to control variables is usually learned by a neural network, see e.g. [8], [9].

When faster and more agile motion is required, kinematic control is often not sufficient. Dynamic controllers must be used. A dynamic control approach for soft robots is model predictive control in combination with both physics-based models [10], [11] and data-driven models [12], [13], [14]. Also, alternatives such as PD controllers [15], [16], sliding mode controllers [17], gain-scheduling controllers [1], [18], other model-based controllers [19] and even reinforcement learning approaches [20], [21], [22] find applications in this context. It should be noted that dynamic controllers are typically more computationally expensive than kinematic controllers and require more data and/or parameters during tuning. Therefore, in applications dynamic controllers are still very limited.

This paper focuses on the open-loop trajectory control of a simple tendon-actuated beam-shaped soft robot. Open-loop approaches are particularly interesting in soft robotics because they do not require sensors, which are usually difficult to integrate into soft robots. A combination of a kinematic data-driven control approach and a dynamic model-based control approach is used for the trajectory control. Therefore, the required control forces are divided into a kinematic and a dynamic part. The kinematic control forces are calculated using learned inverse kinematics represented by a neural network (NN). To calculate the required dynamic control forces, the servo-constraints approach for dynamic model inversion is used. The effectiveness of this simple control approach for the control of fast motions of soft robots is evaluated by experiments.

The paper is organized as follows: First, the test system and the modeling of the test system are presented. Then the control setup is shown and finally experimental results on the controller performance are presented.

II. Model of Test System

In this paper, the tip position trajectory tracking of a simple beam-shaped soft robot is considered in simulation

[1] Institute of Mechanics and Ocean Engineering, Hamburg University of Technology, 21073 Hamburg, Germany malte.grube@tuhh.de; svenja.druecker@tuhh.de; robert.seifried@tuhh.de

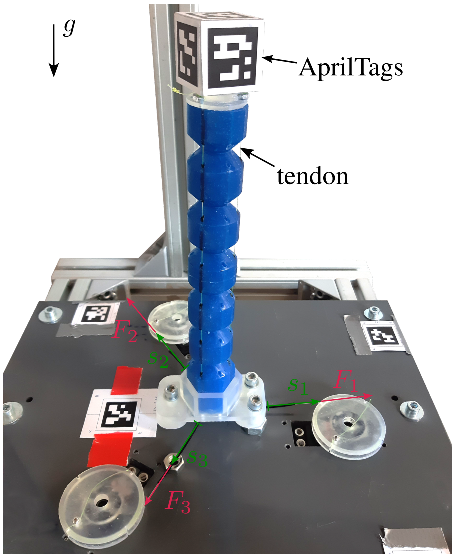

Fig. 1: Soft robot used for experiments.

and experiment. The experimental setup and the model used are described below. The model is used for the design of the controllers. The soft robot used is shown in Fig. 1. It is redundantly actuated by three tendons, allowing the robot tip to move on a hemisphere. The soft robot has a length of $202\,\mathrm{mm}$ and a radius of $30\,\mathrm{mm}$. The soft robot is made of silicone of type "HT45". It is actuated by three servos through tendons evenly distributed around the circumference of the soft robot. This allows to directly control the tendon length s_q for all three tendons. Here $q = 1 \ldots 3$ is the index of the tendon. A cube with AprilTag [23] fiducial markers is attached to the top of the soft robot to allow camera tracking of the tip position. The soft robot has two independent system outputs $\boldsymbol{z}$, in this work the x and y coordinates of the soft robot's tip are chosen.

A. Piecewise constant curvature Parametrization

The deformation of the soft robot is described with a piecewise constant curvature (PCC) model in the formulation of [24]. The soft robot is discretized into $N = 1$ segments of constant curvature with length $L_i = 202\,\mathrm{mm}$. Here i is the index of the segment. Note, that for $N = 1$ the PCC model is identical to the constant curvature model. PCC models are very popular in soft robotics because they can describe large deformations of beam-shaped robots with few parameters. Deformations other than bending, such as elongation, are of minor importance for most soft robots and are neglected here. The parameterization of a PCC segment is visualized in Fig. 2. The curvature is described by the rotations μ_i and ν_i around the x- and $y-$axis. Furthermore, the coordinate u describes the position of a cross section along the centerline. Here, $u = 0$ describes the start of the segment and $u = 1$ describes the tip of the segment.

The deformation of each segment can be described as a rotation around the axis

$$\boldsymbol{w} = \begin{bmatrix} \mu_i & \nu_i & 0 \end{bmatrix}^{\mathrm{T}} \tag{1}$$

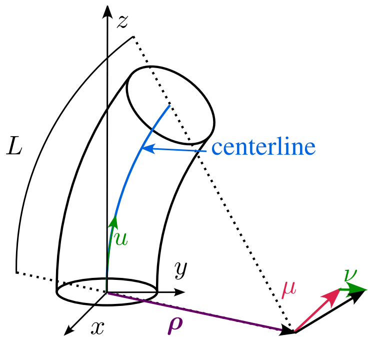

Fig. 2: PCC parametrization.

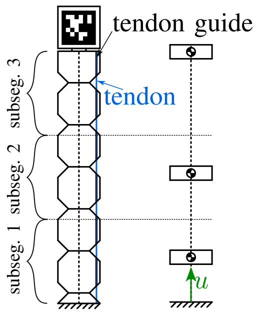

Fig. 3: Discretization of the soft robot for inertia forces.

which is located at position $\boldsymbol{\rho}_i$. The total bending angle φ_i of a segment and the vector $\boldsymbol{\rho_i}$ can be obtained by

$$\varphi_i = ||\boldsymbol{w}|| = \sqrt{\mu_i^2 + \nu_i^2}, \tag{2}$$

$$\boldsymbol{\rho_i} = \frac{L_i}{\varphi_i^2} \begin{bmatrix} \mu_i & \nu_i & 0 \end{bmatrix}^{\mathrm{T}}. \tag{3}$$

With this parametrization the position vector $\boldsymbol{r}_{u,i}$ and rotation matrix $\boldsymbol{R}_{u,i}$ of a cross section on the centerline of segment i at position u are

$$\boldsymbol{r}_{u,i} = \begin{bmatrix} \rho_{i,x}\sigma_{u,i} & \rho_{i,y}\sigma_{u,i} & ||\boldsymbol{\rho}_i|| \sin\varphi_{u,i} \end{bmatrix}^{\mathrm{T}}, \tag{4}$$

$$\boldsymbol{R}_{u,i} = \begin{bmatrix} \sigma_{u,i}\tilde{\nu}_i^2 + 1 & -\sigma_{u,i}\tilde{\mu}_i\tilde{\nu}_i & \tilde{\nu}_i \sin(\varphi_{u,i}) \\ -\sigma_{u,i}\tilde{\mu}_i\tilde{\nu}_i & \sigma_{u,i}\tilde{\mu}_i^2 + 1 & -\tilde{\mu}_i \sin(\varphi_{u,i}) \\ -\tilde{\nu}_i \sin(\varphi_{u,i}) & \tilde{\mu}_i \sin(\varphi_{s,i}) & \cos(\varphi_{u,i}) \end{bmatrix}. \tag{5}$$

Here $\varphi_{u,i} = u\varphi_i$, $\sigma_{u,i} = \cos(\varphi_{u,i}) - 1$, $\tilde{\nu}_i = \frac{\nu_i}{\varphi_i}$ and $\tilde{\mu}_i = \frac{\mu_i}{\varphi_i}$.

B. Internal Forces

For an accurate description of the internal forces, the soft robot segment is divided into $N_{\mathrm{subseg}} = 3$ subsegments of equal length, see Fig. 3. In the following, the subsegments are denoted by the index j. The mass and inertia properties of each subsegment are concentrated in the center of gravity of each subsegment, which is located at u_j. The cube on top of the soft robot is added to the top segment. The stiffness and damping properties are distributed among the elastic links between them. For simplicity, a linear material behavior is assumed. As shown in [25], this is a sufficiently good approximation for typical deformations of soft robots.

The internal torques $\boldsymbol{\ell}_{j,\mathrm{bnd}}$ resulting from the bending stiffness of the individual subsegments can be calculated as

$$\boldsymbol{\ell}_{j,\mathrm{bnd}} = \ell_{j,\mathrm{bnd}} \boldsymbol{R}_{u_j,1} \begin{bmatrix} -\sin(\theta_i) & \cos(\theta_i) & 0 \end{bmatrix}^{\mathrm{T}} \tag{6}$$

with

$$\ell_{j,\mathrm{bnd}} = EI_{xx,\mathrm{loc}} \frac{\sqrt{\mu_i^2 + \nu_i^2}}{L_i}, \tag{7}$$

$$\theta_i = \operatorname{atan2}(\nu_i, \mu_i). \tag{8}$$

The internal damping torques $\boldsymbol{\ell}_{j,\mathrm{dmp}}$ acting on the individual subsegments can be calculated from the angular velocity $\boldsymbol{\omega}_j$ and the damping coefficient d and result in

$$\boldsymbol{\ell}_{j,\mathrm{dmp}} = \begin{cases} d(\boldsymbol{\omega}_j) & j = 1 \\ d(\boldsymbol{\omega}_j - \boldsymbol{\omega}_{j-1}) & \text{otherwise.} \end{cases} \quad (9)$$

The the angular velocity $\boldsymbol{\omega}_j$ follows from the PCC-parametrization. The damping coefficient d has to be determined experimentally. Finally, gravity acting in negative z-direction is added.

C. Actuation

The soft robot is redundantly actuated by three servos via three tendons. With the servos the length of the three tendons is controlled. The length change of the tendons towards the neutral position is described by s_q, where $q = 1 \ldots 3$ is the tendon index. Since the soft robot is very stiff in the longitudinal direction, it can be assumed that no deformations are possible in the longitudinal direction. It follows that the three tendon lengths cannot be chosen independently. Also, s_q must be chosen so that the tendons do not become loose or break. It is also important to note that tendons cannot transmit tensile forces.

The tendon length change s_q can be decomposed into

$$s_q = s_{\mathrm{geo},q} + \underbrace{s_{\mathrm{stiff},q} + s_{\mathrm{dmp},q} + s_{\mathrm{inertia},q}}_{s_{\mathrm{el}} = F_q / c} + s_{\mathrm{err},q}. \quad (10)$$

Here $s_{\mathrm{geo},q}$ describes the change in length of the tendon due to the change in geometry of the soft robot during deformation. The term $s_{\mathrm{el}} = F_q/c$ summarizes the change in length due to the elasticity of the tendon and the elasticity of the soft robot in longitudinal direction, where c is a virtual spring constant that has to be determined experimentally. Further, s_{el} can be divided into a part resulting from the tendon forces due to the stiffness of the soft robot $s_{\mathrm{stiff},q}$, a part caused by damping forces $s_{\mathrm{dmp},q}$, and a part caused by inertia forces $s_{\mathrm{inertia},q}$. Additionally, when applying this to the physical model there remains an error $s_{\mathrm{err},q}$ which mainly results from fabrication inaccuracies and parameters not known exactly. If the soft robot moves very slowly, $s_{\mathrm{dmp},q}$ and $s_{\mathrm{inertia},q}$ disappear. For simplicity, in the simulation model the tendon force F_q is used instead of the length change of the tendon s_q. However, with (10) there is a direct relationship between s_q and F_q. The actuation is described in more detail in [26] and will therefore not be further described here.

D. Equations of Motion

For the derivation of the equations of motion with 2 degrees of freedom the generalized coordinates $\boldsymbol{y} = \begin{bmatrix}\boldsymbol{\mu}^{\mathrm{T}} & \boldsymbol{\nu}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2N}$ are chosen. The equations of motion can now be obtained by following the Newton-Euler formalism and can be written as

$$\boldsymbol{M}(\boldsymbol{y},t)\ddot{\boldsymbol{y}} + \boldsymbol{k}(\boldsymbol{y},\dot{\boldsymbol{y}},t) = \boldsymbol{q}(\boldsymbol{y},\dot{\boldsymbol{y}},t) + \boldsymbol{B}(\boldsymbol{y},t)\boldsymbol{u}(t). \quad (11)$$

Here, $\boldsymbol{M}(\boldsymbol{y},\boldsymbol{t}) \in \mathbb{R}^{2\times 2}$ is the mass matrix, $\boldsymbol{k}(\boldsymbol{y}) \in \mathbb{R}^2$ summarizes the Coriolis, centrifugal and gyroscopic forces and $\boldsymbol{q}(\boldsymbol{y}) \in \mathbb{R}^2$ represents the internal forces and torques. The tendon forces used for actuation are summarized in the vector $\boldsymbol{u}(t) = \begin{bmatrix}F_1(t) & F_2(t) & F_3(t)\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$, the input distribution matrix $\boldsymbol{B}(\boldsymbol{y},t) \in \mathbb{R}^{2\times 3}$ provides a mapping between the tendon forces $\boldsymbol{u}$ and the generalized coordinates.

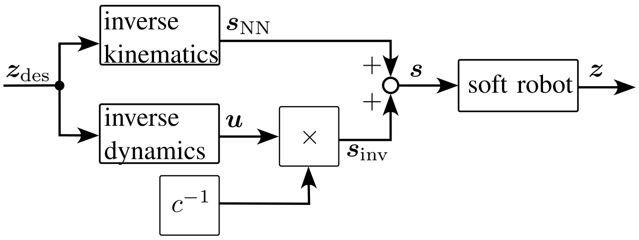

Fig. 4: Structure of the controller.

III. Control Setup

In this paper, an open-loop control approach with learned inverse kinematics and model-based inverse dynamics is chosen. The structure of the controller is shown in Fig. 4. In the following, first both parts of the controller are described separately, followed by an explanation of the combination of both parts.

A. Kinematic Controller

An inverse kinematics approach based on neural networks is chosen as the kinematic controller. A small neural network with two hidden layers is trained to calculate the required tendon length change $\boldsymbol{s} = \begin{bmatrix}s_1 & s_2 & s_3\end{bmatrix}$ of the desired tip position $\boldsymbol{z}_{\mathrm{des}}$ of the soft robot. The two hidden layers have a $\tanh$ activation function with 10 neurons each. The output layer has a linear activation function. Note, that the soft robot is redundantly actuated as described in Sec. II-C. Only two of the tendon lengths can be chosen independently. This is especially important for the collection of training data.

1) Training: To generate the training data, in a first step 218 tip-position points $\boldsymbol{z}_{\mathrm{tip,model}}$ evenly distributed over the workspace are sampled. In a second step, the forward kinematics of the model presented in Sec. II are used to calculate consistent values for the tendon length $\boldsymbol{s}_{\mathrm{model}}$ required to reach these points by the model. These values are already a good guess and can be used directly to train the neural network. However, because these values are purely model-dependent, they do not account for manufacturing inaccuracies and other unmodeled behavior of the soft robot. Therefore, it makes sense to use measured values of the physical robot for training. For this purpose, the calculated tendon lengths $\boldsymbol{s}_{\mathrm{model}}$ are used in a next step to approach each of the tip position points $\boldsymbol{z}_{\mathrm{tip,model}}$ one after the other. For each of the points the actual position of the tip-position $\boldsymbol{z}_{\mathrm{tip,meas}}$ is measured with the camera tracking system. Finally, the calculated tendon length values $\boldsymbol{s}_{\mathrm{model}}$ and the measured tip position values $\boldsymbol{z}_{\mathrm{tip,meas}}$ are used as a training set for the neural network.

Fig. 5 shows the projection into the xy-plane of the tip position points $\boldsymbol{z}_{\mathrm{tip,model}}$ calculated with the model and the measured tip position points $\boldsymbol{z}_{\mathrm{tip,meas}}$. It can be clearly seen that the overall behavior of the physical robot and the model agrees well. However, there are differences between

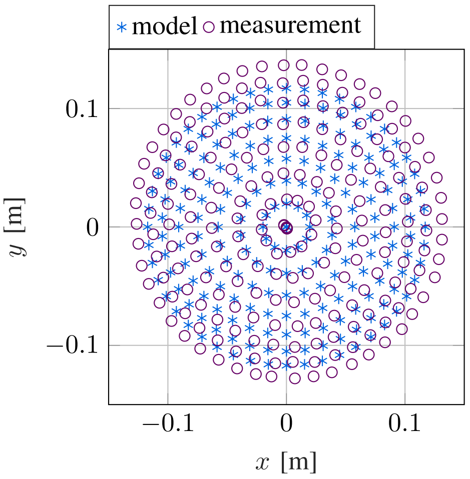

Fig. 5: Projection into the xy-plane of the NN training data.

the physical robot and the model that need to be taken into account to allow accurate trajectory tracking. Note that all parameters of the model are nominal values taken from a CAD model of the soft robot. With some parameter tuning, the performance of the model can usually be greatly improved. However, in soft robotics there are usually some effects that are very difficult to include in the model. Since the model is only used to obtain consistent values for the tendon length, parameter tuning is not required here. The entire training process takes less than $10\,\mathrm{min}$.

B. Dynamic Controller

As a dynamic controller the dynamic model inversion based on the servo-constrains approach is chosen. For this purpose, the dynamic model presented in Sec. II has to be adapted. In a first step, (11) is extended by an additional algebraic equation

$$\boldsymbol{s}(\boldsymbol{y},t) = \boldsymbol{z}(\boldsymbol{y},t) - \boldsymbol{z}_{\mathrm{des}}(t) = \boldsymbol{0} \tag{12}$$

which enforces the system output $\boldsymbol{z}$, which is the x- and y-coordinate of the tip-position trajectory, to be identical with the desired system output $\boldsymbol{z}_{\mathrm{des}}$. Equation (11-12) represent the inverse model. The unknowns of this system are the desired position coordinates $\boldsymbol{y}$, the desired velocity coordinates $\dot{\boldsymbol{y}}$ and the required system input $\boldsymbol{u}$. Since this leads to a system of DAEs of index 3, which is difficult to solve, the constraint equation is differentiated once to enforce the constraints on the velocity level only

$$\dot{\boldsymbol{s}}(\boldsymbol{y},\boldsymbol{v},t) = \dot{\boldsymbol{z}} - \dot{\boldsymbol{z}}_{\mathrm{des}} = \boldsymbol{0}. \tag{13}$$

This can also be written as

$$\dot{\boldsymbol{s}}(\boldsymbol{y},\boldsymbol{v},t) = \underbrace{\frac{\partial \boldsymbol{z}(\boldsymbol{y},t)}{\partial \boldsymbol{y}}}_{\boldsymbol{C}_{\mathbf{T}}} \boldsymbol{v} - \dot{\boldsymbol{z}}_{\mathrm{des}} = \boldsymbol{0}. \tag{14}$$

Here $\boldsymbol{C}_{\mathbf{T}}$ is the translational Jacobi-matrix of the end point position. In a second step, the number of system inputs must be adapted to the number of constraints.

As described in Sec. II, the robot has three control inputs, only two of which are independent. One possible choice is to apply a pulling force to two of the tendons and set the applied force for the third tendon to zero. Since tendons can only transmit pulling forces, the tendon force set to zero must be adapted to the required direction of the tendon force. This can be done by defining a new reduced system input $\boldsymbol{u}_{\mathrm{red}} = \begin{bmatrix} u_{\mathrm{red},1} & u_{\mathrm{red},2} \end{bmatrix}^{\mathrm{T}}$ with

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_{\mathrm{red}} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathbf{1}}^{\mathrm{T}} & \boldsymbol{\Phi}_{\mathbf{2}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \boldsymbol{u}_{\mathrm{red}}, \tag{15}$$

$$\boldsymbol{\Phi}_{\mathbf{1}} = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} & u_{\mathrm{red},1} \leq 0 \\ \begin{bmatrix} 0 & -1 & -1 \end{bmatrix} & \text{otherwise} \end{cases}, \tag{16}$$

$$\boldsymbol{\Phi}_{\mathbf{2}} = \begin{cases} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} & u_{\mathrm{red},2} \leq 0 \\ \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} & \text{otherwise} \end{cases}. \tag{17}$$

The reduced system input $\boldsymbol{u}_{\mathrm{red}}$ can be interpreted as a scaled projection of the three tendon forces in $x-$ and y-direction of the global coordinate frame.

Finally, using index reduction, the inverse model can be written as a first-order DAE in the form

$$\boldsymbol{v} = \dot{\boldsymbol{y}} \tag{18}$$

$$\boldsymbol{M}(\boldsymbol{y},\boldsymbol{t})\dot{\boldsymbol{v}} = \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y},t)\boldsymbol{\Phi}\boldsymbol{u}_{\mathrm{red}}(t) \tag{19}$$

$$\boldsymbol{C}_{\mathrm{T}}\boldsymbol{v} = \dot{\boldsymbol{z}}_{\mathrm{des}}. \tag{20}$$

In the following, this system is solved using the ode23t solver from MATLAB, which is based on the trapezoidal rule. Other solvers can also be used. In [27] an overview of the performance of different solvers for feedforward control using the the servo-constraints approach is given. In a post-processing step, the geometric length change $s_{\mathrm{geo},q}$ of the tendons is calculated from the position coordinates $\boldsymbol{y}$, and the tendon forces $\boldsymbol{u}$ are calculated from the reduced tendon forces $\boldsymbol{u}_{\mathrm{red}}$ using (15). This then allows to calculate the tendon length change

$$s_{\mathrm{inv},q} = s_{\mathrm{geo},q} + F_q/c. \tag{21}$$

C. Combination of the Controllers

For the combination of the two controllers it is important to consider (10) again. The change in tendon length calculated by the neural network can be written as

$$s_{\mathrm{NN},q} = s_{\mathrm{geo},q} + s_{\mathrm{stiff},q} + s_{\mathrm{err},q}. \tag{22}$$

It includes the change in tendon length due to the geometry and stiffness of the soft robot, as well as a learned approximation of unmodeled effects. Dynamic effects from damping and inertia are neglected. The tendon length change calculated by inverse dynamics can be written as

$$s_{\mathrm{inv},q} = s_{\mathrm{geo},q} + \underbrace{s_{\mathrm{stiff},q} + s_{\mathrm{dmp},q} + s_{\mathrm{inertia},q}}_{s_{\mathrm{el}}=F_q/c}. \tag{23}$$

This includes all modeled effects on the tendon length change but does not include unmodeled effects. In a post-processing step, a reduced version can also be calculated to

$$s_{\mathrm{inv,q,red}} = s_{\mathrm{dmp},q} + s_{\mathrm{inertia},q} = F_{q,\mathrm{red}}/c. \tag{24}$$

This is done by inserting the generalized coordinates $\boldsymbol{y}$ computed by the inverse model into (19), setting the stiffness of the soft robot to zero, and solving this linear system of equations for $\boldsymbol{u}_{\text{red.}}$. Now the outputs of the two controllers can simply be summed to get the change in tendon length of the combined controller.

IV. Experimental Evaluation

In the following, the agile trajectory tracking performance of the proposed combined controller is investigated and compared with the performance of the NN-based kinematic controller, which is one of the standard approaches in soft robotics. In a first step, a triangular trajectory with rounded corners is considered. The soft robot starts with zero velocity at the rightmost point of the triangle. In a first lap the velocity is slowly increased until the desired velocity is reached, then three laps with constant velocity and period time T are made and in a final lap the velocity is reduced to zero again. As a control error, for each time step the error E is defined as

$$E(t) = ||\boldsymbol{z}_{\text{des}}(t) - \boldsymbol{z}_{\text{exp}}(t)|| \tag{25}$$

where $\boldsymbol{z}_{\text{exp}}$ is the measured tip position obtained from the experiment.

In Fig. 6, the trajectory tracking result for the NN-based controller and for the proposed combination of both controllers is shown for a period time of $T = 1\,\text{s}$ representing a fast motion. For reference, the trajectory tracking result for the NN-based controller and a period time of $T = 5\,\text{s}$ representing a slow motion is also shown. It can be clearly seen that with the NN-based controller good results can only be achieved for the period time of $T = 5\,\text{s}$. A mean error of $E_{\text{mean}} = 5.1\,\text{mm}$ and a maximum error of $E_{\max} = 9.6\,\text{mm}$ could be achieved. This can be explained by the fact that this is the simplest control problem since the trajectory tracking is slow enough that inertia and damping forces can be neglected. With the combination of both controllers comparable results could be obtained for this slow motion. However, the contribution of the inversion-based controller is so small compared to the contribution of the NN-based controller that it has no impact. Therefore, the results for the combined controller and $T = 5\,\text{s}$ are not shown. For the fast motion with a period time of $T = 1\,\text{s}$, it can be clearly seen that the NN-based controller is not able to follow the trajectory. The resulting trajectory is much too large, rotated against the desired trajectory, and contains some unwanted extra loops. In contrast, the combination of both controllers shows a good trajectory tracking result also for the fast motion. A mean error of $E_{\text{mean}} = 8.6\,\text{mm}$ and a maximum error of $E_{\max} = 18.4\,\text{mm}$ could be achieved. These are very good results for a fast-moving soft robot. In Fig. 7 the contribution of the NN-based and the inversion-based controller to the control output of the combined controller is shown for the first tendon as an example. The results for the other two tendons are qualitatively comparable. For a more detailed examination of the influence of the dynamics on the actuation the mean tendon force for the trajectory tracking of a circle with a radius of $r = 90\,\text{mm}$ is examined for different velocities. The forces are calculated from the inverse model from (18-20) for different angular velocities $\omega = 2\pi/T$. The results are shown in Fig. 8. It can be seen that for slow motions with angular velocities smaller than $\omega = 0.6\,1/\text{s}$, there is almost no dependence of the mean tendon force on the period time. Here the inertial forces can be neglected, and the tendon forces are dominated by forces due to the stiffness of the soft robot. For faster motions the tendon force decreases and reaches its minimum at $\omega = 2.9\,1/\text{s}$ and starts to increase quadratically with increasing angular velocity. This behavior can be explained by the increasing influence of the inertial forces, especially the centripetal forces, which also increase quadratically with increasing angular velocity for decreasing period times.

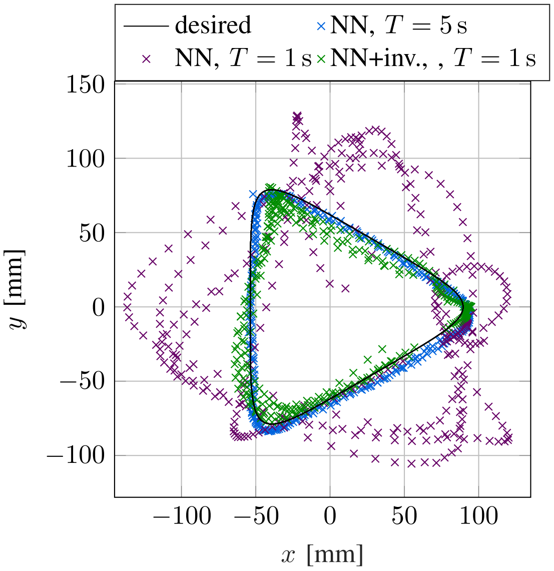

Fig. 6: Trajectory tracking results of a triangular trajectory with the NN-based and the combined controller for different period times T.

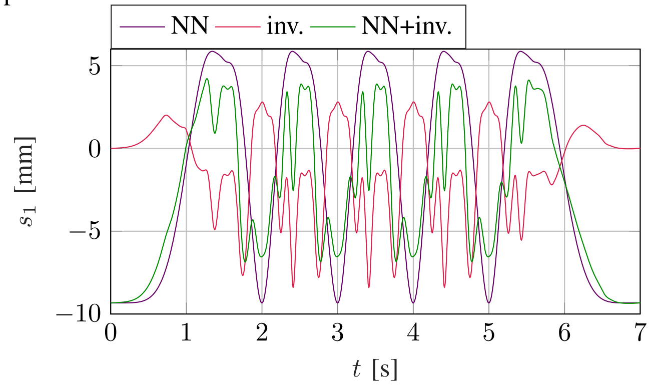

Fig. 7: Contribution of the two controllers to the tendon length change s_1 of the first tendon for the triangular trajectory with $T = 1\,\text{s}$ (fast motion).

This analysis is supported by the phase relationship between the tendon force F_q and the tendon length change s_q as a function of the angular velocity ω. For slow motions the tendon force F_q and tendon length change s_q have no phase shift, a large tendon length change s_q also requires a large

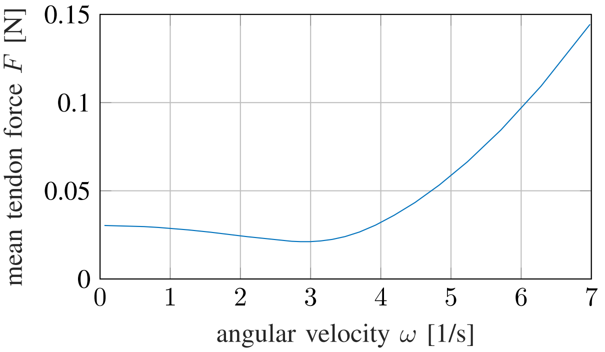

Fig. 8: Mean tendon force for a circular trajectory and different angular velocities ω.

tendon force F_q. However, for very fast motions a phase shift of $\approx 180°$ between F_q and s_q can be observed. This can be explained by the fact that for slow motions the forces resulting from the stiffness of the soft robot dominate, while for fast motions the inertial forces dominate. In the case of a circular trajectory, both forces point in opposite directions, leading to the phase shift of the tendon forces.

V. Conclusion

In this contribution the open-loop trajectory tracking control of a soft robot with a combination of a neural network based and an inversion based dynamic controller was presented. The NN is used to represent the inverse kinematics for quasi-static trajectory tracking. Forces resulting from damping and inertia are not included here, which limits the applicability of this controller alone to slow movements. Therefore, the NN-based controller is combined with a controller based on the servo-constraints approach for inverting the full dynamic model of the soft robot. For the modeling the piecewise constant curvature approach was used. This allows to consider the forces resulting from inertia and damping. It is shown in simulation and experiments that this extends the applicability of the combined controller to much faster and more agile trajectory tracking. Initial results show that the method can be applied to other systems, such as underactuated systems.

References

[1] Grube, M., Wieck, J.C., Seifried, R.: Comparison of modern control methods for soft robots. Sensors **22**(23) (2022) 9464

[2] George Thuruthel, T., Ansari, Y., Falotico, E., Laschi, C.: Control strategies for soft robotic manipulators: A survey. Soft Robotics **5**(2) (2018) 149–163

[3] Lee, C., Kim, M., Kim, Y.J., Hong, N., Ryu, S., Kim, H.J., Kim, S.: Soft robot review. IJCAS **15**(1) (2017) 3–15

[4] Camarillo, D.B., Carlson, C.R., Salisbury, J.K.: Configuration tracking for continuum manipulators with coupled tendon drive. IEEE Transactions on Robotics **25**(4) (2009) 798–808

[5] Bailly, Y., Amirat, Y.: Modeling and control of a hybrid continuum active catheter for aortic aneurysm treatment. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. (2005) 924–929

[6] Jones, B., Walker, I.: Kinematics for multisection continuum robots. IEEE Transactions on Robotics **22**(1) (2006) 43–55

[7] Renda, F., Armanini, C., Mathew, A., Boyer, F.: Geometrically-exact inverse kinematic control of soft manipulators with general threadlike actuators' routing. IEEE Robotics and Automation Letters **7**(3) (2022) 7311–7318

[8] Giorelli, M., Renda, F., Ferri, G., Laschi, C.: A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. (2013) 5033–5039

[9] Fang, G., Tian, Y., Yang, Z.X., Geraedts, J., Wang, C.: Jacobian-based learning for inverse kinematics of soft robots. IEEE/ASME Transactions on Mechatronics **27**(6) (2020) 5296–5306

[10] Best, C.M., Gillespie, M.T., Hyatt, P., Rupert, L., Sherrod, V., Killpack, M.D.: A new soft robot control method: Using model predictive control for a pneumatically actuated humanoid. IEEE Robotics & Automation Magazine **23**(3) (2016) 75–84

[11] Ouyang, B., Mo, H., Chen, H., Liu, Y., Sun, D.: Robust model-predictive deformation control of a soft object by using a flexible continuum robot. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE (2018) 613–618

[12] Hyatt, P., Killpack, M.D.: Real-time nonlinear model predictive control of robots using a graphics processing unit. IEEE Robotics and Automation Letters **5**(2) (2020) 1468–1475

[13] Bruder, D., Fu, X., Gillespie, R.B., Remy, C.D., Vasudevan, R.: Data-driven control of soft robots using koopman operator theory. IEEE Transactions on Robotics **37**(3) (2021) 948–961

[14] 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). (2018) 39–45

[15] Kapadia, A., Walker, I.D.: Task-space control of extensible continuum manipulators. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. (2011) 1087–1092

[16] Falkenhahn, V., Hildebrandt, A., Neumann, R., Sawodny, O.: Model-based feedforward position control of constant curvature continuum robots using feedback linearization. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). (2015) 762–767

[17] Kapadia, A.D., Walker, I.D., Dawson, D.M., Tatlicioglu, E.: A model-based sliding mode controller for extensible continuum robots. In: Proceedings of the 9th WSEAS international conference on Signal processing, robotics and automation. (2010) 113–120

[18] Wu, K., Zheng, G.: Fem-based gain-scheduling control of a soft trunk robot. IEEE Robotics and Automation Letters **6**(2) (2021) 3081–3088

[19] Santina, C.D., Katzschmann, R.K., Bicchi, A., Rus, D.: Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. The International Journal of Robotics Research **39**(4) (2020) 490–513

[20] Engel, Y., Szabo, P., Volkinshtein, D.: Learning to control an octopus arm with gaussian process temporal difference methods. In Weiss, Y., Schölkopf, B., Platt, J., eds.: Advances in Neural Information Processing Systems. Volume 18., MIT Press (2005)

[21] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In Xing, E.P., Jebara, T., eds.: Proceedings of the 31st International Conference on Machine Learning. Volume 32 of Proceedings of Machine Learning Research., Bejing, China, PMLR (22–24 Jun 2014) 387–395

[22] Marchese, A.D., Tedrake, R., Rus, D.: Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. The International Journal of Robotics Research **35**(8) (2016) 1000–1019

[23] Olson, E.: Apriltag: A robust and flexible visual fiducial system. In: 2011 IEEE international conference on robotics and automation, IEEE (2011) 3400–3407

[24] Allen, T.F., Rupert, L., Duggan, T.R., Hein, G., Albert, K.: Closed-form non-singular constant-curvature continuum manipulator kinematics. In: 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), IEEE (2020) 410–416

[25] Grube, M., Seifried, R.: Simulation of soft robots with nonlinear material behavior using the cosserat rod theory. In: 8th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2022, SCIPEDIA (2022)

[26] Allen, T.F., Rupert, L., Duggan, T.R., Hein, G., Albert, K.: Closed-form non-singular constant-curvature continuum manipulator kinematics. In: 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), IEEE (2020) 410–416

[27] Drücker, S., Seifried, R.: Trajectory-tracking control from a multibody system dynamics perspective. Multibody system dynamics **58**(3) (2023) 341–363