2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Model Predictive Control for a Tendon-Driven Surgical Robot with Safety Constraints in Kinematics and Dynamics

Francesco Cursi*[1,3], *Student Member, IEEE*, Valerio Modugno[2], Petar Kormushev[3], *Member, IEEE*

***Abstract*— In fields such as minimally invasive surgery, effective control strategies are needed to guarantee safety and accuracy of the surgical task. Mechanical designs and actuation schemes have inevitable limitations such as backlash and joint limits. Moreover, surgical robots need to operate in narrow pathways, which may give rise to additional environmental constraints. Therefore, the control strategies must be capable of satisfying the desired motion trajectories and the imposed constraints. Model Predictive Control (MPC) has proven effective for this purpose, allowing to solve an optimal problem by taking into consideration the evolution of the system states, cost function, and constraints over time. The high nonlinearities in tendon-driven systems, adopted in many surgical robots, are difficult to be modelled analytically. In this work, we use a model learning approach for the dynamics of tendon-driven robots. The dynamic model is then employed to impose constraints on the torques of the robot under consideration and solve an optimal constrained control problem for trajectory tracking by using MPC. To assess the capabilities of the proposed framework, both simulated and real world experiments have been conducted.**

I. Introduction

Accuracy and precision are of uttermost importance to ensure safety in many robotic applications, especially in minimally invasive surgery, where little (or preferably no) damage should be caused to the patient's body. Moreover, effective control strategies are needed to guarantee safe execution of surgical tasks [1]. Due to mechanical design and actuation, robots have inherent limitations such as joint position, velocity, acceleration, and torque bounds. In addition, the operational environment may also lead to other constraints due to limited working volume or safety margins [2]. This is particularly true for minimally invasive surgery, where motion often occurs in very narrow and constrained spaces. Controllers capable of ensuring constraint satisfaction and task execution are therefore vital in these scenarios.

Different approaches have been investigated to guarantee limit avoidance in robotic systems. The Gradient Projection Method [3]–[5] uses the projection of a secondary task onto the null space of the primary task to guarantee the execution of the desired motion, while minimizing a secondary desired cost function. This method works only with redundant robots and does not always guarantee constraint satisfaction, since bounds are converted to soft bounds with low priority [6]. The Augmented Jacobian method [7] allows constraints to be placed at the same priority level of the main task by augmenting the robot Jacobian. Many other approaches have also been developed such as nonlinear variables transformation [8], [9], adding repulsive forces pushing the joints far from bounds [10], using barrier functions [11], Saturation in the Null Space [12], constrained stochastic optimization [13].

Fig. 1: The Micro-IGES surgical robotic tool [16]

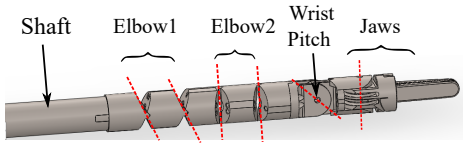
An optimal way to deal with many and various constraints is to formulate the control problem as a Quadratic Programming (QP) optimization with a desired cost function to minimize (e.g. tracking error) with respect to the control variable, and bounds to satisfy [14]. Despite the great effectiveness of finding optimal solutions, this method is only locally temporally optimal, since it doesn't take into account the future evolution of the controlled system [15].

In order to improve the optimality of the solution, Model Predictive Control (MPC) can be adopted. In fact, it allows to include the evolution of the system within a defined prediction horizon and thus obtain smoother solutions.

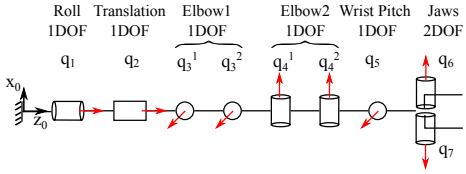
MPC might be employed for tracking control tasks where a desired trajectory is known or at least can be defined within the prediction horizon (i.e. for reference paths that change during the task execution). This is the typical case of many surgical applications where motion tasks are usually known in advance, such as in knot-tying, suturing, or tumor resection. Even though the surgical environment is less structured, it is still possible to estimate how it is evolving, for instance, by tracking tissue deformation [17], and thus still have some tracking reference.

Different types of mechanical transmissions have been used in the design of surgical robots, with the vast majority

*Corresponding author
[1] Hamlyn Centre, Imperial College London, Exhibition Road, London, UK
[2] Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy
[3] Robot Intelligence Lab, Imperial College London, 25 Exhibition Road, London, UK
Email: [f.cursi17,p.kormushev]@imperial.ac.uk,modugno@diag.uniroma1.it



(a) CAD model



(b) Kinematic model

Fig. 2: Micro-IGES surgical tool: 2a) CAD model; 2b) kinematic model

being tendon-driven [18]. Many research efforts focused on modelling this kind of transmission analytically [19]–[22]. Nonetheless, due to the highly complex nonlinearities in tendon-driven systems, other researchers employed machine learning approaches [23] such as Artificial Neural Networks [24], Gaussian Mixture Regression, K-nearest Neighbour Regression, and Extreme Machine Learning [25].

Thus far very little work has been conducted on optimal control of surgical robots, especially in the case of constrained systems, both kinematically and dynamically. Even though in robotic surgery motions are typically not very fast, dynamic constraints may rise to limit the magnitude of the applied end-effector force or to limit tendon forces, as for tendon-driven systems. Recently, Faroni et al. [15] presented a work on MPC for controlling an articulated industrial robot under only kinematic constraints (on joint positions, velocities, accelerations).

The main contribution of our manuscript is to apply an MPC approach to the Micro-IGES [16] surgical robot in order to allow it to follow a desired path (as could be for tumor resection), while satisfying different constraints. In this work, however, we are not directly addressing the kinematic mapping, which is described in [26], but rather providing a framework which allows to satisfy imposed constraints, given a certain kinematic model. Ongoing work is focusing on improving the kinematic model accuracy and will then be merged with this presented work.

The paper is structured as follows. Section II presents the Micro-IGES robotic surgical tool (Figure 1). Section III describes the approach to solve the MPC for trajectory tracking under the imposed kinematic and dynamic constraints. In Section IV the results for modelling the dynamics of the Micro-IGES tendon-driven robot are shown, along with a discussion of the results of the motion control strategy. Lastly, conclusions are drawn in Section V.

II. Tendon-driven system

In this Section an overview of the Micro-IGES surgical robotic tool is presented, describing its kinematic and dynamic models.

A. Micro-IGES Surgical Robotic Tool

The Micro-IGES [16] (Figure 2) is a surgical robotic tool, composed of a rigid shaft (27 cm) and a flexible section (39 mm in zero/home configuration). The shaft is responsible for the roll and translation Degrees of Freedom (DOFs). The articulated end, instead, consists of 2 elbows for pitch and yaw, with each elbow made of a pair of coupled joints, a 1 DOF revolute joint for the wrist pitch, and the jaws. The jaws provide two more DOFs: one for the wrist yaw and one for the gripper's opening/closing. Each joint of the articulated part is driven by an antagonistic pair of tendons, with each pair being connected to the corresponding driving capstan at the proximal drive unit. The coupling of the two pairs of joints of the elbows occurs at the driving unit: the two capstans that drive the two serial joints for each DOF of the elbow (pitch and yaw) are coupled by a series of gears with 1:2 ratio. Due to the current setup, in this work, the translation DOF cannot be used, therefore only 5 independent DOFs are considered (Roll, Elbow 1, Elbow 2, Wrist Pitch, Wrist Yaw). In order to control the Wrist Yaw DOF, with null gripping angle, the two jaws need to move equally (their motion is not independent).

B. Robot Kinematic and Dynamic Model

The nonlinearities in tendon transmission make the mathematical derivation of the system kinematics and dynamics tedious. In addition, tendon-driven systems for minimally invasive surgery usually lack sensors at the distal side, therefore joint values cannot be measured and controlled directly. Because of the generally nonlinear motor to joint (and joint to motor) mapping $\mathbf{q} = \mathbf{l}(\theta)$, being $\theta \in \mathbb{R}^{n_m}$ the vector of motor positions and $\mathbf{q} \in \mathbb{R}^{n_j}$ the vector of joint positions, the kinematic model of the robot can be rewritten as $\mathbf{P} = \mathbf{P}(\theta)$, with $\mathbf{P} \in \mathbb{R}^3$ being the Cartesian end-effector position. n_m and n_j are the number of motors and of joints of the system. The mapping from motor positions to end-effector Cartesian pose results to be highly nonlinear due to the hysteresis, backlash, tendon elongation effects caused by the tendon transmission [21].

The system dynamics, in absence of external interactions (i.e. no contact with the environment), can be rewritten as:

$$\tau_{\mathbf{mot}} = \mathbf{L}(\mathbf{q})^{\mathbf{T}} \tau_{\mathbf{j}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \Gamma_{\mathbf{d}}(\theta, \dot{\theta}, \ddot{\theta}) \ , \tag{1}$$

where $\tau_{\mathbf{mot}}$ is the vector of motor torques and $\tau_{\mathbf{j}} \in \mathbb{R}^{n_j}$ is the vector of joint torques, which is typically known for articulated robots [27]. The matrix $\mathbf{L}$, with $L_{i,j} = \frac{\partial q_i}{\partial \theta_j}$, is the motor to joint differential matrix [28]. $\Gamma_{\mathbf{d}}$ takes into consideration all the dynamic effects, including the nonlinearities in the tendon transmission. Because of these nonlinearities, the dynamic model of the robot mapping the motor values to the motor torque values is build by means of feedforward Artificial Neural Networks (ANN). Due to the small accelerations usually required in robotic surgery, in this work we assume that the motor torques only depend on the motor positions and velocities, meaning $\tau_{\mathbf{mot}} = \tau_{\mathbf{mot}}(\theta, \dot{\theta})$, and, therefore,

ANN are used to find the mapping $\begin{bmatrix}\theta & \dot{\theta}\end{bmatrix} \rightarrow \tilde{\Gamma}_{\mathbf{d}}$. It thus results that the control problem can be formulated as a function of the motor positions, velocities, and accelerations, which can be directly measured and controlled. In this work, the vector of controllable motor positions are defined by $\theta = \begin{bmatrix}\theta_R & \theta_{e1} & \theta_{e2} & \theta_W & \theta_{j1}\end{bmatrix}$, corresponding to the robot joints described in II-A.

III. Method

In this Section we describe the proposed approach for solving the optimal control problem.

A. Model Predictive Control

Model predictive control consists in formulating an optimal problem with a cost function to be minimized over a finite prediction horizon, with respect to the control inputs. Once a solution is found, only the first control action is executed and the procedure is repeated by shifting the horizon forward. The existence of the horizon allows to take into consideration the evolution of the desired cost function, of the constraints, of the system states.

Since the control interface of the Micro-IGES robot accepts desired positions, in this work we formulate the tracking problem in terms of a cost function minimizing the error between the desired Cartesian pose and the current one. Moreover, only the 3D robot Cartesian position is considered.

In order to ensure safety conditions, constraints on joint positions, velocities, accelerations, and torques can be imposed. Due to these limitations, the desired Cartesian position may not be achieved, leading to a deformation of the followed path. To reduce this issue, a scaling factor s can be introduced [15]. Considering a trajectory defined by a curvilinear abscissa $\xi(t)$ as $\mathbf{P_d}(\xi)$, starting from the current time $t = t_k$, the evolution of the desired position can be described by $\mathbf{P_d}(\mathbf{t_{k+1}}) = \mathbf{P_d}(\mathbf{t_k}) + \frac{\partial \mathbf{P_d}}{\partial \xi}\frac{\partial \xi}{\partial t}\Delta t = \mathbf{P_d}(\mathbf{t_k}) + s_k \Delta \mathbf{P_d}(\mathbf{t_k})$. The scaling factor $s_k = \frac{\partial \xi}{\partial t}$ allows to reduce the desired rate of change of the path (which equates to a dilation of the execution time execution), without modifying the path itself. As a matter of fact, at each time instant ξ is computed as $\xi_{k+1} = \xi_k + s_k \Delta t$, where Δt is the sampling time.

The optimal control problem is then formulated as:

$$\min_{\substack{\ddot{\theta}_0,\dots\ddot{\theta}_{\mathbf{N}-\mathbf{1}} \\ s_1\dots s_N \\ \sigma_1\dots\sigma_N}} \quad \frac{1}{2}\sum_{k=1}^{N} ||\mathbf{P_k}(\theta_\mathbf{k}) - \mathbf{P_{d,k}} - s_k \Delta \mathbf{P_{d,k}}||^2_{\mathbf{W_p}} \tag{2a}$$

$$+ W_s(s_{ref} - s_k)^2 + ||\tilde{\Gamma}_{\mathbf{d,k}}||^2_{\mathbf{W_t}} + ||\ddot{\theta}_{\mathbf{k-1}}||^2_{\mathbf{W_a}} + ||\sigma_\mathbf{k}||^2_{\mathbf{W}_\sigma}$$

$$\text{s.t.} \quad \begin{bmatrix}\dot{\theta}_\mathbf{k} \\ \ddot{\theta}_\mathbf{k}\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\theta_\mathbf{k} \\ \dot{\theta}_\mathbf{k}\end{bmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix}\ddot{\theta}_\mathbf{k} \tag{2b}$$

$$\theta_\mathbf{m} \leq \theta_\mathbf{k} \leq \theta_\mathbf{M} \tag{2c}$$

$$\dot{\theta}_\mathbf{m} \leq \dot{\theta}_\mathbf{k} \leq \dot{\theta}_\mathbf{M} \tag{2d}$$

$$\ddot{\theta}_\mathbf{m} \leq \ddot{\theta}_{\mathbf{k-1}} \leq \ddot{\theta}_\mathbf{M} \tag{2e}$$

$$\tau_{\mathbf{mot,m}} - \sigma_\mathbf{k} \leq \tilde{\Gamma}_{\mathbf{d,k}}(\theta_\mathbf{k}, \dot{\theta}_\mathbf{k}) \leq \tau_{\mathbf{mot,M}} + \sigma_\mathbf{k} \tag{2f}$$

$$0 \leq s_k \leq 1 \tag{2g}$$

$$\mathbf{0} \leq \sigma_\mathbf{k} \tag{2h}$$

where N is the number of timesteps in the prediction horizon, $\tilde{\Gamma}_\mathbf{d}$ is the vector of learned dynamic torques (as described in II-B), the subscripts m, M indicate lower and upper bounds, and $\mathbf{W_p}, W_s, \mathbf{W_t}, \mathbf{W_a}, \mathbf{W}_\sigma$ are the weights for each component of the cost function. Due to possible infeasibilities, some slack variables σ are also added in the torque constraints (2f). This allows the bounds to be relaxed, if necessary. s_{ref} is a user-defined reference scaling factor, which is generally set to 1. In order to solve the nonlinear MPC problem, the fast nonlinear MPC ACADO toolkit [29] has been used. Even though the robot states are $\theta = \begin{bmatrix}\theta_R & \theta_{e1} & \theta_{e2} & \theta_W & \theta_{j1}\end{bmatrix} \in \mathbb{R}^5$, the torque vector $\tilde{\Gamma}_\mathbf{d}$ is a six-dimensional-vector, since it includes also the torque for the second jaw.

B. Torque Linearization

The dynamic constraints in (2f) are highly nonlinear, as it is computed by means of an Artificial Neural Network (ANN) as described in II-B. In order to have a more robust model estimation, less affected by possible outliers in the data, the method proposed in [30] is employed in this work to learn the mapping from $\begin{bmatrix}\theta & \dot{\theta}\end{bmatrix} \rightarrow \tilde{\Gamma}_\mathbf{d}$.

Since ACADO toolkit is a symbolic solver, it would require adding this constraint in an analytical form. In principle, this would be possible since the network architecture (number of layers and nodes) and the activation functions are all known. However, for large ANNs, the computational effort due to the large number of operations ACADO needs to perform to obtain the network output and then to formulate the whole MPC in a symbolic form, becomes very high. Because of this, ACADO toolkit takes a very long time to generate the MPC code. Therefore, the following simplification is adopted in order to overcome this issue.

At each time instant $t = t_k$, the dynamic torques of the system can be linearized as:

$$\tilde{\Gamma}_{\mathbf{d,k}} \simeq \tilde{\Gamma}_{\mathbf{d,k-1}} + \mathbf{M_{p,k}}(\theta_\mathbf{k} - \theta_{\mathbf{k-1}}) + \mathbf{M_{v,k}}(\dot{\theta}_\mathbf{k} - \dot{\theta}_{\mathbf{k-1}}) \ , \tag{3}$$

where $\mathbf{M_{p,k}} = \frac{\partial \tilde{\Gamma}_\mathbf{d}}{\partial \theta}\Big|_{\mathbf{k-1}}$, $\mathbf{M_{v,k}} = \frac{\partial \tilde{\Gamma}_\mathbf{d}}{\partial \dot{\theta}}\Big|_{\mathbf{k-1}}$, for $k = 0...N$.

These matrices can be analytically computed by exploiting the cascade structure of a feedforward ANN. In practice The final derivative of the network output with respect to inputs can be calculated iteratively by applying the chain rule to the derivatives of each layer. For more details refer to Appendix I. Those values are functions of the motor states and control variables, and, as such, change over time. However, by expressing them as a function of the motor states and control at the instant precedent to the current prediction horizon ($k = 0$), they can be considered constant within the prediction horizon, resulting in $\tilde{\Gamma}_{\mathbf{d,k-1}} = \tilde{\Gamma}_{\mathbf{d,-1}} = \tilde{\Gamma}_\mathbf{d}(\theta_{-1}, \dot{\theta}_{-1})$, $\mathbf{M_{p,k}} = \mathbf{M_{p,0}} = \frac{\partial \tilde{\Gamma}_\mathbf{d}}{\partial \theta}\Big|_{\mathbf{-1}}$, $\mathbf{M_{v,k}} = \mathbf{M_{v,0}} = \frac{\partial \tilde{\Gamma}_\mathbf{d}}{\partial \dot{\theta}}\Big|_{\mathbf{-1}}$. θ_{-1} and $\dot{\theta}_{-1}$ represent, respectively, the current motors' positions and velocities.

The dynamic constraint can then be rewritten as

$\hat{\tau}_\mathbf{m} \leq \mathbf{M_{p,0}}\theta_\mathbf{k} + \mathbf{M_{v,0}}\dot{\theta}_\mathbf{k} \leq \hat{\tau}_\mathbf{M}$, with $\hat{\tau}_{\mathbf{m(M)}} = \tau_{\mathbf{mot,m(M)}} - \tilde{\Gamma}_{\mathbf{d,-1}} + \mathbf{M_{p,0}}\theta_{-1} + \mathbf{M_{v,0}}\dot{\theta}_{-1}$.

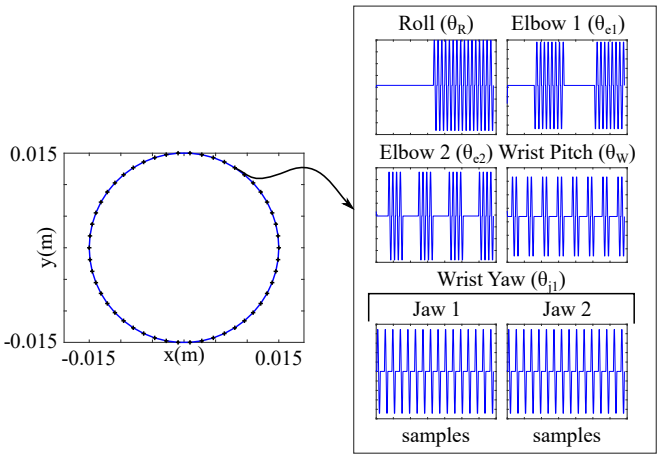


Fig. 3: Motor excitation for learning the dynamic model. The motor values are in *rad*.

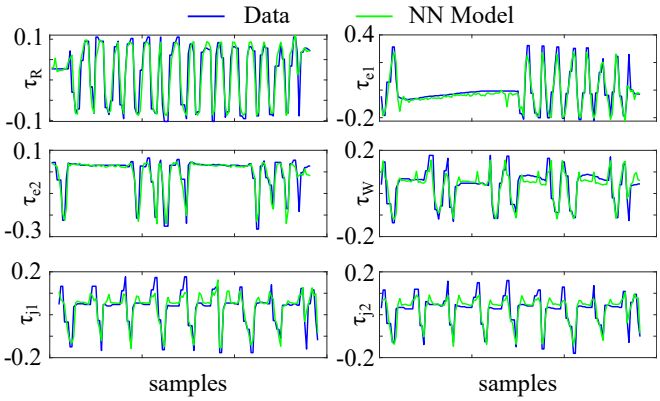


Fig. 4: Results on a subset of the test set for the computed dynamic model of the Micro-IGES. All values are in *mNm*.

IV. Experimental Settings and Results

In this Section we present the results for modelling the dynamics of the Micro-IGES robot along with the results for the control, both in a simulated environment and in real experiments.

A. Micro-IGES Dynamic Model

In order to build the dynamic model of the Micro-IGES $\tilde{\Gamma}_{\mathbf{d}}$, feedforward neural networks are used. To have better results we design 6 independent networks, one for each torque component. In this work, each network has the same structure consisting of 2 hidden layers with 10 and 30 neurons each. However, different structures for each network could have been used too. The input of each network is the 10-dimensional vector of motor positions and velocities $\begin{bmatrix}\theta & \dot{\theta}\end{bmatrix}$ and the output of each ANN is the torque value of each single motor. In order to limit the value of the network output, the activation function of each output layer has been set to be a sigmoid function. This allows to restrict the network estimated values within the motors maximum allowances of ± 4.3 *mNm*. This consideration prevents the ANN from predicting values that are clearly wrong due to a poor extrapolation.

Motor	RMSE		
i	Train	Test	Validation
1	0.028	0.028	0.029
2	0.063	0.065	0.064
3	0.037	0.037	0.039
4	0.040	0.041	0.040
5	0.049	0.048	0.049
6	0.039	0.039	0.038

TABLE I: RMSE (in *mNm*) between the learned torque models of each motor and the real collected data for the Micro-IGES robot on the different datasets.

For the data acquisition, the actual robot is commanded to move along a circular path of 15 *mm* in radius, controlling only the x, y Cartesian components (expressed with respect to the robot base frame) while keeping the z component constant. The path is discretized in 55 points. In order to collect a richer dataset, at each point each joint combination is then excited. This consists in exciting the joints with a sinusoidal motion with an amplitude of 5°. From the motor to joint mapping, the corresponding motor positions are computed. Each joint can have a state 0 (still) or 1 (moving). Consequently, for each Cartesian point, 2^5 joint combinations are obtained. In total 17442 data points have been collected. Figure 3 shows the commanded motor values for the data acquisition. To learn the dynamic model, the dataset is divided into a training (75%), validation (15%), and test (10%) sets. The Root Mean Squared Errors (RMSE) between the acquired data and the learned models are shown in Table I, whereas Figure 4 shows the results for the learned model on a subset of the training set.

B. Motion Control

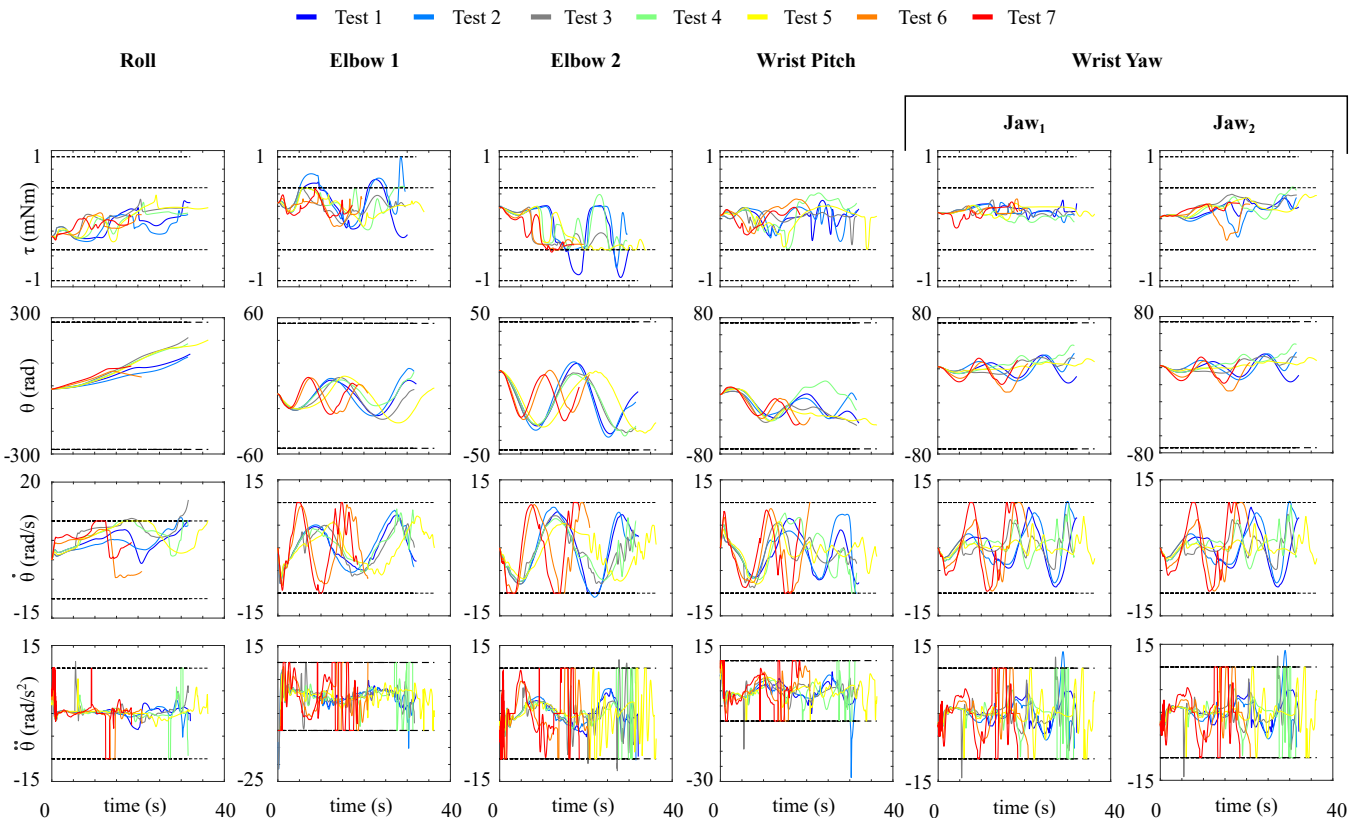
1) Simulation Results: The proposed control approach is based on the MPC formulation in (2) where the computed dynamic model of the Micro-IGES robot has been employed to impose the torque constraints while performing trajectory tracking, as it would be in case of tumor resection. VREP [31] simulator has been used for this purpose. The path to follow is a circle of 15*mm* in radius (same as the one used for the data acquisition), however, for the control also the z component is specified. The robot needs to track the circle while keeping z constant, with respect to the robot base frame. The motion task is thus specified as:

$$\begin{aligned} \mathbf{P_d} &= \begin{bmatrix} r\cos(\alpha) & r\sin(\alpha) & \tilde{z} \end{bmatrix}^T \\ \alpha &= \alpha_f \xi, \ \xi(t) = \frac{t}{T} \ , \end{aligned} \tag{4}$$

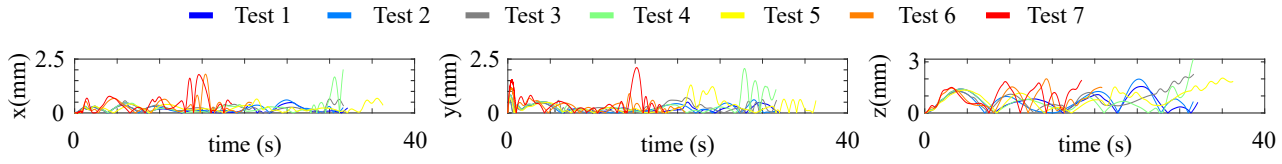
where $\tilde{z} = 0.0514$ m, $\alpha_f = 4\pi$ (the robot makes two loops), and T is the desired period of the motion. A first order polynomial is chosen for σ since the control problem is directly solved at the position level, and, consequently, no condition on the Cartesian velocity or acceleration are imposed. The sampling time has been set to 10*ms* and 10 steps are used for the prediction horizon of the MPC. The motor positions are bounded between $\pm\begin{bmatrix}280 & 55 & 47 & 74 & 74\end{bmatrix} rad$, whereas the velocity,

TABLE II: Different path tracking tests have been run in simulation to validate the framework. Here T, T_{act} are the desired and the actual motion times; $\tau_{M(m)}$, $\dot{\theta}_{M(m)}$, $\ddot{\theta}_{M(m)}$ are the motor torque, velocities, and acceleration limits (all set equal for each motor); $|\varepsilon_P|_{max}, |\tau|_{max}, |\varepsilon_{lin}|_{max}$ are the maximum absolute position error, maximum absolute torque, and maximum absolute torque linearization error; $\bar{s}$ is the average scaling factor.

Test	T (s)	T_{act} (s)	$\tau_{M(m)}$ (mNm)	$\dot{\theta}_{M(m)}$ (rad/s)	$\ddot{\theta}_{M(m)}$ (rad/s^2)	$\|\varepsilon_P\|_{max}$ (mm)			$\|\tau\|_{max}$ (mNm)	s_{ref}	$\bar{s}$	$\|\varepsilon_{lin}\|_{max}$ (mNm)
						x	y	z				
1	30	32.1	1 (-1)	10 (-10)	10 (-10)	0.5	0.8	1.6	0.949	1	0.934	0.084
2	30	31.6	1 (-1)	100 (-100)	100 (-100)	0.6	0.6	2.0	1.0004	1	0.949	0.038
3	30	31.6	0.5 (-0.5)	100 (-100)	100 (-100)	0.7	0.7	2.3	0.503	1	0.948	0.054
4	30	31.6	0.5 (-0.5)	10 (-10)	10 (-10)	2.0	2.1	3.2	0.530	1	0.949	0.024
5	30	36.3	0.5 (-0.5)	10 (-10)	10 (-10)	0.7	1.3	2.0	0.501	0.85	0.827	0.049
6	15	20.9	0.5 (-0.5)	10 (-10)	10 (-10)	1.8	1.2	2.0	0.507	0.85	0.718	0.124
7	15	18.5	0.5 (-0.5)	10 (-10)	10 (-10)	1.8	2.1	1.9	0.538	1	0.811	0.063



(a) Motor values



(b) Position errors

Fig. 5: Simulation results for the different tests. The dashed black lines are the imposed motor limits. Accelerations for Test 2 and 3 fall outside the shown bounds simply because their bounds are set to 100 rad/s^2 (see Table II).

acceleration, and torque bounds have been set to different values to assess the capabilities of the control framework. The bounds on the motor positions are a consequence of the motor to joint mapping. As a matter of fact, due to the routing of the tendons around the capstan, the motors may need to complete more than one full turn in order for the joints to reach their limits.

To validate the framework, different tests have been run. The weights in the MPC cost function have been set to $\mathbf{W_p} = diag([10^8 \quad 10^8 \quad 10^8])$, $W_s = 10$, $\mathbf{W_t} = diag([10^2 \quad 10^2 \quad 10^2 \quad 10^2 \quad 10^2 \quad 10^2])$,

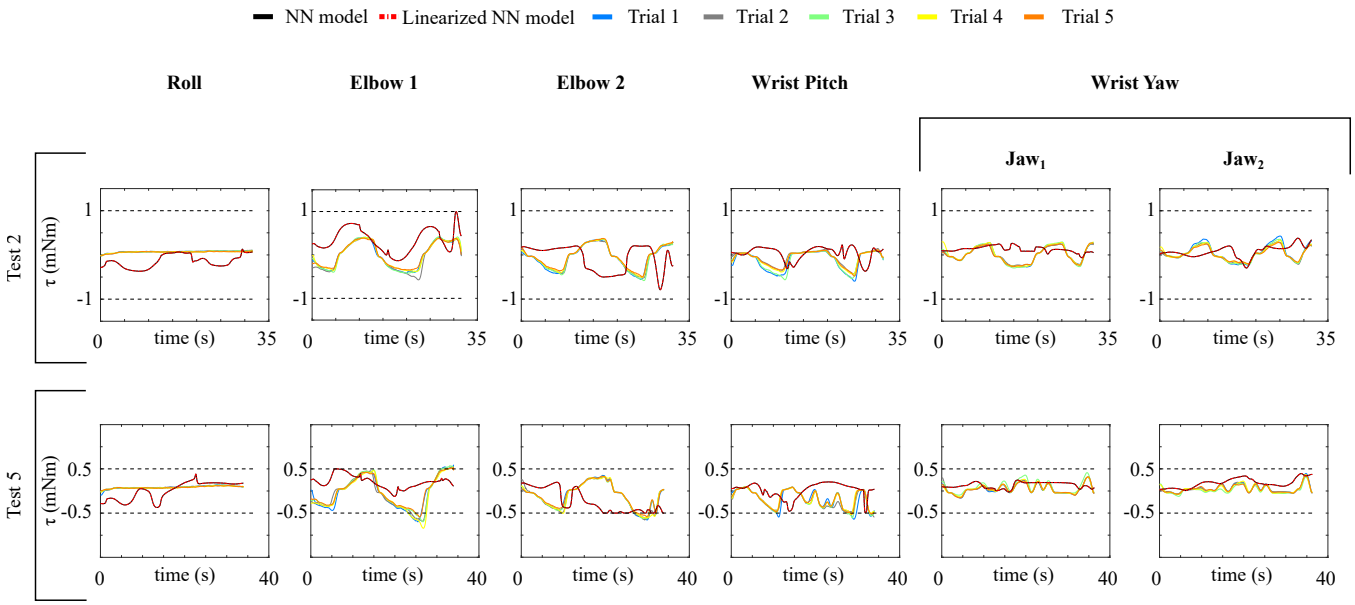


Fig. 6: Torque values for the experiments on the real system for both Test 2 and Test 5 as described in (Table II). The dashed black lines are the imposed torque bounds.

$\mathbf{W_a} = diag([10^{-2} \quad 10^{-2} \quad 10^{-2} \quad 10^{-2} \quad 10^{-2}])$, $\mathbf{W}_\sigma = diag([10^6 \quad 10^6 \quad 10^6 \quad 10^6 \quad 10^6 \quad 10^6])$ and they have not been changed in each test. Table II reports the different setups and the results for each test. Figure 5 shows the results both for the motor values (Figure 5a) and the Cartesian position errors (Figure 5b) for each test.

Due to the scaling factor, each motion lasts longer than the specified period T. This is the cost to pay in order to satisfy at best the imposed constraints. For each test the motor positions, velocities, and accelerations are always within the imposed bounds. The bounds on the torque, instead, are sometimes relaxed due to the presence of the slack variables as discussed in Section III-A. Smaller scaling factors, however, achieved by means of smaller $\tilde{\mathbf{s}}$, allow to reduce the bound relaxations.

Regarding the tracking accuracy, good results are obtained in each test. The errors are generally small (in the order of few millimeters or even less). Larger errors occur in the Test 4. This is because the system undergoes simultaneous saturations of multiple joints and the optimal scaling factor is not enough to allow the system to reduce the motor excitations. Reducing the scaling factor by setting $\tilde{s}$ to 0.85 allows to have better performances. Also having larger acceleration bounds may help, since it reduces the risk of motor saturation. Achieving large accelerations, however, has the disadvantage of increasing the torque linearization errors. Nonetheless, in the tests run the linearized torque is always close to the expected ANN output. Higher accuracy can be achieved by increasing the weights in $\mathbf{W_p}$. Yet, this may increase the bounds violations, especially when tight bounds are imposed. Optimal weight tuning, such as the recent work by Mehndiratta et al. [32], may be useful to solve this issue, even though it is a challenging task which requires more in-depth studies.

2) Experimental Results: The proposed control scheme has also been tested on the real system. For the real world examples, two exemplary tests were run: Test 2 and Test 5. These two tests were chosen because they allowed to examine the control approach under different bound conditions. The robot was commanded with the motor values computed through the MPC control strategy resulting from Test 2 and Test 5 from Table II. Each test was repeated 5 times. Due to inaccuracies in the kinematic model the correct execution of the desired trajectory is not guaranteed, and, therefore, results are not reported. However, improving the kinematic model was out of the scope of this work, where we were only focusing on assessing the capabilities of the framework to satisfy the imposed constraints. Another work is being carried out in order to improve the kinematic accuracy of the robot through online model adaptation.

Table III and Figure 6 show the results for the tests considered. The motor positions, velocities, and accelerations are guaranteed to be within the bounds, being the same as in the simulation case. In both cases, the torque linearization proved effective, with the linearized torque models being almost coincident with the actual ANN models. Moreover, for the Test 2, the learned ANN models behave quite well, being close to the measured values for almost all motors (the Elbow 2 motor is the one where the model performances are the worst) with the maximum RMSE between the ANN torques and the actual measured torques being $\begin{bmatrix}0.256 & 0.452 & 0.505 & 0.330 & 0.230 & 0.212\end{bmatrix}$ mNm. In addition, also the real motor torques reside within the imposed bounds.

With regards to Test 5, instead, the maximum RMSE are $\begin{bmatrix}0.195 & 0.435 & 0.408 & 0.363 & 0.143 & 0.185\end{bmatrix}$ mNm. Even though the ANN models are inside the bounds, due to the

	$\lvert\tau\rvert_{\max}$ (mNm)					
	$\lvert\tau_1\rvert_{\max}$	$\lvert\tau_2\rvert_{\max}$	$\lvert\tau_3\rvert_{\max}$	$\lvert\tau_4\rvert_{\max}$	$\lvert\tau_5\rvert_{\max}$	$\lvert\tau_6\rvert_{\max}$
Test 2	0.108	0.574	0.581	0.594	0.296	0.427
Test 5	0.141	0.844	0.658	0.637	0.414	0.397

TABLE III: Maximum absolute values of the motor torques for each experimental test.

model inaccuracies, the actual torques slightly violate the constraints, with the maximum absolute torque values shown in Table III. This means that more accurate models are still needed. Yet, if the ANN models are accurate enough, then constraints satisfaction can be guaranteed.

V. Conclusions

In conclusions, the proposed framework allows to easily impose the learned dynamic model as a constraint in an MPC formulation for robot trajectory tracking. Simulation results showed that the framework proved successful in allowing the robot to accurately follow a desired trajectory while satisfying the imposed bounds both on the kinematics and dynamics, also under different bound ranges. This is important in application scenarios like minimally invasive surgery, where high motion accuracy and safety must be guaranteed. However, one of the limitations resides in the inaccuracy of the learned model. Because of that, constraint satisfaction may not be guaranteed on the real system if the learned model is not accurate enough. Online adaptive learning techniques will be used in future work to solve this issue, along with a more probabilistic approach to estimate the model uncertainty. Moreover, in this work the weights in the MPC cost function have been set by the authors heuristically. However, optimal weight tuning may improve the efficacy of the proposed approach, improving accuracy and bound satisfaction. This, however, is a challenging task which will also be addressed in future work. Finally, in this work the path to follow was supposed to be fixed. However, the framework can be generalized also to time changing paths, as long as some future estimates are available, for instance by modelling the tissue deformation as in [17].

Appendix I
Artificial Neural Network Computation

Artificial Neural Networks (ANN) are able to approximate any suitably smooth function, given enough hidden layers [33]. Feedforward networks consist of different layers of neurons. The first layer is the input layer, the last one is the output layer, and all the others in between are called hidden layers. Each layer has several neurons, each one receiving inputs from the neurons of the previous layer and sending an output to the neurons of the following layer. Given a dataset of input points $\mathbf{x} \in \mathbb{R}^{n_{in}}$ and output points $\mathbf{y} \in \mathbb{R}^{n_{out}}$, and a network with one input, one hidden, and one output layer, n_{in} inputs, M nodes in the hidden layer, and n_{out} outputs, then approximated mapping $\mathbf{y} \sim \mathbf{f}(\mathbf{x})$ for each output of the network is computed as:

$$y_k(\mathbf{x},\mathbf{w}) = \tilde{h}\Big(\sum_{i=1}^{M} w_{k,j}^{(2)} h\Big(\sum_{i=1}^{n_{in}} w_{i,j}^{(1)} x_i + w_{j,0}^{(1)}\Big) + w_{k,0}^{(2)}\Big) \ , \quad (5)$$

where $\tilde{h}, h$ are the activation functions, and $\mathbf{w}$ are the network weights.

Due to the parametric nature of ANN, it is easily possible to compute the derivatives of the network output with respect to the network weights through back-propagation. Additionally, it is also possible to compute in a similar way the derivatives of the output with respect to the inputs. For each network layer, with input $\mathbf{x_i}$ and output $\mathbf{y_i}$, the derivative of the output with respect to the input can be computed as:

$$\frac{\partial \mathbf{y_i}}{\partial \mathbf{x_i}} = \frac{\partial \mathbf{y_i}}{\partial \mathbf{h_i}} \frac{\partial \mathbf{h_i}}{\partial \mathbf{z_i}} \frac{\partial \mathbf{z_i}}{\partial \mathbf{x_i}} \ , \quad (6)$$

where $\mathbf{z_i} = \mathbf{W_i x_i} + \mathbf{W_{i,0}}$, with $\mathbf{W_i}$ being the matrix of weights of the layer and $\mathbf{W_{i,0}}$ the biases, and $\mathbf{h_i}$ the activation function. The first two partial derivatives can be easily computed analytically once the activation function is chosen (for instance sigmoid), and $\frac{\partial \mathbf{z_i}}{\partial \mathbf{x_i}} = \mathbf{W_i}$. Based on the structure of the neural network as a cascade of layers, the final derivative of the network output with respect to the network inputs can be calculated iteratively by applying the chain rule to the derivatives of each layer.

References

[1] J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," IEEE Transactions on Robotics and Automation, vol. 12, no. 3, pp. 453–465, 6 1996.

[2] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Viability and Feasibility of Constrained Kinematic Control of Manipulators," Robotics, vol. 7, no. 3, p. 41, 7 2018.

[3] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 7, no. 12, pp. 868–871, 1977. [Online]. Available: http://ieeexplore.ieee.org/document/4309644/

[4] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in 5th IEEE-RAS International Conference on Humanoid Robots, 2005. IEEE, pp. 238–244. [Online]. Available: http://ieeexplore.ieee.org/document/1573574/

[5] A. Atawnih, D. Papageorgiou, and Z. Doulgeri, "Kinematic control of redundant robots with guaranteed joint limit avoidance," Robotics and Autonomous Systems, vol. 79, pp. 122–131, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2016.01.006

[6] F. Flacco and A. D. Luca, "Motion control of redundant robots under joint constraints: Saturation in the null space," in IEEE International Conference on Robotics and Automation, 2012, pp. 285–292.

[7] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in Proceedings. 1985 IEEE International Conference on Robotics and Automation, vol. 2. Institute of Electrical and Electronics Engineers, pp. 722–728. [Online]. Available: http://ieeexplore.ieee.org/document/1087234/

[8] M. Charbonneau, F. Nori, and D. Pucci, "On-line joint limit avoidance for torque controlled robots by joint space parametrization," IEEE-RAS International Conference on Humanoid Robots, pp. 899–904, 2016.

[9] D. A. Drexler and I. Harmati, "Joint Constrained Differential Inverse Kinematics Algorithm for Serial Manipulators," Periodica Polytechnica Electrical Engineering, vol. 56, no. 4, p. 95, 2012. [Online]. Available: https://pp.bme.hu/eecs/article/view/7163

[10] V. Mohan, P. Morasso, G. Metta, and G. Sandini, "A biomimetic, force-field based computational model for motion planning and bimanual coordination in humanoid robots," Autonomous Robots, vol. 27, no. 3, pp. 291–307, 2009.

[11] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained robot control using control barrier functions," IEEE International Conference on Intelligent Robots and Systems, vol. 2016-Novem, pp. 279–285, 2016.
[12] F. Flacco, A. De Luca, and O. Khatib, "Control of Redundant Robots under Hard Joint Constraints: Saturation in the Null Space," IEEE Transactions on Robotics, vol. 31, no. 3, pp. 637–654, 2015.
[13] V. Modugno, U. Chervet, G. Oriolo, and S. Ivaldi, "Learning soft task priorities for safe control of humanoid robots with constrained stochastic optimization," in 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). IEEE, 11 2016, pp. 101–108.
[14] A. Escande, N. Mansard, and P.-b. Wieber, "Hierarchical quadratic programming : Fast online humanoid-robot motion generation," International Journal of Robotics Research, vol. 33, no. 7, pp. 1006–1028, 2014.
[15] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, "Predictive Inverse Kinematics for Redundant Manipulators With Task Scaling and Kinematic Constraints," IEEE Transactions on Robotics, vol. 35, no. 1, pp. 278–285, 2 2019.
[16] J. Shang, K. Leibrandt, P. Giataganas, V. Vitiello, C. A. Seneci, P. Wisanuvej, J. Liu, G. Gras, J. Clark, A. Darzi, and G.-Z. Yang, "A Single-Port Robotic System for Transanal MicrosurgeryDesign and Validation," IEEE Robotics and Automation Letters, vol. 2, no. 3, pp. 1510–1517, 2017.
[17] F. Zhong, Y. Wang, Z. Wang, and Y. H. Liu, "Dual-Arm Robotic Needle Insertion with Active Tissue Deformation for Autonomous Suturing," IEEE Robotics and Automation Letters, vol. 4, no. 3, pp. 2669–2676, 7 2019.
[18] H. M. Le, T. N. Do, and S. J. Phee, "A survey on actuators-driven surgical robots," Sensors and Actuators, A: Physical, 2016.
[19] M. Kaneko, M. Wads, H. Maekawaw, and K. Tanie, "A New Consideration on Tendon-Tension Control System of Robot Hands," 1991.
[20] T. N. Do, T. Tjahjowidodo, M. W. Lau, and S. J. Phee, "Nonlinear friction modelling and compensation control of hysteresis phenomena for a pair of tendon-sheath actuated surgical robots," Mechanical Systems and Signal Processing, 2015.
[21] ——, "An investigation of friction-based tendon sheath model appropriate for control purposes," Mechanical Systems and Signal Processing, 2013.
[22] T. N. Do, T. Tjahjowidodo, M. W. S. Lau, and S. J. Phee, "Performance Control of Tendon-Driven Endoscopic Surgical Robots With Friction and Hysteresis," 2017.
[23] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control Strategies for Soft Robotic Manipulators: A Survey."
[24] X. Li, L. Cao, A. M. H. Tiong, P. T. Phan, and S. J. Phee, "Distal-end force prediction of tendon-sheath mechanisms for flexible endoscopic surgical robots using deep learning," Mechanism and Machine Theory, vol. 134, pp. 323–337, 4 2019.
[25] W. Xu, J. Chen, H. Y. Lau, and H. Ren, "Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators," The International Journal of Medical Robotics and Computer Assisted Surgery, vol. 13, no. 3, p. e1774, 9 2017.
[26] K. Leibrandt, P. Wisanuvej, G. Gras, J. Shang, C. A. Seneci, P. Giataganas, V. Vitiello, A. Darzi, and G.-Z. Yang, "Effective Manipulation in Confined Spaces of Highly Articulated Robotic Instruments for Single Access Surgery," Icra 2017, vol. 2, no. 3, pp. 1–1, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7855734/
[27] B. Siciliano and O. Khatib, "Introduction," in Springer Handbook of Robotics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–4. [Online]. Available: http://link.springer.com/10.1007/978-3-540-30301-5_1
[28] R. M. Murray, Z. Li, and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation, 1994. [Online]. Available: http://www.cds.caltech.edu/http://www.crcpress.com/product/isbn/9780849379819.
[29] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit-An open-source framework for automatic control and dynamic optimization," Optimal Control Applications and Methods, vol. 32, no. 3, pp. 298–312, 5 2011.
[30] F. Cursi and G.-Z. Yang, "A Novel Approach for Outlier Detection and Robust Sensory Data Model Learning," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 11 2019, pp. 4250–4257.
[31] "Coppelia Robotics V-REP: Create. Compose. Simulate. Any Robot." [Online]. Available: http://www.coppeliarobotics.com/
[32] M. Mehndiratta, E. Camci, and E. Kayacan, "Automated Tuning of Nonlinear Model Predictive Controller by Reinforcement Learning," in IEEE International Conference on Intelligent Robots and Systems. Institute of Electrical and Electronics Engineers Inc., 12 2018, pp. 3016–3021.
[33] K. Hornik, "Approximation capabilities of multilayer feedforward networks," Neural Networks, vol. 4, no. 2, pp. 251–257, 1 1991.