SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2021, Vol. 64 132203:1–132203:18
https://doi.org/10.1007/s11432-019-2735-6

Tracking control of redundant manipulator under active remote center-of-motion constraints: an RNN-based metaheuristic approach

Ameer Hamza KHAN[1], Shuai LI[2] & Xinwei CAO[3*]

[1]*Department of Computing, Hong Kong Polytechnic University, Hong Kong* 999077, *China;*
[2]*Department of Electronics and Electrical Engineering, Swansea University, Swansea* SA18EN, *UK;*
[3]*School of Management, Shanghai University, Shanghai* 201900, *China*

Received 25 July 2019/Revised 22 October 2019/Accepted 2 December 2019/Published online 5 February 2021

Abstract In this paper, we propose a recurrent neural network (RNN) for the tracking control of surgical robots while satisfying remote center-of-motion (RCM) constraints. RCM constraints enforce rules suggesting that the surgical tip should not go beyond the region of incision while tracking the commands of the surgeon. Violations of RCM constraints can result in serious injury to the patient. We unify the RCM constraints with the tracing control by formulating a single constrained optimization problem using a penalty-term approach. The penalty-term actively rewards the optimizer for satisfying the RCM constraints. We then propose an RNN-based metaheuristic optimization algorithm called "Beetle Antennae Olfactory Recurrent Neural Network (BAORNN)" for solving the formulated optimization problem in real time. The proposed control framework can track the surgeon's commands and satisfy the RCM constraints simultaneously. Theoretical analysis is performed to demonstrate the stability and convergence of the BAORNN algorithm. Simulations using LBR IIWA14, a 7-degree-of-freedom robotic arm, are performed to analyze the performance of the proposed framework.

Keywords tracking control, surgical robots, RCM constraints, metaheuristic optimization, recurrent neural network, RNN, redundant manipulator

Citation Khan A H, Li S, Cao X W. Tracking control of redundant manipulator under active remote center-of-motion constraints: an RNN-based metaheuristic approach. Sci China Inf Sci, 2021, 64(3): 132203, https://doi.org/10.1007/s11432-019-2735-6

1 Introduction

With recent advances in robotics and control theory, redundant robotic manipulators have gained increased research attention [1–3], which has delivered several applications for real-world scenarios [4–6]. One important application includes robot-assisted minimally invasive surgery [7, 8]. The ability to perform surgery with the assistance of a robotic arm greatly increases the precision and accuracy of the surgical process and enhances the agility of the surgeon. An important distinction that makes the robot-assisted surgery challenging, compared to other applications, is the remote center-of-motion (RCM) constraint [8,9]. RCM constraints require that the surgical tip, attached to the end-effector of the robotic arm, must pass through the small incision on the patient's body and treat the incision point as the center of motion. Figure 1 illustrates the concept of RCM constraints. To fulfill these constraints, it is necessary to use a redundant robotic arm [10], because the extra degree of freedom (DOF) provided by redundant joints allows us to enforce arbitrary constraints [11, 12], such as obstacle avoidance [13, 14] and RCM constraints, on the motion of the end-effector. However, it is well-known in the literature that incorporating additional constraints with redundancy resolution is an intricate technical challenge [15], especially because a closed-form mathematical expression for the inverse-kinematic of a redundant robotic arm does not exist [16].

* Corresponding author (email: xinweicao@shu.edu.cn)

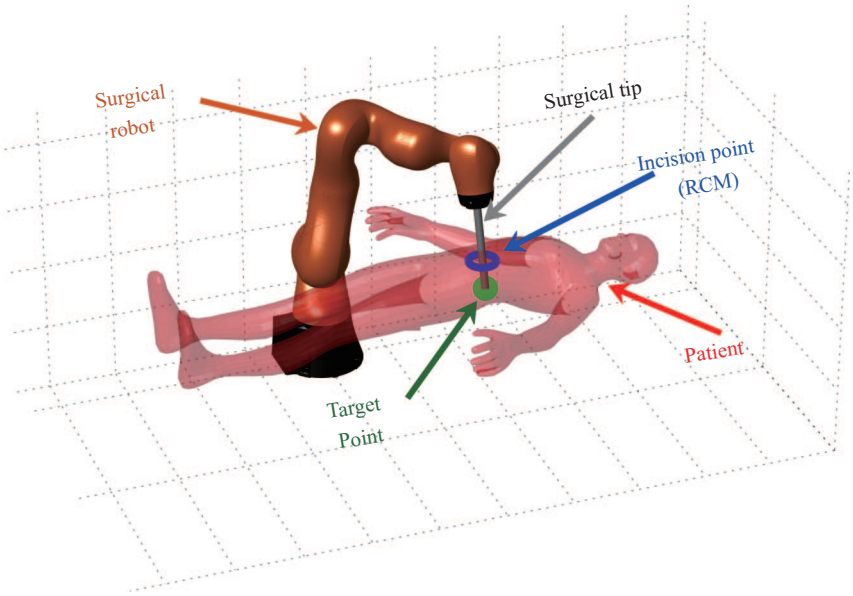

Figure 1 (Color online) Illustration of a surgical robot with a surgical tip at its end-effector. The surgical tip passes through the point of incision on the patient's body.

Traditionally, there have been two types of techniques used to enforce RCM constraints: passive techniques that use mechanical mechanisms to restrict the end-effector [17,18] and active techniques that implement software constraints using a control algorithm [8, 19, 20]. Active techniques are the primary focus of control-theory research because it provides a high degree of motion flexibility and low-cost compared with passive techniques. Ortmaier and Hirzinger [19] proposed an inverse-kinematic controller for a specific type of robotic arm. Similarly, Aghakhani et al. [8] proposed a general kinematic formulation of the RCM constraints for an arbitrary robotic arm. However, both methods considered RCM constraints in a task space, whereas control actions are only available in a joint space. The transformation of RCM constraints from task to joint spaces requires pseudo-inverse Jacobian operation, which is computationally expensive. Similarly, other optimization-driven approaches have been proposed in [21,22]. However, these approaches model RCM constraints as equality constraints in their optimization problem, which does not actively reward the optimization algorithm for satisfying the constraints. We address this issue using the penalty-term approach, which rewards the optimizer for satisfying the constraints.

In addition to RCM constraints, the literature on surgical robots has considered several other aspects. For example, Su et al. [23] formulated a null-space controller that satisfies the RCM constraints while maximizing the manipulability of the manipulators. In other words, it avoided the singularity in joint space. Similarly, other objectives can be incorporated into the control algorithm for robots. Another aspect of surgical robot development is the use of flexible materials to fabricate the end-effector [24–26]. It has gained increased attention because of the widespread popularity of flexible actuators. Using a flexible actuator as an end-effector enhances the safety of robots and assists in several surgical procedures, such as endoscopy [27], which are otherwise very difficult to perform with rigid-actuators.

Redundancy resolution [28–30] for tracking control [31–35] of robotic arms is a well-studied problem. For a given redundant robotic arm, if we specify a pose (i.e., position and orientation) of the end-effector in a Cartesian task space, infinite many angles are traversed in joint space to reach that pose. The Jacobian matrix pseudo-inverse (JMPI) [36–38] is a traditional redundancy resolution algorithm. However, traditional JMPI can only be used in conjunction with equality constraints. Most practical constraints on robotic arms (e.g., joint-angle limits, obstacle avoidance, and RCM constraints) can only be modeled as inequality constraints. Recent approaches to redundancy resolution have formulated it as a constrained optimization problem [15, 39–41]. For example, Ding et al. [42] minimized the joint torques. He et al. [43] and Wang et al. [44] used neural network-based approaches for tracking control of the redundant robotic arms. Li et al. [28] proposed a dual recurrent neural network (RNN) for solving the quadratic optimization problem for tracking control of multiple manipulators during real-time. Adaptive control techniques have also been proposed in [45–49]. These adaptive approaches have an advantage in that they adapt to variations in the manipulator model if a minor mechanical fault occurs in the

manipulator [50–52]. These algorithms adapt the parameters during runtime based on the input-output data from the manipulator. However, these actions are very computationally expensive because they require an additional loop to update parameters, and they usually require high-end embedded processors for implementation. Optimization-driven approaches have also been applied to resolve RCM constraints for redundant robotic manipulators [21, 22, 53]. The challenges being addressed in this paper include the following.

(1) Tracking control: moving the end-effector of the surgical tip as commanded by the surgeon.

(2) RCM constraints: restrict the motion of the surgical tip at the point of incision on the patient's body.

(3) Joint-angle constraints: the algorithm should generate joint-space control actions such that they do not violate the joint-angle limits.

In this paper, we leverage the ability of optimization-driven algorithms to achieve an arbitrary goal by properly formulating the optimization problem [15, 39]. We present a novel formulation for the objective function using position-level kinematic-model, which not only guarantees that the surgical tip reaches the target point as commanded by the surgeon, but also ensures that the surgical tip remains constrained to the RCM point. We introduce an inequality constraint into our optimization problem to ensure the joint-angle limits. To solve the formulated optimization problem in real-time, we propose a metaheuristic optimizer, a beetle antennae olfactory RNN (BAORNN). The proposed algorithm leverages the well-known [54–56] ability of metaheuristic optimization algorithms to efficiently solve constrained nonconvex optimization problems as compared with gradient-based optimizers [57–59]. The BAORNN algorithm is based on the beetle antennae olfactory (BAO) algorithm [60, 61], a nature-inspired algorithm motivated by the food foraging behaviors of beetles. Owing to its recent formulation, BAO has shown potential application in several practical cases [14, 62, 63]. We formulated the BAORNN algorithm using an RNN approach that will enable fast prototyping that leverages hardware acceleration, distributed processing, and software optimization offered by modern computing frameworks. Specifically, the proposed formulation allows the introduction of the concept of "virtual robots", which is a novel feature compared with traditional algorithms. These virtual robots virtually anticipate the effect of joint-space actions before moving the actual robotic arm. It should be noted that the proposed controller is formulated on a position level. Hence, it does not rely on the manipulation of the Jacobian matrix, which is a computationally expensive task. It contrasts traditional velocity-level controllers, which require the calculation of the pseudo-inverse of a Jacobian matrix at each time step. The proposed algorithm completely avoids the Jacobian manipulation by considering position-level control. The proposed algorithm is capable of handling the nonlinear kinematic model of the manipulators, because metaheuristic optimization algorithms are well-known for their ability to solve nonlinear optimization problems without estimating the derivative. The main highlights of this paper include the following.

(1) We present a novel formulation of the objective function to satisfy the RCM constraints.

(2) The objective function is formulated at the position-level, as opposed to the velocity-level control in traditional algorithms, increasing the computational efficiency of the proposed algorithm.

(3) The objective function combines tracking control and RCM constraints by using the penalty-term approach. This approach has the added advantage of rewarding the metaheuristic optimizer for satisfying the RCM constraints, thus accelerating the rate of convergence.

(4) We propose BAORNN algorithm to solve the formulated constrained optimization problem in real-time to track the time-varying reference trajectory accurately.

(5) We present the theoretical analysis on the convergence of the BAORNN algorithms.

(6) We conduct extensive simulation analysis on IIWA14 (KUKA LBR), a 7-DOF robotic arm to demonstrate the performance of the proposed algorithm.

The remainder of this paper is organized as follows. Section 2 describes the tracking-control problem with the formulation of RCM constraints and joint-angle limits. In Section 3, we formulate the BAORNN algorithm, and its theoretical analysis is presented. Section 4 outlines the simulation methodology and presents the results with a detailed discussion. Section 5 concludes this paper.

2 Problem formulation

In this section, we present the mathematical formulation of the tracking control as an optimization problem while incorporating the RCM constraints. Additionally, we discuss the joint-angle constraints

and how to account for them with the controller.

2.1 Tracking control

Consider that the surgeon wants to move the end of a surgical tip along a path inside the patient's body. The problem of computing the required trajectory in joint space, which moves the surgical tip along a designated path, considers an m-DOF robotic arm in an n-dimensional task-space. The mapping from the joint space to the task space is defined as

$$\boldsymbol{x}(t) = f(\boldsymbol{\theta}(t)), \tag{1}$$

where $f(\cdot)$ is a forward-kinematic model of the robotic arm, $\boldsymbol{x}(t) \in \mathbb{R}^n$ and $\boldsymbol{\theta}(t) \in \mathbb{R}^m$ are vectors denoting the task and joint space trajectories, respectively. For a redundant robotic arm, $m > n$. The mapping $f(\cdot)$ is usually a nonlinear function and depends on mechanical design and Denavitâ-Hartenberg (DH) parameters of the robotic arm. However, the path commanded by the surgeon exists in the task space. Therefore, we are interested in mapping from the task space to the joint space using an inverse kinematic model. Inverse mapping is based upon (1):

$$\boldsymbol{\theta}(t) = f^{-1}(\boldsymbol{x}(t)), \tag{2}$$

where $f^{-1}(\cdot)$ denotes the inverse of function $f(\cdot)$. Let us define the task-space path commanded by the surgeon as $\boldsymbol{x}_r(t)$: a reference path. To track this path, the corresponding trajectory in the joint space, $\boldsymbol{\theta}_r(t)$, must satisfy the following equation:

$$\boldsymbol{x}_r(t) = f(\boldsymbol{\theta}_r(t)). \tag{3}$$

The task of tracking control is to solve this equation for $\boldsymbol{\theta}_r(t)$. This equation could be trivially solved if a closed-form expression for $f^{-1}(\cdot)$ exists. However, for a redundant robotic arm, infinite trajectories exist in the joint-space that satisfies (3).

The redundancy resolution deals with the calculation of an optimal joint-space trajectory out of infinitely many trajectories. The redundancy resolution can be modeled as the following optimization problem:

$$\min_{\boldsymbol{\theta}(t)} \quad g_{\mathrm{tr}}(\boldsymbol{x}_r(t), \boldsymbol{\theta}(t)), \tag{4}$$

where $g_{\mathrm{tr}}(\cdot)$ denotes the tracking objective function defined as

$$g_{\mathrm{tr}}(\boldsymbol{x}_r, \boldsymbol{\theta}) = ||\boldsymbol{x}_r - f(\boldsymbol{\theta})||_2^2, \tag{5}$$

where $\boldsymbol{x}_r$ is a point on the reference path, and $\boldsymbol{\theta}$ is the joint-angle vector.

Remark 1. While formulating the objective function (5), we only consider the kinematic model of the manipulator. The kinematic model has been widely used to develop controllers in several recent studies [64,65]. Even commercially available manipulators, such as Adept Quattro 650HS [66], ABB IRB 360 [67], and the UR 10 manipulator [68], have leveraged several applications of the kinematic control algorithms.

2.2 RCM-constraints

Next, we provide a mathematical formulation of RCM constraints as a penalty term for the tracking control problem of (4). The RCM constraints require that the surgical tip remains incident to the point of incision on the patient's body. The policy to satisfy RCM constraints is based on the following principle: minimize the perpendicular distance between the RCM point and the line coincident to the surgical tip. The principle is illustrated in Figure 2. To mathematically formulate this principle, let us define the position vector of each joint in the task space as $\overrightarrow{\boldsymbol{P}}_1(\boldsymbol{\theta})$, $\overrightarrow{\boldsymbol{P}}_2(\boldsymbol{\theta})$, $\overrightarrow{\boldsymbol{P}}_3(\boldsymbol{\theta})$, ..., $\overrightarrow{\boldsymbol{P}}_m(\boldsymbol{\theta})$, where each $\overrightarrow{\boldsymbol{P}}_i(\boldsymbol{\theta}) \in \mathbb{R}^n$, $i \in \{1, 2, \ldots, m\}$, and the position vector of joints is the function of joint angles. Note that $\overrightarrow{\boldsymbol{P}}_m(\boldsymbol{\theta})$ corresponds to the end of the surgical tip and $\overrightarrow{\boldsymbol{P}}_{m-1}(\boldsymbol{\theta})$ corresponds to the start of the

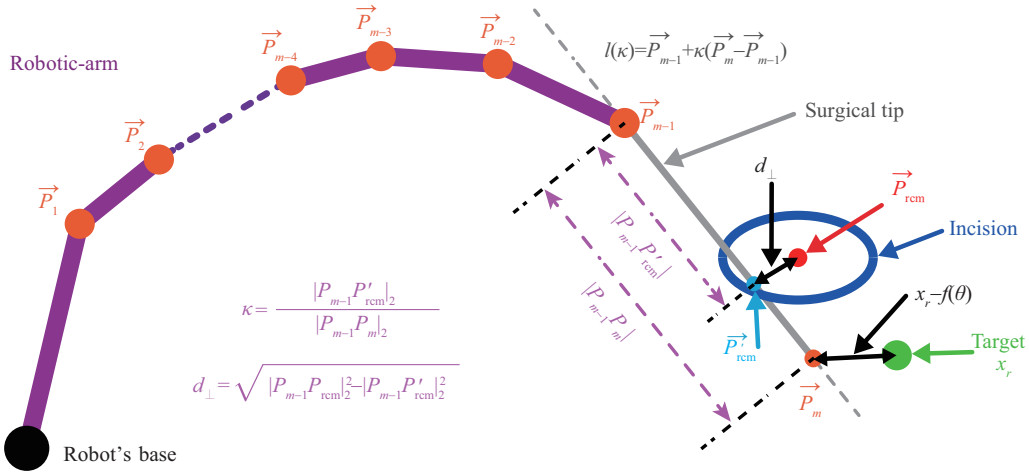

Figure 2 (Color online) Illustration of RCM constraints. Different terms defined for RCM constraints formulation in Subsection 2.2 are depicted here.

surgical tip. Based on these two points, the parametric linear equation coincident to the surgical tip can be written as

$$l(\kappa) = \overrightarrow{\boldsymbol{P}}_{m-1} + \kappa(\overrightarrow{\boldsymbol{P}}_m - \overrightarrow{\boldsymbol{P}}_{m-1}), \tag{6}$$

where dependencies are dropped for simplicity of notation. κ is the parameter used to trace the line in the n-dimensional task space. The portion of line corresponding to $0 < \kappa < 1$ overlaps with the surgical tip.

Let $\overrightarrow{\boldsymbol{P}}_{\text{rcm}}$ be the position vector of the RCM point at the location of incesion. The perpendicular distance between the surgical tip and the RCM point is given by

$$d_\perp(\boldsymbol{\theta}, \overrightarrow{\boldsymbol{P}}_{\text{rcm}}) = \sqrt{|\boldsymbol{P}_{m-1}\boldsymbol{P}_{\text{rcm}}|_2^2 - \left(\boldsymbol{P}_{m-1}\boldsymbol{P}_{\text{rcm}} \cdot \frac{\boldsymbol{P}_{m-1}\boldsymbol{P}_m}{||\boldsymbol{P}_{m-1}\boldsymbol{P}_m||_2}\right)^2}, \tag{7}$$

where $\boldsymbol{P}_{m-1}\boldsymbol{P}_{\text{rcm}} = \overrightarrow{\boldsymbol{P}}_{\text{rcm}} - \overrightarrow{\boldsymbol{P}}_{m-1}$ and $\boldsymbol{P}_{m-1}\boldsymbol{P}_m = \overrightarrow{\boldsymbol{P}}_m - \overrightarrow{\boldsymbol{P}}_{m-1}$, and the symbol $\cdot$ denotes the dot product of two position vectors. The proof of (7) follows the Pythagorean theorem. Let $\overrightarrow{\boldsymbol{P}'}_{\text{rcm}}$ denote the point on the line $l(\kappa)$ closest to the RCM point (i.e., $d_\perp(\boldsymbol{\theta}, \overrightarrow{\boldsymbol{P}}_{\text{rcm}}) = |\boldsymbol{P}'_{\text{rcm}}\boldsymbol{P}_{\text{rcm}}|_2^2$).

To enforce the RCM constraints, we must perform two tasks. First, we minimize the distance between the surgical tip and the RCM point. Second, we ensure that the point $\overrightarrow{\boldsymbol{P}'}_{\text{rcm}}$ lies on the surgical tip. For the first task, we formulate the following optimization problem:

$$\min_{\boldsymbol{\theta}(t)} \quad g_{\text{rcm}}(\boldsymbol{\theta}(t), \overrightarrow{\boldsymbol{P}}_{\text{rcm}}), \tag{8}$$

where $g_{\text{rcm}}(\cdot)$ is the penalty term for RCM constraints defined as

$$g_{\text{rcm}}(\boldsymbol{\theta}, \overrightarrow{\boldsymbol{P}}_{\text{rcm}}) = \left[d_\perp(\boldsymbol{\theta}, \overrightarrow{\boldsymbol{P}}_{\text{rcm}})\right]^2, \tag{9}$$

where $(\cdot)^2$ is used to avoid the square-root operation because the distance $d_\perp \geqslant 0$ is always set. This penalty term ensures that the line $l(\kappa)$ and the RCM point are incident. To achieve the second task (i.e., keeping $\overrightarrow{\boldsymbol{P}'}_{\text{rcm}}$ on the surgical tip), we must constrain the value of κ in (6) in the range of $[0, 1]$ as shown in Figure 2. The figure shows that when $\kappa = 0$, the surgical tip is entirely inside the body of the patient, and when $\kappa = 1$, it is outside the patient. To keep the surgical tip partially inside the patient, we require

$$\kappa > 0 \quad \text{and} \quad \kappa < 1. \tag{10}$$

The value of κ also follows from the Pythagorean theorem:

$$\kappa = \left(\boldsymbol{P}_{m-1}\boldsymbol{P}_{\text{rcm}} \cdot \frac{\boldsymbol{P}_{m-1}\boldsymbol{P}_m}{|\boldsymbol{P}_{m-1}\boldsymbol{P}_m|_2}\right)^2 \Big/ \left(|\boldsymbol{P}_{m-1}\boldsymbol{P}_m|_2^2\right). \tag{11}$$

Combining (8) and (10), we formulate a general constrained optimization problem:

$$\min_{\boldsymbol{\theta}(t)} \quad g_{\mathrm{rcm}}(\boldsymbol{\theta}(t), \overrightarrow{\boldsymbol{P}}_{\mathrm{rcm}}) \qquad \text{s.t.} \quad \kappa > 0, \quad \kappa < 1. \tag{12}$$

It is worth noting that κ is also a function of joint angles $\boldsymbol{\theta}$.

2.3 Joint-angle constraints

Another important constraint on the motion of the robotic arm is the joint-angle limitation. The joints of the robotic arm are only capable of rotating inside a mechanically allowed limit, depending on the mechanical geometry of its link and the type of actuators. To satisfy the joint-angle constraints, the joint-space trajectory must satisfy the following relation:

$$\boldsymbol{\theta}^- < \boldsymbol{\theta}(t) < \boldsymbol{\theta}^+, \tag{13}$$

where $\boldsymbol{\theta}^- \in \mathbb{R}^m$ and $\boldsymbol{\theta}^+ \in \mathbb{R}^m$ denote the lower and upper limits on the joint angles, respectively.

2.4 Unified optimization problem

Until now, we mathematically formulated three components of the problems of tracking control (4), RCM constraints (12), and joint-angle limits (13). Next, we unify them into a single constrained optimization problem as follows:

$$\begin{aligned} &\min_{\boldsymbol{\theta}(t)} \quad g_{\mathrm{tr}}(\boldsymbol{x}_r(t), \boldsymbol{\theta}(t)) + \Lambda g_{\mathrm{rcm}}(\boldsymbol{\theta}(t), \overrightarrow{\boldsymbol{P}}_{\mathrm{rcm}}) \\ &\text{s.t.} \quad \boldsymbol{\theta}^- < \boldsymbol{\theta}(t) < \boldsymbol{\theta}^+, \quad \kappa > 0, \quad \kappa < 1, \end{aligned} \tag{14}$$

where Λ is a constant parameter that controls the priority between tracking the target point and satisfying the RCM constraints. The effect of Λ on the performance of the surgical robotic arm is presented in detail in Section 4.

To simplify the notation in Section 3, we define a combined objective function $g(\cdot)$ as

$$g(\overrightarrow{\boldsymbol{P}}_{\mathrm{rcm}}, \boldsymbol{x}_r, \boldsymbol{\theta}) = g_{\mathrm{tr}}(\boldsymbol{x}_r, \boldsymbol{\theta}) + \Lambda g_{\mathrm{rcm}}(\boldsymbol{\theta}, \overrightarrow{\boldsymbol{P}}_{\mathrm{rcm}}). \tag{15}$$

Based on the above, the optimization problem can be written in complete form as

$$\begin{aligned} &\min_{\boldsymbol{\theta}(t)} \quad ||\boldsymbol{x}_r(t) - f(\boldsymbol{\theta}(t))||_2^2 \; + \; \Lambda \left(\left| \boldsymbol{P}_{m-1}\boldsymbol{P}_{\mathrm{rcm}} \right|_2^2 - \left(\boldsymbol{P}_{m-1}\boldsymbol{P}_{\mathrm{rcm}} \cdot \frac{\boldsymbol{P}_{m-1}\boldsymbol{P}_m}{||\boldsymbol{P}_{m-1}\boldsymbol{P}_m||_2} \right)^2 \right) \\ &\text{s.t.} \quad \boldsymbol{\theta}^- < \boldsymbol{\theta}(t) < \boldsymbol{\theta}^+, \quad \kappa > 0, \quad \kappa < 1. \end{aligned} \tag{16}$$

The solution to this problem is the required joint-space trajectory $\boldsymbol{\theta}_r(t)$.

3 Control design

In this section, the formulation of the BAORNN algorithm is presented with the corresponding RNN architecture. The formulated algorithm numerically solves the optimization problem (16).

3.1 BAORNN algorithm

After the formulation of the optimization problem in Section 2, we now propose the BAORNN algorithm to calculate the solution numerically in real-time. The BAORNN algorithm is based on BAO algorithm [60], a metaheuristic optimization algorithm inspired by the food-foraging behavior of a beetle. A beetle uses its olfactory sense of its antennae to find its way toward food in a previously unknown environment. At each step, the beetle senses the magnitude of smell at both antennae's locations and uses the differences of magnitude to estimate an optimal direction toward the food source. This use of olfactory senses prior to actually taking a step inspired us to introduce the concept of "virtual robots", analogous to the olfactory sense of antennae, into the BAORNN algorithm.

Suppose that, at time-step k, the joint angles of the robotic arm are $\boldsymbol{\theta}_k$. We generate a normally distributed normalized random direction vector, $\vec{\boldsymbol{b}} \in \mathbb{R}^m$, $|\vec{\boldsymbol{b}}|_2^2 = 1$. The random vector is analogous to an antenna of the beetle. As both antennae are located on opposite sides of the beetle's head, the location of their endpoints are

$$\boldsymbol{\theta}_{kL} = \boldsymbol{\theta}_k + \lambda_k \vec{\boldsymbol{b}}, \qquad \boldsymbol{\theta}_{kR} = \boldsymbol{\theta}_k - \lambda_k \vec{\boldsymbol{b}}, \tag{17}$$

where λ_k represents the antenna length, $\boldsymbol{\theta}_{kL}$ and $\boldsymbol{\theta}_{kR}$ are the locations of antennae at time-step k. However, these vectors may not satisfy the joint-angle limits and RCM constraints given in the optimization problem (16). Thus, we define a projection function:

$${}^{\Omega}\boldsymbol{\theta}_{kX} = \mathcal{P}_{\Omega}(\boldsymbol{\theta}_{kX}), \tag{18}$$

where $\mathcal{P}_{\Omega}(\cdot)$ denotes the projection function, ${}^{\Omega}\boldsymbol{\theta}_{kX}$ is the projection of $\boldsymbol{\theta}_{kX}$ on the constrained set $\Omega = \{\boldsymbol{\theta} \in \mathbb{R}^m | \boldsymbol{\theta}^- < \boldsymbol{\theta} < \boldsymbol{\theta}^+ \wedge \kappa \in [0,1]\}$, and $X \in \{L, R\}$. We define the projection function as follows:

$$\mathcal{P}_{\Omega}(\boldsymbol{\theta}) = \begin{cases} \max\{\boldsymbol{\theta}^-, \min\{\boldsymbol{\theta}, \boldsymbol{\theta}^+\}\}, & \text{if} \quad \kappa \in [0,1], \\ \boldsymbol{\theta}_k, & \text{otherwise}, \end{cases} \tag{19}$$

where κ is the same as that defined in (6). The objective function in (15) is evaluated at projected antennae's locations ${}^{\Omega}\boldsymbol{\theta}_{kX}$ using virtual robots (i.e., calculated using the mathematical model instead of actually moving the robotic arm):

$${}^{v}g_{kX} = g(\overrightarrow{\boldsymbol{P}}_{\text{rcm}}, \boldsymbol{x}_r(t), {}^{\Omega}\boldsymbol{\theta}_{kX}), \tag{20}$$

where ${}^{v}g_{kX}$ ($X \in \{L, R\}$) denotes the value of the objective function at both antennae's locations. The superscript ^{v} denotes that the value is calculated using virtual robots.

The following update rule uses the calculated values ${}^{v}g_{kX}$ at both antennae's locations to take the next step inside the joint space toward a direction where the value of the objective function is decreasing:

$${}^{\Omega}\boldsymbol{\theta}'_{k+1} = \mathcal{P}_{\Omega}(\boldsymbol{\theta}_k - \delta_k(\lambda_k)\text{sign}({}^{v}g_{kL} - {}^{v}g_{kR})\vec{\boldsymbol{b}}), \tag{21}$$

where ${}^{\Omega}\boldsymbol{\theta}'_{k+1}$ is the updated location projected on Ω. $\delta_k(\lambda_k)$ denotes the actual Euclidean step size. Note that the step size depends on the antenna length λ_k. The dependency is discussed later. The objective function is then evaluated at updated location ${}^{\Omega}\boldsymbol{\theta}'_{k+1}$, again using the virtual robot:

$${}^{v}g'_{k+1} = g(\overrightarrow{\boldsymbol{P}}_{\text{rcm}}, \boldsymbol{x}_r(t), {}^{\Omega}\boldsymbol{\theta}'_{k+1}). \tag{22}$$

The value ${}^{v}g'_{k+1}$ is then compared to the value g_k from the previous time step. If the value is improved (i.e., the new value g'_{k+1} is smaller), the robotic arm changes angles to ${}^{\Omega}\boldsymbol{\theta}'_{k+1}$. Otherwise, it remains at its current position. Mathematically, this can be written as

$$\boldsymbol{\theta}_{k+1} = \begin{cases} {}^{\Omega}\boldsymbol{\theta}'_{k+1}, & \text{if} \quad {}^{v}g'_{k+1} < g_k, \\ \boldsymbol{\theta}_k, & \text{if} \quad {}^{v}g'_{k+1} \geqslant g_k. \end{cases} \tag{23}$$

Similarly, we assign the value to g_{k+1} for use in the next iteration in (23),

$$g_{k+1} = \begin{cases} {}^{v}g'_{k+1}, & \text{if} \quad {}^{v}g'_{k+1} < g_k, \\ g_k, & \text{if} \quad {}^{v}g'_{k+1} \geqslant g_k. \end{cases} \tag{24}$$

After changing the joint angles to $\boldsymbol{\theta}_{k+1}$, the iterative procedure is repeated for next time steps. Algorithm 1 systematically lists the steps of the BAORNN algorithm.

The choice of the hyper-parameters λ_k and $\delta_k(\lambda_k)$ where k denotes the time step, effects the speed of convergence. By empirical analysis, we find that the following rules for selection of hyper-parameters provide a reasonable convergence rate:

$$\lambda_k = \sqrt{g'_k}, \quad \delta_k(\lambda_k) = \lambda_k.$$

Algorithm 1 BAORNN algorithm: tracking control & obstacle avoidance

Require: $f(\cdot)$: kinematic model of robotic arm. $\boldsymbol{x}_r(t) \in \mathbb{R}^n$: the reference trajectory of surgical tip. $\vec{\boldsymbol{P}}_{\rm rcm}$: position vector of RCM point. Λ: parameter defined in (14).

$\boldsymbol{\theta}_0 \leftarrow$ initial joint coordinates.
$k \leftarrow 0$.
$k_{\rm stop} \leftarrow$ maximum number of time-steps allowed.
while $k < k_{\rm stop}$ **do**
  Generate a normally distributed normalized random direction vector $\vec{\boldsymbol{b}} \in \mathbb{R}^m$.
  Use (17) to calculate the locations of antennae.
  Use (19) to project the locations of these antennae on the constrained set $\Omega = \{\boldsymbol{\theta} \in \mathbb{R}^m | \boldsymbol{\theta}^- < \boldsymbol{\theta} < \boldsymbol{\theta}^+ \wedge \kappa \in [0,1]\}$.
  Use (20) to evaluate the values of objective function using virtual robots as defined in (20).
  Use (21) to compute the updated location in joint space.
  Use (23) to check whether the updated location results cause any improvement.
  Move the joint of robotic arm to $\boldsymbol{\theta}_{k+1}$ and update the value of g_{k+1} using (24).
  $k \leftarrow k+1$.
end while
return $\boldsymbol{\theta}_r(t)$: an optimal trajectory in joint space.

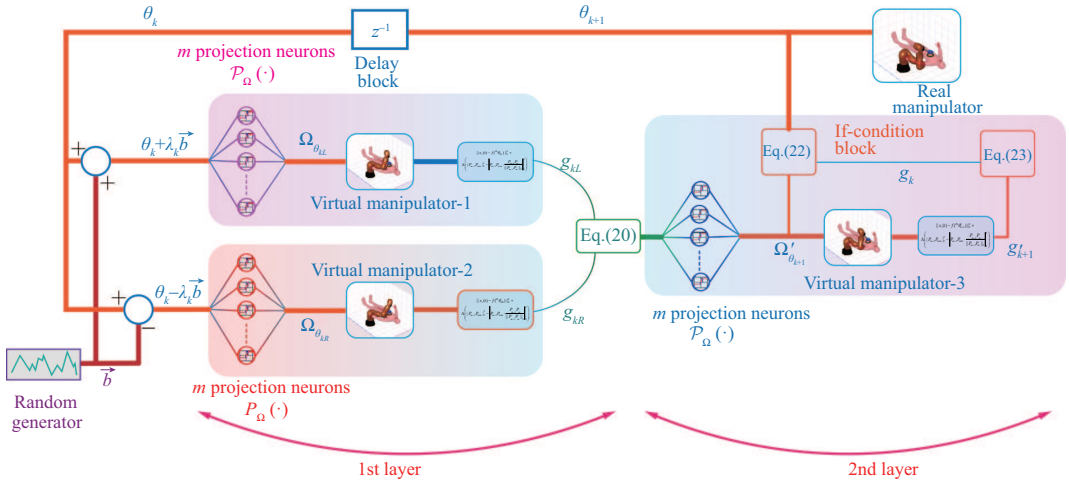

Figure 3 (Color online) Topology of the RNN of the proposed BAORNN algorithm. The diagram illustrates the working of the algorithm formulated in Section 3.

Figure 3 shows the representation of the algorithm as an RNN. The RNN topology comprises two layers with a temporal feedback connection between the output of the second layer and the input of the first layer with an input delay. The network topology comprises a total of $3m+6$ neurons, of which $3m$ are projection neurons, shown as small circles. They represent the functionality of the projection function $\mathcal{P}_\Omega(\cdot)$ as their activation function. Similarly, a total of 3 neurons, shown as curved rectangular boxes, represent the functionality of "virtual manipulator", with $f(\cdot)$ being their activation function. These blocks implement the forward kinematic model of the manipulator and are used to search the joint space for the optimal trajectory without actually moving the manipulator's joints. These blocks take joint angles as input and output the task-space coordinates of the manipulator's links. Additionally, 3 more neurons, shown as curved boxes (filled in cyan), represent the objective function $g(\cdot)$ as their activation function. The "random" block depicts a random-number generator and provides the normally distributed normalized direction vector $\vec{\boldsymbol{b}}$ for the algorithm. The diagram also contains an "if-condition" block to implement the functionality of (23) and (24).

3.2 Computational complexity

Next, the computational complexity of the BAORNN algorithm is estimated. The generation of an m-dimensional random vector $\vec{\boldsymbol{b}}$ which is the first step of the proposed algorithm, requires $\alpha_1 m$ floating-point operations, where α_1 represents the number of operations required to generate a single floating-point random number. Note that m is the number of joints in the manipulator. The second step given in (17) requires a total of $2m$ additions and $2m$ multiplications (i.e., $4m$ floating-point operations). The third step given in (18) requires the calculation of κ twice along with $4m$ comparisons. Suppose the number of floating-point operations required to calculate κ is α_2. Its value can then be determined

from (11). Therefore, the total number of operations in this step is $2\alpha_2 + 4m$. The fourth step is the most computationally expensive step of the algorithm because it requires the calculation of the objective function as given in (15). A careful analysis of (15) reveals that the number of floating-point operations required for its evaluation is linearly proportional to the number of joints in the manipulator m. This is approximately equal to $8m$. Because objective function is evaluated twice during the fourth step, it gives us a total of $16m$ floating-point operations. The fifth step given in (21) requires a total of $\alpha_2 + 4m$ floating-point operations. The sixth step again requires the calculation of the objective function (i.e., $8m$ additional floating-point operations). The last step as given in (23) and (24), just requires two comparisons. After adding the floating-point operations required in each step, the final count becomes $(\alpha_1 m + 4m + 2\alpha_2 + 4m + 16m + \alpha_2 + 4m + 8m + 2) = (\alpha_1 + 36)m + 3\alpha_2 + 2$.

The above analysis shows that, in general, the algorithm has a complexity of $O(m)$ (i.e., linear in terms of the number of joints of the manipulator). For $m = 7$ (i.e., for the case if IIWA14 manipulator), the number of floating-point operations required is in the order of hundreds. Even low-end embedded processors are capable of performing such computations in a few microseconds.

3.3 Theoritical analysis

Theorem 1. For the tracking control of a redundant robotic arm under RCM constraints, starting from an initial angle $\boldsymbol{\theta}_0$, the joint-space trajectory $\boldsymbol{\theta}_r(t)$, generated by BAORNN algorithm, is stable. That is,

$$g_{k+1} \leqslant g_k, \quad \forall k \geqslant 0, \tag{25}$$

and the values of objective function are monotonically decreasing.

Proof. See Lemma 1 of [61].

Theorem 2. For the tracking control of a redundant robotic arm under RCM constraints, starting from an initial angle $\boldsymbol{\theta}_0$, the end-effector trajectory $f(\boldsymbol{\theta}_r(t))$ is convergent to the reference trajectory $\boldsymbol{x}_r(t)$. That is,

$$f(\boldsymbol{\theta}(t)) \to \boldsymbol{x}_r(t), \quad \text{as} \quad t \to \infty. \tag{26}$$

Proof. See Theorem 1 of [61].

4 Simulation results and discussion

In this section, we present the simulation methodology used to test the performance of the BAORNN algorithm for implementing RCM constraints. A KUKA LBR IIWA14 robotic arm is used as the test-bench. A three-dimensional model of the robotic arm with a surgical tip attached to the end effector is shown in Figure 4.

4.1 Simulation methodology

An accurate computational model of the IIWA14 is provided by the MATLAB Robotic System Toolbox [69]. The computational model is a realistic testbench used to verify the accuracy of an algorithm without using a hardware platform. To test the tracking performance of the BAORNN algorithm, we define two reference trajectories (i.e., circular and linear reference) inside the patient's body to simulate the commands provided by a surgeon. These two reference trajectories depict the most common tasks performed by a surgeon during surgery. The RCM point is also specified on the patient's body in the region of the incision. The simulation setup is shown in Figure 5.

A linear reference trajectory is defined using the following equation:

$$\boldsymbol{x}_r^{\text{Line}}(t) = \overrightarrow{\boldsymbol{L}}_1 + \frac{t}{T}\left(\overrightarrow{\boldsymbol{L}}_2 - \overrightarrow{\boldsymbol{L}}_1\right),$$

where $\overrightarrow{\boldsymbol{L}}_1$ and $\overrightarrow{\boldsymbol{L}}_2$ denote the position vectors to the endpoint of the linear trajectory for $t \in [0\ T]$. Similarly, the following equation is used to define the circular reference trajectory inside the patient's body:

$$\boldsymbol{x}_r^{\text{circle}}(t) = \overrightarrow{\boldsymbol{C}} + r\cos(2\pi t/T)\overrightarrow{\boldsymbol{A}} + r\sin(2\pi t/T)\overrightarrow{\boldsymbol{B}}, \tag{27}$$

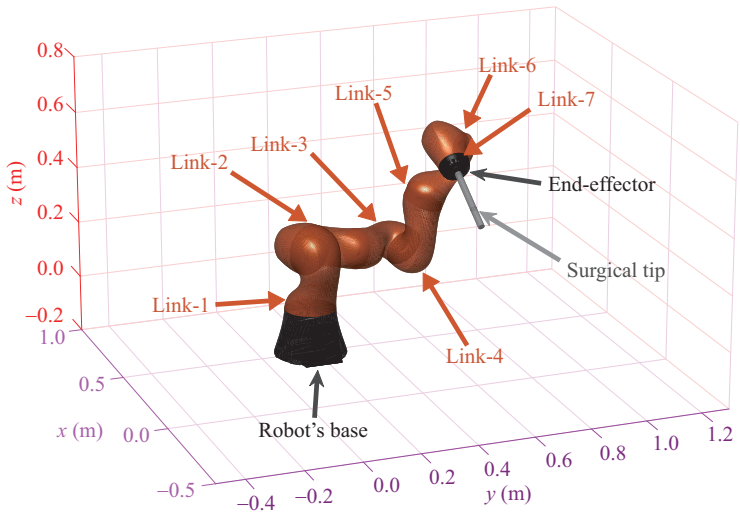

Figure 4 (Color online) Simulation model of IIWA14 7-DOF robotic arm. It has a total of seven revolution joints. The surgical tip attached to the end effector is also shown.

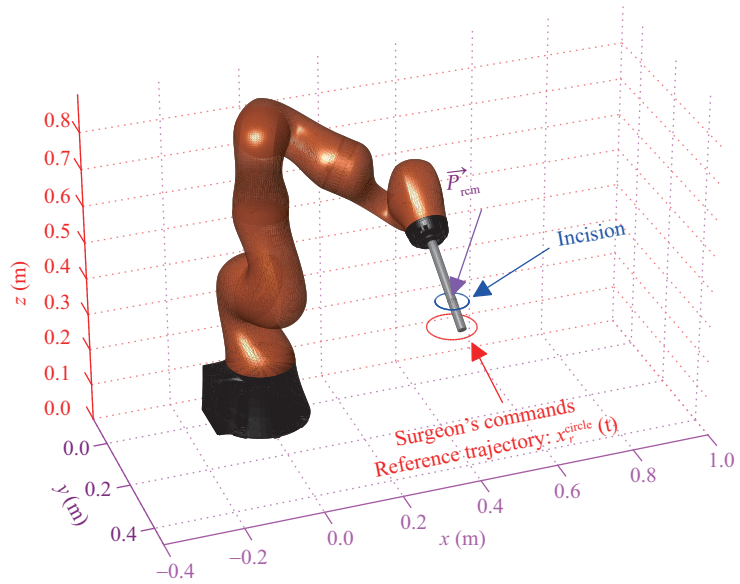

Figure 5 (Color online) Illustration of the simulation methodology. A circular reference trajectory (red circle) is shown, which is used to simulate the surgeon's commands. The point of the incision (blue circle) is also shown. The RCM point lies at the center of the incision.

where $\overrightarrow{\boldsymbol{C}}$ denotes the center of the circular trajectory, whereas $\overrightarrow{\boldsymbol{A}}$ and $\overrightarrow{\boldsymbol{B}}$ are two orthogonal unit vectors (i.e., $|\overrightarrow{\boldsymbol{A}}|_2 = 1$, $|\overrightarrow{\boldsymbol{B}}|_2 = 1$ and $\overrightarrow{\boldsymbol{A}} \cdot \overrightarrow{\boldsymbol{B}} = 0$), which define the plane of circular trajectory. r is the radius of the circular trajectory, and T denotes its time-period. The endpoints of the linear trajectory $\overrightarrow{\boldsymbol{L}}_1$ and $\overrightarrow{\boldsymbol{L}}_2$ are chosen such that it lies below the RCM point inside the patient's body. Similar considerations are made for choosing the value of the center $\overrightarrow{\boldsymbol{C}}$, radius r, $\overrightarrow{\boldsymbol{A}}$, and $\overrightarrow{\boldsymbol{B}}$.

To systematically analyze the performance of the proposed algorithm, the first set of experiments are conducted without considering the RCM constraints (i.e., setting $\Lambda = 0$ in (14)) for both reference trajectories. The second set of simulation results are compiled by setting a higher value for Λ. The Λ controls the weight of the RCM penalty term in the objective function. A higher value of Λ would ensure that the RCM constraint remains strict, but tracking performance could be degraded. The value of Λ must therefore be tuned to maintain a reasonable trade-off between tracking performance and compliance with RCM constraints. In the following discussion, the value of Λ is chosen after extensively analyzing the results of several simulations. The impact of Λ on the performance of the proposed algorithm is discussed.

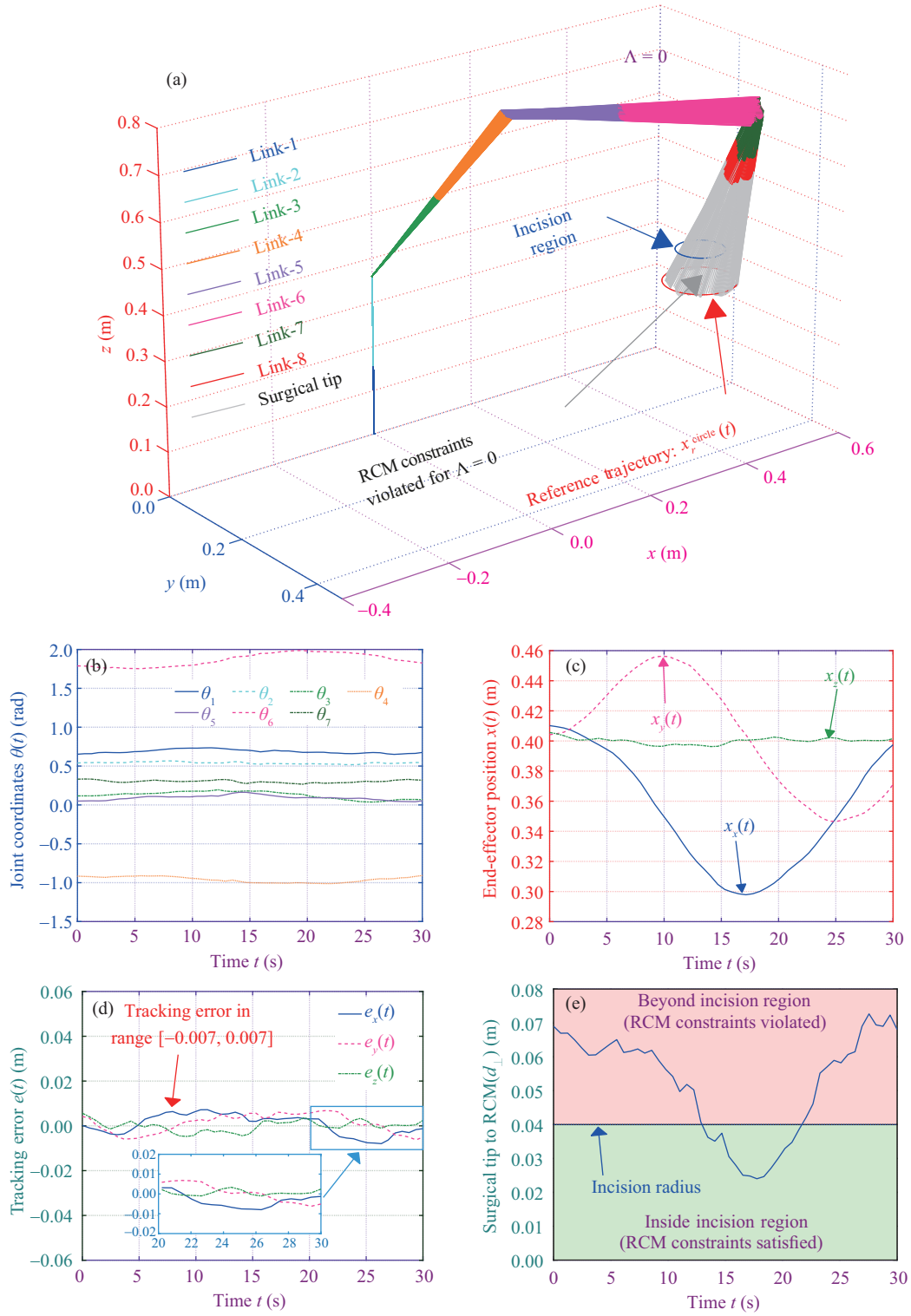

Figure 6 (Color online) Performance of the circular reference trajectory for the value of $\Lambda = 0$ (i.e., RCM constraints not enforced). It can be seen that the surgical tip goes beyond the incision region.

4.2 Trajectory tracking results

As discussed in Subsection 4.1, we first perform simulations without considering the RCM constraints. Figure 6 shows the results of the circular trajectory, and Figure 7 shows the results of the linear trajectory. The next set of experiments involves setting a high value of Λ (i.e., increasing the impact of RCM penalty

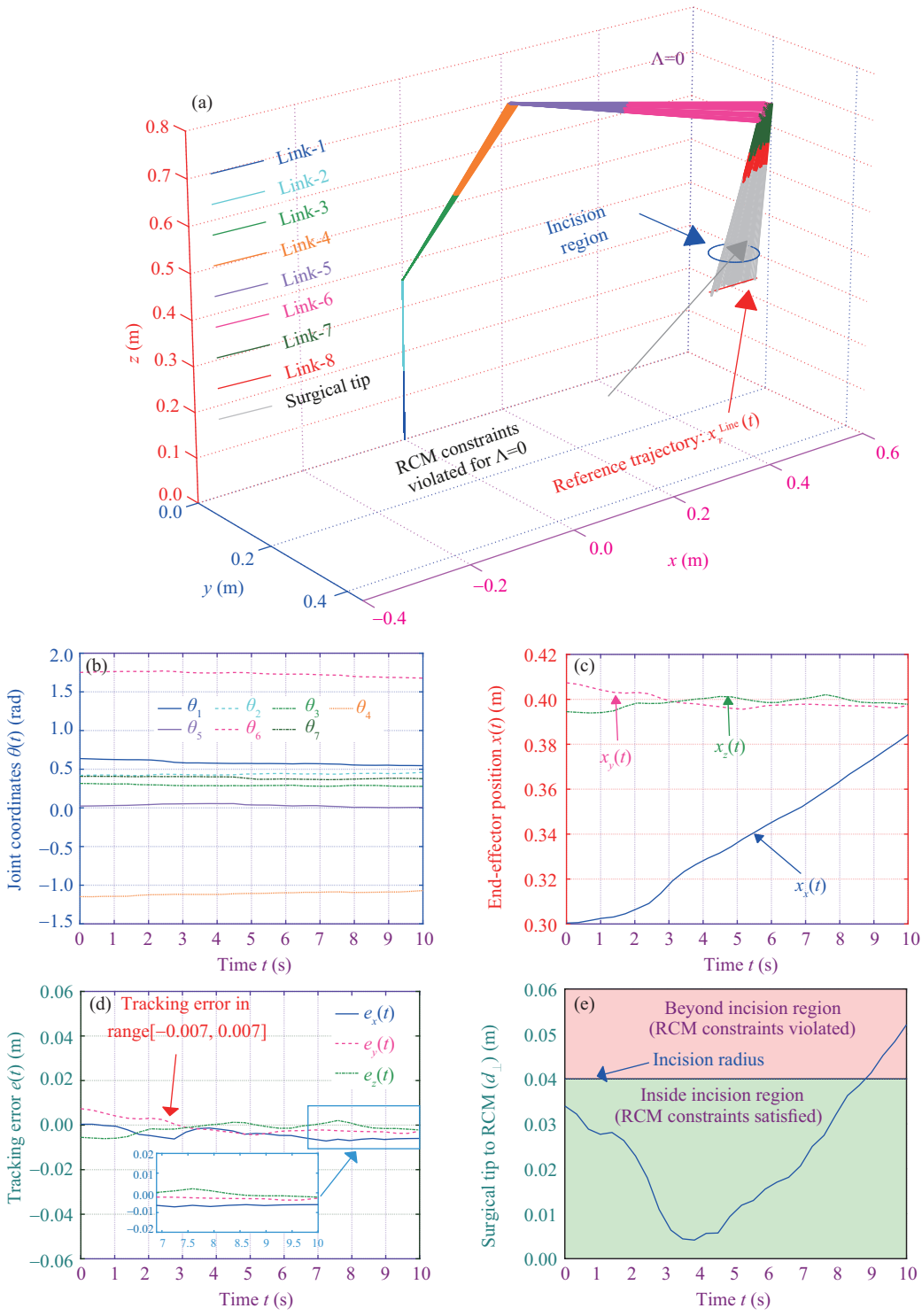

Figure 7 (Color online) Performance of the linear reference trajectory for the value of $\Lambda = 0$.

term). Figure 8 shows the results of the circular trajectory, and Figure 9 shows the results of the linear trajectory with different values of Λ.

First, we consider the results for circular trajectories. It can be seen in Figure 6(a) that, although the end of the surgical tip accurately tracks the reference trajectory, it moves outside the incision region. This is because, in this case, we set $\Lambda = 0$, which implies that the optimization problem (14) ignores the RCM penalty term. This effect can be clearly seen in Figure 6(e), where red region indicates that the distance between surgical tip and the RCM point is greater than the incision region. Because a large

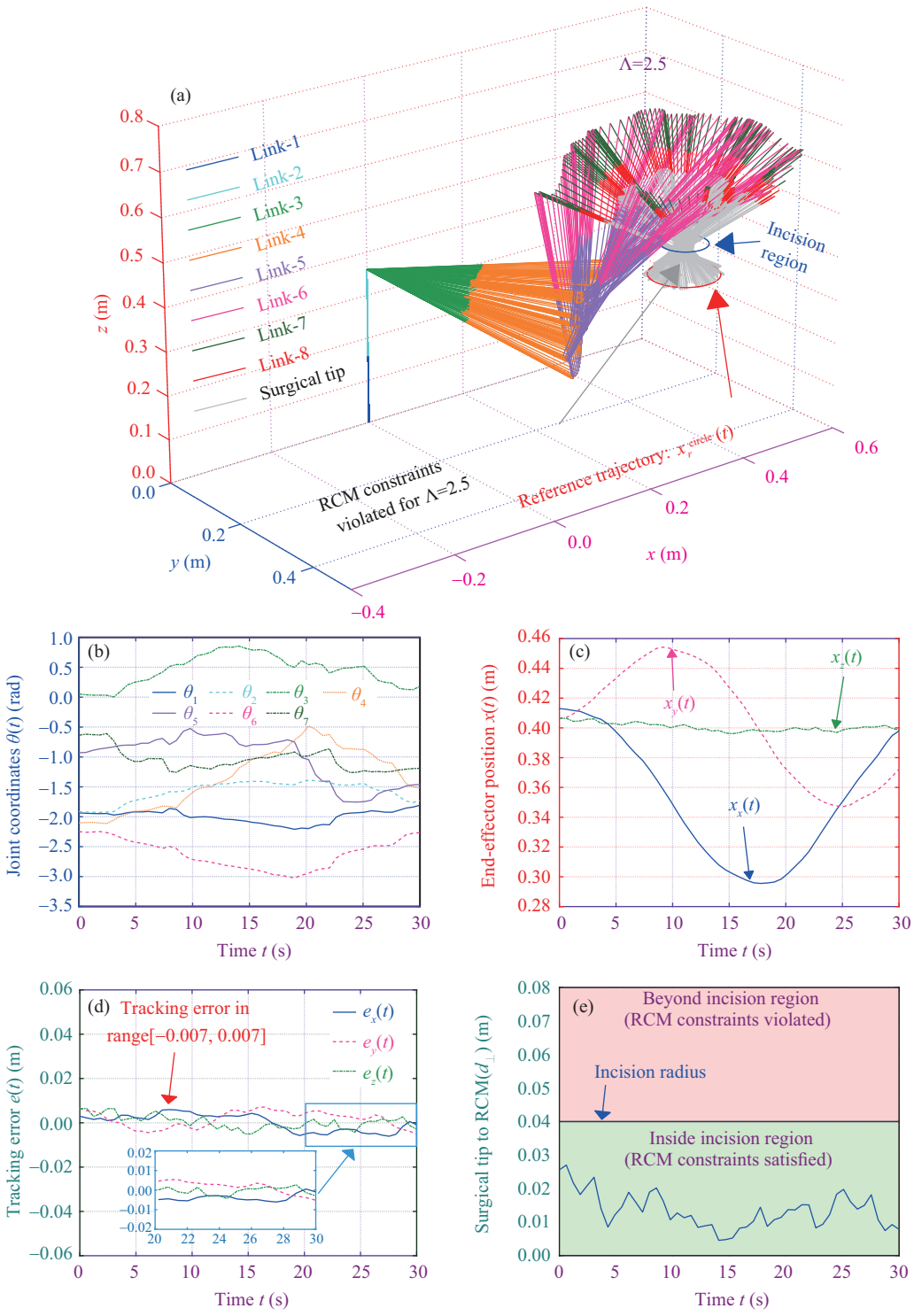

Figure 8 (Color online) Performance of the circular reference trajectory for the value of $\Lambda = 2.5$ (i.e., RCM constraints enforced, as defined in (14)). In this case, both (a) and (e) show that the RCM constraints are respected.

part of the surgical tip trajectory lies in the red region, it indicates bad performance that can potentially injure the patient. Figures 6(b) and (c) show the profiles of the joint- and task-space trajectories of the robotic arm. Figure 6(d) shows the tracking error. It can be seen that the reference tracking error shows a very unsmooth profile. Such behavior can be explained in terms of the stochastic nature of the metaheuristic optimization algorithm. Such an unsmooth response is typical of metaheuristic algorithms. We then perform the experiments for the circular reference trajectory using $\Lambda = 2.5$ (i.e., higher weight

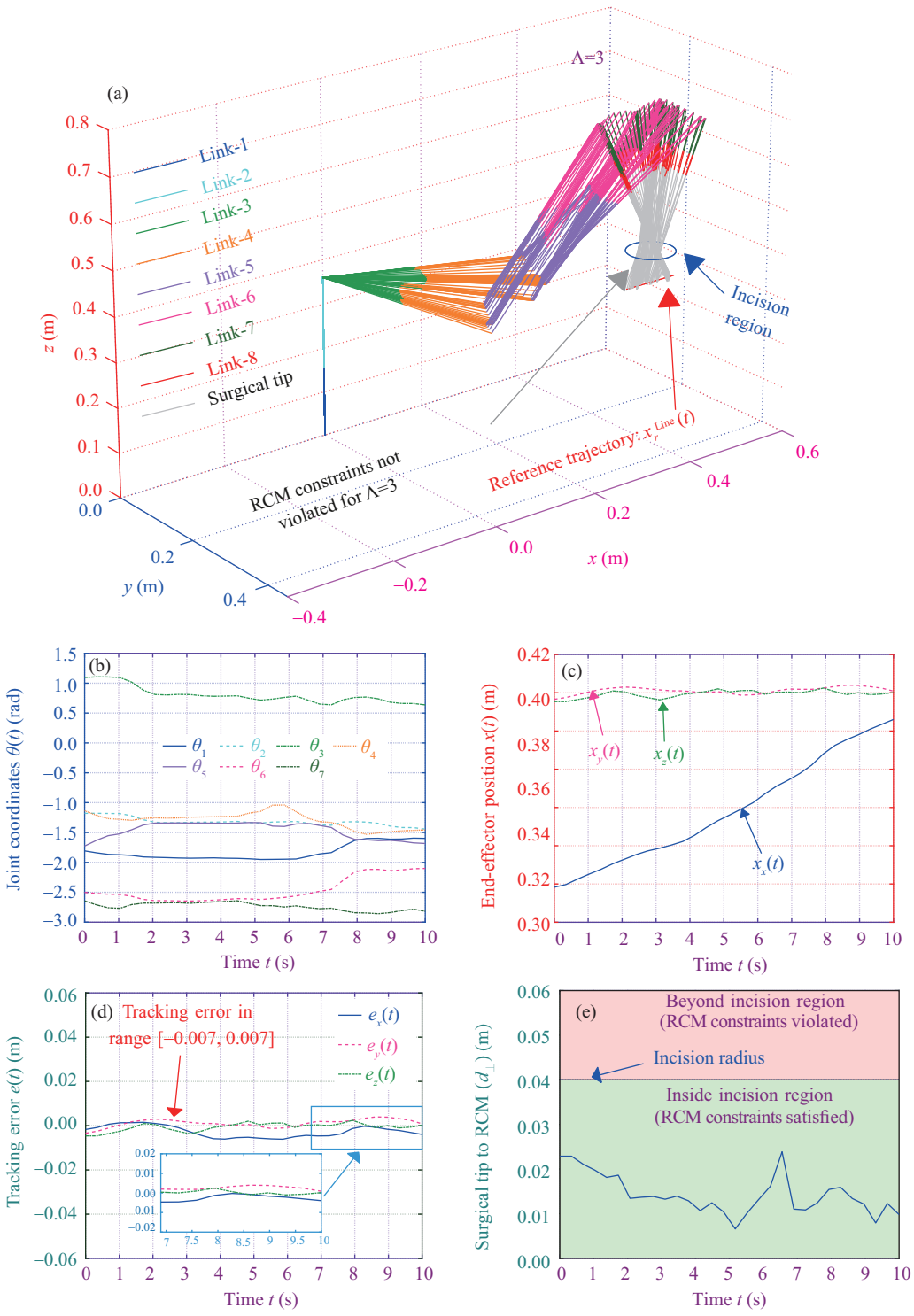

Figure 9 (Color online) Performance of the linear reference trajectory for the value of $\Lambda = 3$.

to the RCM penalty term in (14)). It can be seen in Figure 8(a) that, in this case, the surgical tip remains inside the region of the incision (blue circle) closer to the RCM point while tracking the reference trajectory. Such a response is desirable for a surgical robot, because it does not cause any harm to the patient. Similar results can be drawn from Figure 8(e), which shows that the entire profile for $d_\perp$ lies inside the green region. It is worth noting that the tracking performance is comparable in both cases, as shown in Figures 6(d) and 8(d), respectively. However, compliance with RCM constraints is clearly superior with $\Lambda = 2.5$, as depicted in Figures 6(e) and 8(e).

Figure 10 (Color online) Simulation results for (a) circular reference trajectory with $\Lambda = 2.5$ and (b) linear reference trajectory with $\Lambda = 3$.

From the results of the linear reference trajectory shown in Figures 7 and 9, similar results can be drawn. Figures 7(a) and (e) show that the surgical tip violates the boundary of the incision region with $\Lambda = 0$. Figures 9(a) and (e) show that the surgical tip complies with the RCM constraints. Similar to the circular trajectory, the tracking error shows comparable results, even in this case, as shown in Figures 7(d) and (e). To visualize the working of the surgical robot using the BAORNN algorithm, we compile a few images of the simulated model as shown in Figure 10. Figure 10(a) shows the performance for the circular trajectory with $\Lambda = 2.5$ (i.e., corresponding to Figure 8). Figure 10(b) shows the performance for the linear trajectory with $\Lambda = 3$ (i.e., corresponding to Figure 9).

5 Conclusion and future work

In this paper, we proposed a framework for the tracking control of surgical robots while satisfying RCM constraints. We first presented a formulation of a constrained optimization problem using the penalty-term approach, which unified the tracking control and RCM constraints. The optimization problem performed two tasks: minimize the distance between the surgical tip and target point; and minimize the perpendicular distance between the surgical tip and the RCM point. This formulation boosted the performance of the optimizer by actively rewarding the joint-space configuration, satisfying the RCM constraints. To get a numerical solution of the optimization problem in real time, we proposed an RNN-based metaheuristic optimizer BAORNN. The theoretical analysis of the stability and convergence of the proposed algorithm was then presented. Extensive simulation results using a realistic computational model of IIWA14, a 7-DOF robotic arm, demonstrated that the proposed algorithm could track the reference trajectory while accurately satisfying the RCM constraints.

Potential future work should build upon the contribution of the current paper by adding other optimization objectives apart from the RCM constraints. For example, the collision avoidance constraint should be considered. Although the proposed controller satisfied the RCM constraints, ensuring the safety

of the patient under normal situations is key. Unfortunately, it did not actively try to avoid collisions of the manipulator with the patient and the environment. Therefore, if the patient attempted to move or some other moving object appeared in the surroundings of the manipulator, it would pose a safety risk upon collision. Additionally, the current work assumes a constant weight Λ of the RCM penalty term in the formulation of the objective function. It currently relies on a human to tune this parameter. It would be beneficial to investigate an adaptive mechanism to adjust the value of Λ automatically. Another potential direction is found in the consideration of a flexible manipulator as the end-effector of the surgical manipulator. Flexible robots are gaining popularity because of their inherent compliance. Using them as end-effectors will enhance the safety of several surgical tasks, including endoscopy.

References

1 Yang C, Jiang Y, Li Z, et al. Neural control of bimanual robots with guaranteed global stability and motion precision. IEEE Trans Ind Inf, 2017, 13: 1162–1171
2 Liu Y J, Tong S. Adaptive NN tracking control of uncertain nonlinear discrete-time systems with nonaffine dead-zone input. IEEE Trans Cybern, 2015, 45: 497–505
3 Hu G, Gans N, Fitz-Coy N, et al. Adaptive homography-based visual servo tracking control via a quaternion formulation. IEEE Trans Contr Syst Technol, 2010, 18: 128–135
4 Yang C, Zeng C, Cong Y, et al. A learning framework of adaptive manipulative skills from human to robot. IEEE Trans Ind Inf, 2019, 15: 1153–1161
5 La H M, Dinh T H, Pham N H, et al. Automated robotic monitoring and inspection of steel structures and bridges. Robotica, 2019, 37: 947–967
6 Yang C, Peng G, Cheng L, et al. Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks. IEEE Trans Syst Man Cybern Syst, 2019. doi: 10.1109/TSMC.2019.2920870
7 Kim U, Lee D H, Kim Y B, et al. S-surge: novel portable surgical robot with multiaxis force-sensing capability for minimally invasive surgery. IEEE/ASME Trans Mechatron, 2017, 22: 1717–1727
8 Aghakhani N, Geravand M, Shahriari N, et al. Task control with remote center of motion constraint for minimally invasive robotic surgery. In: Proceedings of 2013 IEEE International Conference on Robotics and Automation, 2013. 5807–5812
9 Kuo C H, Dai J S. Kinematics of a fully-decoupled remote center-of-motion parallel manipulator for minimally invasive surgery. J Med Dev, 2012, 6: 021008
10 Jin L, Li S, Luo X, et al. Neural dynamics for cooperative control of redundant robot manipulators. IEEE Trans Ind Inf, 2018, 14: 3812–3821
11 La H M, Sheng W. Multi-agent motion control in cluttered and noisy environments. J Commun, 2013, 8: 32–46
12 La H M. Multi-robot swarm for cooperative scalar field mapping. In: Proceedings of Handbook of Research on Design, Control, and Modeling of Swarm Robotics, 2016. 383–395
13 La H M, Lim R, Sheng W. Multirobot cooperative learning for predator avoidance. IEEE Trans Contr Syst Technol, 2015, 23: 52–63
14 Khan A H, Li S, Luo X. Obstacle avoidance and tracking control of redundant robotic manipulator: an RNN-based metaheuristic approach. IEEE Trans Ind Inf, 2020, 16: 4670–4680
15 Guo D, Zhang Y. Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators. IEEE Trans Ind Electron, 2014, 61: 6903–6914
16 Tevatia G, Schaal S. Inverse kinematics for humanoid robots. In: Proceedings of IEEE International Conference on Robotics and Automation Symposia Proceedings, 2000. 294–299
17 Chen G, Wang J, Wang H. A new type of planar two degree-of-freedom remote center-of-motion mechanism inspired by the peaucellier-lipkin straight-line linkage. J Mech Des, 2019, 141: 015001
18 Nisar S, Endo T, Matsuno F. Design and optimization of a 2-degree-of-freedom planar remote center of motion mechanism for surgical manipulators with smaller footprint. Mechanism Machine Theor, 2018, 129: 148–161
19 Ortmaier T, Hirzinger G. Cartesian control issues for minimally invasive robot surgery. In: Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000. 565–571
20 Sandoval J, Poisson G, Vieyres P. A new kinematic formulation of the RCM constraint for redundant torque-controlled robots. In: Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017. 4576–4581
21 Sandoval J, Poisson G, Vieyres P. Improved dynamic formulation for decoupled cartesian admittance control and RCM constraint. In: Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016. 1124–1129
22 Yang D, Wang L, Xie Y, et al. Optimization-based inverse kinematic analysis of an experimental minimally invasive robotic surgery system. In: Proceedings of 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2015. 1427–1432
23 Su H, Shuai L, Jagadesh M, et al. Manipulability optimization control of a serial redundant robot for robot-assisted minimally invasive surgery. In: Proceedings of IEEE International Conference on Robotics and Automation, 2019. 1–6
24 Lai W, Cao L, Xu Z, et al. Distal end force sensing with optical fiber bragg gratings for tendon-sheath mechanisms in flexible endoscopic robots. In: Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018. 1–5
25 Bruno D, Calinon S, Caldwell D G. Learning autonomous behaviours for the body of a flexible surgical robot. Auton Robot, 2017, 41: 333–347
26 Calinon S, Bruno D, Malekzadeh M S, et al. Human-robot skills transfer interfaces for a flexible surgical robot. Comput Methods Programs Biomed, 2014, 116: 81–96
27 Xu K, Simaan N. Actuation compensation for flexible surgical snake-like robots with redundant remote actuation. In: Proceedings of IEEE International Conference on Robotics and Automation, 2006. 4148–4154

28 Li S, Chen S, Liu B, et al. Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. Neurocomputing, 2012, 91: 1–10

29 Jin L, Li S, La H M, et al. Manipulability optimization of redundant manipulators using dynamic neural networks. IEEE Trans Ind Electron, 2017, 64: 4710–4720

30 Yang C, Wu H, Li Z, et al. Mind control of a robotic arm with visual fusion technology. IEEE Trans Ind Inf, 2018, 14: 3822–3830

31 He W, Huang H, Ge S S. Adaptive neural network control of a robotic manipulator with time-varying output constraints. IEEE Trans Cybern, 2017, 47: 3136–3147

32 Wang H, Chen B, Lin C. Adaptive neural tracking control for a class of stochastic nonlinear systems. Int J Robust Nonlin Control, 2014, 24: 1262–1280

33 Xiao L, Li S, Lin F J, et al. Zeroing neural dynamics for control design: comprehensive analysis on stability, robustness, and convergence speed. IEEE Trans Ind Inf, 2019, 15: 2605–2616

34 Wang H, Liu X, Liu K. Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems. IEEE Trans Neural Netw Learn Syst, 2016, 27: 510–523

35 Jing L D, Zhang J F. Tracking control and parameter identification with quantized ARMAX systems. Sci China Inf Sci, 2019, 62: 199203

36 Liao B, Liu W. Pseudoinverse-type bi-criteria minimization scheme for redundancy resolution of robot manipulators. Robotica, 2015, 33: 2100–2113

37 Jin L, Zhang Y. Discrete-time zhang neural network of O(τ3) pattern for time-varying matrix pseudoinversion with application to manipulator motion generation. Neurocomputing, 2014, 142: 165–173

38 Liegeois A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans Syst Man Cybern, 1977, 7: 868–871

39 Zanchettin A M, Bascetta L, Rocco P. Achieving humanlike motion: resolving redundancy for anthropomorphic industrial manipulators. IEEE Robot Automat Mag, 2013, 20: 131–138

40 Cha S H, Lasky T A, Velinsky S A. Kinematic redundancy resolution for serial-parallel manipulators via local optimization including joint constraints. Mech Based Des Struct Mach, 2006, 34: 213–239

41 Cavallo A, Russo A, Canciello G. Hierarchical control for generator and battery in the more electric aircraft. Sci China Inf Sci, 2019, 62: 192207

42 Ding H, Tso S K. A fully neural-network-based planning scheme for torque minimization of redundant manipulators. IEEE Trans Ind Electron, 1999, 46: 199–206

43 He W, Yan Z, Sun Y, et al. Neural-learning-based control for a constrained robotic manipulator with flexible joints. IEEE Trans Neural Netw Learn Syst, 2018, 29: 5993–6003

44 Wang H, Liu P X, Bao J, et al. Adaptive neural output-feedback decentralized control for large-scale nonlinear systems with stochastic disturbances. IEEE Trans Neural Netw Learn Syst, 2019, : 1–12

45 Na J, Ren X M, Zheng D D. Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer. IEEE Trans Neural Netw Learn Syst, 2013, 24: 370–382

46 He W, Yin Z, Sun C. Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier lyapunov function. IEEE Trans Cybern, 2017, 47: 1641–1651

47 Yang C, Jiang Y, Na J, et al. Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics. IEEE Trans Fuzzy Syst, 2019, 27: 574–588

48 Na J, Jing B, Huang Y, et al. Unknown system dynamics estimator for motion control of nonlinear robotic systems. IEEE Trans Ind Electron, 2020, 67: 3850–3859

49 Wang H, Liu X P, Xie X, et al. Adaptive fuzzy asymptotical tracking control of nonlinear systems with unmodeled dynamics and quantized actuator. Inf Sci, 2018. doi: 10.1016/j.ins.2018.04.011

50 Yang C, Jiang Y, He W, et al. Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. IEEE Trans Ind Electron, 2018, 65: 8112–8123

51 Na J, Mahyuddin M N, Herrmann G, et al. Robust adaptive finite-time parameter estimation and control for robotic systems. Int J Robust Nonlin Control, 2015, 25: 3045–3071

52 Wang H, Liu P X, Zhao X, et al. Adaptive fuzzy finite-time control of nonlinear systems with actuator faults. IEEE Trans Cybern, 2020, 50: 1786–1797

53 Li M, Kapoor A, Taylor R H. A constrained optimization approach to virtual fixtures. In: Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. 1408–1413

54 Parejo J A, Ruiz-Cortés A, Lozano S, et al. Metaheuristic optimization frameworks: a survey and benchmarking. Soft Comput, 2012, 16: 527–561

55 Yang X-S. Engineering Optimization: An Introduction With Metaheuristic Applications. Hoboken: John Wiley & Sons, 2010

56 Ren Z, Li P, Fang J, et al. SBA: an efficient algorithm for address assignment in zigbee networks. Wirel Pers Commun, 2013, 71: 719–734

57 Fang J, Zhang L, Li H. Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing. IEEE Trans Image Process, 2016, 25: 2920–2930

58 Fang J, Li H B. Distributed estimation of Gauss-Markov random fields with one-bit quantized data. IEEE Signal Process Lett, 2010, 17: 449–452

59 Fang J, Shen Y, Li F, et al. Support knowledge-aided sparse Bayesian learning for compressed sensing. In: Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015. 3786–3790

60 Jiang X, Li S. BAS: beetle antennae search algorithm for optimization problems. 2017. ArXiv: 1710.10724

61 Zhang Y, Li S, Xu B. Convergence analysis of beetle antennae search algorithm and its applications. 2019. ArXiv: 1904.02397

62 Zhu Z, Zhang Z, Man W, et al. A new beetle antennae search algorithm for multi-objective energy management in microgrid. In: Proceedings of the 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2018. 1599–1603

63 Yin X, Ma Y. Aggregation service function chain mapping plan based on beetle antennae search algorithm. In: Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering, 2018. 225–230
64 Zhang Y, Li S, Zou J, et al. A passivity-based approach for kinematic control of redundant manipulators with constraints. IEEE Trans Ind Inf, 2020, 16: 3029–3038
65 Chen D, Zhang Y, Li S. Tracking control of robot manipulators with unknown models: a jacobian-matrix-adaption method. IEEE Trans Ind Inf, 2018, 14: 3044–3053
66 Wu G. Kinematic analysis and optimal design of a wall-mounted four-limb parallel Schönflies-motion robot for pick-and-place operations. J Intell Robot Syst, 2017, 85: 663–677
67 Al-Naimi I, Taeim A, Alajdah N. Fully-automated parallel-kinematic robot for multitask industrial operations. In: Proceedings of 2018 15th International Multi-Conference on Systems, Signals & Devices, 2018. 390–395
68 Menon A, Prakash R, Behera L. Adaptive critic based optimal kinematic control for a robot manipulator. In: Proceedings of International Conference on Robotics and Automation (ICRA), 2019. 1316–1322
69 Corke P I. A robotics toolbox for MATLAB. IEEE Robot Automat Mag, 1996, 3: 24–32