Data-Driven Steering of Concentric Tube Robots in Unknown Environments via Dynamic Mode Decomposition

Citation for published version:
Thamo, B, Hanley, D, Dhaliwal, K & Khadem, M 2023, 'Data-Driven Steering of Concentric Tube Robots in Unknown Environments via Dynamic Mode Decomposition', *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 856-863. https://doi.org/10.1109/LRA.2022.3231490

Digital Object Identifier (DOI):
10.1109/LRA.2022.3231490

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE Robotics and Automation Letters

Data-driven Steering of Concentric Tube Robots in Unknown Environments via Dynamic Mode Decomposition

Balint Thamo[1,2], David Hanley[1,2], Kevin Dhaliwal[2], Mohsen Khadem[1,2]

***Abstract*—Concentric Tube Robots (CTRs) are a type of continuum robot capable of manipulating objects in restricted spaces and following smooth trajectories. CTRs are ideal instruments for minimally invasive surgeries. Accurate control of CTR's motion in presence of contact with tissue and external forces will allow safe deployment of the robot in a variety of minimally invasive surgeries. Here, we propose a data-driven controller that can repeatedly and precisely direct the robot along predetermined deployment trajectories. The proposed controller doesn't rely on a mathematical model of the robot and employs Extended Dynamic Mode Decomposition (EDMD) to learn the nonlinear dynamics of the robot and the interaction forces on the fly. This enables the robot to follow desired trajectories in the presence of unknown perturbations, such as external forces. Experiments are carried out to evaluate the accuracy of the controller in steering the robot on arbitrary trajectories. Results demonstrate that the robot can track trajectories with a mean accuracy of 2.4 mm in repeated trials. Furthermore, we simulate scenarios where the robot is in contact with a rigid obstacle and is cutting through phantom tissue. Results show the robot can reach various static targets with a minimum accuracy of 2 mm.**

***Index Terms*—Model Learning for Control; Learning from Experience; Surgical Robotics: Steerable Catheters/Needles**

I. Introduction

CONCENTRIC tube robots (CTRs) are comprised of several nested pre-curved tubes. Shape and position of CTR end-effector can be controlled by axially rotating and translating the tubes (Fig. 1). Thanks to their small footprint and inherent flexibility, CTRs show great potential for manipulating objects in constrained environments. CTRs have been developed for use as steerable needles and robotic manipulators in a variety of surgical applications [1]. Precise control of the robot's motion is crucial to the safe deployment of CTRs.

A. Background

The kinematic model of the CTR proposed in [2], [3] is widely used to control the robot motion. The model can be used to pre-compute the forward kinematics model solutions over the entire workspace for real-time open-loop guidance of the robot [2] or estimate the differential kinematics of the robot for closed-loop control of the robot [4]. Researchers have developed dynamic models of CTRs [5] that can simulate CTRs nonlinear dynamics and accurately estimate robot motion under external forces. The models include several parameters representing mechanical characteristics of the robot that often are not easy to identify independently. Moreover, the models require accurate knowledge of external forces acting on the robot. Unfortunately, these forces are not always known, especially when the robot operates in an unknown environment or is in contact with a deformable object. Therefore, the majority of model-based approaches for controlling the motion of the CTRs assume the robot is moving in free space [2], [4], [6].

In [7], a radically different approach was used to estimate contacts using real-time medical images and machine learning. The controller employed this feedback to autonomously navigate the inner walls of the heart during cardiac surgery. Several researchers employed CTRs for teleoperated surgical interventions [8], [9]. During the teleoperation, the CTR is remotely-controlled by a user under visual feedback. The user can compensate for controller errors caused by contacts or external forces. In another work [10], a deep neural network was used to estimate robot contact forces at its tip as a function of the robot shape. This information can be used to update the robot's model for accurate control of the robot. However, application of these methods are limited where there is a lack of visual feedback or the robot is obscured by anatomical obstacles.

Another approach for safe deployment of CTRs is based on model-based motion planning with obstacle avoidance [11]–[13]. The motion planners employ pre-operative medical images to develop a cloud point representing obstacles [11], [12] or create a 3D map of robot task space [13]. Next, this data is used to generate collision free paths for the CTR. Finally, model-based control approaches are employed to steer the robot on the pre-planned path.

Data-driven controllers that learn the complex dynamics of the robot and its interaction with the environment can overcome the aforementioned difficulties. Commonly, these methods employ machine learning to learn inverse/forward kinematic or dynamic models [14], [15] or learn a direct control policy for moving the robot using Reinforcement Learning (RL) techniques with/without prior knowledge about geometric models [16]–[18]. The major disadvantages of these methods are the requirement for numerous training data. For example, a deep-neural-network based approach was proposed in [19] to learn inverse kinematics of CTRs and a dataset of 100000 samples were used to train the network. Gathering large data sets with a real robot is not always feasible. Additionally, relying only on simulation dataset for training leads to unsatisfactory results when the model is deployed on

Manuscript received: July 26, 2022; Revised: December 15, 2022; Accepted: November 30, 2022

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Medical Research Council [MR/T023252/1].

[1] School of Informatics, University of Edinburgh, UK.

[2] The Translational Healthcare Technologies Group in Centre for Inflammation Research, The Queen's Medical Research Institute, University of Edinburgh, UK

Digital Object Identifier (DOI): see top of this page

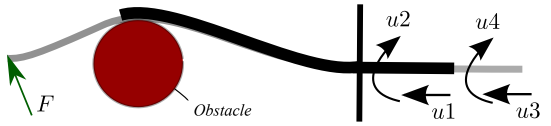


Figure 1. Illustration of a CTR with two tubes in contact with an obstacle and under external forces. The actuation variables u_i denote the proximal base rotation, and translation of the tubes.

the real robot [20]. Moreover, these models that are trained offline on experimental datasets [19] or simulated [18] datasets cannot capture the robot's behaviour in contact with the environment or external forces, as it would require a very large training dataset considering numerous robot configurations with various forces.

Some researchers [21] proposed online learning paradigms for control of flexible robots. In our previous work [22], we followed a similar approach to develop a model-less algorithm to estimate the inverse kinematics of the CTR on the fly using only the position measurement of the robot's tip. However, the proposed work was only tested in computer simulations. These methods require considerable time for the algorithm to converge and learn before reaching the desired accuracy. Additionally, the performance of the algorithm is very susceptible to the quality of feedback signals, which are commonly obtained using electromagnetic trackers or stereo cameras and have a low signal-to-noise ratio. Therefore, these methods often fail in practice due to the poor quality of the feedback signal or the slow learning phase. As a result, none of these methods have been tested on real CTRs.

B. Contribution

In this paper, we aim to control the motion of a CTR with unknown dynamics in contact with an unknown environment as shown in Fig. 1. The goal is to follow a desired trajectory without having prior knowledge of the external forces acting on the robot or obstacles restricting the robot's motion. It is assumed that we can only measure the Cartesian coordinates of the robot's tip position using commercially available electromagnetic trackers (EMT). To this end, we propose a data-driven method that overcomes the difficulties of the model-based approaches, including predicting the effects of unknown external forces, robot's dynamics, and unexpected disturbances that might happen in a real setting during the robot's motion. Additionally, the proposed controller requires only 250 samples to effectively learn the robot dynamics, which is significantly less than previous learning-based algorithms. We study the performance of the controller in extensive simulations and experiments. We consider a variety of scenarios including the robot under external forces, the robot in contact with an obstacle, and the robot cutting through phantom tissue, simulating percutaneous needle-based interventions. Our algorithm is available online.[1]

The remainder of the paper is organised as follows. Sections II details the proposed learning-based controller. In Section III, an extensive simulation study is performed to tune the parameters of the controller. Experimental evaluation of the controller and discussion of the results are presented in Section IV. Concluding remarks appear in Section V.

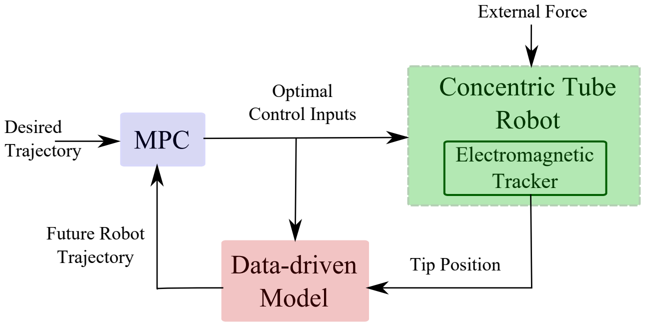


Figure 2. A block diagram of the proposed control strategy.

II. Methodology

Here, we present a holistic approach to model the dynamics of a closed-loop controlled CTR. We use several samples from the system to develop a linear state space model of the CTR, i.e., a model that accepts joint inputs and measurements of the robot's tip position over a fixed period of time and predicts the future position of the robot's tip. Each sample at time t_k contains the Cartesian coordinates of the robot's end-effector position x and the robot control inputs u at time t_k. The proposed model is linear, however, it captures the nonlinear dynamics of the robot and interaction forces. In particular, rather than describing the evolution of a dynamical system's state directly, which may be a nonlinear mapping, the Koopman operator [23] is employed to describe the evolution of continuous scalar valued functions of the state. Koopman operator has been previously proposed for data-driven modelling of nonlinear systems. A review of Koopman applications in nonlinear control can be found in [24]. Later, we apply a numerical approximation algorithm known as Extended Dynamic Mode Decomposition (EDMD) [25] to discretize the linear system. EDMD is an efficient numerical method previously proposed for reducing dimensionality of linear systems [26]. The final model, is a linear dynamical system that allows established linear control design methodologies to be used to design controllers for the robot. Finally, we employ a linear model predictive controller that employs the data-driven dynamic model to control the position of the CTR. A block diagram of the proposed control strategy is shown in Fig. 2.

A. Data-driven Modelling via Dynamic Mode Decomposition

This section describes a data-driven approach for modeling the dynamics of a CTR as a linear state space system. The basic idea is to lift (or embed) the nonlinear dynamics into a higher dimensional space where its evolution is approximately linear. For this purpose, we use a linear operator known as the Koopman operator. Later, we apply the Extended Dynamic Mode Decomposition (EDMD) to compute a finite-dimensional approximation of the operator to form an input/output discrete dynamical system representing the motion of the CTR.

[1] https://github.com/SIRGLab/CTR-EDMD.git

Without loss of generality, we assume the CTR is composed of two tubes. We can summarize the dynamic behaviour of the CTR as

$$x^+ = f(x,u), \tag{1}$$

where $x \in R^3$ is the state of the system representing the Cartesian coordinates of the robot's tip, $u \in R^4$ are the 4 control inputs shown in Fig. 1, x^+ is the successor state and f is the dynamic transition mapping.

The Koopman operator is an infinite-dimensional and linear operator originally developed to describe the evolution of nonlinear autonomous systems [23]. More specifically, it describes the evolution of functions of the states of a nonlinear system. We call these functions of states *observables*. To this end, we define a set of scalar valued observables as a mapping

$$g(x,u) : \mathbb{R}^3 \times \mathbb{R}^4 \to \mathbb{R}, \tag{2}$$

where g belongs to an infinite dimensional Hilbert space $\mathcal{H}$ [27]. Now, the Koopman operator with inputs and control (KIC) can be defined as a linear operator $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ such that

$$\mathcal{K}g(x,u) \equiv g(f(x,u),u^+). \tag{3}$$

Linear operators from Hilbert space to Hilbert space have associated eigenvalues and eigenfunctions, i.e.,

$$\mathcal{K}\phi_i(x,u) = \lambda_i \phi_i(x,u), \tag{4}$$

where ϕ_i is the i^{th} eigenfunction and λ_i is the i^{th} eigenvalue. Consequently, the observable functions can be written as a linear combination of all the eigenfunctions (since they form a basis)

$$g(x,u) = \sum_{i=1}^{\infty} \phi_i(x,u) v_i \tag{5}$$

where v_i is called the i^{th} Koopman mode associated with the i^{th} Koopman eigenfunction. In what follows, we approximate the infinite-dimensional KIC system with a finite dimensional system of size N:

$$\mathcal{K}g(x,u) \approx \sum_{i=1}^{N} \lambda_i \phi_i(x,u) v_i = Kg(x,u), \tag{6}$$

where K denotes the linear map from the observables' space to the finite dimensional approximation of the KIC's resulting Hilbert space, which we call the *lifted space*.

We compute this approximation to the Koopman operator using a method called extended dynamic mode decomposition [26]. First, the observables are partitioned into functions of the state Y_x, input Y_u, and both $Y_{x,u}$

$$Y = \begin{bmatrix} Y_x \\ Y_u \\ Y_{x,u} \end{bmatrix} = \begin{bmatrix} g_1(x,0) \\ \vdots \\ g_l(x,0) \\ g_{l+1}(0,u) \\ \vdots \\ g_{l+p}(0,u) \\ g_{l+p+1}(x,u) \\ \vdots \\ g_{l+p+j}(x,u) \end{bmatrix} = \mathbf{g}(x,u), \tag{7}$$

where $l,\ p,$ and j denote the number of basis functions used to estimate the states, inputs, and input-output coupled dynamics, respectively. Applying the Koopman operator to the observables gives

$$Z = \begin{bmatrix} Z_x \\ Z_u \\ Z_{x,u} \end{bmatrix} = \mathbf{g}(f(x,u),u^+) \tag{8}$$

We want the EDMD approximation of KIC to be linear and affine. Therefore, we set $Y_{x,u} = 0$. Keeping $Y_{x,u}$ would result (in the simplest case) in a bilinear approximation to the dynamical system. Predictors with bilinear form are not immediately suited for control design. Therefore, we decide to set $Y_{x,u} = 0$. This potentially reduces the model's accuracy. However, results in a linear control system that is computationally more efficient. Additionally, we are not concerned with the internal dynamics of the input, thus $Z_{x,u} = Z_u = 0$. We note that by definition then $Z_x = Y_x^+$.

The choice of functions in Y_x, Y_u and Z_x are a design choice. A common choice is polynomial functions of varying order [28]. In this work, we set the state vector observables to be

$$\begin{aligned} Y_x = [x_1 \quad & x_2 \quad x_3 \quad x_1^2 \quad x_2^2 \quad x_3^2 \\ x_1^2 x_2 \quad & x_1^2 x_3 \quad x_2^2 x_1 \quad x_2^2 x_3 \quad x_3^2 x_1 \quad x_3^2 x_2]^T \end{aligned} \tag{9}$$

and the input vector observables to be

$$Y_u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T. \tag{10}$$

Assuming that we collect a dataset of tip positions and control inputs at m consecutive time samples, we get

$$\begin{bmatrix} Z^0 & Z^1 & ... & Z^m \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} Y_x^0 & Y_x^1 & ... & Y_x^m \\ Y_u^0 & Y_u^1 & ... & Y_u^m \end{bmatrix} \tag{11}$$

where $K = \begin{bmatrix} A & B \end{bmatrix}$, superscripts $0, \cdots, m$ denote the time steps, with m commonly known the as DMD horizon showing the number of samples used to estimate the linear model. The elements of $Z_{x,u}$ are ignored here since we have set those to zero (as well as $Y_{x,u}$). We note that functions of states in Y_x, Y_u, and Z_x are computed directly from the tip measurements and control inputs. Now we can use the EDMD algorithm to estimate A and B. EDMD aims to calculate A and B so that we arrive at the linear differential equation

$$Y_x^+ = AY_x + BY_u. \tag{12}$$

Given the Datasets $Z^0, Z^1, ..., Z^m$, $Y_x^0, Y_x^1, ..., Y_x^m$, and $Y_u^0, Y_u^1, ..., Y_u^m$, first we Construct an input matrix

$$Y = \begin{bmatrix} Y_x Y_u \end{bmatrix} = \begin{bmatrix} Y_x^0 & Y_x^1 & ... & Y_x^m \\ Y_u^0 & Y_u^1 & ... & Y_u^m \end{bmatrix}, \tag{13}$$

and an output matrix

$$Z = \begin{bmatrix} Z^0 & Z^1 & ... & Z^m \end{bmatrix}. \tag{14}$$

Next, we compute the singular value decomposition (SVD) of the input matrix and partition the result into the first l rows associated with Y_x and the next p rows associated with Y_u.

$$Y = U\Sigma V^* = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \Sigma V^*. \tag{15}$$

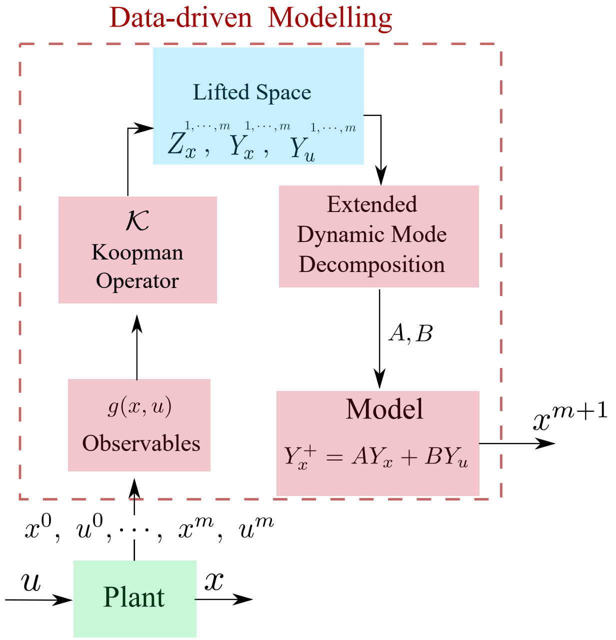


Figure 3. A block diagram of the proposed data-driven modelling approach.

Similarly, we compute the SVD of the output matrix

$$Z = \hat{U}\hat{\Sigma}\hat{V}^*. \tag{16}$$

Finally, we can estimate matrices A and B as

$$\begin{aligned} A &= \hat{U}^* Z V \Sigma^{-1} U_1^*, \hat{U} \\ B &= \hat{U}^* Z V \Sigma^{-1} U_2^*. \end{aligned} \tag{17}$$

Fig. 3 summarizes the proposed modelling approach. The model given in (12) will accept the m samples of input u and output x passed through the observables as Y_u and Y_x and predicts the successor observable states Y_x^+. We note that, based on (9), the first three elements of Y_x^+ will be the predicted future output of the plant, i.e., x^{m+1}. Additionally, rank of A describes the dimension of the lifted space (N in (6)) and is equal to the number of state vector observables. Based on (9), N is equal to 12. One can always select more observables to increase the model accuracy at the cost of reducing the EDMD's computational efficiency. In Section III, simulations are performed to demonstrate that 12 observables are sufficient for estimating the lifted space of the CTR.

B. Model Predictive Control

Here, we employ a model predictive control (MPC) algorithm that uses the linear data-driven model of CTRs in (12) to steer the CTR over a pre-defined desired trajectory x_d. The model predictive controller aims to find the control inputs (u) to drive the states (x) to some reference value over a fixed horizon by solving the following quadratic program:

$$\begin{aligned} &\text{minimize: } \sum_{i=1}^{M} (x^i - {x_d}^i)^T Q (x^i - {x_d}^i) + {u^i}^T R u^i \\ &\text{with respect to: } u \\ &\text{subject to:} \\ &Y_x^+ = AY_x + BY_u \\ &u_L < u^i < u_U. \end{aligned} \tag{18}$$

Table I
MECHANICAL PARAMETERS OF THE CTR USED IN SIMULATIONS.

	Tube 1	Tube 2
Inner Diameter[mm]	0.7	1.4
Outer Diameter[mm]	1.1	1.8
Length[mm]	431	332
Curvature[1/m]	21.3	13.1
Young's Modulus, E[GPa]	64.3	52.5
Shear Modulus, G[GPa]	25	21.4

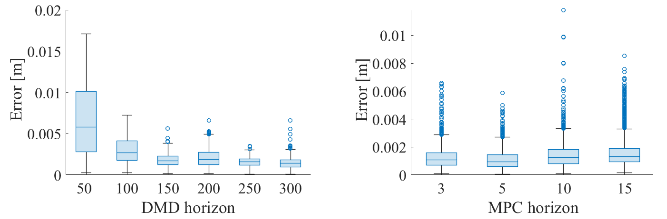
where $i = 1, \cdots, M$ denotes the time instant, M is the MPC horizon, Q and R are positive definite matrices penalizing tracking error and the control inputs. u_L and u_U denote the upper and lower limits of the joint inputs. In this work, we used the algorithm in [29] to solve the MPC problem. The algorithm employs the Nesterov Accelerated Gradient method to rapidly estimate the gradient of the cost function and improve the computational efficiency of the MPC. The gradient is later used in a gradient descent algorithm for updating control variables while minimising the cost function.

III. SIMULATION STUDY

In this section, we perform simulations to evaluate the performance of the proposed controller. Simulation environment for the CTR is developed using the mathematical model of the CTR presented in [3]. The robot is composed of two tubes. Mechanical characteristics of the robot used in the simulations are given in Table I. These are the parameters of an actual robot used in the experiments. The robot's tip was steered to follow a square trajectory with 20 mm base at a velocity of 1 mm/s.

In the first set of simulations of this square trajectory, we evaluated the effect of DMD horizon (m) on the accuracy of the controller. Six different horizons ranging from 50 to 300 were selected. To simulate the effects of noise in the sensory feedback, we added random Gaussian noise with standard deviation of 2 mm to the measured tip position. Results are summarised in Fig. 4(a). Furthermore, we simulated a scenario where the robot's tip's Cartesian coordinates are compared with only the first three states of the Lifted Space, which based on (9) correspond to the robot tip coordinates. The robot is moved randomly and the DMD horizon was set to 250. After the 250 samples, the robot tip position is compared with the first three states of the lifted space. We note that this is an open-loop simulation. The average error measured over ten trials is 0.3 mm with a standard deviation of 0.1 mm. Based on these results, we selected the DMD horizon m to be equal to 250 to be later used in the experiments, as it was found to achieve the minimum error.

In the next simulation, we evaluated the effects of MPC Horizon (M). Four different horizons ranging from 3 to 15 sample times were selected. The result of trajectory tracking error for various MPC horizons are shown in Fig. 4(b). The results did not show any significant difference between the different horizons. We selected the largest horizon (i.e., 15). The proposed data-driven approach accepts the system states, i.e, robot tip position, and corresponding control inputs for a fixed period of time (DMD horizon). Next, it develops a linear map between control inputs and robot states in a



(a) (b)

Figure 4. (a) Error bars comparing CTR's tip position error with respect to the DMD Horizon. Starting from the middle of the box, the line inside the box corresponds to the median error, box ends indicate the 25th and 75th percentiles, and the ends of the dashed lines are the maximum and minimum errors. (b) Error bars comparing CTR's tip position error with respect to the MPC Horizon.

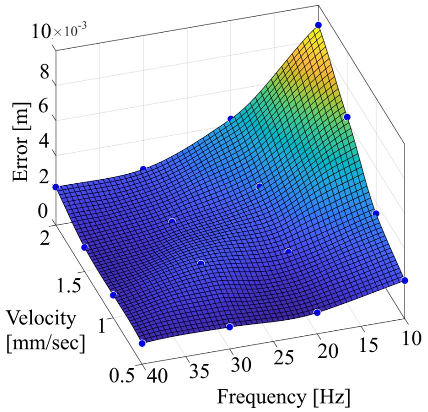


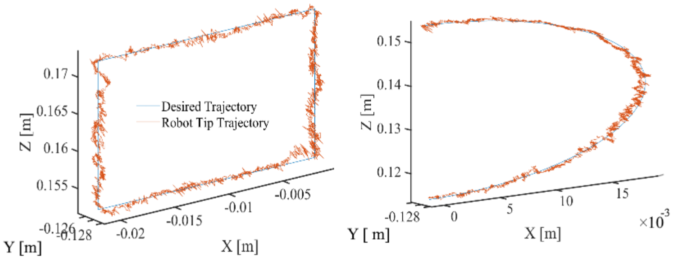
Figure 5. Tip position error with respect to sampling frequency and desired velocity.

higher dimensional space called the lifted space. As shown in Fig. 4(a), an optimal number of 250 samples are sufficient to learn the robot behaviour moving at a velocity of 1 mm/sec. We note that this is a continuous learning process. Therefore, a change in external disturbance will affect the robot states and the EDMD has the ability to adapt quickly and update the model accordingly. However, if the robot is moving too fast, the sampling frequency must be increased so that the model can capture the rapidly varying dynamics of the system. To investigate the effect of sampling frequency on the proposed model and controller, we performed 16 trials; in each trial the robot was asked to follow the same trajectory at velocities of 0.5, 1, 1.5, and 2 mm/sec, with sampling frequencies varying from 10 to 40 Hz. We note that the maximum sampling frequency of the electromagnetic tracking sensor used in the experiment is 40 Hz. Therefore, we limit the sensor frequency in the simulations to 40 HZ. Results are summarized in Fig. 5. The simulation demonstrates that the tracking error generally increases with respect to sampling frequency and tip velocity. Moreover, it is evident that the tracking error at higher velocities can be minimized by selecting a higher sampling frequency. Based on this analysis, we selected 1 mm/s as the robot's desired tip velocity and 40 Hz as the sampling frequency in our experiments.

The proposed approach can be generalised to a CTR with 3 tubes. For a 3-tube CTR, the control inputs consist of three translational movements and three rotational movements. Accordingly, we can update the input vector observables in (10) as follows

$$Y_u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{bmatrix}^T. \tag{19}$$

We performed a simulation study on a CTR with 3 tubes. The parameters of the robot were selected based on a real CTR [4]. First, the robot's tip was controlled to follow a square trajectory with a base length of 20 mm. Next, the robot's tip was steered to follow a circular trajectory with a 30 mm radius. 2 mm Gaussian noise was added to the tip position to simulate realistic sensory feedback. Results are shown in Fig. 6. The root mean squared error is 1.06 mm for the square trajectory and 1.27 mm for the circular trajectory. The results demonstrate that the proposed method offers similar performance for a CTR with 3 tubes.



(a) (b)

Figure 6. Results of simulation on a CTR with 3 tubes. (a) Tracking a square trajectory. (b) Tracking a circular trajectory.

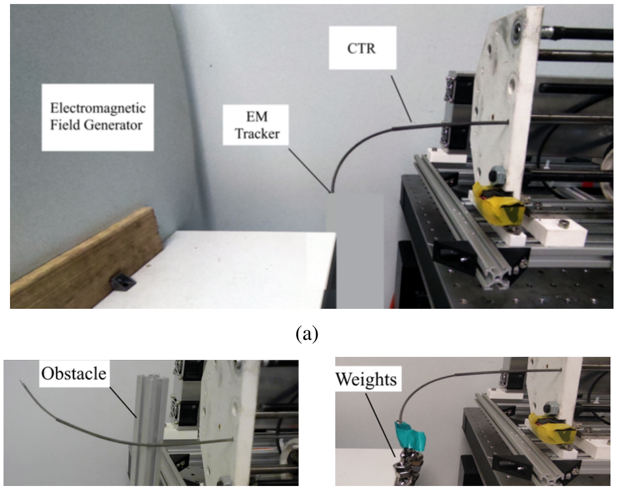
IV. Experiments and Discussions

Several experiments were performed to evaluate the performance of the proposed controller. The experimental setup consists of a concentric tube robot with two tubes and a 5-DOF electromagnetic tracker (Aurora, NDI) attached to its tip (Fig. 7(a)). Based on the manufacturer datasheet, the electromagnetic tracker has a mean accuracy of 0.7 mm and maximum error of 1.8 mm. The controller was implemented in Robot Operating System (ROS) in C++ and tested on a Desktop Computer with Intel(R) Core(TM) i9-12900K CPU processor and 32.0 GB of Memory.

The following four scenarios were considered in the experiments:

(S_1) *Robot following a pre-defined trajectory in free space:* The controller was tested on the CTR in free space. The robot's tip was steered to follow three different trajectories: (i) a square trajectory with 20 mm base length, (ii) a circular trajectory with a 30 mm radius and (iii) a longer trajectory towards the edge of the robot's workspace, which was given as a sequence of equally distanced random points. 10 trials are performed for each trajectory.

(S_2) *Robot under unknown external forces:* The controller was tested while a point load of 20 grams was applied at the tip of the robot (Fig. 7(c)). The controller is tasked to follow a square trajectory with 20 mm base length. 10 trials are performed.



(a)

(b)

(c)

Figure 7. (a) Experimental setup. An electromagnetic tracker is placed at the tip of the robot to measure the robot's tip position. (b) Robot in contact with an obstacle. (c) 20 gram weight is connected to the robot's tip.

Table II
COMPARISON OF ERROR BETWEEN DESIRED AND ACTUAL TRAJECTORIES. MEAN ERROR AND THE STANDARD DEVIATION OF ERROR ARE REPORTED OVER 10 TRIALS. THE VALUES ARE ALL IN MILLIMETER.

	Scenario S1			Scenario S2
Trajectory	Square	Circular	Point series	Square with weight
Mean error	1.40	0.748	2.40	2.00
std	0.97	0.450	1.70	1.70

(S_3) *Robot in the presence of an obstacle:* The robot's tip was steered to follow a sequence of points while a fixed obstacle was obstructing the robot's motion (Fig. 7(b)).

(S_4) *Robot in contact with phantom tissue:* The robot was required to perform multiple tasks while it was in contact with a phantom tissue.

Based on the simulation study presented in Section. III, the sampling frequency and the desired tip velocity in all the scenarios were selected as 40 Hz and 1 mm/sec, respectively. In addition to this, we selected the MPC parameters $\mathbf{Q} = 5 \times 10^7\mathbf{I}$, and $\mathbf{R} = \mathbf{I}$,. The MPC horizon was set to 15. The DMD horizon was set to 250 as well. Prior to the experiments, the robot's joint inputs were altered randomly for 10 seconds, while the corresponding tip positions were recorded. This data-set was used to initially update the data-driven model.

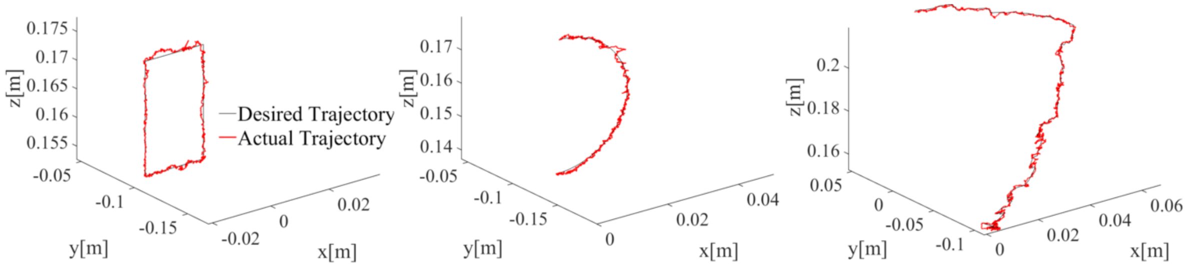
Representative results of the trajectory tracking in the first scenario (S_1) are shown in Fig. 8(a-c). The proposed data-driven algorithm is capable of following various trajectories with high accuracy. The mean tracking error and standard deviation of error are reported in Table II.

Additionally, we compared the proposed controller with the *hybrid controller* from our previous work [22]. The hybrid controller receives the model-based prediction of the robot's differential kinematics (i.e., Jacobian) and updates the Jacobian iteratively based on the feedback of position of the robot received from the electromagnetic sensor. As the controller relies on the robot model, we first performed a system identification experiment to identify model parameters accurately. Manual backbone segmentation established the base and shape of the CTR relative to the aligned calibration grid. Matching backbone points were selected in both images, and then triangulated to provide the 3-D point cloud. The extracted 3-D backbones were used to calibrate for the CTR model parameters, namely, Young's and shear moduli of the tubes. The parameters were identified by fitting the kinematic model given in [3] to the shape of the robot estimated via the cameras at 25 different configurations. The identified parameters of the model are given in Table I and are the same as those used in the simulations. Later, 20 more trials were performed to tune controller parameters and learning rate. The following controller parameters were chosen: control gain was set to $\mathbf{K}_p = 3\mathbf{I}$, learning rates λ_1, λ_2, and χ [22] were selected as 10, 10, and diag$[0.001,\ 0.001,\ 0.01,\ 0.1]$, respectively. These values were tuned via trial and error to achieve the minimum tracking error. Fig. 9 shows a comparison between the tracking error of the hybrid controller and the proposed data-driven controller. As it can be seen, both controllers offer similar performance. Of note, in contrast to the hybrid controller, the proposed data-driven controller does not require any information from the model.

In the second scenario (S_2), a weight of 20 grams was attached to the robot's tip. Of note, the 20 gram weight at the tip of the CTR causes a significant change in the CTR's shape and tip position as shown in Fig. 7(c). The controller was able to adapt to these changes without any prior knowledge of the weight and accurately follow the desired trajectory. The mean error and the standard deviation of the error over 10 trials are 1.7 mm and 2 mm, respectively.

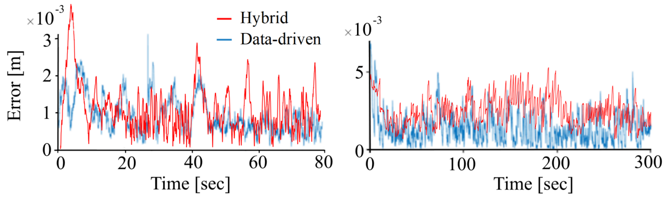
In the third scenario (S_3), the robot was tasked to follow several target points while it was in continuous contact with an obstacle. The target positions were selected to be at least 1 cm apart. The controller stops once it reaches the vicinity of the target and the error becomes less than 2 mm. This value was selected based on the accuracy of the electro-magnetic sensor (1.8 mm). Results are shown in Fig. 11. In all cases, the robot was able to reach the desired target positions with maximum error of 2 mm.

In the fourth scenario (S_4), the robot was navigated in the presence of a phantom tissue. We simulated two clinical scenarios. The first case study simulated percutaneous needle-based interventions such as prostate brachytherapy, where a needle is steered on a straight line to reach a deeply nested target in soft tissue [30]. The targets were selected inside the tissue within the robot's workspace, 65 mm from the entry point (Fig. 12(b)). The 2nd case study simulates percutaneous lung biopsy, where a needle is inserted through the skin, puncturing the pleura to reach the peripheral lung for sampling. Commonly, multiple incisions are used to reach multiple suspicious areas. Here, the robot is tasked to cut through a phantom tissue with a 25 mm thickness, simulating skin and pleura. Later it is tasked to reach 6 different points spread across its workspace from the same point of entry. We demonstrate that we can reach multiple sites via a single entry point (Fig. 12(b)). The phantom tissue in all the case studies is made following the recipe given in [31]. The tissue is made by mixing bovine gelatin powder with water at a temperature



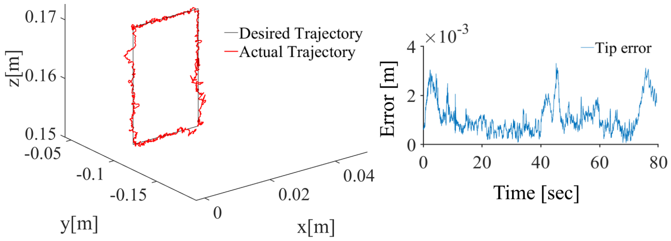
(a) (b) (c)

Figure 8. Representative experimental results for trajectory tracking in the 1st scenario (S_1). (a) Square Trajectory. (b) Circular Trajectory. (c) Long trajectory towards the edge of robot workspace.



(a) (b)

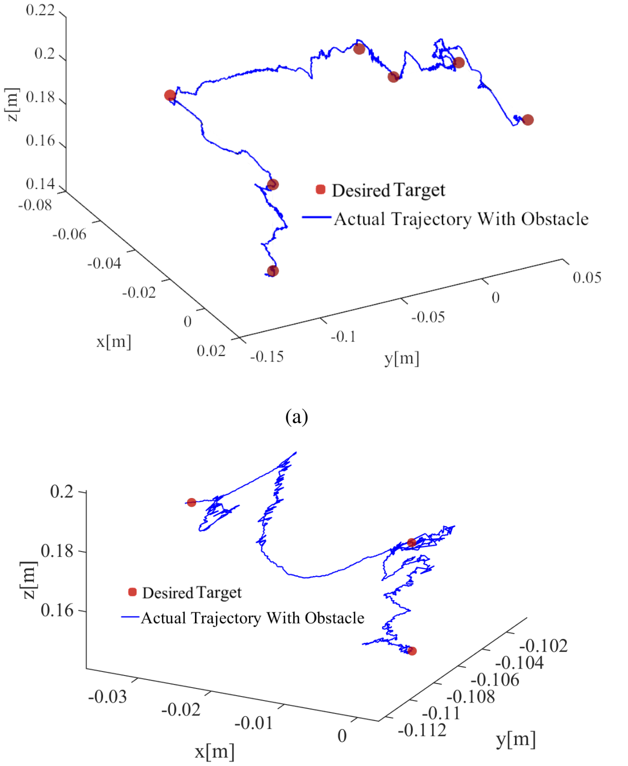
Figure 9. A comparison between the error of the proposed data-driven controller (blue line) and a hybrid controller developed in [22] (red line). (a) Following a square trajectory. (b) Following a circular trajectory.



(a) (b)

Figure 10. Result for the second experimental scenario (S_2), an 20g weight is attached to the robot's tip while it's following a square trajectory. (a) Robot tip trajectory. (b) Tracking error.

of 70°C. The weight ratio of gelatin-to-water in the mixture is 18% and the tissue's Young's modulus of elasticity was estimated to be 59 kPa. The elasticity of the synthetic tissues is similar to what is found in animal tissue. Although navigating the CTR inside a soft tissue is particularly challenging, the first case study shows that the robot is able to move along its tip direction following a straight line with a length of 65 mm inside the tissue (Fig. 12(a)). Figure 12(c) shows a representative result from this experiment. Over five insertions, the mean error of following the straight trajectory was 1.75 mm. As demonstrated by the second experiment, the robot's tip can reach target locations in different directions within its workspace despite making continuous contact with soft tissue. The robot was able to move its tip within 2 mm of the target in all 6 cases. (Fig. 12(d)).



(a)

(b)

Figure 11. Experimental results for the 3rd scenario (S_3). The CTR is tasked to reach several target points shown by red circles while it is in contact with an unknown obstacle.

V. Concluding Remarks

In this paper, we introduced a data-driven control strategy for autonomous steering of Concentric Tube Robots (CTR). The control strategy relies only on a limited data-set and is capable of rapidly learning the robot's nonlinear dynamics resulting in fast and accurate convergence. We studied the performance of the controller in extensive simulations and experiments. We consider a variety of scenarios, including the robot under external forces and the robot in contact with an obstacle. The proposed controller was capable of following a variety of trajectories within the robot's workspace with a maximum mean error of 2.4 mm at a frequency of 40 Hz and a velocity of 1 mm/s. Additionally, the controller was capable of steering the robot in the presence of unknown

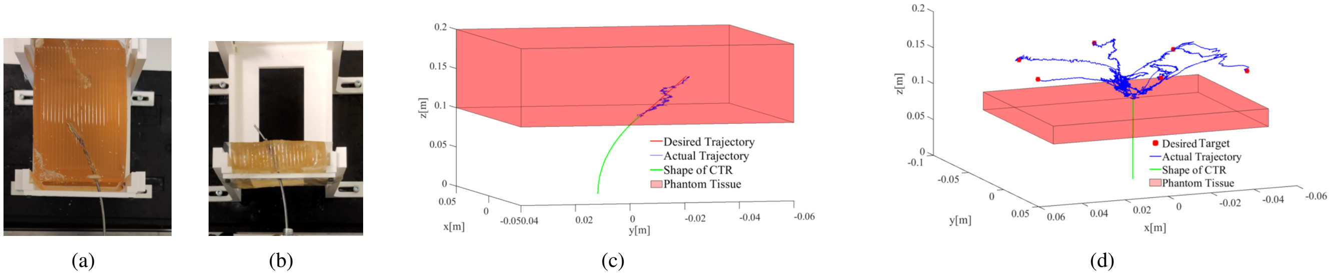


Figure 12. Experimental setup and results for the fourth scenario (S_4) (a) Robot's tip following a straight line while cutting through phantom tissue. (b) Robot cuts through the tissue to reach multiple target positions. (c) Results for robot cutting tissue along a straight line. (b) Results for robot's tip reaching 6 different target positions after cutting through tissue.

obstacles with a maximum error of 2 mm. Future work focuses on experimentally validating the proposed approach on CTRs with 3 tubes and investigating application of the DMD for controlling the robot's orientation in surgical tasks requiring higher dexterity.

References

[1] A. W. Mahoney, H. B. Gilbert, and R. J. W. III, *A Review of Concentric Tube Robots: Modeling, Control, Design, Planning, and Sensing*, ch. Chapter 7, pp. 181–202.

[2] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 209–225, 2009.

[3] D. C. Rucker, B. A. Jones, and R. J. Webster, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 769–780, 2010.

[4] M. Khadem, J. O'Neill, Z. Mitros, L. d. Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots for enhanced force/velocity manipulability," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 2197–2204.

[5] J. Till, V. Aloi, K. E. Riojas, P. L. Anderson, R. J. Webster III, and C. Rucker, "A dynamic model for concentric tube robots," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1704–1718, 2020.

[6] D. C. Rucker and R. J. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 945–950.

[7] G. Fagogenis, M. Mencattelli, Z. Machaidze, B. Rosa, K. Price, F. Wu, V. Weixler, M. Saeed, J. E. Mayer, and P. E. Dupont, "Autonomous robotic intracardiac catheter navigation using haptic vision," *Science Robotics*, vol. 4, no. 29, p. eaaw1977, 2019.

[8] A. Razjigaev, R. Crawford, J. Roberts, and L. Wu, "Teleoperation of a concentric tube robot through hand gesture visual tracking," in *IEEE Int. Conf. on Robotics and Biomimetics*, 2017, pp. 1175–1180.

[9] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster, "A telerobotic system for transnasal surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 996–1006, 2014.

[10] H. Donat, S. Lilge, J. Burgner-Kahrs, and J. J. Steil, "Estimating tip contact forces for concentric tube continuum robots based on backbone deflection," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 4, pp. 619–630, 2020.

[11] L. G. Torres, C. Baykal, and R. Alterovitz, "Interactive-rate motion planning for concentric tube robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1915–1921.

[12] A. Kuntz, M. Fu, and R. Alterovitz, "Planning high-quality motions for concentric tube robots in point clouds via parallel sampling and optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2205–2212.

[13] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric tube robots: Rapid, stable path-planning and guidance for surgical use," *IEEE Robotics & Automation Magazine*, vol. 24, no. 2, pp. 42–53, 2017.

[14] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2020, pp. 417–423.

[15] D. Bruder, C. D. Remy, and R. Vasudevan, "Nonlinear system identification of soft robot dynamics using koopman operator theory," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6244–6250.

[16] R. Morimoto, S. Nishikawa, R. Niiyama, and Y. Kuniyoshi, "Model-free reinforcement learning with ensemble for a soft continuum robot arm," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*. IEEE, 2021, pp. 141–148.

[17] D. Büchler, S. Guist, R. Calandra, V. Berenz, B. Schölkopf, and J. Peters, "Learning to play table tennis from scratch using muscular robots," *IEEE Transactions on Robotics*, 2022.

[18] K. Iyengar, G. Dwyer, and D. Stoyanov, "Investigating exploration for deep reinforcement learning of concentric tube robot control." *Int. J. of Computer Assisted Radiology and Surgery*, 2020.

[19] N. Liang, R. M. Grassmann, S. Lilge, and J. Burgner-Kahrs, "Learning-based inverse kinematics from shape as input for concentric tube continuum robots," in *IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 1387–1393.

[20] G. Fang, Y. Tian, Z.-X. Yang, J. M. Geraedts, and C. C. Wang, "Efficient jacobian-based inverse kinematics with sim-to-real transfer of soft robots by learning," *IEEE/ASME Transactions on Mechatronics*, 2022.

[21] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Trans. on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.

[22] B. Thamo, F. Alambeigi, K. Dhaliwal, and M. Khadem, "A hybrid dual jacobian approach for autonomous control of concentric tube robots in unknown constrained environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2021, pp. 2809–2815.

[23] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[24] S. E. Otto and C. W. Rowley, "Koopman operators for estimation and control of dynamical systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 59–87, 2021.

[25] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[26] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *J. of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[27] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing koopman theory to allow for inputs and control," *SIAM J. on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[28] ——, "Dynamic mode decomposition with control," *SIAM J. on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[29] M. Kögel and R. Findeisen, "A fast gradient method for embedded linear predictive control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1362–1367, 2011, 18th IFAC World Congress.

[30] M. Khadem, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli, "Robotic-assisted needle steering around anatomical obstacles using notched steerable needles," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1917–1928, 2018.

[31] ——, "A two-body rigid/flexible model of needle steering dynamics in soft tissue," *IEEE/ASME Trans. on Mechatronics*, vol. 21, no. 5, pp. 2352–2364, 2016.