

# Improved Recurrent Neural Network-based Manipulator Control with Remote Center of Motion Constraints: Experimental Results

Hang Su<sup>a</sup>, Yingbai Hu<sup>b</sup>, Hamid Reza Karimi<sup>a,\*</sup>, Alois Knoll<sup>b</sup>, Giancarlo Ferrigno<sup>a</sup> and Elena De Momi<sup>a</sup>

<sup>a</sup>Politecnico di Milano, Milan, 20133, Italy

<sup>b</sup>Technical University of Munich, Munich, 85748, Germany

## ARTICLE INFO

### Keywords:

Recurrent Neural Network  
Remote Center of Motion  
Redundant Manipulator  
Robot-assisted Minimally Invasive Surgery

## ABSTRACT

In this paper, an improved recurrent neural network (RNN) scheme is proposed to perform the trajectory control of redundant robot manipulators using remote center of motion (RCM) constraints. Firstly, learning by demonstration is implemented to model the surgical operation skills in the Cartesian space. After that, considering the kinematic constraints associated with the optimization control of redundant manipulators, we propose a novel RNN-based approach to facilitate accurate task tracking based on the general quadratic performance index, which includes managing the constraints on RCM joint angle, and joint velocity, simultaneously. The results of the conducted theoretical analysis confirm that the RCM constraint has been established successfully, and accordingly. The corresponding end-effector tracking errors asymptotically converge to zero. Finally, demonstration experiments are conducted in a laboratory setup environment using KUKA LWR4+ to validate the effectiveness of the proposed control strategy.

## 1. Introduction

In the past decades, common commercial industrial serial robots with redundant manipulators have been successfully adopted and further developed in precise automation processes for a variety of applications [? ?]. Thanks to their lower cost with respect to specialized surgical robots, it has attracted increased research interest for their applications in the medical field, especially in minimally invasive surgery (MIS). These surgical procedures require a small incision in the abdominal wall allowing the insertion of a surgical tool. A robotic implementation of these techniques provides the prospect of an improvement in the control and precision of the surgical tool while reducing trauma to patients [?]. The small incision produces a constraint, commonly known as the Remote Center of Motion (RCM) constraint, on the inserted robot end effector [?]. While a mechanical implementation is generally considered safer but requires complex structures and calibration procedures, a programmable RCM restricting the movement by the control algorithm is

cheaper and more flexible and is, therefore, a preferable option [?]. However, maintaining the RCM constraint during the surgical operations becomes a challenging problem for the surgical robotics society.

Generally, robot-assisted surgical operations can be classified according to the following two modes: 1) a small incision is opened, and the surgical tooltip crosses the abdominal wall by hands-on insertion, and 2) a surgeon controls the tooltip of a slave robot performing a surgical operation.

In the second mode, the physical interaction between the trocar and the abdominal wall is inevitable [?]. Due to the possibility of uncertain disturbances during the physical interaction, the accuracy of the surgical tip and the RCM constraint may be affected, while both are of vital importance to assure the safety of the surgical operation. To comply with the high requirements associated with accuracy, surgical robots are required to learn and adapt the interaction according to the complex environment. Especially, in the case when the interaction is with a rigid object, it may cause considerable force and instability. Impedance control is one of the widely used control methods to improve the robustness of robotic systems [?]. In the related literature, researchers have reported on uncertainties associated with impedance control considering adaptive control considering the environmental dynamics in robotic systems [? ?]. Similar to impedance control, hybrid position/force control has been proven as one of the most efficient control methods to achieve compliancy of the robot's behavior in the operational space, which can modify the desired trajectory based on the external interaction force. This type of control allows making the interaction between the robot and the unknown environment safer. To address the properties of the complex environment such as uncertain disturbance, compensatory items are implemented in control systems [? ? ?].

In addition to the safety issue associated with robot ma-

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.

\*Corresponding author

[hangsu@polimi.it](mailto:hangsu@polimi.it) (H. Su); [yingbai.hu@tum.de](mailto:yingbai.hu@tum.de) (Y. Hu);

[hamidreza.karimi@polimi.it](mailto:hamidreza.karimi@polimi.it) (H.R. Karimi); [knoll@in.tum.de](mailto:knoll@in.tum.de) (A. Knoll); [giancarlo.ferrigno@polimi.it](mailto:giancarlo.ferrigno@polimi.it) (G. Ferrigno); [elena.demomi@polimi.it](mailto:elena.demomi@polimi.it) (E. De Momi)

[hangsu@polimi.it](mailto:hangsu@polimi.it) (H. Su); <https://www.mecc.polimi.it/en/research/faculty/faculty/prof-hamid-reza-karimi/>

(H.R. Karimi); <https://www6.in.tum.de/en/people/prof-dr-ing-habil-alois-knoll/> (A. Knoll); <http://nearlab.polimi.it/medical/giancarlo/> (G. Ferrigno);

<http://nearlab.polimi.it/medical/elenadem/> (E. De Momi)

ORCID(s): 0000-0002-6877-6783 (H. Su); 0000-0003-2452-3570 (Y. Hu); 0000-0001-7629-3266 (H.R. Karimi); 0000-0001-7629-3266 (G. Ferrigno); 0000-0002-8819-2734 (E. De Momi)

nipulation, the RCM constraint should be maintained. This problem was discussed in a number of related research works. General kinematic formalization in the joint space to solve the RCM constraint was presented in [? ]. The novel control architecture for redundant robots was proposed by Sandoval [? ] aiming to solve the fixed RCM constraints through the Cartesian admittance control without considering the movement of the RCM.

In practice, in the case of the robot control systems with RCM constraints, such as MIS, RCM is inevitably kinematic and dynamic. Specifically, with regard to RCM constraints, uncertain disturbances from the kinematic level and interaction force are the major challenges related to providing the accuracy and safety of the robot controller execution. Due to nonlinearity and uncertainties [? ], neither perfect knowledge of a system nor a perfect model can be presumed. RCM constraints do not only affect the accuracy of the robot end effector but may also lead to the instability of the robot control. Therefore, it is essential to investigate control strategies and improve the stability and accuracy of robotic systems aiming to resolve these challenges. Based on the robot control theory, RCM constraints have been studied with regard to several control schemes [? ].

However, the recent results corresponding to the active RCM constraints required to be solved are still unsatisfactory, except for the several schemes that introduced adaptive control theory to address this problem. The efforts and prevalence of applying adaptive control algorithms to solve the RCM constraint are still relatively limited, basically, in the initial stage [? ]. However, adaptive control methods [? ] have been widely investigated in general. Compared with the traditional control methods, the adaptive controls are characterized by the powerful approximation ability [? ]. In [? ], the adaptive neural network-based controller is proposed to approximate uncertainties and nonlinearity aiming to improve dynamically sub-structured systems. In [? ], complex functions are estimated using a fuzzy logic system employed together with the backstepping control approach to optimize its performance. **When it comes to the self-learning optimal problem for discrete-time nonlinear systems, an adaptive critic scheme was investigated in [? ], where an iterative dual heuristic dynamic programming (DHP) strategy was employed to the event-based mechanism. Besides, in order to improve the performance of the reinforcement learning, an approximate neural optimal control scheme was presented in [? ] for the continuous-time nonlinear systems.**

**Application of quadratic programming (QP) problem in practice is still a major challenge. The distributed methods of discrete-time and continuous-time have validated the effectiveness for solving the QP scheme. For example, a network model combined with the supervised learning and deep reinforcement learning was discussed in [? ]. In order to achieve the guaranteed convergence and flexible structure, an iterative low-complexity scheme was proposed to comply with the quadratic program in [? ]. In addition, to solve the active-set with the tight bound more effectively, an active-set approach was applicable to the QP problem in**

**term of the low cost in [? ]. However, these methods are too complicated in calculation, and are not suitable for the fast response in robot system. Therefore, this paper presents a novel RNN-based algorithm to facilitate accuracy for the general quadratic performance.**

In recent years, neural learning enhanced by adaptive control schemes have attracted great attention [? ]. The nonlinear terms of kinematics have been compensated by applying the Nussbaum function and a well defined smooth function in [? ]. In [? ], the adaptive neural impedance control scheme has been designed for an  $n$ -link robotic manipulator with uncertainties generated by the kinematic constraints. An auxiliary system has been introduced in the controller design to cope with the effect of the predefined constraints.

In the present research, a neural-learning enhanced Cartesian admittance control scheme based on the recurrent neural network (RNN) approximation is proposed to improve the accuracy of the end effector and to comply with the active RCM constraint. The main contributions of this paper are highlighted as follows:

1. Learning by demonstration is adopted to model the surgical operation skills in the Cartesian space.
2. **The uncertainties existing in the robotic system due to the RCM constraints are compensated by using a RNN-based adaptive controller scheme.**
3. Experimental demonstration using KUKA LWR4+ is performed to evaluate the applicability of the improved RNN controller.

**The remainder of this paper is organized as follows. Problem statement and preliminaries are presented in Section 2. In Section 3, the control development using neural networks in the presence of RCM constraints is discussed. Finally, experimental results are provided in Section 4, and the conclusions are drawn in Section 5.**

## 2. Preliminaries and Problem Description

In MIS surgical procedures, a surgeon needs to manage the end-effector applied to the patient's organs, such as a 5 mm incision or suturing according to the desired trajectory. One of the key issues associated with skill transfer is to establish the accurate representation of the actuator trajectory. In this section, we discuss the process of skill transfer, focusing on the dynamic movement primitive (DMP), Gaussian mixture Modeling (GMM), and Gaussian mixture regression (GMR). In addition, the RCM and its kinematic control problem investigated in this paper are also described.

### 2.1. Dynamic Movement Primitive

At present, DMP is a general approach used in artificial and biological systems that implies identifying movement primitives for the motor control in robotics and biology [? ]. The DMP method is represented as a set of equations, and it can be used to model different linear or nonlinear motions, which is convenient to imitate the learning of the complex

movement fusion. The DMP is expressed as follow:

$$\begin{aligned}\ddot{S}_t &= K^p (g - S_t) - K^v \dot{S}_t + F(v_t) \\ \dot{v}_t &= \alpha_v v_t \\ F(v_t) &= h_t^T(v_t) \omega (g - S_0)\end{aligned}\quad (1)$$

subjected to

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

where  $[S_t, \dot{S}_t, \ddot{S}_t]$  is the Cartesian space trajectory;  $S_0$  and  $g$  denote the initial and goal positions of an attractor point in the Cartesian space, respectively;  $K^p$  and  $K^v$  are the stiffness matrix, and the damping term of DMP in the 3D Cartesian space, respectively;  $\omega$  is the shape parameter of DMP;  $\alpha_v$  is the scale parameter of the canonical system, where  $v_t$  asymptotically decays from 1 to 0;  $\sigma_i$  and  $c_i$  are the width and center of the  $i$ -th Gaussian kernel.

It should be noted that DMP consists of the following two parts: a linear spring damper part  $K^p (g - S_t) - K^v \dot{S}_t$  and a nonlinear part  $F(v_t)$ , which can be applied to model the trajectories based on learning by demonstrations even in the case of a nonlinear system. Therefore, DMP is deemed applicable to imitate the human motions owing to its feature of convergence to the attractor point  $g$ .

## 2.2. Gaussian Mixture Model

In this subsection, the GMM is presented to encode the trajectories obtained from learning. GMM is a probability-based statistical model that can be used to describe the probability density distribution of a high-dimensional dataset by the sum of weights corresponding to multiple Gaussian models [31]. In the present study, GMM is used to describe the position density in the Cartesian space and to obtain the nonlinear items in DMP by regression corresponding to each GMM. The DMP framework of multi-demonstrations defined in (1) is reformulated in terms of the  $K$  component Gaussian model as follows:

$$\ddot{S} = \sum_{k=1}^K h_k (K_k^p (\mu_k^S - S) - K_k^v \dot{X} + F)\quad (3)$$

where  $\mu_k^S$  and  $h_k$  are the mean and weighted of the  $k$ -th component in Gaussian mixture model.  $K_k^p$  and  $K_k^v$  are the  $k$ -th item stiffness, and the damping matrix of DMP in the Cartesian space, respectively. The Cartesian space data points obtained from demonstrations are defined as follows:  $v_j = (v_{t,j}, v_{S,j})$  ( $j = 1, \dots, N$ ), where  $N$  is the length of a dataset. Each data point includes the time temporal value  $v_{t,j}$  and the position value  $v_{S,j}$ . To encode the dataset of the position distribution  $P(v_t, v_S)$ , the following GMM model is defined as below:

$$p(v_j) = \sum_{k=1}^K p(k) p(v_j|k)\quad (4)$$

where  $K$  is the number of the Gaussian model;  $p(k)$  denotes the prior probability, and  $p(v_j|k)$  is the conditional probability density function.

The manipulator operates in the 3D space, so that the parameters in (4) can be denoted as follows:

$$\begin{aligned}p(k) &= \lambda_k \\ p(v_j|k) &= \frac{1}{\sqrt{(2\pi)^3 |\Sigma_k|}} e\left(-\frac{1}{2} (v_j - \mu_k)^T \Sigma_k^{-1} (v_j - \mu_k)\right)\end{aligned}\quad (5)$$

We define the GMM parameters as  $\Theta = \{\lambda_k, \mu_k, \Sigma_k, E_k\}$  where  $\lambda_k, \mu_k, \Sigma_k$ , and  $E_k$  are the prior probability, mean variable, covariance variable, and the cumulated posterior probability, respectively. According to the Bayes theorem, the cumulated posterior probability  $E_k$  can be expressed as follows:

$$E_k = \sum_{j=1}^N p(k|v_j)\quad (6)$$

$$p(k|v_j) = \frac{p(k)p(v_j|k)}{\sum_{m=1}^K p(m)p(v_j|m)}$$

Then, the log-likelihood of the GMM model  $\Theta$  is defined,

$$\mathcal{L}_\Theta = \frac{1}{N} \sum_{j=1}^N \log(p(v_j))\quad (7)$$

where  $p(v_j) = \sum_{k=1}^K p(k)p(v_j|k)$ . To estimate GMM parameters  $\Theta = \{\lambda_k, \mu_k, \Sigma_k, E_k\}$ , the expectation maximization algorithm described in [32–33] is employed to train the model parameters, and therefore, we obtain the model parameters after their convergence. The iteration is completed when  $\frac{\mathcal{L}_\Theta^{(t+1)}}{\mathcal{L}_\Theta^{(t)}} \leq 0.01$ .

## 2.3. Gaussian Mixture Regression

In fact, the aim of training is to obtain the regression parameter  $F$  from the target dataset. As soon as the GMM obtains the multi-demonstration probability distributions, the GMR is applied to reconstruct the general form of the dataset.

To estimate the conditional expectation value, the observation parameters are defined as follows:  $v = \{v_t, v_S\}$  where  $v_S$  is the spatial variable at the time step  $v_t$ . Therefore, the purpose of the regression is to estimate the conditional expectation of  $v_S$  when the time step  $v_t$  is firstly initiated.

With regard to the multi-demonstrations obtained from learning, the GMM  $\Theta$  encodes the set of trajectories in the Cartesian space obtained from a robot. The  $k$  component of GMM is defined as follows,

$$\mu_k = \{\mu_{t,k}, \mu_{S,k}\}, \quad \Sigma_k = \begin{pmatrix} \Sigma_{tt,k} & \Sigma_{tS,k} \\ \Sigma_{St,k} & \Sigma_{SS,k} \end{pmatrix}\quad (8)$$

where  $\mu_k$  and  $\Sigma_k$  are the mean and covariance matrices of the  $k$ -component GMM, respectively. When the time step  $v_t$

is set, the expected distribution  $v_{S,k}$  of the  $k$ -th component is expressed as below:

$$\begin{aligned} p(v_{S,k}|v_t, k) &= \mathcal{N}(v_{S,k}; \hat{v}_{S,k}, \hat{\Sigma}_{SS,k}) \\ \hat{v}_{S,k} &= \mu_{S,k} + \sum_{S_{t,k}} (\Sigma_{tt,k})^{-1} (v_t - \mu_{t,k}) \\ \hat{\Sigma}_{SS,k} &= \Sigma_{SS,k} - \Sigma_{S_{t,k}} (\Sigma_{tt,k})^{-1} \Sigma_{tS,k} \end{aligned} \quad (9)$$

where  $\hat{v}_{S,k}$  and  $\hat{\Sigma}_{SS,k}$  are derived on the basis of the probability. According to the GMM parameters  $\Theta = \{\lambda_k, \mu_k, \Sigma_k, E_k\}$ , the condition probability density is obtained as follows:

$$\begin{aligned} p(v_S|v_t) &= \sum_{k=1}^K h_k \mathcal{N}(v_S; \hat{v}_{S,k}, \hat{\Sigma}_{SS,k}) \\ h_k &= \frac{p(k)p(v_t|k)}{\sum_{i=1}^K p(i)p(v_t|i)} = \frac{\lambda_k \mathcal{N}(v_t; \mu_{t,k}, \Sigma_{tt,k})}{\sum_{i=1}^K \lambda_i \mathcal{N}(v_t; \mu_{t,i}, \Sigma_{tt,i})} \end{aligned} \quad (10)$$

$$(11)$$

From (8) and (9), estimation of the condition expectation  $v_S$  and the covariance matrix are derived as follows:

$$\hat{v}_S = \sum_{k=1}^K h_k \hat{v}_{S,k}, \quad \hat{\Sigma}_{SS} = \sum_{k=1}^K h_k^2 \hat{\Sigma}_{SS,k} \quad (12)$$

Therefore, the movement  $\hat{v} = \{\hat{v}_t, \hat{v}_S\}$  can be generated by estimating  $\{\hat{v}_S, \hat{\Sigma}_{SS}\}$  at the time step  $v_t$ .

## 2.4. Manipulator Kinematics Model

The forward kinematics model of a redundant manipulator is formulated as follows:

$$r = f(\theta) \quad (13)$$

where  $r \in \mathbb{R}^m$  is the Cartesian space position; and  $f(\cdot)$  represents the nonlinear mapping function, which is used to map the joint space to the Cartesian space. Generally, the joint angle  $\theta$  of a manipulator is physically constrained; however, there exists strong nonlinearity at the position level.

The equation (13) is nonlinear and difficult to solve, as the mapping is not performed in a point-to-point manner. However, it is possible to obtain the linear mapping between the desired velocity  $\dot{r}_d \in \mathbb{R}^m$  and the joint velocity  $\dot{q} \in \mathbb{R}^n$ . The time derivative of (13) is defined as below:

$$J\dot{q} = \dot{r}_d \quad (14)$$

where  $J \in \mathbb{R}^{m \times n}$  is the Jacobian matrix. Let the joint constraints be described as follows:

$$\theta^- \leq \theta \leq \theta^+ \quad (15)$$

$$\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+ \quad (16)$$

where  $\theta^-$  and  $\theta^+$  denote the lower and upper bounds of the corresponding element of the joint angle vector  $\theta$ , respectively;  $\dot{\theta}^-$  and  $\dot{\theta}^+$  denote the lower and upper bounds of the joint velocity vector, respectively.

**Figure 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.

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

$$\lambda(\theta^- - \theta) \leq \dot{\theta} \leq \lambda(\theta^+ - \theta) \quad (17)$$

where  $\lambda > 0$  is the constant coefficient. Therefore, by combining (16) and (17), the joint limits can be reformulated as below:

$$\begin{aligned} \pi^- &\leq \dot{\theta} \leq \pi^+, \dot{\theta} \in \Omega \\ \pi_i^- &= \max\{\dot{\theta}_i^-, \lambda(\theta_i^- - \theta_i)\} \\ \pi_i^+ &= \min\{\dot{\theta}_i^+, \lambda(\theta_i^+ - \theta_i)\} \end{aligned} \quad (18)$$

For the Cartesian space control of a redundant robot, the inverse kinematics is expressed as follows:

$$\theta = f^{-1}(r) \quad (19)$$

where  $f^{-1}(r)$  denotes the inverse nonlinear mapping from the Cartesian space to the joint space. It should be noted that the joint trajectory  $\theta$  is difficult to be obtained by the inverse kinematics provided in (19), due to the high computational complexity. To address this issue, we propose a novel dynamic neural network based method aiming to solve the problem associated with the redundant degree of freedom, as described in Section. 3.

The nonlinear function  $f(\cdot)$  corresponding to serial manipulators can be obtained using the Denavit-Hartenberg convention. In the case of the redundant manipulators control, there exists  $n > m$ . Therefore, the first optimization problem  $F_1(r(t), \theta(t))$  associated with task tracking can be defined as follows:

$$\min \frac{1}{2} \theta^T W \theta \quad (20)$$

$$\text{s.t. } f(\theta) = r_d \quad (21)$$

$$\theta^- \leq \theta \leq \theta^+ \quad (22)$$

where  $r_d$  is the desired trajectory of a surgical task;  $W = I$ .

## 2.5. Remote Center of Motion

During a surgical tracking task, the surgical tooltip of the robot needs to pass through the RCM. Figure 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.

For an  $n$ -DoF robot manipulator, the mapping from its joint space to the Cartesian coordinate  $r_{n-1} \in \mathbb{R}^m$  of its end-effector position  $r_n \in \mathbb{R}^m$  can be described by a nonlinear function as follows:

$$\begin{aligned} r_{n-1} &= f_{n-1}(\theta) \\ r_n &= f_n(\theta) \end{aligned} \quad (23)$$

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 seek 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:

$$\overrightarrow{r_{n-1}r_{rcm}} \times \overrightarrow{r_{n-1}r_n} = 0 \quad (24)$$

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

$$\begin{aligned} e_{rcm} &= \frac{\overrightarrow{r_1 r_{rcm}} \times \overrightarrow{r_{n-1} r_n}}{L} \\ &= \frac{(r_{rcm} - r_{n-1}) \times (r_n - r_{n-1})}{L} \end{aligned} \quad (25)$$

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

We define the Cartesian coordinates  $r_{n-1} = [x_{n-1}, y_{n-1}, z_{n-1}]^T$ ,  $r_n = [x_n, y_n, z_n]^T$  and  $r_{rcm} = [x_{rcm}, y_{rcm}, z_{rcm}]^T$ . From (25), the RCM error model can be expanded as follows:

$$\begin{aligned} e_{rcm} \cdot L &= \\ &\begin{bmatrix} (y_n - y_{n-1})(z_{n-1} - z_{rcm}) - (y_{n-1} - y_{rcm})(z_n - z_{n-1}) \\ (x_{n-1} - x_{rcm})(z_n - z_{n-1}) - (x_n - x_{n1})(z_{n-1} - z_{rcm}) \\ (x_n - x_{n-1})(y_{n-1} - y_{rcm}) - (x_{n-1} - x_{rcm})(y_n - y_{n-1}) \end{bmatrix} \end{aligned} \quad (26)$$

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  $F_2(e_{rcm}(t), \theta(t))$  of the RCM constrains are defined as,

$$\min \frac{1}{2} \theta^T W \theta \quad (27)$$

$$\text{s.t. } e_{rcm} = 0 \quad (28)$$

where the matrix  $W = I$ .

*Assumption 2.1:* In the actual experiment, the RCM constraint  $r_{rcm}$  is known or can be identified by optical tracking.

## 2.6. Problem Reformulation in Terms of Quadratic Programming

In this subsection, we focus on the kinematic control of serial manipulators considering forward kinematics, and focusing on the RCM (28) and joint velocity level constraints (14). The end task and RCM constraints should be considered simultaneously. We expected to find the optimization solution at the velocity level such that the coordinate tooltip tracking error  $\|f_n(\theta) - r_n\|^2$  asymptotically converges to zero

and the RCM deviation error  $\|e_{rcm}\|^2$  remains constrained within a predefined area.

Considering the RCM and end task constraints and end tasks constraints simultaneously, the new optimization problem based on  $F_1(r_n(t), \theta(t))$  and  $F_2(r_{rcm}(t), \theta(t))$  is defined as follows:

$$\begin{aligned} \min \quad & F = F_1(r_n(t), \theta(t)) + F_2(r_{rcm}(t), \theta(t)) \quad (29) \\ \text{s.t.} \quad & \theta^- \leq \theta \leq \theta^+ \\ & \dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+ \end{aligned}$$

However, it is difficult to obtain the optimization solution for the problem (29), as the objective function is non-convex with regard to variable  $\theta$ . Therefore, it is necessary to convert this non-convex optimization problem to a convex one.

Firstly, the optimization problem (29) can be converted as velocity level optimization problem defined as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \dot{\theta}^T W \dot{\theta} \quad (30) \\ \text{s.t.} \quad & J \dot{\theta} = \dot{r}_{nd} \\ & \pi^- \leq \dot{\theta} \leq \pi \end{aligned}$$

Then, the time derivative of the RCM error model in (26) is reformulated as below:

$$J_{rcm} \dot{q} = \dot{e}_{rcm} \quad (31)$$

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

Therefore, the multi-tasks optimization problem can be reformulated as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \dot{\theta}^T W \dot{\theta} \quad (32) \\ \text{s.t.} \quad & J \dot{\theta} = v_d \\ & J_{rcm} \dot{\theta} = 0 \\ & \pi^- \leq \dot{\theta} \leq \pi^+ \end{aligned}$$

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

The angle joint 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 (36) as follows:

$$J \dot{\theta} = -k_1 (f_n(\theta) - r_{nd}) + \dot{r}_{nd} \quad (33)$$

$$J_{rcm} \dot{\theta} = -k_2 (r_{rcm}) \quad (34)$$

where  $v_d = -k_1 (f_n(\theta) - r_{nd}) + \dot{r}_{nd}$ ,  $v_{rcm} = -k_2 (r_{rcm})$ . The optimization problem in (32) is rewritten as,

$$\begin{aligned} \min \quad & \frac{1}{2} \dot{\theta}^T W \dot{\theta} \quad (35) \\ \text{s.t.} \quad & J \dot{\theta} = v_d \\ & J_{rcm} \dot{\theta} = v_{rcm} \\ & \pi^- \leq \dot{\theta} \leq \pi^+ \end{aligned}$$

**Figure 2: Framework of the proposed RNN.**

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 (32) can be reformulated as below:

$$\begin{aligned} \min \quad & \frac{c_0}{2} \dot{\theta}^T \dot{\theta} + \frac{c_1}{2} \|J\dot{\theta} - v_d\|^2 + \frac{c_2}{2} \|J_{rcm}\dot{\theta} - v_{rcm}\|^2 \\ \text{s.t.} \quad & J\dot{\theta} = v_d \\ & J_{rcm}\dot{\theta} = v_{rcm} \\ & \pi^- \leq \dot{\theta} \leq \pi^+ \end{aligned} \quad (36)$$

where  $c_0 > 0$ ,  $c_1 > 0$  and  $c_2 > 0$  are the constants used to balance the different priorities of multi-tasks.

### 3. Neural Network Design and Stability Analysis

In this section, the RNN [?] is applied to solve the multi-tasks optimization problem according to the RCM constraints defined in (36). We first transfer the quadratic programming problem formulated in (36) to the equivalent relationship problem, and then design the RNN to solve it.

#### 3.1. Recurrent neural network for the Quadratic Programming Problem

To obtain the equivalent relationship problem from (36), the Lagrange function is constructed as follows:

$$\begin{aligned} \mathcal{L}(\dot{\theta}, \mu_1, \mu_2) = & \frac{c_1}{2} \|J\dot{\theta} - v_d\|^2 + \frac{c_2}{2} \|J_{rcm}\dot{\theta} - v_{rcm}\|^2 \\ & + \frac{c_0}{2} \dot{\theta}^T \dot{\theta} + \mu_1^T (v_d - J\dot{\theta}) + \mu_2^T (v_{rcm} - J_{rcm}\dot{\theta}) \end{aligned} \quad (37)$$

where  $\mu_1 \in \mathbb{R}^m$  and  $\mu_2 \in \mathbb{R}^m$ . The gradient of  $\mathcal{L}$  is defined as  $\nabla \mathcal{L} = \left[ \frac{\partial \mathcal{L}}{\partial \dot{\theta}}, \frac{\partial \mathcal{L}}{\partial \mu_1}, \frac{\partial \mathcal{L}}{\partial \mu_2} \right]^T$ . Therefore, the gradient  $\nabla \mathcal{L}$  can be derived as follow,

$$\nabla \mathcal{L} = \begin{cases} \frac{\partial \mathcal{L}}{\partial \dot{\theta}} = c_1 J^T (J\dot{\theta} - v_d) + c_2 J_{rcm}^T (J_{rcm}\dot{\theta} - v_{rcm}) \\ \quad + c_0 \dot{\theta} + (-J^T) \mu_1 + (-J_{rcm}^T) \mu_2 \\ \frac{\partial \mathcal{L}}{\partial \mu_1} = J\dot{\theta} - v_d \\ \frac{\partial \mathcal{L}}{\partial \mu_2} = J_{rcm}\dot{\theta} - v_{rcm} \end{cases} \quad (38)$$

According to the Karush–Kuhn–Tucker (KKT) condition defined in [?], if  $\nabla \mathcal{L}$  is continuous, the solution of (38) should satisfy the following requirement,

$$\nabla \mathcal{L} = 0 \quad (39)$$

Considering joint angle constraints  $\dot{\theta} \in \Omega$ , the equation  $\frac{\partial \mathcal{L}}{\partial \dot{\theta}}$  in (38) can be equivalently reformulated as the following form:

$$\dot{\theta} = P_{\Omega} \left( \dot{\theta} - \frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) \quad (40)$$

where  $P_{\Omega}(\cdot)$  denotes the projection operator on set  $\Omega$ .

Then, substituting equation (40) by (39), the nonlinear equations for the two constraint optimization problems can be reformulated as follows:

$$\begin{cases} 0 = -\dot{\theta} + P_{\Omega} \left\{ \dot{\theta} - (c_0 \dot{\theta} + c_1 J^T (J\dot{\theta} - v_d) \right. \\ \quad \left. + c_2 J_{rcm}^T (J_{rcm}\dot{\theta} - v_{rcm}) + (-J^T) \mu_1 + (-J_{rcm}^T) \mu_2 \right\} \\ 0 = J\dot{\theta} - v_d \\ 0 = J_{rcm}\dot{\theta} - v_{rcm} \end{cases} \quad (41)$$

With regard to the equivalence optimization problem defined in (40), the neuro-dynamics method is applied to solve the multi-task optimization problem. The RNNs [?] are constructed as follows,

$$\begin{cases} \gamma \dot{\theta} = -\dot{\theta} + P_{\Omega} \left\{ \dot{\theta} - (c_0 \dot{\theta} + c_1 J^T (J\dot{\theta} - v_d) + \right. \\ \quad \left. c_2 J_{rcm}^T (J_{rcm}\dot{\theta} - v_{rcm}) + (-J^T) \mu_1 + (-J_{rcm}^T) \mu_2 \right\} \\ \gamma \lambda_1 = J\dot{\theta} - v_d \\ \gamma \lambda_2 = J_{rcm}\dot{\theta} - v_{rcm} \end{cases} \quad (42)$$

where  $\gamma$  is the constant and positive value used to scale the convergence rate. In the actual experiment,  $\gamma$  should be set within an appropriate range, as setting too large a value may lead to a slow convergence rate [?]. The framework of the proposed RNN is represented in Figure. 2, where the green line denotes the connection between RCM constraint and state variable  $\theta$ ; the red lines denotes the connection between end-tasks constraint and state variable  $\theta$ .

#### 3.2. Convergence Analysis

As mentioned above, the quadratic programming problem defined in (36) is equivalent to the nonlinear equation in (40). Therefore, in this subsection, we prove the stability and convergence of the RNN in terms of the quadratic programming problem optimization defined in (42). Firstly, we define the decision variable  $u = [\dot{\theta}, \mu_1, \mu_2]^T \in \mathbb{R}^{n+2m}$ .

**Theorem 1.** *If there exists the optimal solution  $u^* = [\dot{\theta}^*, \mu_1^*, \mu_2^*]^T$ , starting from any initial state  $u(0)$ , then  $u_t = [\dot{\theta}, \mu_1, \mu_2]^T$  globally converges to the KKT theoretical point  $u^*$ .*

**Figure 3:** Learning tumor resection from human demonstrations.

**Figure 4:** Gaussian components of GMM.

*Proof.* The RNN defined in (42) is reformulated using the state variable  $u$ ,

$$\gamma \dot{u} = -u + P_{\Omega} \{u - G(u)\} \quad (43)$$

where the projection operator is defined as follows:  $P_{\Omega} \in \mathbb{R}^{n+2m}$ ,  $G(u) = \left[ \frac{\partial \mathcal{L}}{\partial \theta}, \frac{\partial \mathcal{L}}{\partial \mu_1}, \frac{\partial \mathcal{L}}{\partial \mu_2} \right]^T$ .

Based on the equation (39),  $G(u)$  can be expressed as follows:

$$G(u) = \begin{bmatrix} G_1 \\ J\dot{\theta} - v_d \\ J_{rcm}\dot{\theta} - v_{rcm} \end{bmatrix} \in \mathbb{R}^{n+2m}$$

where  $G_1 = c_0 \dot{\theta} + c_1 J^T (J\dot{\theta} - v_d) + c_2 J_{rcm}^T (J_{rcm}\dot{\theta} - v_{rcm}) + (-J^T) \mu_1 + (-J_{rcm}^T) \mu_2$ . The gradient of  $G(u)$  is obtained as following,

$$\nabla G(u) = \begin{bmatrix} \frac{\partial G_1}{\partial \theta} & \frac{\partial G_1}{\partial \mu_1} & \frac{\partial G_1}{\partial \mu_2} \\ J & \mathbf{0} & \mathbf{0} \\ J_{rcm} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (44)$$

where

$$\frac{\partial G_1}{\partial \theta} = c_0 I_n + c_1 J^T J + c_2 J_{rcm}^T J$$

$$\frac{\partial G_1}{\partial \mu_1} = -J^T$$

$$\frac{\partial G_1}{\partial \mu_2} = -J_{rcm}^T$$

Evidently,  $\nabla G(u)$  exists, and it can be concluded that  $G(u)$  is continuously differentiated. Moreover, we can further obtain the following:

$$\begin{aligned} & \frac{1}{2} (\nabla G(u) + \nabla G^T(u)) \\ & = \begin{bmatrix} c_0 I_n + c_1 J^T J + c_2 J_{rcm}^T J & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \geq 0 \end{aligned}$$

It can be further concluded that the gradient matrix  $\nabla G(u)$  is a semi-definite matrix. Therefore, the variable matrix  $G(u)$  is monotone. The proof is completed.  $\square$

**RCM:** the robot tool passes through a small incision (RCM) on the obstacle surface. During the robot manipulation, the tooltip position  $r_1$  is required to follow the desired reference trajectory, while the tool shaft should comply with the kinematic constraint on the obstacle surface.

**Figure 5:** Regression results corresponding to the nonlinear item  $F$ .

**Figure 6:** Motion trajectories obtained using the KUKA simulator.

**Figure 7:** Motion trajectories.

**Figure 8:** RNN optimization results obtained using the skills derived from learning.

## 4. Experiment

The parameters of RNN is chosen as:  $c_0 = 0.1$ ,  $c_1 = 20$ ,  $c_2 = 20$ ,  $\gamma = 0.01$ ,  $k_1 = 7$ ,  $k_2 = 7$ . In order to evaluate the proposed control scheme, comparison experiments are carried out. The magnitude of the Cartesian position error  $E_{end}$  and the RCM constraint error  $\|E_{rcm}\|$  are recorded for analysis. For the null space based solution, the parameters can be found in our previous works [? ].

Firstly, there are six samples of tumor resection collecting from human demonstrations, and the learning results are shown in Figure. 3–Figure. 5. Then, as it is shown in Figure. 6, a demonstration using the KUKA simulator is performed to check the feasibility of the proposed optimization framework.

To consider some methods described in the related works, the experiment performance are displayed in Figure 13. The detailed description of the configuration and development of the system can be found in our previous works [? ].

The operative procedure is organized as follows:

1. The subject uses the hands-on control to move the robot to perform the multiple demonstrated surgical tasks;
2. The skill learning is activated to learn surgical operations;
3. The robot autonomously tracks the set trajectory  $r_d$  using different optimization solutions to reproduce the learned surgical tasks.

The approaches described in the related works are also implemented and applied to the same trajectory for the purpose of comparison. Figures 7 and 8 represent the motion trajectories and the Cartesian trajectory of the real experiments, respectively. It should be noted that the real trajectory converges quickly to the desired trajectory. Figures 9 and 10 represent the comparison of the performance in terms of the tracking error and RCM error estimated real time during the tracking task. In Figure. 9, 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 within 4 mm compared with other considered approaches. Figure 10 demonstrates that the proposed RNN has the appropriate performance to ensure compliance with the RCM constraint within 3 mm.

**Figure 9:** Results of the comparative experiments: the tracking error of the end effector.

**Figure 10:** Results of the comparison experiments: distribution of the RCM constraint error.

**Figure 11:** Joint position of the manipulator.

**Figure 12:** Joint velocity of manipulator.

The joint position and joint velocity are represented in Figures 11 and 12, respectively.

## 5. Discussion and Conclusion

In this paper, we present a novel optimization control method based on RNN for a manipulator in the autonomous laparoscopic robotic surgery. 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 a way to learn tumor resection skills in the Cartesian space from human demonstrations. To control the hierarchical tasks efficiently using the learning skills to comply with RCM constraint, the hierarchical operational space formulation corresponding to a surgical task is investigated. The new optimization problem is formulated as the 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, an experimental evaluation has been performed to test the proposed method on the virtual surgical tasks related to a kidney phantom. In future work, in order to improve the robustness and accuracy, we will consider the global manipulability optimization combined with the RCM constraint for the surgical robot.

**Figure 13:** Experimental setup scene: 1) hands-on control to enable the robot manipulator to learn how to remove tumors by demonstrations; 2) autonomous tracking is activated to run the application.