



RESEARCH ARTICLE

Neural-learning-enhanced Cartesian Admittance control of robot with moving RCM constraints

Hang Su¹ , Yunus Schmirander², Sarah Elena Valderrama-Hincapié¹, Wen Qi^{3,4,*} ,
Salih Ertug Ovur⁵ and Juan Sandoval⁶

¹Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133, Milan, Italy, ²Department of Computer Science, University of Innsbruck, Innsbruck, Austria, ³School of Future Technology, South China University of Technology, Guangzhou, China, ⁴Pazhou Lab, Guangzhou, China, ⁵Department of Electrical and Electronic Engineering, Imperial College London, London, UK, and ⁶Department of GMSC, Pprime Institute, CNRS, ENSMA, University of Poitiers, UPR 3346, Poitiers, France

*Corresponding author. E-mail: wenqi@scut.edu.cn

Received: 20 July 2022; **Revised:** 28 October 2022; **Accepted:** 2 November 2022; **First published online:** 20 December 2022

Keywords: adaptive neural control, Cartesian admittance control, null space, moving remote center of motion

Abstract

In this manuscript, a scheme for neural-learning-enhanced Cartesian Admittance control is presented for a robotic manipulator to deal with dynamic environments with moving remote center of motion (RCM) constraints. Although some research has been implemented to address fixed constrained motion, the dynamic moving movement constraint is still challenging. Indeed, the moving active RCM constraints generate uncertain disturbance on the robot tool shaft with unknown dynamics. The neural-learning-enhanced decoupled controller with disturbance optimisation is employed and implemented to maintain the performance under the kinematic uncertain and dynamic uncertain generated. In addition, the admittance Cartesian control method is introduced to control the robot, providing compliant behaviour to an external force in its operational space. In this proposed framework, a neural-learning-enhanced disturbance observer is investigated to calculate the external factor operating on the end effector premised on generalised momentum in order to ensure accuracy. Finally, the experiments are implemented using a redundant robot to validate the efficacy of the suggested approach with moving RCM constraints.

1. Introduction

Commercially available manufacturing serial robots with multiple manipulators have indeed been effectively accepted and further improved in accurate automation processes for a wide range of application areas during the last few decades. Their relatively lower cost in comparison to specialist surgical robots has sparked greater interest in medicinal uses [1, 2], especially in minimally invasive surgery (MIS) procedures. A tiny incision inside the abdominal wall is required for these operations to facilitate the insertion of a medical instrument. A robotic application of these procedures offers the possibility of improved surgical tool control and precision while lowering patient trauma [3]. The minor incision places a limitation upon the inserted robot end effector referred to as the remote center of motion (RCM) constraint [4]. While a mechanical implementation is typically deemed safer, it needs bulky, costly structures plus calibration processes, a programmable RCM regulating motion by the control algorithm is less costly and more versatile, making it a better alternative [5, 6]. It is true that more than the surgical scenario, many industrial applications involve the RCM. For example, a safety region convolutional neural network (CNN) method adapted with the RCM method was implemented on Baxter robot, a seven-degrees of freedom redundant manipulator, demonstrating the feasibility of RCM control in industrial area [7].

Although many works have been performed and implemented to maintain the movement constraints, few works have considered a moving RCM constraint, presenting the challenge with dynamic kinematic

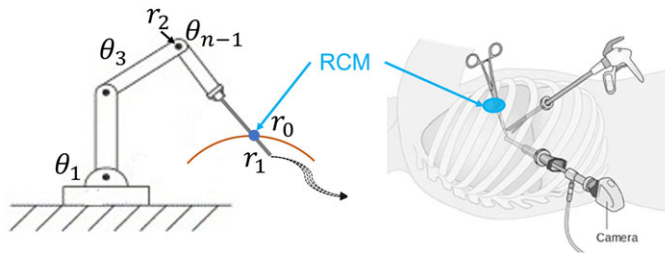


Figure 1. Remote center of motion in medical application.

and dynamic disturbance. Since the moving RCM forges a dynamic kinematic obstacle, it is uncomplimentary to make the robotic tool contact and interact with the hole, which delivers a dynamic interaction. This makes it unlike a fixed RCM and becomes an additional complex issue. Additionally, for safer robot operation, the moving RCM constraint must be preserved. Ortmaier *et al.* employed the inverse kinematic control to prevent any force exerted on the RCM constraint [8]. In ref. [9], a general kinematic formalisation in joint space to address the RCM constraint was offered. A new control architecture was put forward by Sandoval [8] for redundant robots to tackle fixed RCM constraints with Cartesian admittance control [10] without taking into account the movement of the RCM. But nearly all of these works consider only fixed movement constraints.

In recent years, more attention has been drawn to neural learning improved adaptive control methods [11–13]. The nonlinear kinematic terms were compensated by integrating a smooth function in refs. [14, 15]. In ref. [16], an adaptive neural resistance control technique was developed for a n -link robotic manipulator with kinematic constraints. A supplementary system was added to the controller architecture to address the effect of the above limits.

In this work, a neural-learning boosted Cartesian admitting control system that relies on the neural approximation is suggested to improve end effector accuracy while still obeying the active RCM requirement. This paper's control framework includes:

1. A Cartesian admittance control is used in robotic devices to produce compliant behaviour while considering environmental dynamics in Cartesian space.
2. A decoupled controller is employed to maintain the RCM constraints on its kinematic level.
3. An adaptive controller compensates for the uncertainty in the robotic system caused by the shifting RCM restrictions.

Different from the previous works, this work considers a moving movement constraint and conducts a demonstration and analysis to check its feasibility to use an adaptive approximation to compensate for the disturbances generated by uncertain RCM point movement.

The rest of this manuscript is structured as follows. Section 2 contains the research problem and preliminary information. Section 3 discusses control creation with neural networks in the context of moving RCM limitations. Finally, in Section 4, the experimental data are displayed, and in Section 5, the conclusions are formed.

1.1. Problem statement and preliminaries

1.2. Moving RCM constraints

1.2.1. Kinematics analysis

To track the trajectory from the actual Cartesian position to the target position during the robot manipulation, as shown in Fig. 1, the end effector must go through the small incision P on the obstacle wall, representing the RCM constraint, where \mathbf{r}_0 is the RCM position, \mathbf{r}_1 is the actual Cartesian position, and \mathbf{r}_2 is the actual position of the position of joint θ_{n-1} . To respect the RCM constraint, \mathbf{r}_0 should always

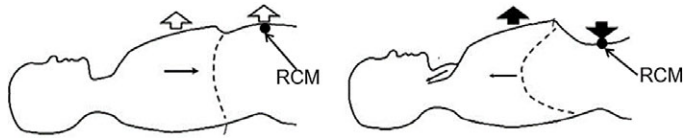


Figure 2. Moving remote center of motion due to breathing.

be on a straight line from the tooltip’s position, r_1 , to the position of the joint holding the tool, r_2 . In most previous works, no movement was applied to this point has been frequently assumed to simplify the complexity of the application. However, for an actual clinical operation, the patient’s abdominal wall may not be fixed due to breathing or heart beating, as it is shown in Fig. 2. The relation can be derived as

$$(r_0 - r_1) \times (r_2 - r_1) = 0 \tag{1}$$

Assumption 2.1: The RCM constraint position $r_0 \in R^3$ is known or can be detected with optical tracking.

Consider an n -degrees-of-freedom manipulator, the projection from its joint space to the Cartesian coordinate $r_1 \in R^m$ of its end effector is characterised by the following nonlinear function:

$$\begin{aligned} r_1 &= f_1(\theta) \\ r_2 &= f_2(\theta) \end{aligned} \tag{2}$$

where $\theta \in R^n$ can be seen as the joint angle vector. The nonlinear function $f_1(\cdot)$ and $f_2(\cdot)$ for serial manipulators can be obtained through a systematic method known as the D-H convention. For excessive manipulators, $n > m$. Straightforwardly, computing the time derivatives on both sides of (1) yields

$$\begin{aligned} \dot{r}_1 &= J_1(\theta)\omega \\ \dot{r}_2 &= J_2(\theta)\omega \end{aligned} \tag{3}$$

where $\dot{r}_1 = dr_1/dt$, $\dot{r}_2 = dr_2/dt$, $J_1(\theta) = \partial f_1(\theta)/\partial \theta$, $J_2(\theta) = \partial f_2(\theta)/\partial \theta$, and $\omega = \dot{\theta} = d\theta/dt$.

1.2.2. Dynamic modelling

F_e is the outside force created through the physical contact [17, 18] of the RCM constraint and the robot tool shaft, which maybe be modelled using a mass-damping-spring model:

$$M_p \ddot{P} + H_p(\dot{P}) \dot{P} + G_p(P) = F_e \tag{4}$$

where M_p , H_p , G_p are the inertia, damping and stiffness matrices of the obstacle wall around the RCM constraint, $F_e \in R^3$ is the outside force from the physical contact, and P is the actual contact point between the surgical tool shaft and the obstacle. F_e can be decomposed into two components $F_{eX} \in R^3$ and $F_{eN} \in R^3$ [19], where F_{eX} is the projected outside force on the tooltip in the task space, and F_{eN} is the external force acting on the corresponding null space of the tooltip.

Assumption 2.2: The time-varying external force $F_e(t) \in R^3$ from physical interaction between the robot tool shaft and the obstacle wall is continuous in terms of time and constrained by a constant β as follows:

$$\exists \beta \in R, \|F_e(t)\| \leq \beta, \forall t \geq 0 \tag{5}$$

1.3. Robot modelling

The dynamic model of the n degrees of freedom (DoFs) serial manipulator in the Lagrangian formulation can be expressed as [20]

$$M(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) \dot{\theta} + g(\theta) - \tau_e = \tau_c \tag{6}$$

where $\theta \in R^n$ is the joint vector, $M(\theta) \in R^{n \times n}$ is the positive definite, symmetric and bounded inertia matrix in the robot joint space, $C(\theta, \dot{\theta}) \in R^{n \times n}$ is a matrix which represents the effects for Coriolis and centrifugal and $g(\theta) \in R^n$ is the vector for gravity torques. The torque vectors $\tau_c \in R^n$ and $\tau_e \in R^n$, respectively, symbol the control torques [21] and the outside disturbance torque vectors [22, 23]. For simplicity, it is presumed that the surgical robotic system is different from any pseudoinverse and singularity of the Jacobian matrix, $J_1(\theta) \in R^{m \times n}$, in existence.

Since for Cartesian admittance control, the task space expresses the expected behaviour of the serial robot, and the generalised formulation can be written as [19, 24, 25]

$$M_r \ddot{r} + H_r(r, \dot{r}) \dot{r} + J_1^{-T}(\theta) g(\theta) - F_{eT} = F_c \tag{7}$$

where $r = r_1 \in R^m$ is the vector of the task space coordinates,

$$M_r(r) = J_1^{-T} M(\theta) J_1^{-1}, \tag{8}$$

$$H_r(r, \dot{r}) = J_1^{-T} [C(\theta, \dot{\theta}) - M(\theta) J_1^{-1} \dot{J}] J^{-1}, \tag{9}$$

$$F_{eT} = J^{-T} \tau_e. \tag{10}$$

The matrix $M_r(r) \in R^{m \times m}$ is the Cartesian inertia matrix, $H_r(r, \dot{r}) \in R^{m \times m}$ is the Coriolis and centrifugal force effects in the task space and $F_{eT} \in R^m$ is the external force in the task space, which is bounded by β .

$$\exists \beta \in R, \|F_{eT}\| \leq \beta, \forall t \geq 0 \tag{11}$$

Property 2.1: The Cartesian inertia matrix, $M_r(r)$, as defined in (8), positive and symmetric, and it is also bounded as

$$\lambda_1 \|A\| \leq A^T M_r(r) A \leq \lambda_2 \|A\|, \forall A, r \in R^m \tag{12}$$

where λ_1 and λ_2 are positive constants, and A is assumed as non-singular matrix.

Property 2.2: The Cartesian matrix, $H_r(r, \dot{r})$, and the time derivative of the Cartesian inertia matrix, $\dot{M}_r(r)$, fulfill the criterion:

$$A^T [\dot{M}_r(r) - 2H_r(r, \dot{r})] A = 0, \forall A, \theta, \dot{\theta} \in R^n \tag{13}$$

where A is assumed as a non-singular matrix.

1.4. Cartesian admittance control

In some applications in which the objective is to control the end effector to follow a path in free-space, position control methods are appropriate. Nevertheless, in some instances with a rigid interaction between the environment and the end effector, a large force will be generated and compromise the safety of the operation. This could be avoided by putting in place a compliant controller on the robot. Among the most commonly used compliant control approaches, Cartesian admittance control is an efficient mechanical admittance method [26], featuring damping, stiffness and mass terms:

$$F_{ext} = M_d \ddot{r}_e + D_d \dot{r}_e + K_d r_e \tag{14}$$

where $r_e = r_{1c} - r_{1d}$, M_d, D_d and K_d are the symmetric and positive definite matrices of the desired inertia, damping and stiffness, respectively. $r_{1d} \in R^m$ is the desired trajectory, and $r_{1c} \in R^m$ is the desired trajectory determined by the external force F_{ext} on the end effector. It can be seen that $r_{1c} = r_{1d}$ when there is no external interaction on the end effector, that is, $F_{ext} = 0$. Meanwhile, when $F_{ext} \neq 0$ indicates external interaction on the end effector, the desired trajectory will be modified as a new trajectory that can be seen as the adaptation relation between the external force and the target admittance model in [27]. As it is shown in Fig. 3, the proposed framework is constructed with a decoupled control and adaptive approximation layer.

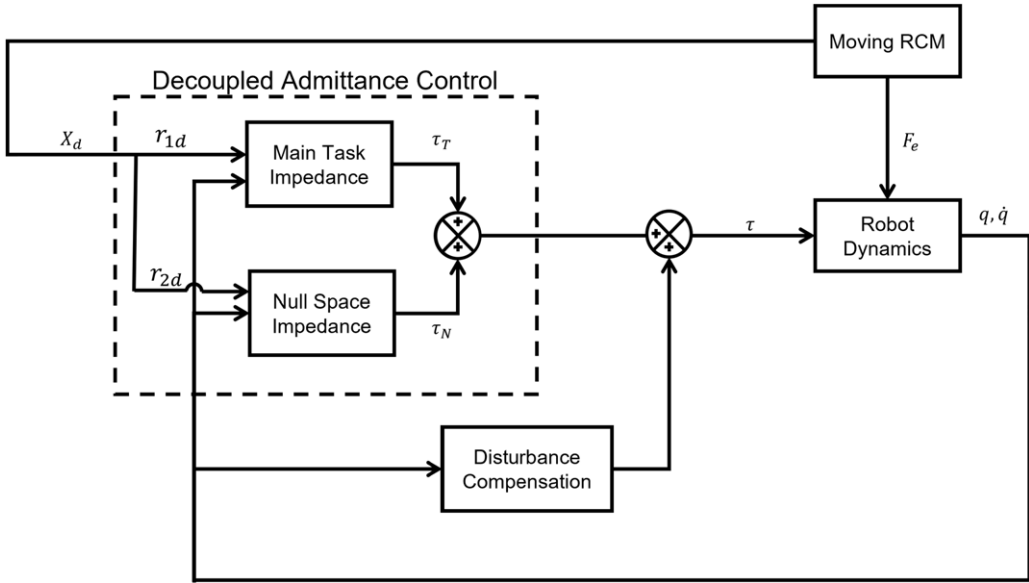


Figure 3. The proposed control architecture.

1.5. Null-space projection and decoupling control

By specifying the joint angular velocity in the joint space and the end effector velocity in the workspace:

$$\dot{r}_1 = J_1(\theta) \dot{\theta} \tag{15}$$

where $\dot{r}_1 \in R^3$ is the real Cartesian velocity for the end effector, and $J(\theta) \in R^{3 \times n}$ is the Jacobian matrix from the base to the end effector. To drive the end effector ($r_1 \in R^3$) to reach the desired position ($r_{1d} \in R^3$), the task control torque, τ_T , can be defined as

$$\tau_T = J_1^T(K_{r1}\tilde{r}_1 - D_{r1}\dot{r}_1) \tag{16}$$

where $J(\theta) \in R^{3 \times n}$ is the Jacobian matrix from the base to the end effector, $\tilde{r}_1 = r_{1d} - r_1(\theta) \in R^{3 \times n}$, $K_{r1} \in R^{3 \times 3}$ is the diagonal stiffness matrix and $D_{r1} \in R^{3 \times 3}$ is the diagonal damping matrix.

Because the number of the degrees of freedom is larger than three, the redundant serial robot can be employed to complete additional tasks by the null-space controller ($\tau_N \in R^n$), which can be specified as

$$\tau_N = (I - J_1^T(\theta)J_1(\theta)_M^+) \tau_{T_2} \tag{17}$$

where $\tau_{T_2} \in R^{n-3}$ serves as the torque to complete the additional assignment with redundant degrees, $J(\theta)_M^+$ is the inertia-weighted pseudoinverse matrix [24] defined as

$$J(\theta)_M^+ = (J_1(\theta)M(\theta)^-1J_1(\theta)^T)^-1J_1(\theta)M(\theta)^-1 \tag{18}$$

1.6. Adaptive neural approximation

The neural approximation has been well known to be adopted for the approximation of the uncertain disturbances [28], and here introduces a kind of feedforward neural web which is constructed with three general layers, that is, the hidden layer, the input layer and the output layer. Regarding the minimum MSE, it attains a globally optimised solution to the adjustable weights. The inputs variables (Z_1, \dots, Z_m) are distributed from the input layer to the hidden layer. The hidden layer consists of N neurons, and each neuron computes the Euclidean distance between the inputs and the centres. Every neuron from the hidden layer employs a radial basis function ξ as its nonlinear activation function. In this paper [20], the

Gaussian function is depicted as the most commonly used activation function in the RBFNN algorithm. The hidden layer conducts a nonlinear transform of the input, the output layer serves as a mixture of the weighted hidden layer outputs that calculates the approximated output variable (y_1, \dots, y_j) . The output of RBFNN (y_1, \dots, y_j) is computed as

$$y_i = \sum_{k=1}^N \omega_{ki} \xi_i, \quad i = 1, 2, \dots, j \tag{19}$$

where y_i is the i th output, ω_{ki} is the connection weight from the k th hidden unit to the i th output unit and j is the dimension of the outputs.

The approximation is employed in this study to correct for the Cartesian position mistake. As a result, the input Z is picked with the Cartesian position r . The uncertain disturbance $\Xi(r, \dot{r})$ is represented as

$$\Xi(r, \dot{r}) = \Omega \cdot \xi(r, \dot{r}) + \varepsilon \tag{20}$$

where ε is the smallest approximation error of RBFNN algorithm, $\Omega = [\omega_1, \omega_2, \dots, \omega_l]^T$ is the best constant weight matrix of neural web, l is the number of nodes used in neural networks and $\xi(r, \dot{r}) = [\xi_1, \xi_2, \dots, \xi_l]$ is Gaussian function matrix. According to the RBFNN approximation theory, the approximation error has an upper bound ε^* , i.e. $|\varepsilon| \leq \varepsilon^*$, with a positive constant $\varepsilon^* > 0$.

The Gaussian function matrix employed in the RBFNN algorithm is specified as below:

$$\xi_i(Z) = \exp \left[\frac{-(Z - c_i)^T (Z - c_i)}{b_i^2} \right] \tag{21}$$

Whereas c_i is the central point of receptive field, b_i appears as the width of the Gaussian function. In accordance with the rule of approximation, a positive constant δ such that $\|\xi(Z)\| \leq \delta$ with $\delta > 0$ exists.

The adaptive neural web update law [28] is presented to modify the weights of neural network as:

$$\dot{\Omega} = \gamma [E \xi^T(r, \dot{r}) + \varsigma \Omega] \tag{22}$$

where γ is a diagonal positive definite constant matrix that determines the update pace, E is the error vector and ς is a neural network momentum factor matrix that can boost both training speed and accuracy accuracy and training speed.

2. Control development and stability analysis

In this part, the neural-learning improved admittance control method with RCM constraints has been developed, as displayed in Fig. 4. Given the required Cartesian location, the end effector of the robot should attain r_{1d} without breaching the RCM constraints. The desired trajectory is adapted to the interaction on the end effector, while the external interaction force on the RCM constraints should not influence the desired trajectory and its accuracy. The RCM constraint interaction, shown in Fig. 1, is regarded as the disturbance of the system which is compensated by neural network adaption.

2.1. Remote center of motion

In order for the end effector position from the master ($r_1 \in R^3$) to reach the desired position ($r_{1f} \in R^3$), an interpolation approach is formulated so to move towards the wanted location smoothly as [6]

$$r_{1d} = -k_0(r_1 - r_{1f}) + \dot{r}_{1f} \tag{23}$$

where $k_0 > 0$ serves as a positive coefficient. On the basis of the r_{1d} , we can get the wanted input r_{1c} from the model of admittance control (14).

In accordance with the scene in Fig. 5 with a serial robot, d is the distance between the tool (for a better understanding, the RCM constraint is increased in size.) and the RCM point (r_0). The tooltip position r_1 has been controlled to trace the objective r_{1d} in the patient’s abdomen. The value v_2 represents

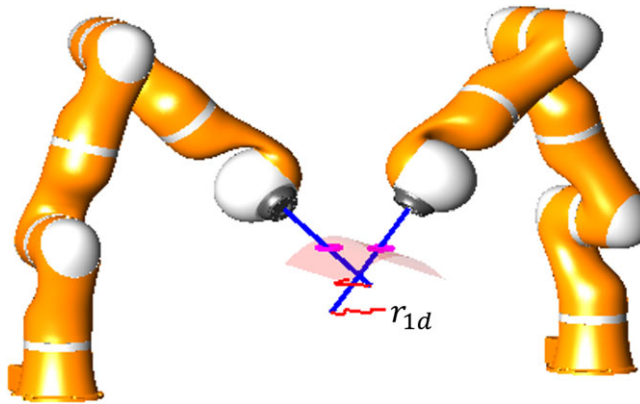


Figure 4. Robots control with RCM constraints.

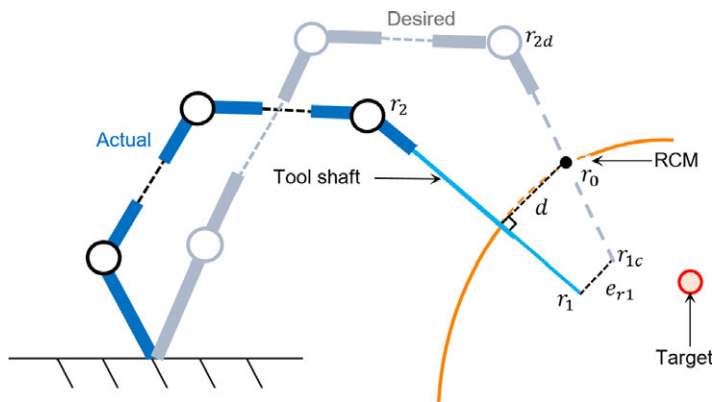


Figure 5. Achieving active RCM constraint with a serial robot.

the velocity for moving the wrist out of its current location r_2 to its expected position r_{2d} , where the tool shaft traverses the RCM point r_0 . Thus, the wanted wrist position r_{2d} can be acquired from (1):

$$r_{2d} = r_{1d} + \frac{r_0 - r_{1d}}{\|r_0 - r_{1d}\|} \|r_2 - r_1\| \tag{24}$$

The whole control framework is shown in Fig. 6. Different from the previous studies, considering the disturbance from the moving RCM, a neural approximation scheme is utilised in the control loop to manage the disruption in the kinematic level, and a Cartesian Admittance control is employed to achieve compliant interaction.

2.2. Controller development

In this part, we create a controller to trace the wanted trajectory with RCM constraints. Firstly, we define a reference trajectory in the Cartesian space and an RCM constraint trajectory which is moving r_0 in the workspace of the robot. Then we define the related error signals, shown in Fig. 5, as follows:

$$\begin{aligned} e_{r1} &= r_{1c} - r_1 \\ e_{r2} &= r_{2d} - r_2 \\ d &= \|(r_0 - r_1) \times \frac{r_2 - r_1}{\|r_2 - r_1\|}\| \end{aligned} \tag{25}$$

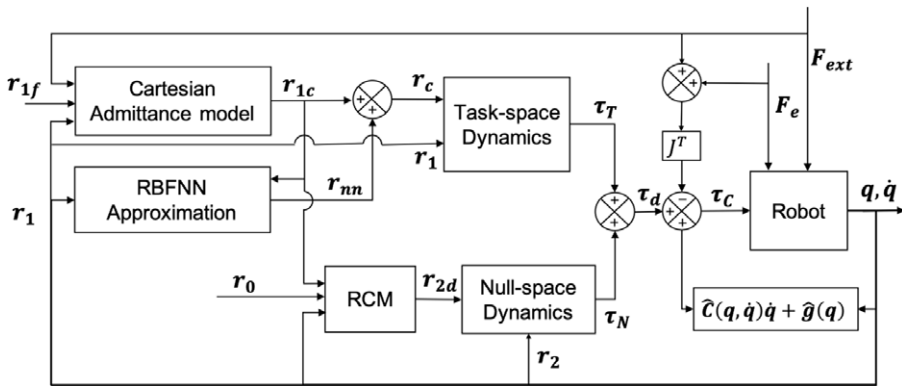


Figure 6. The overview of the control scheme.

According to the decoupled control strategy [19], the optimal torque control, τ_c^* , can be defined as

$$\tau_c^* = \tau_{dynamics} + \tau_T + \tau_N + \vartheta + \tau_e \tag{26}$$

$$\tau_{dynamics} = \hat{C}(q, \dot{q}) \dot{q} + \hat{g}(q) \tag{27}$$

$$\tau_T = J_1^T (K_1 e_{r1} - D_1 \dot{e}_{r1}) \tag{28}$$

$$\tau_N = J_2^T (K_2 e_{r2} - D_2 \dot{e}_{r2}) \tag{29}$$

where $\tau_{dynamics}$ is the robot dynamic torque computed inside the robot motion kernel [29], $\hat{C}(q, \dot{q})$ and $\hat{g}(q)$ are the estimated inertia matrix, Coriolis and Centrifugal effects and gravity term in the joint space, respectively [29]. ϑ is the estimation error of the $\tau_{dynamics}$, τ_T is the task space control input, τ_e is the outside torque in the robot combined space. K_1 is the stiffness matrix in the Cartesian space and D_1 is the damping matrix in the Cartesian space. K_2 is the stiffness matrix for the wrist position control, and D_2 is its damping matrix.

According to the control aim, the reference trajectory should be only modified by external forces F_{ext} on the end effector in (14), and not be influenced by the force on the tool shaft F_e . Since F_{ext} has been considered in the design of the Cartesian admittance control model, the uncertain terms in (26), which represent the disturbances from external force F_e and gravity compensation error, are compensated by the adaptive approximation of artificial neural networks. Instead of approximating the disturbances in the joint space, we propose to estimate the disturbance in the Cartesian space and introduce an adaptive term r_{nn} . Given the desired trajectory r_{1c} from the Cartesian admittance control model, there is an error between the actual position r_1 and the desired position r_{1c} due to the external force F_e on the tool shaft.

The neural network approximation gives:

$$\hat{r}_{nn} = \Omega \cdot \xi(r_1, \dot{r}_1) \tag{30}$$

The updating law Ω of RBFNN is

$$\dot{\Omega} = \gamma [e_{r1} \xi^T(r_1, \dot{r}_1) + \zeta \Omega] \tag{31}$$

where γ is a diagonal positive definite constant matrix that determines update speed, e_{r1} is the error vector and ζ is the neural network momentum factor matrix, which is able to boast both accuracy and training speed.

The final control optimal term is

$$\tau_c^* = \tau_{dynamics} + \tau_T^* + \tau_N \tag{32}$$

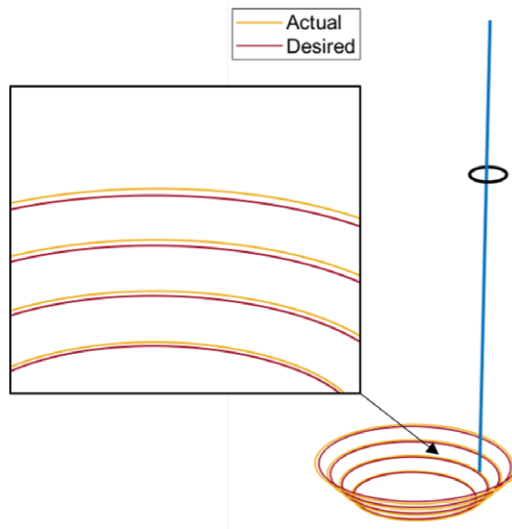


Figure 7. Fixed RCM constraint without neural learning. The desired helix trajectory and the actual trajectory are shown.

$$\tau_T^* = J_1^T(K_1(r_c - r_1) - D_1\dot{e}_{r1}) \tag{33}$$

where $r_c = r_{1c} + r_{nm}$ is the adapted input in the task space.

3. Experimental demonstration

The control was validated by using a KUKA robot, tracking various trajectories using the developed controller defined in (32) with the comparison of the controller developed in ref. [10]. A straight tool with the length $l = 0.22$ m is attached to the robot manipulator. The RCM was set to move in a sinusoidal way, and the RBFNN was introduced to compensate for this uncertain disturbance due to the RCM constraints.

A 7-DOF KUKA LWR4+ robot was controlled to track the desired trajectory. The end effector was utilised to track various trajectories starting from a non-singular home Cartesian location of the end effector $r_{1f,0}$ to a final point $r_{1f,n}$ in the robot workspace. Initially, a fixed RCM was established to validate the decoupled control strategy and then a moving RCM was used to validate and compare the developed neural-enhancing controller and the controller developed in [10].

3.0.1. Test of fixed RCM constraints

Firstly, the experimental setup was tested with fixed RCM constraints and various reference trajectories in its workspace. One of the experiments was shown in Fig. 7 with a helix trajectory. $r_{1f,0} = [-0.2285, 0.4835, 0.2095]$ m and $r_{1f} = r_{1f,0} + [(0.025 + 0.02t) \cos(8\pi t), (0.025 + 0.02t) \sin(8\pi t), 0.02t]$ m. A fixed RCM constraint point was chosen as $r_0 = [-0.2285, 0.4835, 0.2095]$ m. The control parameters were chosen as: $k_0 = 0.1$, $K_1 = \text{diag}[3000, 3000, 3000]$, $D_1 = \text{diag}[30, 30, 30]$, $K_2 = \text{diag}[300, 300, 300]$, and $D_2 = \text{diag}[3.5, 3.5, 3.5]$.

As shown in Fig. 7, the robot tool shaft needed to go through the RCM constraint point at all times during the trajectory tracking of the end effector. The 3-D tracking trajectory was magnified to highlight the tracking error.

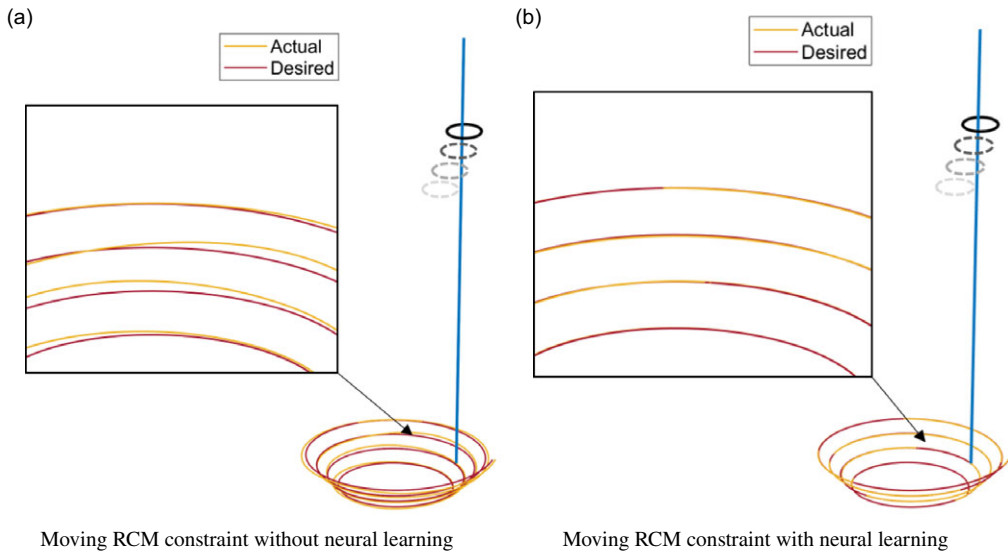


Figure 8. Moving RCM simulations when the (a) null-space and (b) RBFNN controller are implemented for a helix trajectory.

3.0.2. Test of moving RCM constraints

Subsequently, the experimental setup was tested with moving RCM constraints and various reference trajectories in its workspace. The experiment with a helix trajectory with increasing radius was shown in Fig. 8. $r_{1f,0} = [-0.2285, 0.4835, 0.2095]$ m and $r_{1f} = r_{1f,0} + [(0.025 + 0.02t) \cos(8\pi t), (0.025 + 0.02t) \sin(8\pi t), 0.02t]$ m. A moving RCM constraint point was chosen as $r_0 = [-0.2285, 0.4835, 0.2095]$ m. Firstly, the controller developed in ref. [10] was applied to control the end effector tracking with the parameters: $k_0 = 0.1, \mathbf{K}_1 = \text{diag}[3000, 3000, 3000], \mathbf{D}_1 = \text{diag}[30, 30, 30], \mathbf{K}_2 = \text{diag}[300, 300, 300],$ and $\mathbf{D}_2 = \text{diag}[3.5, 3.5, 3.5]$. And the neural-learning-enhanced controller was applied on the same trajectory tracking with parameters set as: $k_0 = 0.1, \mathbf{K}_1 = \text{diag}[3000, 3000, 3000], \mathbf{D}_1 = \text{diag}[30, 30, 30], \mathbf{K}_2 = \text{diag}[300, 300, 300], \mathbf{D}_2 = \text{diag}[3.5, 3.5, 3.5]$. Furthermore, we employed 6 nodes for each input dimension of the neural network, with initial weights $\Omega(0) = [0, 0, 0, 0, 0, 0]$, the learning rate $\gamma = 0.2$ and momentum $\zeta = 0.05$. The centres of the radial basis functions were defined as $c_i = [-3, -1.5, 0, 1.5, 3, 3.5]$ and their widths as $b_i = [2, 2, 2, 2, 2, 2]$.

The performance of the controller in ref. [10] with moving RCM constraint was shown in Fig. 7, and the performance of neural-learning-enhanced controller in (32) was shown in Fig. 7. Compared to the 3-D tracking error with the controller applied in ref. [10], the tracking error was almost eliminated by employing the proposed neural-learning-enhanced controller.

The above Cartesian trajectory tracking error and the RCM constraint error of the helix reference trajectory were further compared in Fig. 9. As shown in Fig. 8, the performance of Cartesian position tracking with neural learning was much better than without neural enhancement, regardless of fixed RCM constraints or moving RCM constraints. It, furthermore, showed a small improvement in the RCM constraint error with neural enhancement in Fig. 8.

To validate the performance with different reference trajectories, a straight line with $x_{EE,d} = x_{EE,0} + [0.013t, 0.1t, 0.0095t]$ m and a sine wave trajectory with $x_{EE,d} = x_{EE,0} + [0.1t, 0.05 * \sin(4\pi t), 0]$ m were designed and simulated with moving RCM constraints additionally to the more complex helix trajectory. The sinusoidal motion of the RCM along the z-axis had an amplitude of $[0.05]$ m with a frequency of $[5]$ Hz. The RMSE of the tracking error of Cartesian position and RCM constraints was computed for each trajectory and were depicted in Table I.

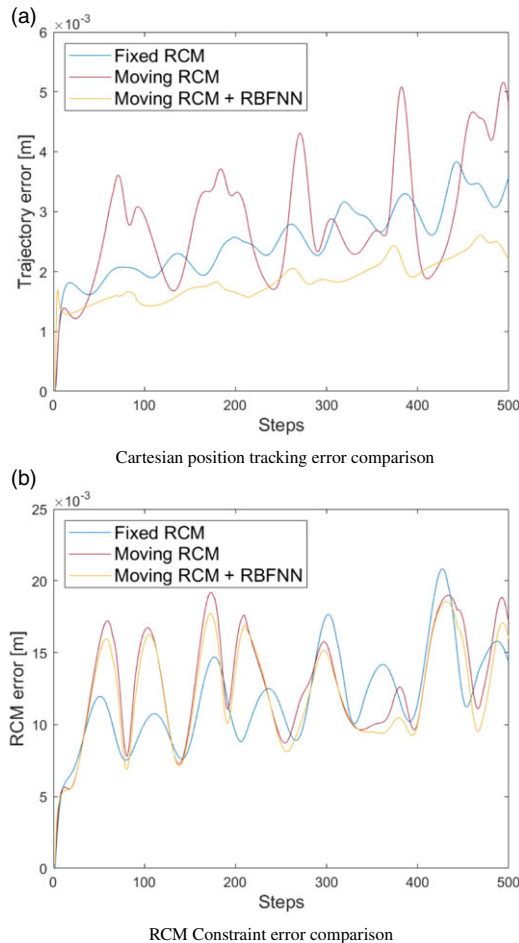


Figure 9. Error comparison under different situations.

3.1. Discussion

Figure 7 illustrated the performance of the trajectory tracking with a fixed RCM constraint and Fig. 8 depicted the performance of the trajectory tracking with a moving RCM constraint. It could be noticed that the 3-D Cartesian position tracking accuracy was worse when the RCM constraint moved using the controller without neural learning. In Fig. 9, a relatively small difference in the RCM constraints was evidenced when using the decoupled controller without neural learning and when implementing the RBFNN-based neural learning strategy. As for the moving RCM, in Table I, the RSME and maximum errors of the different simulated trajectories were summarised. As expected, the RBFNN-based control not only eliminated the trajectory errors but also reduced the RCM errors. In Fig. 8, the RSME errors were evidently lower when the neural approximation was employed. From the results, it could be concluded that the neural-learning-enhanced controller had better accuracy and stability with respect to the controller developed in ref. [10]. Hence, these results demonstrate that promising performance for the moving movement constraints using the neural enhancing solution. The idea of using an adaptive approximation to manage the disturbance generated by the moving RCM constraints can be useful for a similar application scenario more than the surgical application.

Table I. Comparison of the tracking performance between decoupled controller [10] and the neural-learning-enhanced strategy.

	Straight line		Sine		Helix	
	Null	Neural	Null	Neural	Null	Neural
Trajectory RSME	0.058	0.009	0.073	0.030	0.095	0.059
Max trajectory error	4.135	2.611	4.719	1.787	5.159	2.616
RCM RSME	0.260	0.230	0.329	0.302	0.422	0.398
Max RCM error	19.131	16.951	19.759	17.709	19.199	18.55

4. Conclusion

This paper presented a neural-learning-enhanced Cartesian admittance control of the robot with moving RCM constraints, which presents nonlinear uncertain kinematic and dynamic disturbances. The proposed control scheme integrated neural approximation-based disturbance estimation as well as admittance control for a serial robot to operate with a moving RCM constraint. Furthermore, admittance control was introduced to adapt the desired trajectory achieving compliant behaviour in its operational space. The adaptive neural approximation solution demonstrates that the adaptive could enhance the accuracy of the tracking tasks and decrease the Cartesian tracking errors of the constrained robot system. Finally, experiments were validated on a serial robot with moving RCM constraints to validate the efficacy of the presented controller. It showed improved tracking accuracy with the neural-learning-enhanced control strategy. In the near future, we will navigate more challenging issues in a clinical environment (e.g. soft tissue interaction, time delay and dead zone) in our proposed control solution [30]. System stability and tracking accuracy might not be guaranteed under these situations. Full proof of stability and robustness will be provided and validated in a real scenario. At the same time, how to tracking the real-time movement of the RCM point is also a challenging topic. The modelling of the RCM point between its changing trajectory and breathing time is helpful for the construction of stable operating performance. Furthermore, experimental validations and analysis will be performed with physical interaction on the robot's end effector, facilitating the feasibility of the proposed neural-learning-enhanced admittance control scheme. For future works, we will develop a real phantom with a moving RCM generated by simulated human breathing to evaluate the proposed algorithm.

Author contributions. Hang Su, Yunus Schmirander and Wen Qi wrote the paper. Hang Su, Yunus Schmirander, Wen Qi and Juan Sandoval conceived and designed the study. Yunus Schmirander and Sarah Elena Valderrama-Hincapie conducted the data gathering and analysis. Salih Ertug Ovur and Juan Sandoval performed statistical analyses.

Financial support. This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

Conflicts of interest. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

- [1] M. Zhou, Q. Yu, K. Huang, S. Mahov, A. Eslami, M. Maier, C. P. Lohmann, N. Navab, D. Zapp, A. Knoll and M. A. Nasser, "Towards robotic-assisted subretinal injection: A hybrid parallel-serial robot system design and preliminary evaluation," *IEEE Trans. Ind. Electron.* **67**(8), 6617–6628 (2019).
- [2] L. Chen, Z. Jiang, L. Cheng, A. C. Knoll and M. Zhou, "Deep reinforcement learning based trajectory planning under uncertain constraints," *Front. Neurobotics* **16**, 1–10 (2022).
- [3] A. M. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Curr. Opin. Urol.* **19**(1), 102–107 (2009).
- [4] H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno and E. De Momi, "Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results," *Neural Netw.* **131**, 291–299 (2020).

- [5] H. Su, W. Qi, Y. Schmirander, S. E. Ovrur, S. Cai and X. Xiong, “A human activity-aware shared control solution for medical human-robot interaction,” *Assembly Autom.* **42**(3), ahead-of-print (2022).
- [6] H. Su, A. Mariani, S. E. Ovrur, A. Menciassi, G. Ferrigno and E. De Momi, “Toward teaching by demonstration for robot-assisted minimally invasive surgery,” *IEEE Trans. Autom. Sci. Eng.* **18**(2), 484–494 (2021).
- [7] K. Guo, H. Su and C. Yang, “A small opening workspace control strategy for redundant manipulator based on RCM method,” *IEEE Trans. Contr. Syst. Technol.* **30**(6), 2717–2725 (2022).
- [8] T. Ortmaier and G. Hirzinger, “Cartesian Control Issues for Minimally Invasive Robot Surgery,” **In: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)**, vol. 1 (IEEE, 2000) pp. 565–571
- [9] P. J. From, “On the kinematics of robotic-assisted minimally invasive surgery,” *Model. Identification Contr.* **34**(2), 69–82 (2013).
- [10] J. Sandoval, G. Poisson and P. Vieyres, “Improved Dynamic Formulation for Decoupled Cartesian Admittance Control and RCM Constraint,” **In: 2016 IEEE International Conference on Robotics and Automation (ICRA)** (IEEE, 2016) pp. 1124–1129.
- [11] S. S. Ge, C. C. Hang, T. H. Lee and T. Zhang. *Stable Adaptive Neural Network Control*, vol. 13. (Springer Science & Business Media, New York, NY, USA, 2013).
- [12] X. Zhang, W. Pan, R. Scattolini, S. Yu and X. Xu, “Robust tube-based model predictive control with koopman operators,” *Automatica* **137**, 110114 (2022).
- [13] X. Zhang, J. Liu, X. Xu, S. Yu and H. Chen, “Robust Learning-Based Predictive Control for Discrete-Time Nonlinear Systems with Unknown Dynamics and State Constraints,” **In: IEEE Trans. Syst. Man Cybernet.** (2022).
- [14] 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 Trans. Automat. Contr.* **56**(7), 1672–1678 (2011).
- [15] X. Zhang, Y. Jiang, Y. Lu and X. Xu, “A receding-horizon reinforcement learning approach for kinodynamic motion planning of autonomous vehicles,” *IEEE Trans. Intell. Veh.* **7**(3), 556–568 (2022).
- [16] W. He, Y. Dong and C. Sun, “Adaptive neural impedance control of a robotic manipulator with input saturation,” *IEEE Trans. Syst. Man Cybern. Syst.* **46**(3), 334–344 (2016).
- [17] X. Chen, C. Chen, Y. Wang, B. Yang, T. Ma, Y. Leng and C. Fu, “A piecewise monotonic gait phase estimation model for controlling a powered transfemoral prosthesis in various locomotion modes,” *IEEE Robot. Automat. Lett.* **7**(4), 9549–9556 (2022).
- [18] X. Chen, K. Zhang, H. Liu, Y. Leng and C. Fu, “A probability distribution model-based approach for foot placement prediction in the early swing phase with a wearable imu sensor,” *IEEE Trans. Neur. Syst. Rehabil.* **29**, 2595–2604 (2021).
- [19] C. Ott. “Cartesian Impedance Control: The Rigid Body Case,” **In: Cartesian Impedance Control of Redundant and Flexible-Joint Robots** (Springer, Berlin, Heidelberg, Germany, 2008) pp. 29–44.
- [20] C. Yang, X. Wang, L. Cheng and H. Ma, “Neural-learning-based telerobot control with guaranteed performance,” *IEEE Trans. Cybernet.* **47**(10), 3148–3159 (2017).
- [21] Z. Li, C. Deng and K. Zhao, “Human-cooperative control of a wearable walking exoskeleton for enhancing climbing stair activities,” *IEEE Trans. Ind. Electron.* **67**(4), 3086–3095 (2019).
- [22] Z. Li, C. Xu, Q. Wei, C. Shi and C.-Y. Su, “Human-inspired control of dual-arm exoskeleton robots with force and impedance adaptation,” *IEEE Trans. Syst. Man Cybernet. Syst.* **50**(12), 5296–5305 (2018).
- [23] Z. Li, K. Zhao, L. Zhang, X. Wu, T. Zhang, Q. Li, X. Li and C.-Y. Su, “Human-in-the-loop control of a wearable lower limb exoskeleton for stable dynamic walking,” *IEEE/ASME Trans. Mechatron.* **26**(5), 2700–2711 (2020).
- [24] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE J. Robot. Automat.* **3**(1), 43–53 (1987).
- [25] A. Albu-Schaffer, C. Ott, U. Frese and G. Hirzinger, “Cartesian Impedance Control of Redundant Robots: Recent Results with the Dlr-Light-weight-arms,” **In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA ’03**, vol. 3 (IEEE, 2003) pp. 3704–3709.
- [26] S. G. Shuzhi, C. C. Hang and L. Woon, “Adaptive neural network control of robot manipulators in task space,” *IEEE Trans. Ind. Electron.* **44**(6), 746–752 (1997).
- [27] S. P. Buerger and N. Hogan, “Complementary stability and loop shaping for improved human-robot interaction,” *IEEE Trans. Robot.* **23**(2), 232–244 (2007).
- [28] L. Zhang, Z. Li and C. Yang, “Adaptive neural network based variable stiffness control of uncertain robotic systems using disturbance observer,” *IEEE Trans. Ind. Electron.* **64**(3), 2236–2245 (2017).
- [29] G. Schreiber, A. Stemmer and R. Bischoff, “The Fast Research Interface for the Kuka Lightweight Robot,” **In: IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)** (2010) pp. 15–21.
- [30] H. Zhong, Y. Wang, Z. Miao, L. Li, S. Fan and H. Zhang, “A homography-based visual servo control approach for an underactuated unmanned aerial vehicles in gps-denied environments,” *IEEE Trans. Intell. Veh.*, 1–1 (2022).