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

Improving Motion Planning for Surgical Robot with Active Constraints

Hang Su[1], Yingbai Hu[2*], Jiehao Li[1], Jing Guo[3], Yuan Liu[3], Mengyao Li[4]
Alois Knoll[2], Giancarlo Ferrigno[1] and Elena De Momi[1]

***Abstract*— In this paper, an improved motion planning scheme is proposed for surgical robot control with multiple active constraints, including joint constraints, joint velocity constraints and remote center of motion constraints. It introduces an improved recurrent neural network (RNN) to optimize the online motion planning respect to multiple constraints. The demonstrated surgical operation trajectory is derived using teaching by demonstration. An improved motion planning scheme using the novel recurrent neural network is then designed to achieve the accurate task tracking under the multiple constraints. The general quadratic performance index is adopted to represent the constraints. Finally, the effectiveness of the proposed algorithm is demonstrated using KUKA LWR4+ robot in a lab setup environment.**

I. INTRODUCTION

Over the past few years, conventional serial robots with redundant manipulators have been successfully applied and additionally developed in precise automation processes for a variety of applications [1], [2]. For the medical scenario applications, especially in minimally invasive surgery (MIS), it has drawn progressed research interest due to the improvement of the accuracy and their lower cost concerning specialized surgical robots. For example, a novel robotic implementation in the control and precision of the surgical tool was introduced to diminish the trauma of the patients [3].

As we knew, there is a small incision in the abdominal wall to support the insertion of a surgical tool in surgical procedures. The small incision presents a constraint on the inserted robot end-effector, which is named as the Remote Center of Motion (RCM) constraint [4]. Although a mechanical implementation is usually ensured safer with the complex structures and calibration procedures, a programmable RCM limiting the action by the control algorithm is more reasonable and more flexible and is, therefore, a preferable solvsion [5]. Therefore, how to maintain the RCM constraint during surgical operations using a serial robot becomes a challenging problem.

During the surgical operation, it is inevitable to consider the physical interaction between the trocar and the abdominal wall [5]. Accurate position tracking is the main task due to uncertain disturbances involved in the physical interaction, which is of vital importance to assure the safety of the surgical operation. Thus, surgical robots are required to learn and adapt the interaction according to the complex environment to comply with the high requirements associated with accuracy [5], [6]. Furthermore, when it comes to the safety issue associated with robot manipulation, the RCM constraint should be maintained. There are several related works to address this problem in recent years. For example, in [7], a kinematic formalization was discussed to solve the RCM constraint in the joint space.

It is generally to address this problem using adaptive control theory; however, the current performances are still unsatisfactory combined with the active RCM constraints. Moreover, to determine the RCM constraints applying the efforts and prevalence of adaptive control algorithms is still basically insufficient in the initial stage [5]. In the past few years, adaptive control methods have attracted considerable attention to connect the neural networks [8]. For example, the Nussbaum function is utilized to compensate for the nonlinear terms of kinematics in [9]. In order to comply with the uncertainties due to the kinematic constraints, an adaptive neural impedance control approach is introduced in [10] for a n-link robotic manipulator.

In our previous works [11], [12], the neural network based optimization control was presented for convex optimization problem which achieves good performance with multi-constraints. In this paper, to enhance the stability of the end effector and to deal with the active RCM constraint and other constraints at the same time, a neural-learning enhanced control scheme based on the novel recurrent neural network (RNN) approximation is presented, where the learning methods is explored to encode the motion [13], and then neural network optimize the tasks. The main contributions of this article are concluded as follows:

1) The surgical operation trajectory is derived from learning by demonstration in the Cartesian space.
2) An improved RNN-based adaptive controller is introduced to manage the uncertainties due to the RCM constraints.
3) Experimental demonstration using KUKA LWR4+ is performed to evaluate the feasibility of the improved RNN controller.

*This work was supported in part by the European Commission Horizon 2020 research and innovation program, under the project SMARTsurg, grant agreement No. 732515 and in part by the Human Brain Project SGA2, under the Specific Grant Agreement No. 785907. Yingbai Hu is corresponding author.

[1]Hang Su, Giancarlo Ferrigno and Elena De Momi are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133, Milan, Italy. hang.su, giancarlo.ferrigno, elena.demomi@polimi.it

[2]Yingbai Hu and Alois Knoll are with the Department of Informatics, Technical University of Munich, 85748, Munich, Germany. yingbai.hu@tum.de

[3]Jing Guo and Yuan Liu are with the School of Automation, Guangdong University of Technology, 20133, Guangzhou, China. toguojing@gmail.com; eeliuyuan@gdut.edu.cn

[4]Mengyao Li is with Shenzhen Institutes of Advanced Technology, Chinese Academy Sciences, 518055, Shenzhen, China. my.li@siat.ac.cn

II. Learning motor skills from human demonstrations

In this section, the surgeon hold the robot to remove the tumor by demonstrations, and then all the dataset are collected.

A. Dynamic Movement Primitive with GMM

The DMP is represented as a set of equations [14], and it can model different linear or nonlinear motions which are expressed as,

$$\begin{aligned} &\ddot{X}_t = K_p\left(X_g - X_t\right) - K_v\dot{X}_t + F\left(s_t\right) \\ &\dot{s}_t = \alpha_s s_t \\ &F\left(s_t\right) = h_t^{\mathrm{T}}\left(s_t\right)\omega\left(g - X_0\right) \end{aligned} \tag{1}$$

subjected to

$$h_t\left(s_t\right) = \frac{\sum_{i=1}^{N}\psi_i\left(s_t\right)s_t}{\sum_{i=1}^{N}\psi_i\left(s_t\right)_t}, \psi_i\left(s_t\right) = \exp\left(-\frac{1}{2\sigma_i}\left(s_t - c_i\right)^2\right) \tag{2}$$

where $\left[X_t, \dot{X}_t, \ddot{X}_t\right]$ is the Cartesian space trajectory; X_0 and g present the initial position and goal position, respectively; K_p and K_d are the stiffness matrix, damping term of DMP. ω is the shape parameter; α_s is the scale parameter of Canonical system, where s_t asymptotically decays from 1 to $0; \sigma_i$ and c_i andwidth and center of the i -th Gaussian kernels.

In general, the complex movements are linear combinations of simple movement primitive with weighted which is defined as,

$$\ddot{X} = \sum_{k=1}^{K} h_k\left(K_k^p\left(X_g - X\right) - K_k^v\dot{X} + F\right) \tag{3}$$

where h_k denotes the weighted of different components. The Cartesian space data point from demonstrations are defined as: $\zeta_k = \{s_k, X_k\}\,(k = 1, \ldots, M)$, where M is the number of demonstrations. Each datapoint include the time temporal value s_k and position value X_k.

The new dataset of nonlinear item F in DMP framework of multi-demonstrations is defined as: $\{s_k, F_k\}$. To encode the movement of dataset $\{s_k, F_k\}$, the Gaussian Mixture Model (GMM) is proposed to train the trajectories from demonstration. Then the dataset is modeled by K component Gaussian model as,

$$\begin{bmatrix} s \\ F \end{bmatrix} \sim \sum_{k=1}^{K} h_k\mathcal{N}\left(\lambda_k, \Sigma_k\right) \tag{4}$$

$$\lambda_k = \begin{bmatrix} \lambda_k^s \\ \lambda_k^F \end{bmatrix}, \Sigma_k = \begin{bmatrix} \Sigma_k^{ss} & \Sigma_k^{sF} \\ \Sigma_k^{Fs} & \Sigma_k^{FF} \end{bmatrix} \tag{5}$$

where K denotes the Gaussian model; λ_k and Σ_k are mean and covariance matrix of k component GMM.

Then the Gaussian Mixture Regression (GMR) is proposed to reconstruct the general form for the dataset [15], [16]. The estimation of condition expectation $\hat{\lambda}^s$ and covariance matrix $\hat{\Sigma}^{ss}$ are concluded as,

$$\hat{\lambda}^s = \sum_{k=1}^{K} h_k\hat{\lambda}_k^s, \quad \hat{\Sigma}^{ss} = \sum_{k=1}^{K} h_k\hat{\Sigma}_k^{ss} \tag{6}$$

$$h_k\left(s\right) = \frac{\eta_k\mathcal{N}\left(\hat{\lambda}_k^s\left(s\right), \hat{\Sigma}_k^{ss}\left(s\right)\right)}{\sum_{k=1}^{K}\eta_k\left(s\right)\mathcal{N}\left(s|\hat{\lambda}_k^s\left(s\right), \hat{\Sigma}_k^{ss}\left(s\right)\right)} \tag{7}$$

Therefore, the motion $\left\{s, \hat{\lambda}^F\right\}$ can be generated by estimating $\left\{\hat{\lambda}^F, \hat{\Sigma}^{FF}\right\}$ at time step s.

B. Kinematics Model

The model of a redundant manipulator is formulated as below:

$$J\dot{q} = \dot{r}_d \tag{8}$$

where $J \in \mathbb{R}^{m\times n}$ is the Jacobian matrix. Then we have

$$q^- \le q \le q^+ \tag{9}$$

$$\dot{q}^- \le \dot{q} \le \dot{q}^+ \tag{10}$$

where q^-, q^+, $\dot{q}^-$ and $\dot{q}^+$ represent corresponding bounds of the joint angle and the joint velocity vector, respectively.

Then, we convert the position limit to the velocity limit, which is consistent with optimization objective function as below:

$$\lambda\left(q^- - q\right) \le \dot{q} \le \lambda\left(q^+ - q\right) \tag{11}$$

where $\lambda > 0$ is the constant coefficient. Therefore, by combining (10) and (11), the joint limits can be reformulated as below:

$$\begin{aligned} &\pi^- \le \dot{q} \le \pi^+, \dot{q} \in \Omega \\ &\pi_i^- = \max\left\{\dot{q}_i^-, \lambda\left(q_i^- - q_i\right)\right\} \\ &\pi_i^+ = \min\left\{\dot{q}_i^+, \lambda\left(q_i^+ - q_i\right)\right\} \end{aligned} \tag{12}$$

It should be noted that the joint trajectory q is difficult to be obtained by the inverse kinematics provided in (8), due to the high computational complexity and the infinite number of solutions. To address this issue, we reformulate the problem as a novel optimization problem associated with the redundant degree of freedom. Therefore, the first optimization problem associated with task tracking can be defined as follows:

$$\min \quad \frac{1}{2}\dot{q}^T W\dot{q} \tag{13}$$

$$\text{s.t.} \quad J\left(q\right)\dot{q} = \dot{r}_d \tag{14}$$

$$\pi^- \le \dot{q} \le \ \pi^+ \tag{15}$$

where $\dot{r}_d$ is the desired velocity of a surgical task; $W = I$.

C. Remote Center of Motion

During a surgical tracking task, the surgical tooltip of the robot needs to pass through the RCM. Fig.e 1 represents the assumption that the tool should always be inserted into the patient's body at the point r_{rcm}, without affecting the main surgical task.

According to the coordinate $r_{n-1} \in \mathbb{R}^m$, the end-effector position $r_n \in \mathbb{R}^m$ can be addressed as below:

$$r_{n-1} = f_{n-1}(q), \ r_n = f_n(q) \tag{16}$$

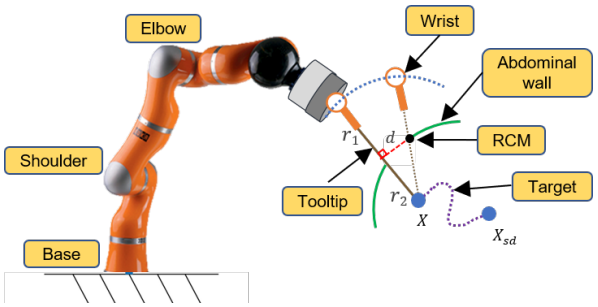

Fig. 1: Remote center of motion: a robot tool passes through a small incision r_{rcm} on the obstacle surface. During the robot manipulation, the tool-tip position needs to follow the desired reference trajectory, while the tool shaft should respect the kinematic constraint of the RCM.

To comply with the RCM constraint, r_{rcm} should be always on the straight line between r_{n-1} and r_n, where r_n is the position of the tooltip and r_{n-1} is the joint position of holding the tool. In an actual surgical operation, we seel to keep the error of RCM equal to zero. The vectors of line 1 and line 2 are defined as follows: $\overrightarrow{r_{n-1}r_n} = r_n - r_{n-1}$, $\overrightarrow{r_{n-1}r_{rcm}} = r_{rcm} - r_{n-1}$, respectively.

According to the geometric relationship, the relation is derived as below:

$$\overrightarrow{r_{n-1}r_{rcm}} \times \overrightarrow{r_{n-1}r_n} = 0 \tag{17}$$

From the relationship between the RCM error e_{rcm} and the vector projection, the error of RCM can be further represented as follows:

$$e_{rcm} = \frac{\overrightarrow{r_1 r_{rcm}} \times \overrightarrow{r_{n-1}r_n}}{L} \tag{18}$$

where $L = \|r_n - r_{n-1}\|$ is the length of the last link.

The time derivative of the RCM error model in (18) is reformulated as below:

$$J_{rcm}\dot{q} = \dot{e}_{rcm} \tag{19}$$

where $J_{rcm} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix corresponding to the RCM error model.

With regard to the RCM constraint task, we seek to maintain the distance of the RCM error e_{rcm} at the minimum value. Therefore, the second optimization problem with the RCM constraints are defined as,

$$\min \quad \frac{1}{2}\dot{q}^T W \dot{q} \tag{20}$$

$$\text{s.t.} \quad J_{rcm}(q)\,\dot{q} = 0 \tag{21}$$

D. Problem Reformulation in Terms of Quadratic Programming

In this subsection, the kinematic control of serial manipulators considering forwarding kinematics and focusing on the RCM (21) and joint velocity level constraints (14) is considered. The end task and RCM constraints should be considered simultaneously. We expected to find the optimization solution for the velocity constraint. Then, the coordinate tooltip tracking error asymptotically converges to zero and the RCM deviation error remains constrained within a predefined area.

Considering the RCM and end task constraints and end tasks constraints simultaneously, the new multi-tasks optimization problem based on (13)–(15) and (20)–(21) is defined as

$$\begin{aligned} \min \quad & \frac{1}{2}\dot{q}^T W \dot{q} \\ \text{s.t.} \quad & J\dot{q} = v_d \\ & J_{rcm}\dot{q} = 0 \\ & \pi^- \le \dot{q} \le \pi^+ \end{aligned} \tag{22}$$

where $v_d = \dot{r}_{nd}$.

The joint angle drift can occur due to the loss of explicit information on r_n and e_{rcm}. Therefore, we design the feedback controller to restrict the movement of the robot in terms of the end effector and RCM velocity constraint in (26) as follows:

$$J\dot{q} = -k_1\left(f_n(q) - r_{nd}\right) + \dot{r}_{nd} \tag{23}$$

$$J_{rcm}\dot{q} = -k_2\left(r_{rcm}\right) \tag{24}$$

where $v_d = -k_1\left(f_n(q) - r_{nd}\right) + \dot{r}_{nd}$, $v_{rcm} = -k_2\left(r_{rcm}\right)$. The optimization problem in (22) is rewritten as,

$$\begin{aligned} \min \quad & \frac{1}{2}\dot{q}^T W \dot{q} \\ \text{s.t.} \quad & J\dot{q} = v_d \\ & J_{rcm}\dot{q} = v_{rcm} \\ & \pi^- \le \dot{q} \le \pi^+ \end{aligned} \tag{25}$$

It should be noted that the multi-tasks have different priorities, which are scaled by corresponding weights. Finally, the multi-tasks optimization scheme defined in (22) can be reformulated as below:

$$\begin{aligned} \min \quad & \frac{\eta_0}{2}\dot{q}^T\dot{q} + \frac{\eta_1}{2}\|J\dot{q} - v_d\|^2 + \frac{\eta_2}{2}\|J_{rcm}\dot{q} - v_{rcm}\|^2 \\ \text{s.t.} \quad & J\dot{q} = v_d \\ & J_{rcm}\dot{q} = v_{rcm} \\ & \pi^- \le \dot{q} \le \pi^+ \end{aligned} \tag{26}$$

where $\eta_0 > 0$, $\eta_1 > 0$ and $\eta_2 > 0$ are the constants used to balance the different priorities of multi-tasks.

III. Neural Network Design and Stability Analysis

In this section, the novel RNN is applied to solve the multi-tasks optimization problem according to the RCM constraints defined in (26). We first transfer the quadratic programming problem formulated in (26) to the equivalent relationship problem, and then design the novel RNN to solve it. The control framework of the developed RNN scheme is described in Fig. 2.

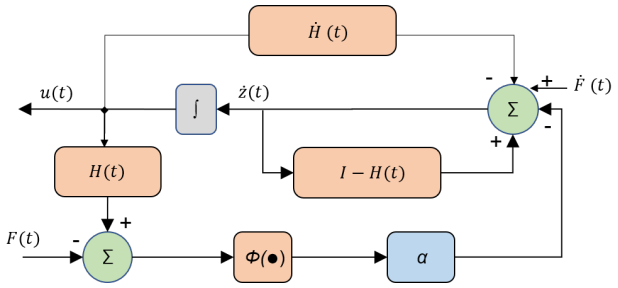

Fig. 2: The block diagram of novel recurrent neural network model.

A. novel recurrent neural network for the Quadratic Programming Problem problem

To obtain the equivalent relationship problem from (26), the Lagrange function is constructed as follows:

$$\mathcal{L}(\dot{q}, \lambda_1, \lambda_2) = \frac{\eta_1}{2}\|J\dot{q} - v_d\|^2 + \frac{\eta_2}{2}\|J_{rcm}\dot{q} - v_{rcm}\|^2 \frac{\eta_0}{2}\dot{q}^T\dot{q} + \lambda_1^T(v_d - J\dot{q}) + \lambda_2^T(v_{rcm} - J_{rcm}\dot{q}) \quad (27)$$

where $\lambda_1 \in \mathbb{R}^m$ and $\lambda_2 \in \mathbb{R}^m$. The gradient of $\mathcal{L}$ is defined as $\nabla\mathcal{L} = \left[\frac{\partial\mathcal{L}}{\partial\dot{q}}, \frac{\partial\mathcal{L}}{\partial\lambda_1}, \frac{\partial\mathcal{L}}{\partial\lambda_2}\right]^T$. Therefore, the gradient $\nabla\mathcal{L}$ can be derived as follow,

$$\nabla\mathcal{L} = \begin{cases} \frac{\partial\mathcal{L}}{\partial\dot{q}} = \eta_1 J^T(J\dot{q} - v_d) + \eta_2 J_{rcm}^T(J_{rcm}\dot{q} - v_{rcm}) \\ \quad +\eta_0\dot{q} + \left(-J^T\right)\lambda_1 + \left(-J_{rcm}^T\right)\lambda_2 \\ \frac{\partial\mathcal{L}}{\partial\lambda_1} = J\dot{q} - v_d \\ \frac{\partial\mathcal{L}}{\partial\lambda_2} = J_{rcm}\dot{q} - v_{rcm} \end{cases} \quad (28)$$

According to the KKT condition defined in [17], if $\nabla\mathcal{L}$ is continuous, the solution of (28) should satisfy the following,

$$\nabla\mathcal{L} = 0 \quad (29)$$

The state decision variable $u(t) = [\dot{q}, \lambda_1, \lambda_2]^T \in R^{n+2m}$. The problem in (29) is equivalent to the following,

$$H(t)u(t) = F(t) \quad (30)$$

where

$$H(t) = \begin{bmatrix} h_1 & -J^T & -J_{rcm}^T \\ J & 0 & 0 \\ J_{rcm} & 0 & 0 \end{bmatrix} \in R^{(n+2m)\times(n+2m)}$$

$$u(t) = [\dot{q}, \lambda_1, \lambda_2]^T, F(t) = [F_1, v_d, v_{rcm}]^T$$

$$h_1 = \eta_0 I_n + \eta_1 J^T J + \eta_2 J_{rcm}^T J_{rcm}$$

$$F_1 = \eta_1 J^T v_d + \eta_2 J_{rcm}^T v_{rcm}$$

The error model of the novel RNN is defined as,

$$e(t) = H(t)u(t) - F(t) \quad (31)$$

To ensure the model error convergence to zero, the corresponding function can be defined as below:

$$\dot{e}(t) = -\alpha P_\Omega(e(t)) \quad (32)$$

$$P_\Omega(e_i(t)) = \frac{(1+\exp(-\xi))(1-\exp(-\xi e_i(t)))}{(1-\exp(-\xi))(1+\exp(-\xi e_i(t)))}$$

where $\alpha > 0$ is the constants which can adjust the convergence rate; $P_\Omega(e_i(t))$ denotes the activation function, and $\xi \geq 2$ which makes $0 \leq |e_i(t)| \leq 1$; Obviously, the error in (32) is convergence to zero with exponential convergence.

To obtain the model of novel RNN, the function in (32) is expanded as,

$$H(t)\dot{u}(t) = -\dot{H}(t)u(t) - \alpha P_\Omega(H(t)u(t) - F(t)) + \dot{F}(t) \quad (33)$$

We further modify the novel recurrent neural network in (33) as,

$$\dot{u}(t) = (I - H(t))\dot{u}(t) - \alpha P_\Omega(H(t)u(t) - F(t)) + \dot{F}(t) \quad (34)$$

For comparison, the traditional gradient descent-based recurrent neural network in [18] is denoted as,

$$\dot{u}(t) = -\alpha H^T(t)(H(t)u(t) - F(t)) \quad (35)$$

For the online solving process, the neural network consists of N neurons and the neural network is designed as,

$$\dot{u}_i = \sum_{j=1}^{N}(I_{ij} - H_{ij}(t))\dot{u}_j(t) - \sum_{j=1}^{N}\dot{H}_{ij}(t)u_j(t) - \alpha P_\Omega\left(\sum_{j=1}^{N}H_{ij}(t)u_j(t) - F_i(t)\right) + \dot{F}_i(t) \quad (36)$$

B. Convergence Analysis

Theorem 1: If there exists the optimal solution $u^* = [\dot{q}^*, \lambda_1^*, \lambda_2^*]^T$, for any initial state $u(0)$, then $u = [\dot{q}, \lambda_1, \lambda_2]^T$ globally converges to the equilibrium point u^*.

Proof: The candidate Lyapunov function is defined as,

$$V(t) = \frac{1}{2}e^T e \quad (37)$$

The $V(t)$ can be described as below:

$$\dot{V}(t) = \frac{dV(t)}{dt} = e^T(t)\dot{e}(t) \quad (38)$$

Equations (32) and (38) are combined as,

$$\dot{V}(t) = -\alpha e^T(t)P_\Omega(e(t)) \quad (39)$$

$$= -\alpha\sum_{i=1}^{N}e_i(t)P_\Omega(e_i(t)) \quad (40)$$

Since $P_\Omega(e(t))$ is the monotone nondecreasing activation function, we can conclude the following,

$$e_i(t)P_\Omega(e_i(t)) = \begin{cases} > 0, & \text{if } \mathrm{e_i(t)} > 0 \text{ or } \mathrm{e_i(t)} < 0 \\ = 0, & \text{if } \mathrm{e_i(t)} = 0 \end{cases} \quad (41)$$

Hence, the time derivative $V(t)$ is formulated as,

$$\dot{V}(t)\begin{cases} < 0, & \text{if } \mathrm{e_i(t)} \neq 0 \\ = 0, & \text{if } \mathrm{e_i(t)} = 0 \end{cases} \quad (42)$$

From the (42), it can conclude that only if $e_i(t) = 0$, $\dot{V} = 0$; otherwise $\dot{V} < 0$. The proof is done. ■

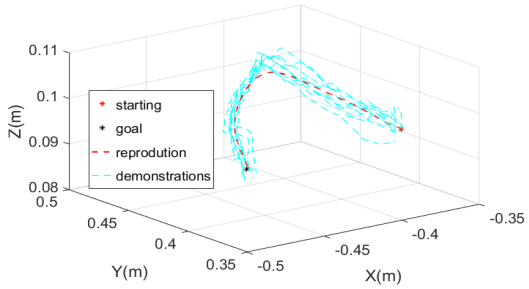

Fig. 3: Learning tumor resection from human demonstrations.

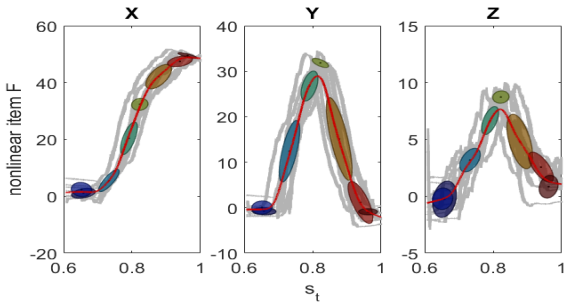

Fig. 4: Regression results corresponding to the nonlinear item F.

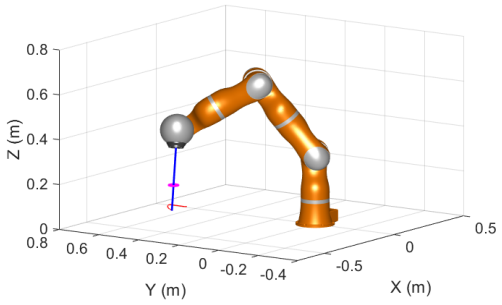

Fig. 5: Motion trajectories obtained using the KUKA simulator.

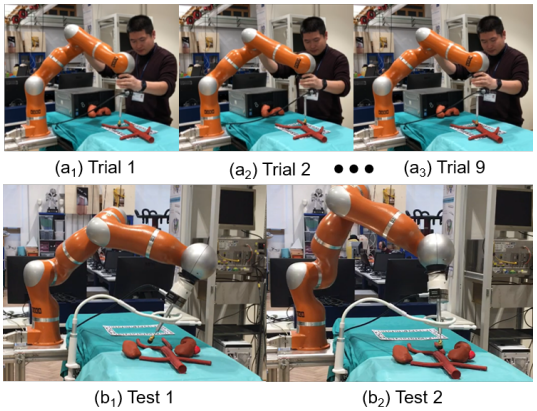

Fig. 6: Experimental demonstration: 1) hands-on control to enable the robot manipulator to learn how to remove tumors by demonstrations; 2) autonomous tracking is activated to demonstrate the surgical operation.

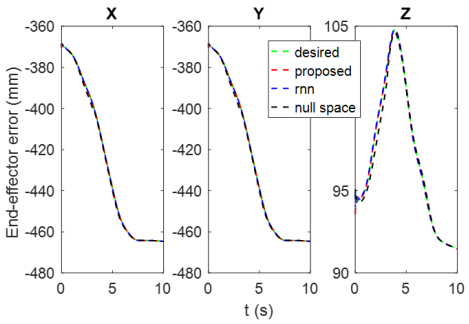

Fig. 7: Trajectories tracking.

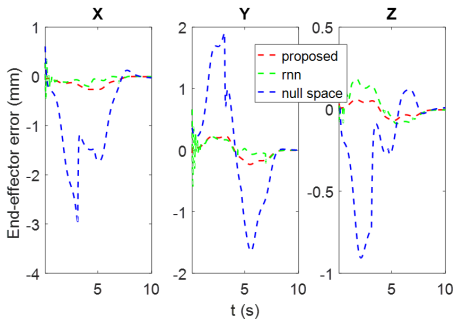

Fig. 8: The comparative performance for the position error of the end effector.

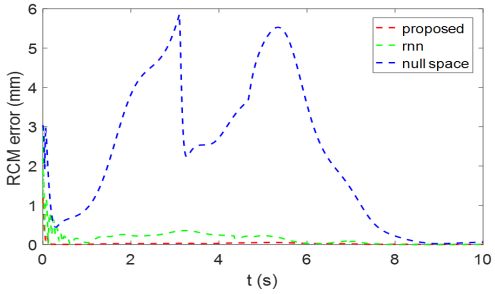

Fig. 9: The comparative results for the distribution of the RCM constraint error.

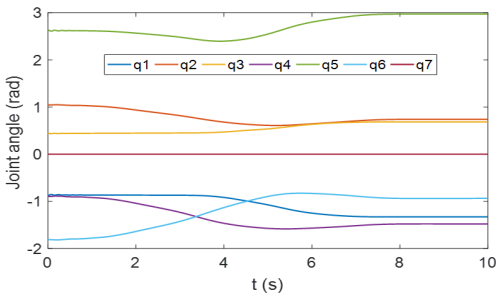

Fig. 10: Joint position of the manipulator.

IV. Simulation and Experimental Demonstration

The parameters of neural networks are chosen as: $\eta_0 = 0.1$, $\eta_1 = 20$, $\eta_2 = 20$, $\gamma = 0.01$, $k_1 = 7$, $k_2 = 7$. In order to evaluate the proposed control scheme, comparison experiments are carried out. The Cartesian position error $\boldsymbol{E}_{end}$ and the RCM constraint error $\|\boldsymbol{E}_{rcm}\|$ are recorded for analysis [19]–[21]. After that, a demonstration using the KUKA simulator is performed to verify the feasibility of the proposed optimization framework firstly, as it is shown in Fig. 5.

Firstly, 9 sample trajectories of tumor resection are collected from human demonstrations, and the learning results are shown in Fig. 3–Fig. 4. Considering the effectiveness of the developed method combined with the related works for the surgucal robot, the experiments are conducted, as shown in Fig. 6. The proposed null space solution in our previous works [5] is performed for comparison.

The approaches described in the related works are also implemented and applied to the same trajectory for the purpose of comparison. Fig. 7 represents the motion trajectories tracking of the real experiments. It should be noted that the real trajectory is able to converge and to track the desired trajectory. Fig. 8-9 represent the comparison of the performance in terms of the tracking error and RCM error estimated in real time during the tracking task. Fig. 10 depicts the actual joints trajectory during the tracking. It can be seen that all errors of the end effector are constrained within the acceptable error range of 4 mm; however, it should be outlined that the proposed RNN achieves the lowest error compared with other considered approaches. The constraint error demonstrates that the proposed RNN has the appropriate performance to ensure compliance with the RCM constraint within 3 mm.

V. Discussion and Conclusion

In this article, we present a novel optimization control method based on improved RNN for a surgical manipulator control under the RCM constraint. It can be used to enable conducting multiple tasks simultaneously, including surgical operation tracking, controlling RCM, and joint limits, etc.. The robot manipulator is developed in such way to learn tumor resection skills in the Cartesian space from human demonstrations. The proposed neural network optimization problem is formulated as a real-time resolution for the given tasks. We observe that it achieves the acceptable convergence performance even in the case of a random initial position. Finally, experimental evaluation has been performed to test the proposed method on the virtual surgical tasks related to the organ phantom.

References

[1] A. M. Okamura, C. Simone, and M. D. O'leary, "Force modeling for needle insertion into soft tissue," *IEEE transactions on biomedical engineering*, vol. 51, no. 10, pp. 1707–1716, 2004.

[2] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. De Momi, "Deep neural network approach in robot tool dynamics identification for bilateral teleoperation," *IEEE Robotics and Automation Letters*, 2020.

[3] H. Su, Y. Schmirander, Z. Li, X. Zhou, J. Li, G. Ferrigno, and E. De Momi, "Bilateral teleoperation control of a redundant manipulator with an rcm kinematic constraint," in *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[4] J. Sandoval, H. Su, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, "Collaborative framework for robot-assisted minimally invasive surgery using a 7-dof anthropomorphic robot," *Robotics and Autonomous Systems*, vol. 106, pp. 95–106, 2018.

[5] H. Su, C. Yang, G. Ferrigno, and E. De Momi, "Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.

[6] H. Su, J. Sandoval, M. Makhdoomi, G. Ferrigno, and E. De Momi, "Safety-enhanced human-robot interaction control of redundant robot for teleoperated minimally invasive surgery," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6611–6616.

[7] P. J. From, "On the kinematics of robotic-assisted minimally invasive surgery," *Modeling, Identification and Control*, vol. 34, no. 2, p. 69, 2013.

[8] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable adaptive neural network control*. Springer Science & Business Media, 2013, vol. 13.

[9] C. Wen, J. Zhou, Z. Liu, and H. Su, "Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1672–1678, 2011.

[10] C. Sun, W. He, W. Ge, and C. Chang, "Adaptive neural network control of biped robots," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 47, no. 2, pp. 315–326, 2016.

[11] Y. Hu, H. Su, L. Zhang, S. Miao, G. Chen, and A. Knoll, "Nonlinear model predictive control for mobile robot using varying-parameter convergent differential neural network," *Robotics*, vol. 8, no. 3, p. 64, 2019.

[12] Z. Li, T. Zhao, F. Chen, Y. Hu, C.-Y. Su, and T. Fukuda, "Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 121–131, 2017.

[13] J. Li, J. Wang, H. Peng, L. Zhang, Y. Hu, and H. Su, "Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction," *Neurocomputing*, vol. 410, pp. 342–353, 2020.

[14] M. Ewerton, G. Maeda, J. Peters, and G. Neumann, "Learning motor skills from partially observed movements executed at different speeds," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 456–463.

[15] J. Silvério, Y. Huang, L. Rozo *et al.*, "An uncertainty-aware minimal intervention control strategy learned from demonstrations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6065–6071.

[16] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.

[17] J. Dutta and C. Lalitha, "Optimality conditions in convex optimization revisited," *Optimization Letters*, vol. 7, no. 2, pp. 221–229, 2013.

[18] H. Su, S. Li, J. Manivannan, L. Bascetta, G. Ferrigno, and E. De Momi, "Manipulability optimization control of a serial redundant robot for robot-assisted minimally invasive surgery," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1323–1328.

[19] H. Su, Y. Hu, Z. Li, K. Alois, G. Ferrigno, and E. De Momi, "Reinforcement learning based manipulation skill transferring for robotassisted minimally invasive surgery," in *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[20] H. Su, S. Ertug Ovur, Z. Li, Y. Hu, K. Alois, J. Li, G. Ferrigno, and E. De Momi, "Internet of things (iot)-based collaborative control of a redundant manipulator for teleoperated minimally invasive surgeries," in *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[21] Z. Li, Z. Ren, K. Zhao, C. Deng, and Y. Feng, "Human-cooperative control design of a walking exoskeleton for body weight support," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 2985–2996, 2019.