A Model-Based Recurrent Neural Network With Randomness for Efficient Control With Applications

Yangming Li, *Member, IEEE*, Shuai Li, *Senior Member, IEEE*, and Blake Hannaford, *Fellow, IEEE*

***Abstract*—Recently, recurrent neural network (RNN) control schemes for redundant manipulators have been extensively studied. These control schemes demonstrate superior computational efficiency, control precision, and control robustness. However, they lack planning completeness. This paper explains why RNN control schemes suffer from the problem. Based on the analysis, this work presents a new random RNN control scheme, which 1) introduces randomness into RNN to address the planning completeness problem, 2) improves control precision with a new optimization target, and 3) improves planning efficiency through learning from exploration. Theoretical analyses are used to prove the global stability, the planning completeness, and the computational complexity of the proposed method. Software simulation is provided to demonstrate the improved robustness against noise, the planning completeness and the improved planning efficiency of the proposed method over benchmark RNN control schemes. Real-world experiments are presented to demonstrate the application of the proposed method.**

***Index Terms*—Motion planning, recurrent neural networks, redundant manipulator, random neural networks, robot.**

I. Introduction

REDUNDANT manipulators demonstrated superior dexterity and are widely applied to intelligent robots. However, the redundant manipulator motion planning problem remains challenging. Actually, it has been proven that this problem is PSPACE-hard, when obstacles exist [1]. The redundant manipulator motion planning problem is to find the optimal path in the manipulator configuration space that delivers the end-effector to the desired target without breaking constraints [2]. The configuration space consists of all feasible arm joint configurations, $\boldsymbol{q}(t) \in \mathbb{R}^m$, where m denotes the degrees of freedom (DoF) of the arm. Given the manipulator model, each joint configuration corresponds to a uniquely defined end-effector pose as $\boldsymbol{r}(t) = f(\boldsymbol{q}(t))$, where $f(\cdot)$ is the kinematic model. Therefore, the configuration space is mapped to the task space [3]. Correspondingly, the n-dimensional task space contains all feasible end-effector poses, $\boldsymbol{r}(t) \in \mathbb{R}^n$.

Searching for the solution in the configuration space given a task and the kinematic model is known as the kinematic control problem [1]. The problem is often solved in the velocity space because the partial differentiation of the kinematic model linearizes and simplifies the problem to

$$\dot{\boldsymbol{r}}_t = J\dot{\boldsymbol{q}}_t \tag{1}$$

where $J = \partial f / \partial \boldsymbol{q}$ is the $n \times m$ Jacobian matrix [2].

For redundant manipulators, an infinite number of solutions satisfy (1), because the redundancy $m - n > 0$. The redundancy corresponds to the self-motion, which is useful in obstacle avoidance [3]. Existing algorithms utilizing the redundancy for obstacle avoidance can be generally divided into two categories. One utilizes "gradient projection" methods to determine the joint velocity vector $\dot{\boldsymbol{q}}_0$ that corresponds the self-motion of avoiding obstacles [3]. The components of $\dot{\boldsymbol{q}}_0$ that are in the null space of J can be selected by $(I - J^\dagger J)$. By adding the selected components to the motion that moves the end-effector $(J^\dagger \dot{\boldsymbol{r}})$, the optimal joint velocity is uniquely defined as $\dot{\boldsymbol{q}} = J^\dagger \dot{\boldsymbol{r}} + (I - J^\dagger J)\dot{\boldsymbol{q}}_0$, where $J^\dagger$ is the Moore–Penrose pseudo-inverse defined as $J^\dagger = J^{\mathrm{T}}(JJ^{\mathrm{T}})^{-1}$ [2], [4]. The second family of algorithms treats the obstacle avoidance as constraints and converts them into tasks, through which the task space is augmented to uniquely define the solution [2], [4]. Mathematically, $f_y(\boldsymbol{q}(t)) \in \mathbb{R}^s$ denotes the constraint of obstacle avoidance, where s is the dimension of the constraints, then the task is augmented as $\boldsymbol{r}_A(t) = [f(\boldsymbol{q}(t))^{\mathrm{T}}, f_y(\boldsymbol{q}(t))^{\mathrm{T}}]^{\mathrm{T}}$, and the solution can be uniquely determined when $s = m - n$, as

$$\dot{\boldsymbol{r}}_{At} = J_A \dot{\boldsymbol{q}}_t \tag{2a}$$

$$J_A = \left[J^{\mathrm{T}}, J_y^{\mathrm{T}}\right]^{\mathrm{T}} \tag{2b}$$

where $J_y = \partial f_y / \partial \boldsymbol{q}$.

Manuscript received May 25, 2018; revised July 26, 2018; accepted August 9, 2018. Date of publication September 10, 2018; date of current version April 3, 2019. This work was partially supported by NSF grant IIS-1637444 and NIH grant 5R21EB016122-02. Paper no. TII-18-1308. *(Corresponding author: Blake Hannaford.)*

Y. Li is with the College of Engineering Technology, Rochester Institute of Technology, Rochester, NY, USA 14623. The major part of this work was done when he was with the BioRobotics Lab at University of Washington, Seattle, WA, USA 98195 (e-mail: ymli@umich.edu).

S. Li is with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: shuaili@polyu.edu.hk).

B. Hannaford is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: blake@uw.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2869588

From the previous paragraph, we know the problem is mathematically well-defined. However, it is still challenging to solve the problem because the configuration spaces are often concave while obstacles exist. Neural network based methods have attracted attention recently, and many reported significant improvement of robot performance [5]–[18].

Most of these recent studies approach the problem from the control theory perspective, and focus on improving control stability [19], [20] or system adaptiveness [21], [22]. In real-world robotic applications, the solutions from the motion planning perspective are preferable because of the requirements on applicability, effectiveness, and efficiency. Among the works that approach the problem from the motion planning perspective, some of them assume obstacle-free environments. For example, Jin *et al.* proposed a recurrent neural network (RNN) based solution to optimize motions for maximizing the manipulability of redundant manipulators [23]. Li *et al.* proposed an RNN control scheme to address the cooperative control problem for distributed redundant manipulators [24]. For environments with obstacles, Zhang and Wang treated the condition of collision avoidance as an additional constraint and solved the motion planning in the velocity space [25]. Guo and Zhang extended the work into the acceleration space, and the proposed scheme guarantees minimum-acceleration-norm [26]. More discussion and comparison on neural network based motion planning can be found in [27]. Although these RNN control schemes have demonstrated improved control precision and efficiency, they model the problem as the constrained optimization; therefore, they suffer from the local minimum problem and lack the planning completeness.

This work aims to address the planning incompleteness problem of these neural network based control schemes in environments with obstacles. Being different from these existing works, this work proposed a novel random RNN control scheme, inspired by the recent finding that neural network randomness correlates with superior learning abilities [28]. Therefore, the proposed method inherits the robustness, the computational efficiency, and the effectiveness of neural network-based control schemes, while it also achieves the probabilistic planning completeness. Furthermore, through learning in the process of exploration, the proposed method balances the random exploration of environments with the heuristic search and improves the planning effectiveness. In summary, the main contributions of this work are as follows.

1) We propose a novel RNN control scheme to address the planning incomplete problem. The proposed method inherits the advantages of classical RNN control schemes, including high precision, the high efficiency, and the high robustness from RNN. Meanwhile, it addressed the local minimum problem and achieved guaranteed probabilistic planning completeness.
2) We introduce short-term memory (STM) model into the proposed method to learn environmental complexity from exploration. The proposed scheme balances the random exploration and the heuristic search to improve planning efficiency.
3) We prove the global stability, the planning completeness, and show the computational complexity of the proposed method.
4) We study the control precision, the robustness against noise and the planning completeness, and the planning efficiency of the proposed method through comparing it with other three algorithms.
5) We demonstrate the application of the proposed method in both software simulation and real-world experiments.

The rest of this paper is organized as follows: Section II presents the proposed RNN control schemes in detail. Section III presents the theoretical analyses of the proposed method. Section IV compares the proposed method with other three schemes in simulation experiment, and further verifies it in real-world experiments. The paper concluded with the discussion in Section V.

II. Random RNN for Redundant Manipulator Motion Planning

A. Solving Kinematic Control in Dual Space With RNN

RNNs refer to networks that have interlayer connections. RNNs are intrinsically parallel and capable of processing sequential or time-varying data [29]. RNNs showed superior robustness and efficiency on solving the quadratic programming problem (QP) [30], [31], thus have been used for redundant manipulator control. The kinematic control of redundant manipulators can be express in the form of QP as

$$\min_{\dot{\boldsymbol{q}}} \dot{\boldsymbol{q}}^{\mathrm{T}} W \dot{\boldsymbol{q}} + \boldsymbol{c}^{\mathrm{T}} \dot{\boldsymbol{q}} \tag{3a}$$

$$\text{s.t. } \dot{\boldsymbol{r}}_d = J\dot{\boldsymbol{q}} \tag{3b}$$

$$\boldsymbol{q} \in \Omega \tag{3c}$$

where W and $\boldsymbol{c}$ are two weighting factors that are $m \times m$ real symmetric matrix and a real-valued m-dimensional vector, respectively, $J = \partial f/\partial \boldsymbol{q} \in \mathbb{R}^{n\times m}$, $\dot{\boldsymbol{r}}_d = \partial \boldsymbol{r}(t)/\partial t \in \mathbb{R}^n$, $\dot{\boldsymbol{q}} = \partial \boldsymbol{q}/\partial t$, $\boldsymbol{q} \in \Omega$, and $\Omega \subset \mathbb{R}^m$ denotes the configuration space of the redundant manipulator.

To simplify the QP problem, we project it into its dual space, through designing the Lagrange multiplier, $\lambda \in \mathbb{R}^n$, to correspond to the constraint (3b). The problem described in (3) is simplified to

$$L(\dot{\boldsymbol{q}}, \lambda) = \dot{\boldsymbol{q}}^{\mathrm{T}} W \dot{\boldsymbol{q}} + \boldsymbol{c}^{\mathrm{T}} \dot{\boldsymbol{q}} + \lambda^{\mathrm{T}}(\dot{\boldsymbol{r}}_d - J\dot{\boldsymbol{q}}). \tag{4}$$

The Karush–Kuhn–Tucker condition ([32, Ch. 5.5.3]) ensures that the solution to (4) equals to the solution to the following equation:

$$\dot{\boldsymbol{q}} = P_\Omega\left(\dot{\boldsymbol{q}} - \frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right)$$

$$\dot{\boldsymbol{r}}_d = J\dot{\boldsymbol{q}} \tag{5}$$

where $P_\Omega(x) = \text{argmin}_{y\in\Omega}||y - x||$ is a projection function from domain Ω' to Ω, where $x \in \Omega'$ and $y \in \Omega$.

Equation (5) naturally matches the neural dynamics of a projected RNN (6), and it can be proved that the equilibrium of the

network equals to the optimal solution of the system described in (3) [33], [34]

$$\epsilon\ddot{\boldsymbol{q}} = -\dot{\boldsymbol{q}} + P_\Omega\left(\dot{\boldsymbol{q}} - \frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right) \tag{6a}$$

$$\epsilon\dot{\lambda} = \dot{\boldsymbol{r}}_d - J\dot{\boldsymbol{q}} \tag{6b}$$

where $\epsilon > 0$ is a scaling factor. These types of projected RNN has simple architecture as they are a single layer neural network, in which neurons are fully connected. The output of the neural network goes to a projection function as explained in (9).

B. Improve Control Precision and Robustness by Closing the Loop of RNN Control Scheme

The control precision and the robustness against noise are critical in motion planning, because 1) the control precision defines the minimum safety distance to obstacles, which shrinks the configuration space; and 2) high control precision and robustness against noise ensure the successful execution of planning results. Classical RNN control schemes have demonstrated improvement on the control precision, the robustness, and the efficiency of redundant manipulator control, compared with their Jacobian inversion based equivalents [35], we identified the problem of error accumulations in these control schemes and designed a new RNN control scheme to overcome the error accumulation problem through introducing a new optimization target. We first introduced the new design and the proof of the effectiveness will be presented in Section III-A.

Intuitively, feeding the tracking errors, $\boldsymbol{e} = \boldsymbol{r}_d - \boldsymbol{r}$, back into the optimization target forces RNN to minimize the errors. Correspondingly, the new optimization target can be designed as

$$\min_{\dot{\boldsymbol{q}}} \left(\dot{\boldsymbol{q}}^{\mathrm{T}}\dot{\boldsymbol{q}} + k\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}\right) \tag{7}$$

where $k > 0$ is a weighting factor and $\boldsymbol{r}$ is the end-effector position [36]. Equation (7) will direct RNNs to minimize the magnitude of $\boldsymbol{e}$, because $\dot{\boldsymbol{q}}^{\mathrm{T}}\dot{\boldsymbol{q}} \geq 0$ and $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} \geq 0$. Control precision and robustness of RNNs based on (7) are analyzed in theory in Section III-A, and empirically compared in Section IV-A.

With the new defined optimization function (7), (4) becomes

$$L(\dot{\boldsymbol{q}}, \lambda) = \dot{\boldsymbol{q}}^{\mathrm{T}}\dot{\boldsymbol{q}} + k\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} + \lambda^{\mathrm{T}}(\dot{\boldsymbol{r}}_d - J\dot{\boldsymbol{q}}). \tag{8}$$

The constraints of joint physical limits can be fulfilled by designing a proper projection function. A projection function has the form of

$$P_\Omega(\boldsymbol{x}) = \begin{cases} \boldsymbol{d}^- & \text{for} \quad x \leq \boldsymbol{d}^-, \\ \boldsymbol{x} & \text{for} \quad \boldsymbol{d}^- < \boldsymbol{x} < \boldsymbol{d}^+, \\ \boldsymbol{d}^+ & \text{for} \quad \boldsymbol{d}^+ \leq \boldsymbol{x}, \end{cases} \tag{9}$$

where we design the boundary conditions as

$$\begin{cases} \boldsymbol{d}^- = \max\left(-c_1(\boldsymbol{q} - \boldsymbol{q}^-), \boldsymbol{w}^-\right), \\ \boldsymbol{d}^+ = \min\left(-c_2(\boldsymbol{q} - \boldsymbol{q}^+), \boldsymbol{w}^+\right). \end{cases} \tag{10}$$

Equation (10) ensures that the joint limits are met. This is because when joints approach the upper ($\boldsymbol{q}^+$) and the lower ($\boldsymbol{q}^-$) bounds of joint limits, the magnitudes of the two terms, $-c_1(\boldsymbol{q} - \boldsymbol{q}^-)$ and $-c_2(\boldsymbol{q} - \boldsymbol{q}^+)$ decrease to zeros, as the speed is regulated by two positive scaling factors, c_1 and c_2. These two scaling factors can be empirically tuned to clamp the joint accelerations, which improves the control precision in mechanics with non-negligible inertial, for example, cable-driven manipulators. w^+ and w^- are the upper and the lower bounds of the joint speed. They can be used to clamp movement speeds, which improve manipulator performance through regulating motion patterns [36].

By substituting (8), (9), and (10) into (6), we have a new RNN control scheme that meets the constraints of joint limits and joint speeds in obstacle free environments.

C. Obstacle Avoidance and Complete Motion Planning with RNN

Because of the flexibility of RNN architecture, the obstacle avoidance problem can be addressed by both *augmenting the task space*, and *converting the obstacle avoidance constraints into bounding conditions* [25], [26]. The former adopts the scheme explained in Section II-A. To be more specific, the Jacobian and the task are augmented by the obstacle avoidance constraints, as explained in (2). These methods require the dimension of the extra tasks, s, to meet the condition $s \leq m - n$. The latter does not have such limitation. In order to explain it, we first introduce the concept of "critical points," as indicated by green points in Fig. 2. The critical points are the points on the manipulator, whose distances to obstacles equal or are smaller than the safety threshold. Let us denote the critical points as $\boldsymbol{r}_{o,i}$, and denote the nearest points on the obstacle by $\boldsymbol{o}_{o,i}$, where i indicates the ith critical point. Then, the algorithms in the latter category avoid a collision through exerting "escaping velocity" on the critical points.

The escaping velocity denotes the velocity that moves the critical points away from the obstacles

$$\dot{\boldsymbol{r}}_{\mathrm{o}} = a(\boldsymbol{r}_o - \boldsymbol{o}_o) \tag{11}$$

where a is a semipositive scaling factor. Any critical point velocity between zero and the escaping velocity will avoid a collision. It is obvious that this method converts the constraints of obstacle avoidance, $\dot{\boldsymbol{r}}_{o,i} = a(\boldsymbol{r}_{o,i} - \boldsymbol{o}_{o,i})$, into a closed set in the configuration space, and in the RNN control scheme, this set bounds neuron activities

$$\begin{cases} \left[\boldsymbol{d}_i^-, \boldsymbol{0}\right], & \text{for} \quad \dot{\boldsymbol{r}}_{o,i} < 0, \\ \left[\boldsymbol{0}, \boldsymbol{d}_i^+\right], & \text{otherwise.} \end{cases} \tag{12}$$

Because in this scheme, the obstacle avoidance does not explicitly "consumed" the manipulator redundancy, algorithms based on this scheme can deal with the infinite number of constraints and are more flexible.

From (12), we know that in classical RNN control schemes, the target globally "drags" the end-effector to move toward it, while obstacles "push" the critical points to move away. Intuitively, this causes the local minimum problem and the planning is guaranteed to be complete. This problem is visualized in Fig. 2. In the figure, RNN schemes failed to drive the manipulator from the position indicated by the solid red circle, to the target indicated by the dashed red circle, because the obstacle made the configuration concave and trapped the manipulator.

The mathematical explanation of this local minimum problem is in Section III-B.

In order to address this problem, the manipulator needs the ability to "jump" out of local minimum, and the neural network randomness can be utilized because a recent study demonstrated that the randomness correlated with superior learning abilities has been used to demonstrate human-level concept learning ability [28].

In order to grant RNNs the randomness, we replace the global attraction from the target with random attractions. The random attractions, $\boldsymbol{r}_{\text{random}} \in \mathbb{R}^{\text{n}}$, is designed as

$$\dot{\boldsymbol{r}}_{\text{d}} = g \cdot (\boldsymbol{r}_{\text{random}} - \boldsymbol{r}) \tag{13}$$

where g is a nonzero weighting factor that regulates the attractions.

D. Balancing Random Exploration and Heuristic Search with STM Model

The local minimum problem was due to the concave configuration space. In sparse environments, the heuristic search has a high probability of success and is efficient, therefore, is preferable. In environments with complex obstacles, the heuristic search is easy to fail and the random exploration is the key to success. However, the full knowledge of the environments is often not available, and it is known, calculating the configuration space for the environment is computational expensive [2]. In this work, we balance the heuristic search and the random exploration, through learning the environmental complexity online.

This environmental complexity can be learned by probabilistic methods, such as the classical theory of probability or Bayesian methods. The classical theory of probability is simple in concept as it counts the percentage of failed exploration, but these algorithms prone to bias [37]. For example, in a scenario that the target is inside a corridor, when the manipulator's initial position is unluckily located, the manipulator reaches obstacles before it reaches the targets. Therefore, the algorithm degenerates to pure random exploration because collisions are frequent. However, we know the configuration space is wide and nicely connected. Bayesian methods do not suffer from this problem, because of the existence of the prior. However, the prior depends on the configuration space and is difficult to achieve [38], [39].

Shunting STM model is a solution to this online learning problem. STM is a robust and powerful tool that describes how living creatures adapt to environmental changes and has been widely used in addressing the system adaptiveness problem [40]. STM was derived from the additive STM model, which is mathematically described as [40], [41]:

$$\frac{dx_i}{dt} = -A_i x_i + \sum_{j=1}^{N} f_j(x_j) B_{ji} z_{ji}^{(+)} - \sum_{j=1}^{N} g_j(x_j) C_{ji} z_{ji}^{(-)} + I_i$$

where $-A_i x_i$ is a passive decay; $\sum_{j=1}^{N} f_j(x_j) B_{ji} z_{ji}^{(+)}$ is the positive feedback, $\sum_{j=1}^{N} g_j(x_j) C_{ji} z_{ji}^{(-)}$ is the negative feedback, and I_i is the input inspiration (see detailed explanation in [40]).

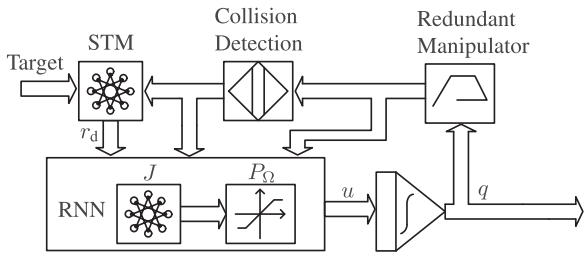

Fig. 1. Architecture of the proposed random RNN. The proposed method consists a single layer RNN and a single cell STM. The STM learns from exploration and controls the RNN to generate precise, efficient, and robust motion planning results for redundant manipulators.

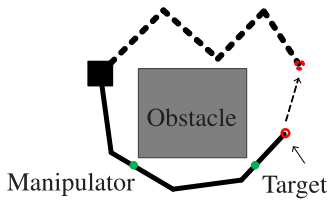

Fig. 2. Explanation to the reason that classical RNN control schemes suffer from the local minimum problem and lack of planning completeness. In the simple two-dimensional environment, classical RNN control schemes fail to find a valid pathway to move from the pose indicated by the solid line to the pose indicated by the dashed line because the configuration space is concave due to the existence of the obstacle.

The intuition is easy to understand through an analogy of "pain" from the collision. If a collision was fed into an STM, the output of the STM can be seen as the memory of "pain." In the beginning, the STM has no pain and lean to the heuristic search. While a collision happens, the neural activity of the STM increases and it tends to random exploration. The "pain" decays with time, and it tends to perform the heuristic search again. In the extreme case that collisions happen very time, STM is still able to perform the heuristic search as the decay is continuous and the neuron activities are guaranteed to be unsaturated. From the description, we know that unlike to some "windowed" methods, STM does not explicitly have a fixed term of memory; instead, it nonlinear decays memory and the old ones will be "flashed" by new ones.

A single cell STM is the simplest model that meets our needs, which can be mathematically described as

$$\frac{dx}{dt} = -Ax + (1 - x)\left(I + \sum_{j=1} w\,[x_j]^{+}\right) \tag{14}$$

where parameters A denotes the passive decay rate; variable x is the neural activity and $x \in [0, 1]$ is guaranteed; the excitatory inputs (inspirations) to neurons are $I + \sum_{j=1} w\,[x_j]^{+}$, I is the excitatory input from the exploration, it can simply be 0 for no collision and 1 for collision; w_j denotes weights of the self-excitatory. The neural activities of STM, RRNN_{STM}, will be used to balance the random search with the heuristic search. The proposed RNN control scheme is explained in Fig. 1 and in Algorithm 1.

In Algorithm 1, RRNN_{RNN} denotes the control scheme described in (8), (9), (10), and (12), and RRNN_{STM} is mathematically explained in (14).

Algorithm 1: Proposed Random RNN Control Scheme.

Input: Target ($\boldsymbol{r}_{\mathrm{T}}$), Manipulator Start Position ($\boldsymbol{r}_0$)
Output: Sequence of Joint States (C)
Init : $\boldsymbol{r}_s$=$\boldsymbol{r}_0$, $\boldsymbol{r}_d$=$\boldsymbol{r}_0$, $T_s = \{[\boldsymbol{r}_s, \boldsymbol{r}_d, \emptyset]\}$, RRNN†, re‡
if $\left(\text{Neural activity of RRNN}_{\text{STM}} > rand\,(0,1)\right)$ **then**
  $\boldsymbol{r}_d$ = random point in the task space
else
  $\boldsymbol{r}_d = \boldsymbol{r}_{\mathrm{T}}$
end if
Select the reached goals($\boldsymbol{r}_d$ in T_s), which is closes to $\boldsymbol{r}_d$, use it as $\boldsymbol{r}_s$
RRNN_{RNN} plan motion for given $\boldsymbol{r}_s$ and $\boldsymbol{r}_d$, and produce a sequence of commands $c_{s,d}$
while re>0 **do**
  $\boldsymbol{r}_r$ = end effector position
  if $(\boldsymbol{r}_d = \boldsymbol{r}_r)$ **then**
    if $(\boldsymbol{r}_d == \boldsymbol{r}_{\mathrm{T}})$ **then**
      Back tracing control sequence $C = \Sigma c_{s,d}$
      Break
    end if
  else
    $\boldsymbol{r}_d = \boldsymbol{r}_r$
  end if
  Feed collision into RRNN_{STM}
  Insert $[\boldsymbol{r}_s, \boldsymbol{r}_d, c_{s,d}]$ into T_s
  $re = re - 1$
end while
return C

†The architecture of RRNN is explained in Fig. 1, which consists RRNN_{RNN} and RRNN_{STM}.
‡A big integer to control the maximum allowed exploration time.

III. Theoretical Analyses

A. Precision and Stability

Section II-B states that the proposed RNN control scheme overcomes the error accumulation and improves the control precision. The classical RNN control scheme is

$$\begin{aligned}\lambda &= \lambda_0 + \frac{1}{\epsilon}\int (\dot{\boldsymbol{r}}_{\mathrm{d}} - J\dot{\boldsymbol{q}})\,dt,\\ &= \lambda_0 + \frac{1}{\epsilon}\left(\boldsymbol{r}_{\mathrm{d}} - \boldsymbol{r}_{d0} - \int J\dot{\boldsymbol{q}}\,dt\right)\end{aligned} \tag{15}$$

where λ_0 and $\dot{\boldsymbol{r}}_{d0}$ denotes the value of λ and $\dot{\boldsymbol{r}}_d$ at $t = 0$, respectively.

By replacing λ in (6) with (15), we have

$$\begin{aligned}\epsilon\ddot{\boldsymbol{q}} &= -\dot{\boldsymbol{q}} + P_\Omega\left(J^{\mathrm{T}}\left(\lambda_0 + \frac{1}{\epsilon}\left(\boldsymbol{r}_{\mathrm{d}} - \boldsymbol{r}_{d0} - \int J\dot{\boldsymbol{q}}\,dt\right)\right)\right)\\ &= -\dot{\boldsymbol{q}} + P_\Omega\left(J^{\mathrm{T}}\left(\lambda_0 + \frac{1}{\epsilon}\left(\boldsymbol{r}_{\mathrm{d}} - \boldsymbol{r}_{\mathrm{d0}} - (\boldsymbol{r} - \boldsymbol{r}_0)\right)\right)\right).\end{aligned} \tag{16}$$

From (16), we know that at any given arbitrary time point $t = 0$, the errors accumulates with time as long as errors $\boldsymbol{e}_0 = \boldsymbol{r}_{d0} - \boldsymbol{r}_0 \neq 0$. With the proposed scheme (7), for any time point t, its initial error accumulation, $\boldsymbol{e}_0 = \boldsymbol{r}_{d0} - \boldsymbol{r}_0 \neq 0$, is fed back into the controller as

$$\epsilon\ddot{\boldsymbol{q}} = -\dot{\boldsymbol{q}} + P_\Omega\left(J^{\mathrm{T}}\left(\frac{1}{\epsilon}(\boldsymbol{r}_{\mathrm{d}} - \boldsymbol{r})\right)\right). \tag{17}$$

We can see that the accumulated errors are canceled out in (17), thus the proposed scheme does not suffer from the problem anymore.

In order to prove the global stability of the proposed method, we define the Lyapunov function as $V = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}/2$, where $\boldsymbol{e} = \boldsymbol{r}_{\mathrm{d}} - \boldsymbol{r}$ denotes the tracking errors [used as feedback in (7)]. Then, from (3a) and (5), we know $\dot{\boldsymbol{r}} = JP_\Omega(\frac{1}{\epsilon}J^{\mathrm{T}}(\boldsymbol{r}_d - \boldsymbol{r}))$. Because $\boldsymbol{r}_d$ is the goal, which is a constant, we have

$$\dot{\boldsymbol{e}} = -JP_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right). \tag{18}$$

By substituting (18) into the defined Lyapunov function, we have

$$\dot{V} = \boldsymbol{e}^{\mathrm{T}}\dot{\boldsymbol{e}} = -\boldsymbol{e}^{\mathrm{T}}JP_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right). \tag{19}$$

Because the defined projection function is a saturation function as $P_\Omega(x) = \text{argmin}_{y\in\Omega}||y - x||$, we have

$$\|P_\Omega(x) - x\|^2 \leq \|y - x\|. \tag{20}$$

Therefore, by projection $\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}$ to zero, we have

$$\begin{aligned}\left\|-\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right\|^2 &\geq \left\|P_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right) - \frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right\|^2\\ &= \left\|P_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right)\right\|^2 + \left\|-\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right\|^2\\ &\quad + 2\left(-\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right)^{\mathrm{T}}P_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right).\end{aligned}$$

The above equation is simplified as

$$2\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right)^{\mathrm{T}}P_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right) \geq \left\|P_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right)\right\|^2.$$

From the definition of the norm, we have

$$-\boldsymbol{e}^{\mathrm{T}}JP_\Omega\left(\frac{1}{\epsilon}J^{\mathrm{T}}\boldsymbol{e}\right) \leq 0. \tag{21}$$

Equation (21) can be used in LaSalle's invariant set principle to prove that $\boldsymbol{e} = 0$ is the only solution to $\dot{V} = 0$, and the proposed control scheme globally converges to zero [42].

B. Probability Completeness

We first introduce two lemmas and a definition to facilitate the analysis.

Remark 3.1: The task space of a manipulator is a bounded connected open set, as $X \subset \mathbb{R}^{\mathrm{n}}$, because of the existence of obstacles, there is an obstacle space as $X_{\text{obs}} \subset X$, and the subspace that is reachable by a manipulator becomes $X_{\text{reach}} \subset X_{\text{free}}$, where X_{free} is the free space as $X_{\text{free}} = X \backslash X_{\text{obs}}$.

TABLE I
COMPARISONS AMONG RNN BASED CONTROL SCHEME FOR REDUNDANT MANIPULATORS†

	Global Convergence	Theoretical Error	Free of Error Accumulation	Free of Matrix Inversion	Physical Limits Avoidance	Smothness Optimization	Obstacle Avoidance	Planning Completeness
Model in [43]	Yes	Nonzero	No	No	No	No	No	No
Model in [44]	Yes	Zero	Yes	Yes	Yes	No	No	No
Model in [36]	Yes	Zero	Yes	Yes	Yes	Yes	No	No
Model in [35]	Yes	Zero	Yes	Yes	Yes	Yes	No	No
Model in [25]	Yes	Zero	No	Yes	Yes	No	Yes	No
Model in [26]	Yes	Zero	No	Yes	Yes	Yes	Yes	No
Proposed	Yes	Zero	Yes	Yes	Yes	Yes	Yes	Yes‡

†Algorithms closely related to the proposed method are compared here.
‡Probabilistic completeness.

Lemma 3.1: For all points in the reachable space ($\forall x_{\mathrm{d}} \in X_{\mathrm{reach}}$), there exists a space, B_{d} for x_{d}, and all points in the space ($\forall x_{\mathrm{j}} \in B_{\mathrm{d}}$), a valid path can be found by the control scheme described in (7).

Proof: Because the point x_{d} is reachable by the manipulator, for any points ($\forall x_{\mathrm{s}} \in X_{\mathrm{reach}}$), there exists at least one valid path $\delta : [0, t] \to x_\delta$, $x_\delta \in X_{\mathrm{reach}}$, and $x_{\delta 0} = x_{\mathrm{s}}$, and $x_{\delta 0} = x_{\mathrm{d}}$. However, because the control scheme defined in (7) is consistently attracted by the target $\boldsymbol{r}_{\mathrm{d}}$, a subset of the path $\delta : [t - n, t] \to x_\delta$ that corresponds to the nonconcave configuration subspace is directly reachable by the control scheme in (7). If we define B_{d} as $\forall x_{\mathrm{j}} \in B_{\mathrm{d}}$ and $x_{\mathrm{j}} \in \delta : [t - n, t] \to x_\delta$, Lemma 3.1 holds truth. ■

Intuitively, B_{d} can be imagined as a basin to x_{d}. We know that, for most of the manipulators, if there are no joint limits, self-collisions and obstacles, $B_{\mathrm{d}} = X_{\mathrm{reach}}$ holds truth for any $x_{\mathrm{d}} \in X_{\mathrm{reach}}$. Those constrains break X_{reach} into basins, and here we have Lemma 3.2.

Lemma 3.2: For a given manipulator and a environment, $\cup B_{\mathrm{d}} = X_{\mathrm{reach}}$.

Proof: Lemma 3.1 proves that $\forall x_{\mathrm{d}} \in X_{\mathrm{reach}}$, B_{d} exists, so $\cup B_{\mathrm{d}} \geq X_{\mathrm{reach}}$ holds truth. Let us define $x \in (\cup B_{\mathrm{d}}) \backslash X_{\mathrm{reach}}$, because x locates on at least one of the valid path, as proved in Lemma 3.1, we have $x \in X_{\mathrm{reach}}$, therefore, $(\cup B_{\mathrm{d}}) \backslash X_{\mathrm{reach}} = \emptyset$, therefore $\cup B_{\mathrm{d}} = X_{\mathrm{reach}}$. ■

Now it is ready to prove that the proposed method has the planning probabilistic completeness.

Definition 3.1: For a given target $x_{\mathrm{d}} \in X_{\mathrm{reach}}$, if the set of valid paths, $\Sigma_\delta = \{\delta : [0, t] \to x_\delta$ equals to $\emptyset$, it is reported in finite time. If $\Sigma_\delta \neq \emptyset$, $P(\Sigma_\delta \cap T_{\mathrm{s}} = \emptyset) = 0$.

Theorem 3.1: For the proposed method, for $\Sigma_\delta \neq \emptyset$, in finite time, $P(\Sigma_\delta \cap T_{\mathrm{s}} = \emptyset) = 0$. ■

Proof: Lemma 3.1 and Lemma 3.2 show that B_{d} can be reached within finite time and Theorem 3.1 is immediate.

In real applications, as described in Algorithm 1, the maximum allowed exploration time needs be set to ensure the algorithm will not take arbitrarily long to explore. After exceeding the time, the algorithm will report the nonexistence of a valid trajectory.

C. Efficiency

If we denote a valid path by the control points as $\{x_{\mathrm{s}}, x_{\mathrm{d\text{-}k}}, \ldots, x_{\mathrm{d}-1}, x_{\mathrm{d}}\}$, where $x_{\mathrm{s}} \in B_{\mathrm{d}-\mathrm{k}}$ and $\{x_{\mathrm{d\text{-}i}} \in B_{\mathrm{d}-\mathrm{i}+1}, \forall i \in [1, \ldots, k]\}$. Then, we have following theorem.

Theorem 3.2: For the given path, $\{x_{\mathrm{s}}, x_{\mathrm{d\text{-}k}}, \ldots, x_{\mathrm{d\text{-}1}}, x_{\mathrm{d}}\}$, the possibility of finding a path grows exponentially in n iterations as $1 - exp(\frac{2k-np}{2})$, where $p = \min\{p_{\mathrm{i}}\}$ and p_{i} denotes the probability of sampling a point in B_{i}.

Proof: Because of the uniform random sampling is adopted in the proposed method, given the volume of B, the probability of sampling a point in B is known as the ratio of the volume of B with respect to the volume of the reachable space. However, precisely calculating the volume B is more computationally expensive than motion planning. Here, we estimate the upper bound of the efficiency instead.

If we approximate p_{i} with p, then the probability of sampling B is independent and identical, and follows Bernoulli distribution with parameter p. Let L denote the event of achieving the given path with n samplings. With the approximation of p, the event L follows a binomial distribution with parameters k and p, as

$$L \sim \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \tag{22a}$$

$$\mu = \mathrm{E}[L] = np. \tag{22b}$$

$$\sigma^2 = \mathrm{Var}(L) = np(1-p). \tag{22c}$$

Therefore, $P(L \leq \frac{k\mu}{np}) < \exp(\frac{\mu}{2}(1 - \frac{k}{np})^2)$ hold truth [45]. We also have

$$\begin{aligned} \frac{\mu}{2}\left(1 - \frac{k}{np}\right)^2 &= \frac{\mu}{2}\left(1 - \frac{2k}{np} + \frac{k^2}{n^2p^2}\right) \\ &= k - \frac{np}{2} - \frac{k^2}{2np}. \end{aligned} \tag{23a}$$

Because $\exp(-\frac{k^2}{2np}) \geq 0$, the probability of missing the path is less than $\exp(\frac{2k-np}{2})$, and the probability of finding the path is bigger than $1 - \exp(\frac{2k-np}{2})$. ■

IV. ILLUSTRATIVE EXAMPLES AND DISCUSSION

Table I compares RNN control schemes for redundant manipulator motion planning. To our best knowledge, the proposed method is the first RNN control scheme that achieved planning completeness. Among all those algorithms, three were selected to compare with the proposed method, based on similarity. The first algorithm [25] solves the redundant manipulator obstacle avoidance problem in the velocity space. For conciseness, we refer the work as "Method 1" in this paper. The second

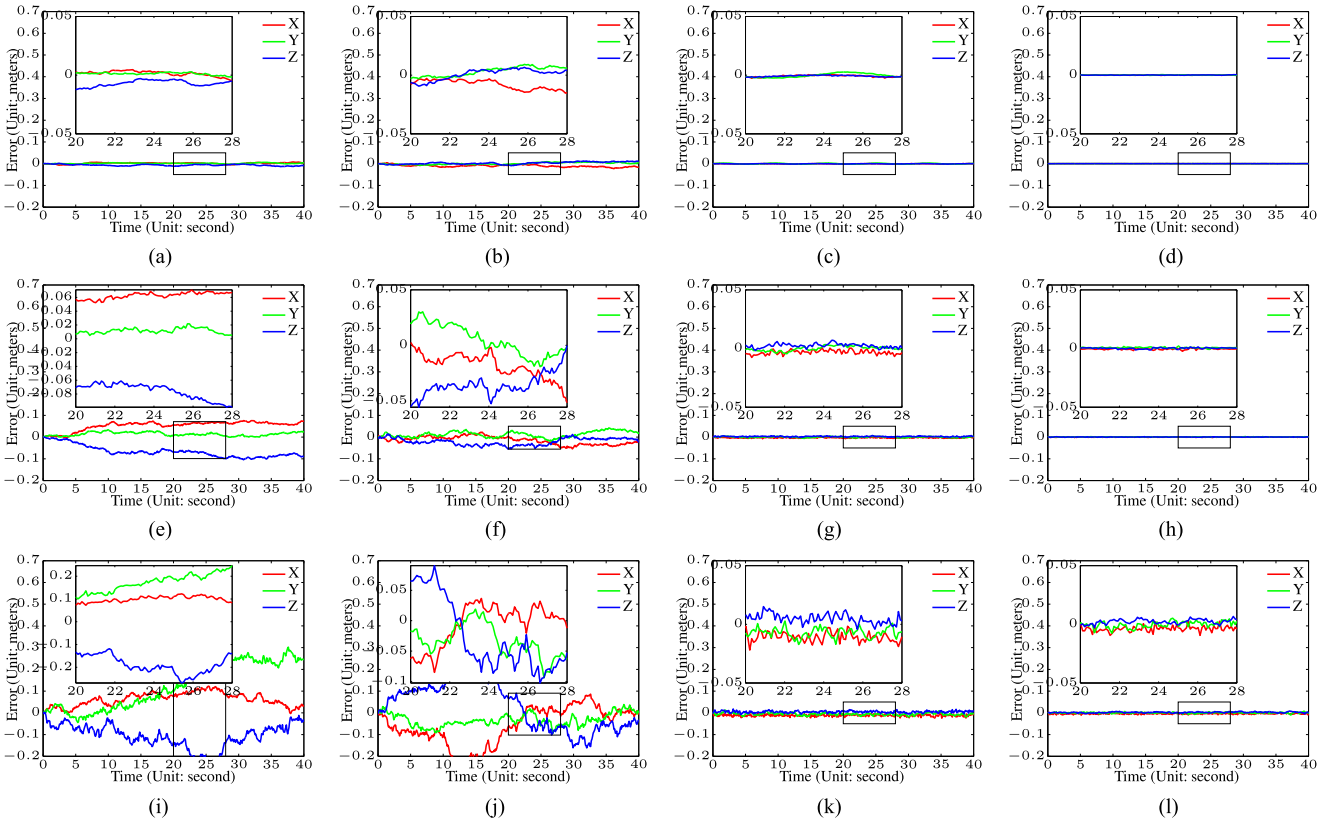

Fig. 3. Tracking precision comparison. Different levels of process noise have been injected to verify the robustness of the RNN control schemes. (a) $\sigma = 0.01$ - Method 1. (b) $\sigma = 0.01$ - Method 2. (c) $\sigma = 0.01$ - Method 3. (d) $\sigma = 0.01$ - Proposed. (e) $\sigma = 0.05$ - Method 1. (f) $\sigma = 0.05$ - Method 2. (g) $\sigma = 0.05$ - Method 3. (h) $\sigma = 0.05$ - Proposed. (i) $\sigma = 0.25$ - Method 1. (j) $\sigma = 0.25$ - Method 2. (k) $\sigma = 0.25$ - Method 3. (l) $\sigma = 0.25$ - Proposed.

algorithm [26] addresses the obstacle avoidance problem in the acceleration space, and we refer to it as "Method2." The third method [44] is not capable of avoiding obstacles, but it demonstrated the superior control precision and the outstanding robustness in obstacle-free environments. In order to provide a benchmark for the proposed method, we included that work in the comparison and refer to it as "Method 3."

The three representative algorithms were compared with the proposed method on the Mitsubishi PA10-7C based simulation. The PA10 redundant manipulator was chosen because it has seven DoF and its mechanics are similar to human arms [46].

In the simulation experiments, the parameters of the proposed method were empirically chosen as $k = 100$, $c_1 = c_2 = 0.5$, $A = 0.9$, and $w = 0.95$. For the other algorithms, we followed the references to set up the parameters [25], [26], [44].

A. Control Precision and Robustness Against Noise

Because "Method 3" is not capable of obstacle avoidance, the four algorithms were compared in obstacle-free environments. Obstacle-free environments can also validate that the proposed method balances the random exploration and the heuristic search, and succeed in planning with one heuristic search attempt.

In the simulation experiments, the manipulator starts from a known position and tracks a circular trajectory. White Gaussian process noise was injected in order to study the robustness against noise. The white noise has three different levels of standard deviation: $\sigma = 0.01$, $\sigma = 0.05$, and $\sigma = 0.25$. End-effector trajectory errors are compared in Fig. 3. From the results, we can see that with the increase of noise, "Method 3" and the proposed method showed stronger robustness against noise, due to their ability against error accumulation. Although the proposed method and "Method 3" have similar performance, we know the proposed method is better from Table II.

TABLE II
RMS POSITION TRACKING ERROR WITH RESPECT TO VARIOUS NOISE LEVEL

	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.25$
Proposed	0.007	0.010	0.010
Scheme1 [25]	0.027	0.092	0.246
Scheme2 [26]	0.025	0.069	0.199
Scheme3 [44]	0.005	0.010	0.011

B. Planning Completeness and Efficiency

Because "Method 3" is not capable of obstacle avoidance, only "Method 1" and "Method 2" were compared with the proposed method,

Two representative scenarios in redundant manipulator planning are adopted in the experiments. Scenario 1 is the environment with a plane-like obstacle and Scenario 2 has a

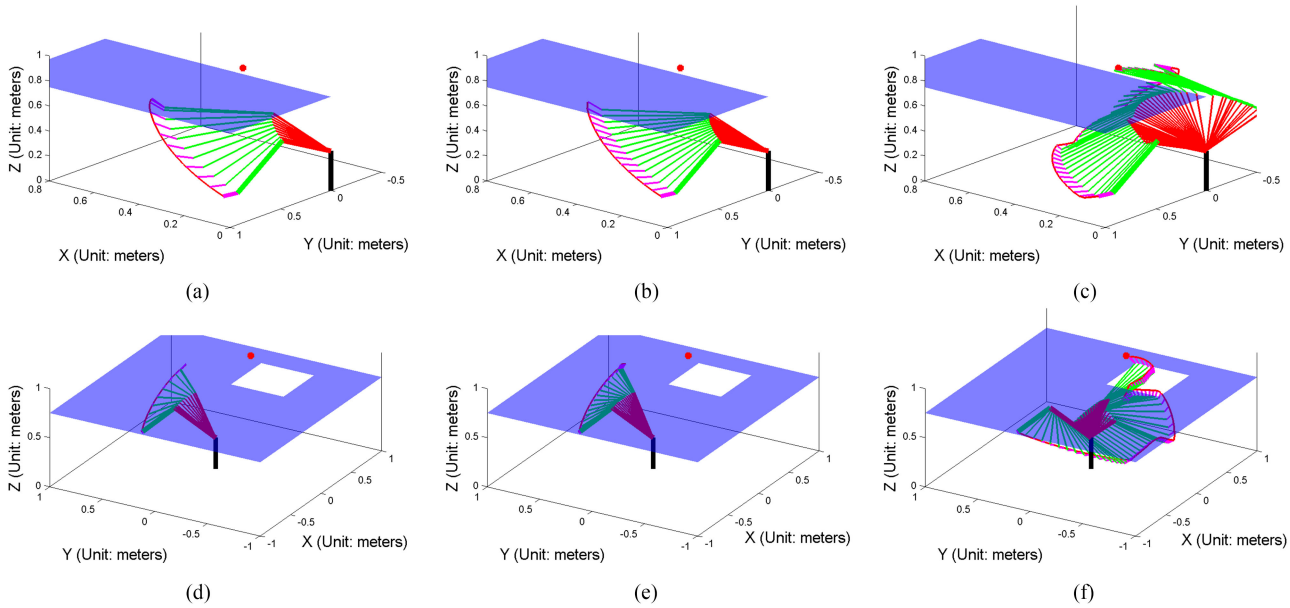

Fig. 4. Example planning results in environments with a plane-shaped obstacle or a window-shaped obstacle. The semitransparent planes denote the obstacle, the red globe denotes the target, and the colored lines indicate the manipulator trajectories. (a) Method 1 in Scenario 1. (b) Method 2 in Scenario 1. (c) Proposed in Scenario 1. (d) Method 1 in Scenario 2. (e) Method 2 in Scenario 2. (f) Proposed in Scenario 2.

TABLE III
PLANNING SUCCESS RATE COMPARISON

	Method1	Method2	Proposed
Scenario 1	92%	92%	100%
Scenario 2	72%	74%	100%

TABLE IV
PLANNING EFFICIENCY COMPARISON

	Without Learning			Proposed		
	RE	HS	Total	RE	HS	Total
Scenario 1	20.80	21.46	42.26	19.88	2.92	*22.80*
Scenario 2	56.66	56.60	113.26	52.76	5.42	*58.18*

RE: Random Exploration, HS: Heuristic Search, Total=RE+HS.
The averaged total numbers of explorations in 50 experiments are compared in the table.

window-shaped obstacle. In both scenarios, the three algorithms command the PA10 manipulator to reach a target behind the obstacles. The manipulator randomly starts from the known initial positions because both "Method 1" and "Method 2" suffer from the error accumulation problem. Table III compares the planning success rates, which is defined as v_s/v_e, where v_s denotes the total number of successes, and $v_e = 50$ denotes the total number of experiments.

Example planning results in scenarios 1 and 2 are shown in Fig. 4(a)–(c) and (d)–(f), respectively. In the figure, the semitransparent blue plane denotes the obstacle, the thick colored lines indicate manipulator initial configurations, and the thin colored lines denote the trajectory. The goal is denoted by a red sphere, and the curved red line segments denote the end-effector trajectories. From Fig. 4 and Table III, it is clear that the proposed method addresses the local minimum problem and achieves the planning completeness.

Planning efficiency of the proposed method was demonstrated by comparing the proposed method with a control scheme described in [47]. This control scheme is not capable of learning from exploration. Table IV compares the efficiency and shows how many random explorations and heuristic searches have been conducted in the planning. Fig. 5 shows how RRNN_{STM} neuron activities change during the exploration and adapt to different environments. From Table IV and Fig. 5, it is clear that the proposed method optimizes the exploration per environment and dramatically increases the planning efficiency.

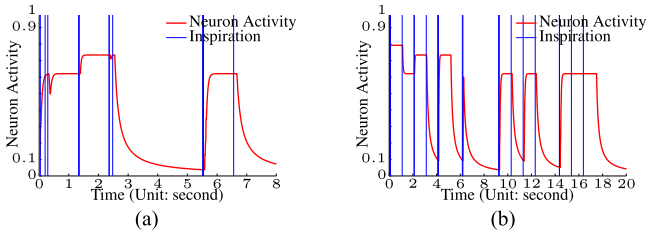

Fig. 5. Example RRNN_{STM} neuron activity changes in exploration. The neural network learns from exploration. In simpler environments, it tends to perform the heuristic search; in complex environments, it leans to the random exploration. (a) Scenario 1. (b) Scenario 2.

C. Real-World Experiment

The proposed method was applied to a Raven II surgical robot. Raven II surgical robots are popular in the robotic surgery community and are deployed at 18 sites worldwide. The challenges of controlling Raven II, as well as the kinematic model, can be found in [8]. The experiment simulates the popular endoscopic robotic surgeries, in which the robot reaches a target through a small orifice. Compared to exiting teleoperated robotic surgeries, the robot starts from an unknown position and autonomously reaches the goal without collision. Fig. 6(a) shows the Raven II surgical robot and 6(b) explains the experimental setup. A box with a 4-cm square opening represents the

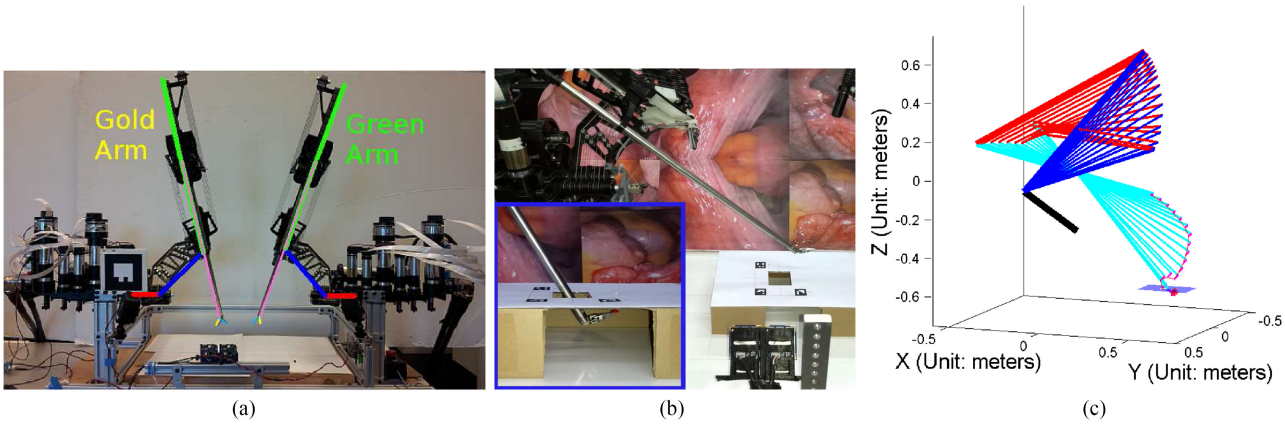

Fig. 6. Applying the proposed method to simulate autonomous robotic endoscopic surgery. The Raven II robot automatically reaches the surgical target under the control of the proposed method. (a) Raven II surgical robot. (b) Experimental setup and the initial manipulator position. The zoomed-in area shows that the manipulator reaches the goal. (c) Surgical robot trajectory.

surgical environmental obstacle and the target is denoted by a pin with a red head. The robot trajectory is recorded by a stereo camera. Fig. 6(c) visualizes the robot trajectory, in which thick colored lines indicate the manipulator initial configuration and the thin lines indicate the end-effector trajectory. The obstacle position is indicated by the semitransparent blue plane and the goal is denoted by a red dot. From the experimental result, it is clear that the proposed method achieved success in such a complex task in robotic surgeries.

V. Conclusion

This paper presents a new RNN control scheme for complete and efficient redundant manipulator motion planning. The proposed scheme addresses the local minimum problem in RNN control scheme, through introducing randomness. Moreover, it avoids the low-efficiency of the random explorations, through online learning from exploration. Rigorous theoretical analyses show the precision, the stability, the planning completeness, and the planning efficiency of the proposed method. Simulation experiments demonstrate that the proposed method has better precision, robustness than other three representative RNN schemes, and more importantly, it achieves the planning completeness and improves the planning efficiency. Real-world experiments demonstrate the application of the proposed method in the endoscopic robotic surgery scenario. The proposed method shares the efficiency and the robustness with other RNN control schemes, meanwhile, it achieves higher control precision. More importantly, the proposed method is the first RNN control scheme that achieves the planning completeness in environments with obstacles and shows the much broader applicability of RNN.

References

[1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.
[2] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2005.
[3] A. K. Bejczy, "Robot arm dynamics and control," 1974. [Online]. Available on: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19740008732.pdf
[4] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*, vol. 41. Princeton, NJ, USA: Princeton Univ. Press, 2005.
[5] Y. Pan, C. Yang, L. Pan, and H. Yu, "Integral sliding mode control: Performance, modification and improvement," *IEEE Trans. Ind. Inf.*, vol. 14, no. 7, pp. 3087–3096, Jul. 2018.
[6] S. Li and Y. Li, "Nonlinearly activated neural network for solving time-varying complex sylvester equation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1397–1407, Aug. 2014.
[7] Y.-J. Liu and S. Tong, "Barrier lyapunov functions for Nussbaum gain adaptive control of full state constrained nonlinear systems," *Automatica*, vol. 76, pp. 143–152, 2017.
[8] Y. Li and B. Hannaford, "Gaussian process regression for sensorless grip force estimation of cable-driven elongated surgical instruments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1312–1319, Jul. 2017.
[9] H. Gao, W. He, C. Zhou, and C. Sun, "Neural network control of a two-link flexible robotic manipulator using assumed mode method," in *IEEE Trans. Ind. Inf.*, 2018, doi: 10.1109/TII.2018.2818120.
[10] Z. Li, Z. Huang, W. He, and C.-Y. Su, "Adaptive impedance control for an upper limb robotic exoskeleton using biological signals," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1664–1674, Feb. 2017.
[11] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8112–8123, Oct. 2018.
[12] B. Sun, D. Zhu, and S. X. Yang, "A bioinspired filtered backstepping tracking control of 7000-m manned submarine vehicle," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3682–3693, Jul. 2014.
[13] H. Xiao *et al.*, "Robust stabilization of a wheeled mobile robot using model predictive control based on neurodynamics optimization," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 505–516, Jan. 2017.
[14] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Trans. Ind. Inf.*, vol. 13, no. 3, pp. 1162–1171, Jun. 2017.
[15] Y. Pan, H. Wang, X. Li, and H. Yu, "Adaptive command-filtered backstepping control of robot arms with compliant actuators," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 3, pp. 1149–1156, May 2018.
[16] G. Wen, C. L. P. Chen, and Y. J. Liu, "Formation control with obstacle avoidance for a class of stochastic multiagent systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5847–5855, Jul. 2018.
[17] W. He, Y. Ouyang, and J. Hong, "Vibration control of a flexible robotic manipulator in the presence of input deadzone," *IEEE Trans. Ind. Inf.*, vol. 13, no. 1, pp. 48–59, Feb. 2017.
[18] Y. Li and B. Hannaford, "Soft-obstacle avoidance for redundant manipulators with recurrent neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–6.
[19] L. Tang, Y. J. Liu, and C. L. P. Chen, "Adaptive critic design for pure-feedback discrete-time mimo systems preceded by unknown backlashlike hysteresis," *IEEE Trans. Neural Netw. Learn. Syst.*, 2018, doi: 10.1109/TNNLS.2018.2805689.
[20] L. Jin, S. Li, X. Luo, and Y. Li, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Trans. Ind. Inf.*, vol. 14, no. 9, pp. 3812–3821, Sep. 2018.

[21] X. Li, Y. Pan, G. Chen, and H. Yu, "Continuous tracking control for a compliant actuator with two-stage stiffness," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 57–66, Jan. 2018.
[22] W. He and Y. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 4, pp. 1174–1186, Apr. 2018.
[23] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, Jun. 2017.
[24] S. Li, J. He, Y. Li, and M. U. Rafique, "Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 2, pp. 415–426, Feb. 2017.
[25] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 752–759, Feb. 2004.
[26] D. Guo and Y. Zhang, "Acceleration-level inequality-based man scheme for obstacle avoidance of redundant robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6903–6914, Dec. 2014.
[27] Y. Zhang and L. Jin, *Robot Manipulator Redundancy Resolution*. New York, NY, USA: Wiley, 2017.
[28] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.
[29] K.-i. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Netw.*, vol. 6, no. 6, pp. 801–806, 1993.
[30] J. Nocedal and S. J. Wright, *Sequential Quadratic Programming*. New York, NY, USA: Springer, 2006.
[31] Z. Zhang and Y. Zhang, "Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 674–686, Apr. 2013.
[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[33] Y. Zhang, J. Wang, and Y. Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 923–931, Dec. 2002.
[34] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 1, pp. 147–154, Feb. 2001.
[35] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1423–1434, Jun. 2017.
[36] Y. Li, S. Li, M. Miyasaka, A. Lewis, and B. Hannaford, "Improving control precision and motion adaptiveness for surgical robot with recurrent neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1–6.
[37] A. N. Kolmogorov, "Foundations of the theory of probability." 1950.
[38] J. M. Bernardo and A. F. Smith, "Bayesian theory," 2001.
[39] Y. Li, S. Li, Q. Song, H. Liu, and M. Q.-H. Meng, "Fast and robust data association using posterior based approximate joint compatibility test," *IEEE Trans. Ind. Inf.*, vol. 10, no. 1, pp. 331–339, Feb. 2014.
[40] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Netw.*, vol. 1, no. 1, pp. 17–61, 1988.
[41] Y. Li, J. Zhang, and S. Li, "STMVO: biologically inspired monocular visual odometry," *Neural Comput. Appl.*, vol. 29, no. 6, pp. 215–225, 2018.
[42] Y. Zhang, J. Wang, and Y. Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 923–931, Dec. 2002.
[43] D. Guo and Y. Zhang, "Li-function activated znn with finite-time convergence applied to redundant-manipulator kinematic control via time-varying jacobian matrix pseudoinversion," *Appl. Comput.*, vol. 24, pp. 158–168, 2014.
[44] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2243–2254, Oct. 2016.
[45] R. Motwani and P. Raghavan, *Randomized Algorithms*. London, U.K.: Chapman & Hall, 2010.
[46] W. Shen, J. Gu, and Y. Ma, "3d kinematic simulation for pa10-7c robot arm based on VRML," in *Proc. IEEE Int. Conf. Autom. Logistics*, 2007, pp. 614–619.
[47] Y. Li, S. Li, and B. Hannaford, "A novel recurrent neural network control scheme for improving redundant manipulator motion planning completeness," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1–6.

Yangming Li (M'17) received the Ph.D. degree in automatic control engineering from University of Science and Technology of China, Hefei, China, in 2010. He is an Assistant Professor with Rochester Institute of Technology, Rochester, NY, USA. Before this appointment, he held the positions as an Acting Instructor with the University of Washington, Associate Professor with the Institute of Intelligent Machines, Chinese Academy of Sciences, and PostDoc with the University of Michigan. His research focuses on improving performance and reliability of environmental perception and modeling, robot manipulator collaboration, and manipulator and environment interaction, through probabilistic methods and learning techniques.

Shuai Li (SM'18) received the B.E. degree in precision mechanical engineering from Hefei University of Technology, Hefei, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, Hefei, China in 2008, and the Ph.D. degree in electrical and computer engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2014.

He is currently a Research Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include dynamic neural networks, robotic networks, and other dynamic problems defined on a graph.

Blake Hannaford (F'05) received the B.S. degree in engineering and applied science from Yale University, New Haven, CT, USA, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, CA, USA.

He has been a Professor of electrical engineering since 1989, and an Adjunct Professor of bioengineering, mechanical engineering, and surgery with the University of Washington, Seattle, WA, USA. From 1986 to 1989, he worked on the remote control of robot manipulators in the Man–Machine Systems Group, Automated Systems Section of the NASA Jet Propulsion Laboratory, Caltech, and supervised that group from 1988 to 1989. His current research interests include surgical robotics, surgical skill modeling, and haptic interfaces.

Dr. Hannaford received the National Science Foundation's Presidential Young Investigator Award and the Early Career Achievement Award from the IEEE Engineering in Medicine and Biology Society.