Improving Control Precision and Motion Adaptiveness for Surgical Robot with Recurrent Neural Network

Yangming Li[1], Shuai Li[2], David Caballero[1], Muneaki Miyasaka[1], Andrew Lewis[3] and Blake Hannaford[4]

Abstract— **Surgical robot research is driven by the desire of improving surgical outcomes. This paper proposed a Recurrent Neural Network based controller to address two problems: 1) improving control precision, 2) increasing adaptiveness for robot motion (explained in Section I). RNN was adopted in this work mainly because 1) the problem formulation naturally matches RNN structure, 2) RNN has advantages as an biologically inspired method. The proposed method was explained in detail and analysis shows that the proposed method is able to dynamically regulate outputs to increase the adaptiveness and the control precision. This paper uses Raven II surgical robot as an example to show the application of the proposed method, and the numeral simulation results from the proposed method and three other controllers show that the proposed method has improved precision, improved high robustness against noise and increased movement smoothness, and it keeps the manipulator links as far away as possible from physical boundaries, which potentially increases surgical safety and leads to improved surgical outcomes.**

Index Terms— **Kinematic Control, Bioinspired Controller, Surgical Robot, Recurrent Neural Network, Surgical Safety**

I. INTRODUCTION

Robotic surgeries became standard of care in many surgeries, because of its advantages, such as improved freedom of movement, amplified 3D endoscopic view, reduced hand tremor, decreased hospitalization and recovery time, reduced pain, and minimized scarring[1], [2], [3].

Although the study on surgical robot has made great progress, the controller study has room for improvement. For example, the Raven II surgical robot, as one of the most popular platform for robotic surgery study, still relies on tuning parameters for PID controllers, in order to achieve desired control performance[4]. In teleoperated robotic surgeries, surgeons use their conscious and unconscious intelligence to adapt to the reduced control precision[5], [6], [7], and in semi-automatic robotic surgeries, problems such as control precision and motion smoothness, remain as a bottleneck of improving system performance[8], [9], [10], [11], [12].

In this work, we aim to improve surgical robot performance through: 1, improving control precision, 2, increasing adaptiveness for robot motion. Robotic systems often desire low control errors, and this is especially true for surgical robot[13], [14]. Adaptiveness, in general, refers to making robots behave like experienced surgeons in various categories of surgeries. Research has correlated motion patterns with surgical skill levels (which is considered an equivalent of surgical outcomes[15], [16]), and the results show experts generally move smoothly and tend to move slower in the neighborhood of critical structures, and the distributions of basic kinematic features, such as velocity, change with respect to anatomical regions[16], [17]. These findings motivate the assumption that increased motion smoothness, the ability to dynamically adjust velocity pattern, and moving slower in the neighborhood of critical structures will make robots behave more like experts and may lead to increased surgical outcomes.

The manipulator control problem has been thoroughly studied and a survey on the topic can be found in [18]. Among these controllers, the Recurrent Neural Networks (RNNs) family algorithms attract our attention, because of the advantages of RNNs and their success in both the control field and the surgical robot field[19], [20], [9]. RNN controllers generally modeled the control problem as a quadratic programming problem with inequality constraints (explained in Section II-B), and they can address the two main limitations of the conventional pseudoinverse-based control algorithms: 1, high computational complexity of performing pseudo-inversion, 2, local instability. Because of those advantages, different RNN controllers have been proposed to address various control problems. For example, Xia *et al.* proposed a solution to the tracking problem of redundant serial manipulators[20]. Zhang *et al.* creatively optimize energy under the joint limit constraints for redundant manipulators with RNN controller[19]. These methods demonstrated improved efficiency, but have restrictions on initial position setup and suffer from the error accumulation problem[21]. Li *et al.* addressed the error accumulation problem through introducing the error into the optimization equation in a novel RNN controller[21]. RNN controllers also have been proposed to address the cooperative control problem[22], the unknown dynamics problem[23], multi-manipulator collaboration problem[24] etc. To our best knowledge, there is no existing solution to address the two problems identified above for surgical robots.

In this work, we propose a novel RNN controller and introduced it into the surgical robot community to address the control precision and the adaptiveness problem. The proposed method optimizes control output with respect to tracking errors and dynamically regulates joint angular speed to improve control precision and adaptiveness. Similar to other RNN controllers, the proposed RNN controller does not require training and the network weights are defined by

[1]Department of Electrical Engineering, University of Washington, Seattle, WA, USA 98195

[2]Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

[3]Applied Dexterity Inc., Seattle, WA, USA 98195

[4]Departments of Electrical Engineering, Bioengineering, Mechanical Engineering, and Surgery, University of Washington, Seattle, WA, USA 98195

robot model (Jacobian). To sum up, the core contributions of the paper are:

- We propose a Recurrent Neural Network based control optimization scheme, which has proven stability and efficiency, does not suffer from error accumulation and is capable of adjust joint movement speed dynamically.
- We demonstrate the application of the proposed method to the Raven II surgical robot.
- We verify the proposed method by comparing it with three other controllers on the precision, the robustness and the motion adaptiveness, through numerical simulation.

This paper is organized as follows: Section II introduces the formulation of the kinematic control problem, using Raven II as an example, and explains how to apply RNN to the problem. Section III presents the proposed method and analyzes the effectiveness. Section IV compares the proposed method with three other controllers and discusses the simulation experiments. Conclusions and future work are drawn in the last section.

II. PRELIMINARIES

A. Problem Formulation

The first step of designing a Recurrent Neural Network based controller for a manipulator is to abstract the manipulator control problem into a constrained optimization problem. This work uses Raven II surgical robot as an example.

The Raven-II surgical robot is a popular platform for robotic surgery study and there are 16 Raven sites worldwide. The Raven-II system has two 3-Degree of Freedom (DOF) spherical positioning mechanisms that are capable of attaching interchangeable 4-DOF instruments, as shown in Fig.1 [4]. The mechanical design and properties of the Raven-II can be found in[4]. The detailed mechanical design can be found in [4] (Fig. 5, 6 and 7). The left and the right positioning joints of the Raven-II robot have differences in geometry and reference frames. Because these differences have no impact on controller design, in this work, only the left (Gold) arm was studied. For quick reference, the Denavit-Hartenberg (DH) parameters are listed in Table I
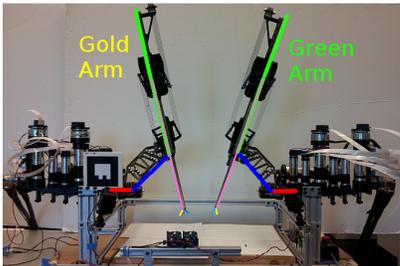

Fig. 1: The Raven II Surgical Robot. Its kinematic design was highlighted in colors and the DH parameters can be found in Table I .

For the robot, the state vector is $s = \{\theta_1, \theta_2, d_3, \theta_4, \theta_5, \theta_6\}$, as there are 5 revolute joints and 1 prismatic joint. Parameter $\alpha_1 = 75°$, $\alpha_2 = 52°$ are defined by Raven-II design and are

TABLE I: Denavit-Hartenberg parameters of Raven-II Left(Gold) Arm With Standard Raven Instrument.

Link(i)	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	α_1	0	0	θ_2
3	$\pi - \alpha_2$	0	d_3	$\pi/2$
4	0	a_3	d_4	θ_4
5	$\pi/2$	0	0	θ_5
6	$\pi/2$	l_w	0	θ_6
7	0	l_G	0	0

constant, and a_3, d_4, l_w and l_G are defined by the adopted instrument. In this paper, the standard Raven-II grasper was used and thus $a_3 = 0$, $d_4 = 458.69$mm, $l_w = 13$mm and $l_G = 7$mm. Given the DH parameters, the forward kinematic model can be easily derived (as shown in [25] Eqn. 3-30). The forward kinematic model defines the nonlinear transformation between the joint space and the Cartesian workspace, as $r(t) = f(s(t))$, and the constraints in the joint space commonly are from both physical limits and application requirements. The function $f(\cdot)$ depends on the manipulator state, and both the end effector state $r \in \mathbb{R}^m$ and the manipulator state $s \in \mathbb{R}^n$ change with time.

B. Recurrent Neural Network for Kinematic Control

RNNs allow in-layer connections, and are able to use their internal memory to process arbitrary sequences of inputs, including time series input. Also, the parallel processing capability is intrinsic to RNNs. All these features made them successful in control problems[19].

Applying Recurrent Neural Network to the corresponding constrained optimization problem is described in Eqn. 1:

$$\min_u u^T u \tag{1a}$$

$$\dot{r}_d = Ju \tag{1b}$$

$$u \in \Omega \tag{1c}$$

, where $J = \partial f / \partial \theta \in \mathbb{R}^{m \times n}$, $\dot{r}_d = \partial r(t) / \partial t \in \mathbb{R}^m$, and $u = \dot{\theta} = \partial \theta / \partial t \in \mathbb{R}^n$, $\Omega \subset \mathbb{R}^n$ is the domain of possible values of u. If we design the Lagrange multiplier, $\lambda \in \mathbb{R}^n$, as the correspondence to the equality constraint (Eqn.1b), we have the Lagrange function as: $L(u, \lambda) = u^T u + \lambda^T (\dot{r}_d - Ju)$. Therefore, the Karush-Kuhn-Tucker condition(Chapter 5.5.3 in [26]) ensures the solution to Eqn. 2 equals to the optimal solution to Eqn.1.

$$u = P_\Omega(u - \frac{\partial L}{\partial u}) \tag{2}$$
$$\dot{r}_d = Ju$$

It has been proven[20] that the dynamics of a projected Recurrent Neural Network (Eqn.3) converge to an equilibrium, which equals the optimal solution of the system described by Eqn.2.

$$\varepsilon \dot{u} = -u + P_\Omega(u - \frac{\partial L}{\partial u}) \tag{3a}$$

$$\varepsilon \dot{\lambda} = \dot{r}_d - Ju \tag{3b}$$

, where $\varepsilon > 0$ is a scaling factor and $P_\Omega(x) = \text{argmin}_{y \in \Omega} ||y - x||$ is a projection function from domain Ω' to Ω, and $x \in \Omega'$.

III. RECURRENT NEURAL NETWORK BASED CONTROLLER FOR SURGICAL ROBOT PERFORMANCE IMPROVEMENT

A. Controller Design

In this section, we demonstrate the design of the proposed RNN controller and the application to the Raven II surgical robot. The objective of the controller design is to ensure the steady-state solution to the neural network is identical to the solution to the kinematic model control, and have desired control precision and motion adaptiveness.

In order to overcome the error accumulation problem, we introduce the position tracking error e:

$$e = r - r_d \tag{4}$$

, where r is the end effector position and r_d is the target position, into the optimization objective function (Eqn.1a) and design the new one as:

$$\min_u (u^T u + k e^T e) \tag{5}$$

, where $k > 0$ is a weighting factor.

The change on the optimization function has no impact on the robot kinematic model, so the system model remains effective:

$$\dot{r}_d = Ju \tag{6}$$

For adaptiveness improvement, manipulator links need to dynamically adjust movement speed according to the relative position between the joint states and the boundary conditions, meanwhile, they 1) meet boundary conditions, 2) generate smooth motion pattern, 3) stay as far as possible from boundaries. These requirements can be achieved by optimizing the objective function (Eqn. 5) under a projection saturation function designed as:

$$P_\Omega(x) = \begin{cases} d^- & \text{for} & x \le d^- \\ x & \text{for} & d^- < x < d^+ \\ d^+ & \text{for} & d^+ \le x \end{cases} \tag{7}$$

, with boundary conditions as:

$$\begin{cases} d^- & = \max\left(-c_1(\theta - \theta^-), w^-\right) \\ d^+ & = \min\left(-c_2(\theta - \theta^+), w^+\right) \end{cases} \tag{8}$$

, where θ denotes the joint angle, θ^+ and θ^- are the upper and the lower bound joint limits (from physical limits and application requirements); w^+ and w^- are the upper and the lower bound of the joint speed, and c_1 and c_2 are two positive scaling factors.

The objective function (Eqn. 5) minimizes the tracking error and the norm of joint speed; term $c(\theta - \theta^\pm)$ ensures the joint speed monotonically decrease with respect to the distance to boundary conditions, which 1) slows down joint speed in the neighborhood of boundary, 2) guarantee the boundary condition will not be broke; term $w^\pm$ grant us the ability to dynamically adjust joint speed and ensure joint speed meet desired motion pattern.

According to the Karush-Kuhn-Tucker condition[26], the optimization problem that corresponds to our control objec-

tive (Eqn.5) can be described as:

$$u = P_\Omega(-kJ^T(r - r_d)) \tag{9a}$$

$$\dot{r}_d = Ju \tag{9b}$$

Therefore, the proposed controller can be mathematically described as:

$$\varepsilon \dot{u} = -u + P_\Omega(-kJ^T(r - r_d)) \tag{10}$$

Eqn.10 is similar to the dynamics equation of single layer feed forward RNN [21], thus it is natural to apply RNN to address the problem.

B. Effectiveness and Stability Analysis

The classical solution (Eqn.2) defined the Lagrange multiplier as the equality constraint, which can be extended as:

$$\begin{aligned} \lambda & = \lambda_0 + \frac{1}{\varepsilon}\int_0^t (\dot{r}_d - Ju)\mathrm{d}t \\ & = \lambda_0 + \frac{1}{\varepsilon}(r_d - r_{d0}) - \frac{1}{\varepsilon}\int_0^t Ju\,\mathrm{d}t \end{aligned}$$

, where λ_0 and r_{d0} are the initial value of λ and the initial value of desired end effector position, respectively. Recall that $\int_0^t Ju\,\mathrm{d}t = r - r_0$[21], therefore, from the projected RNN model (Eqn.3) and the upper equation we know:

$$\begin{aligned} \varepsilon \dot{u} & = -u + P_\Omega\big(J^T(\lambda_0 + \frac{1}{\varepsilon}(r_d - r_{d0}) - \frac{1}{\varepsilon}\int_0^t Ju\,\mathrm{d}t))\big) \\ & = -u + P_\Omega\big(J^T(\lambda_0 + \frac{(r_d - r) - (r_{d0} - r_0)}{\varepsilon})\big) \end{aligned} \tag{11}$$

Eqn.11 reveals that in the control law (Eqn.2), for any given t_0, if $r_{d,t_0} \ne r_{t_0}$, then for any $t > t_0$, there is a biased control error as $e_0 = r_{d,t_0} - r_{t_0}$. In the proposed method know the biased control error does not exist because of the introduction of the tracking error (Eqn.4).

A Lyapunov function is defined as $V = e^T e/2$ for the proposed method, where e is the penalty from tracking error (defined in Eqn. 4). For any given control target r_d, because it is constant:

$$\dot{e} = \dot{r} = Ju = JP_\Omega(-kJ^T e) \tag{12}$$

Substituting Eqn.12 into the defined Lyapunov function as:

$$\dot{V} = e^T \dot{r} = e^T JP_\Omega(-kJ^T e) \tag{13}$$

Because the defined projection function is a saturation function that contains point $(0,0)$, we have: $||y - x|| \ge ||P_\Omega(x) - x||$ for $y \in \Omega$[20], therefore we know:

$$P_\Omega(x) = \begin{cases} \dot{V} = 0, & \text{if and only if} \quad J^T e = 0 \\ \dot{V} < 0, & \text{otherwise} \end{cases} \tag{14}$$

While the Jacobian $J \in \mathbb{R}^{m \times n}$ is non singular for the control problem, we know $J^T e = 0$ if and only if $e = 0$, therefore, the proposed method also has the global stability[19], [20].

IV. SIMULATION RESULT AND DISCUSSION

A. Simulation Setup

Numerical simulation was adopted for verifying the proposed method on the Raven-II surgical robot.

The boundary conditions for the simulation are listed in Table.II. In the table, row 1 and 2 are joint lower and upper limits and row 3 and 4 are joint speed lower and upper limits, respectively. The joint limits are defined by Raven

II mechanical properties, and the speed limits are defined by surgical procedure. Given joint positions, the Jacobian matrix of the forward kinematic model can be used to project the speed limit between the Cartesian space and the Joint space. In order to show that the proposed method can increase surgical safety by dynamically changing boundary conditions with system states, we intentionally chose big constant speed limits as listed in the table.

TABLE II: Simulation Boundary Conditions.

Joint 1 (Shoulder)	Joint 2 (Elbow)	Joint 3 (Insertion)	Joint 4 (Rotation)	Joint 5 (Wrist)	Joint 6 (Finger)
$0°$	$45°$	0.23m	$-182°$	$-70°$	$-105°$
$90°$	$135°$	0.56m	$182°$	$70°$	$105°$
$-50°/s$	$-70°/s$	-0.3m/s	$-100°/s$	$-100°/s$	$-100°/s$
$50°/s$	$70°/s$	-0.2m/s	$100°/s$	$100°/s$	$100°/s$

The proposed method was compared with the algorithms described in [19], [20] and [21], because these algorithms are RNN based solutions and also because of the popularity of these algorithms. For easy reference, we refer the algorithms described in [19], [20] and [21] as controller1, controller2 and controller3, respectively. In the simulation, the parameters of the proposed method were empirically chosen as: $k = 100$ and $c_1 = c_2 = 0.5$. For the other methods, we followed the references to setup the parameters[19], [20], [21].

B. Tracking Precision

Two types of comparisons were made (one is to track a fixed position, another one is to track time-varying trajectories) in order to verify the tracking precision under different circumstances.

In the fixed position tracking experiment, because controller1 and controller2 require the initial position of the manipulator to be aligned with the start tracking position (Eqn. 11), only controller3 was compared with the proposed method, and the experimental result was shown in Fig.2. In Fig.2(a) and (b), the red lines denote the end effector trajectories, the green lines indicate the initial manipulator link positions and the blue lines denote the manipulator poses; in (c) and (d), the red, green and blue solid lines and the cyan dots, black doted lines and magenta lines denoted the joint positions of the 1st to the 6th links, respectively; in (e) and (f), the red lines, the green lines and the blue lines denotes the tracking error in X, Y and Y direction, respectively. From the figure we can see the proposed method and controller3 planned similar but different trajectories for the robot. The proposed method prefers slow speed for links, so it tends to drive links simultaneously. The proposed method also tends to keep all links as far as possible from the physical boundary. This shows the proposed method achieve the behavior of staying away from boundaries. The tracking RMS error was calculated for the last 15 seconds, and it is 0.00076m for the proposed method and 0.00078m for controller3.

In the time-varying trajectories tracking experiments, an acted Raven II teleoperated operation on mouse dissection

(a) Trajectory-Proposed   (b) Trajectory-Controller3

(c) Joint Angle-Proposed   (d) Joint Angle-Controller3

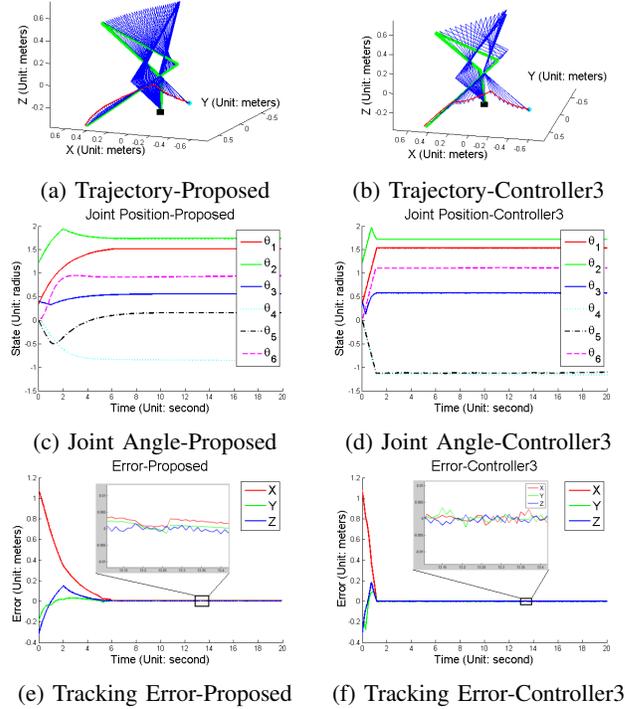(e) Tracking Error-Proposed   (f) Tracking Error-Controller3

Fig. 2: Fixed Position Tracking Comparison. (a) and (b) visualized the planned movements; (c) and (d) shows the joint trajectories (For joint 3, the unit of Y axis is *m*); (e) and (f) shows the end effector tracking errors.
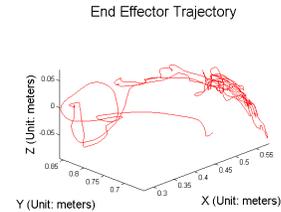

Fig. 3: Time Varying Trajectory.

was injected into the simulator (trajectory shown in Fig.3). For controller1 and controller2, the initial position of the end effector was at the start point of the trajectory. For controller3 and the proposed method, their start points were randomly picked in the workspace to show the advantages. The tracking precisions comparison was shown in Fig.4. In the figure, the red lines, the green lines and the blue lines indicated the tracking error in X, Y and Y direction, respectively. The RMS are 0.030m, 0.026m, 0.019m and 0.016m for controller1, controller2, controller3 and the proposed one, respectively.

C. Robustness

The robustness was studied by showing the tracking errors with respect to different noise levels. Circle tracking was used because it simulates the suturing and the tying knot operation and is very common in surgical robots. Noise had been modeled as additive Gaussian white noise as the system is assumed to be fully tuned. The exact same circle has been

(a) Tracking Error-Proposed   (b) Tracking Error-Controller1

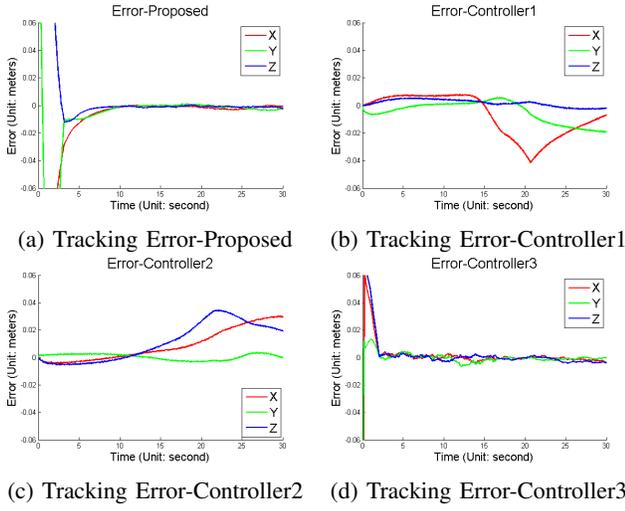(c) Tracking Error-Controller2   (d) Tracking Error-Controller3

Fig. 4: Time-varying Trajectory Tracking Comparison.

used under noise standard deviation $\sigma = 0.01$, $\sigma = 0.05$ and $\sigma = 0.25$. The tracking errors are visualized in Fig.5 and the RMS was compared in Table.III. From the results we can see the proposed method has good robustness against noise.

TABLE III: RMS Position Tracking Error With Respect to Various Noise Level.

	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.25$
Proposed	0.007	0.009	0.010
Controller1[19]	0.030	0.106	0.344
Controller2[20]	0.026	0.073	0.284
Controller3 [21]	0.004	0.009	0.012

D. Adaptiveness Study

The proposed method is able to dynamically adjust boundary conditions with respect to joint positions. We force all 6 links to move in the full range of the physical limits(TableII), and the boundary condition changes were shown in Fig. 6. The proposed method is better on velocity smoothness because we introduced dynamic bounding conditions, which was shown in Fig.6.
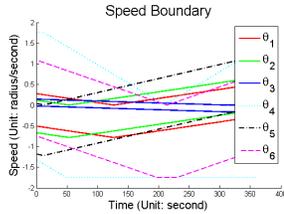

Fig. 6: Joint Velocity Boundary Dynamic Changes. The boundary conditions of the proposed method changes with respect to joint states, which dynamically adjust the manipulator behaviors.

By visualizing the real joint speed changes, we can see the dynamic boundaries force the links to stay away from the physical limits and tend to generate smooth speed, as shown in Fig.7. Clinically, these changes will make the robot move slower in the neighborhood of critical structures and

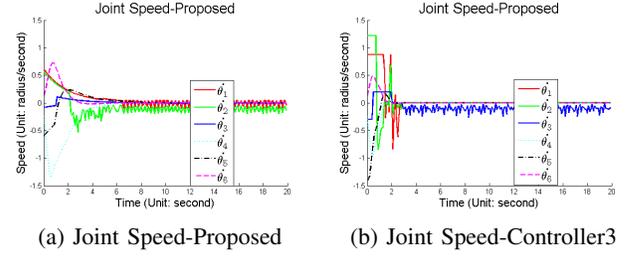behave more gently, which potentially will improve surgical outcome.

(a) Joint Speed-Proposed   (b) Joint Speed-Controller3

Fig. 7: Joint Velocity Comparison. For joint 3, the unit of Y axis is m/s. The boundary condition of the proposed method change dynamically, as a result, the proposed method plans smooth movements, which potentially increase surgical safety.

V. CONCLUSION

Aiming for improving robotic surgery outcomes, this paper proposed a RNN controller to improve the tracking precision and the motion pattern adaptiveness. The effectiveness and the stability of the proposed method were theoretically analyzed and numerical simulations were used to compare the proposed method with three other algorithms to validate the precision, robustness and adaptiveness. The simulation results showed the proposed method has high precision, robustness, and generates smooth motions and repels boundaries.

In the simulation, we only modeled noise as additive and white Gaussian distributed and the speed boundaries are in the joint space and not really correlated with specific surgical procedures. It definitely needs to validate the proposed method with real surgeries and study the surgical outcomes. Meanwhile, the fact that even experienced surgeons do not know how to clearly define the relationship between motion patterns and surgical procedures and they never clearly calculate arm, wrist hand and finger position and velocity strongly driving us to explore deeper in biologically inspired solutions for surgical outcome improvement.

REFERENCES

[1] M. J. Mack, "Minimally invasive and robotic surgery," *Jama*, vol. 285, no. 5, pp. 568–572, 2001.
[2] C. Diaz-Arrastia, C. Jurnalov, G. Gomez, and C. Townsend, "Laparoscopic hysterectomy using a computer-enhanced surgical robot," *Surgical endoscopy*, vol. 16, no. 9, pp. 1271–1273, 2002.
[3] J. Clark, D. P. Noonan, V. Vitiello, M. H. Sodergren, J. Shang, C. J. Payne, T. P. Cundy, G.-Z. Yang, and A. Darzi, "A novel flexible hyperredundant surgical robot: prototype evaluation using a single incision flexible access pelvic application as a clinical exemplar," *Surgical endoscopy*, vol. 29, no. 3, pp. 658–667, 2014.
[4] B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White, "Raven-ii: an open platform for surgical robotics research," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 4, pp. 954–959, 2013.
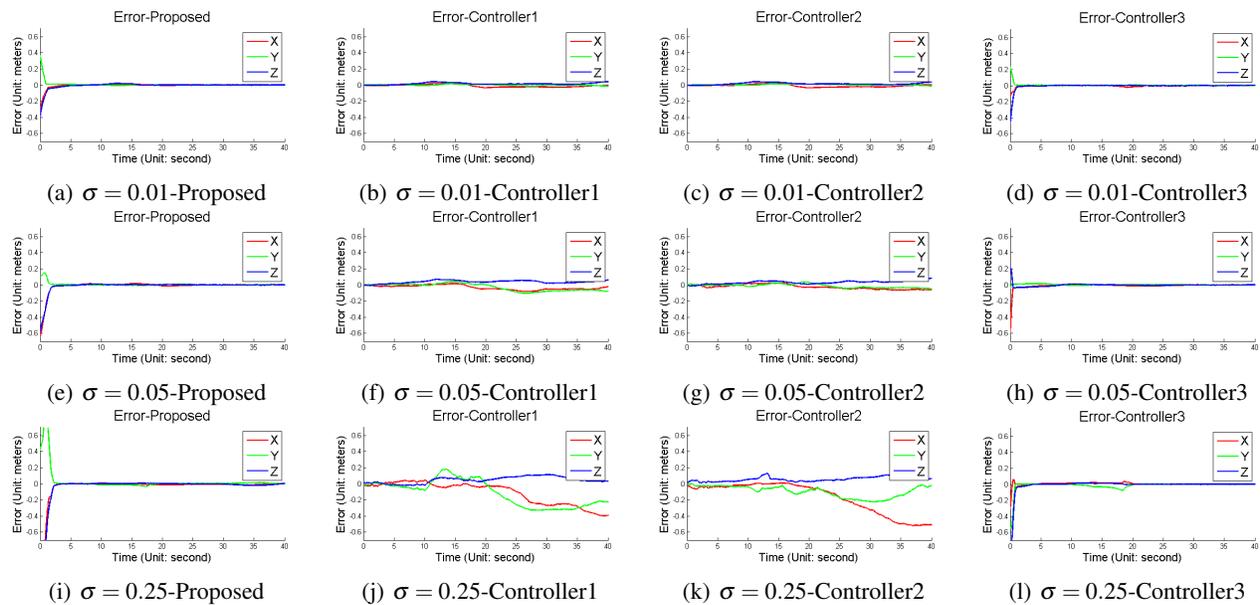
(a) $\sigma = 0.01$-Proposed (b) $\sigma = 0.01$-Controller1 (c) $\sigma = 0.01$-Controller2 (d) $\sigma = 0.01$-Controller3

(e) $\sigma = 0.05$-Proposed (f) $\sigma = 0.05$-Controller1 (g) $\sigma = 0.05$-Controller2 (h) $\sigma = 0.05$-Controller3

(i) $\sigma = 0.25$-Proposed (j) $\sigma = 0.25$-Controller1 (k) $\sigma = 0.25$-Controller2 (l) $\sigma = 0.25$-Controller3

Fig. 5: Comparison on Tracking Errors With Respect to Various Noise Level.

[5] H. W. Schreuder, R. Wolswijk, R. P. Zweemer, M. P. Schijven, and R. H. Verheijen, "Training and learning robotic surgery, time for a more structured approach: a systematic review," *BJOG: An International Journal of Obstetrics & Gynaecology*, vol. 119, no. 2, pp. 137–149, 2012.

[6] C. D. Lallas, Davis, and J. W. Members of the Society of Urologic Robotic Surgeons, "Robotic surgery training with commercially available simulation systems in 2011: a current review and practice pattern survey from the society of urologic robotic surgeons," *Journal of endourology*, vol. 26, no. 3, pp. 283–293, 2012.

[7] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: a current perspective," *Annals of surgery*, vol. 239, no. 1, pp. 14–21, 2014.

[8] C. Coulson, R. Taylor, A. Reid, M. Griffiths, D. Proops, and P. Brett, "An autonomous surgical robot for drilling a cochleostomy: preliminary porcine trial," *Clinical Otolaryngology*, vol. 33, no. 4, pp. 343–347, 2008.

[9] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.

[10] H. Kang and J. T. Wen, "Autonomous suturing using minimally invasive surgical robots," in *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*. IEEE, 2000, pp. 742–747.

[11] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W. D. Boyd, P. Abbeel *et al.*, "Autonomous multilateral debridement with the raven surgical robot," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1432–1439.

[12] G. Moustris, S. Hiridis, K. Deliparaschos, and K. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[13] R. A. Beasley and R. D. Howe, "Increasing accuracy in image-guided robotic surgery through tip tracking and model-based flexion correction," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 292–302, 2009.

[14] M. Haghighipanah, Y. Li, M. Miyasaka, and B. Hannaford, "Improving position precision of a servo-controlled elastic cable driven surgical robot using unscented kalman filter," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2030–2036.

[15] N. Ahmidi, P. Poddar, J. D. Jones, S. S. Vedula, L. Ishii, G. D. Hager, and M. Ishii, "Automated objective surgical skill assessment in the operating room from unstructured tool motion in septoplasty," *International journal of computer assisted radiology and surgery*, vol. 10, no. 6, pp. 981–991, 2015.

[16] R. A. Harbison, Y. Li, A. M. Berens, R. A. Bly, B. Hannaford, and K. S. Moe, "An automated methodology for assessing anatomy-specific instrument motion during endoscopic endonasal skull base surgery," *Journal of Neurological Surgery Part B: Skull Base*, 2016.

[17] R. A. Harbison, A. M. Berens, Y. Li, R. A. Bly, B. Hannaford, and K. S. Moe, "Region-specific objective signatures of endoscopic surgical instrument motion: A cadaveric exploratory analysis," *Journal of Neurological Surgery Part B: Skull Base*, 2016.

[18] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[19] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE transactions on neural networks*, vol. 14, no. 3, pp. 658–667, 2003.

[20] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 1, pp. 147–154, 2001.

[21] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[22] S. Li, J. He, Y. Li, and M. U. Rafique, "Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 2, pp. 415–426, 2017.

[23] Z. Wang, T. Zhou, Y. Mao, and Q. Chen, "Adaptive recurrent neural network control of uncertain constrained nonholonomic mobile manipulators," *International Journal of Systems Science*, vol. 45, no. 2, pp. 133–144, 2014.

[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 transactions on neural networks and learning systems*, 2016.

[25] B. Hannaford, J. Ma, H. King, and S. Kosari, "Kinematic analysis of the ravenii (tm) research surgical robot platform," *Department of Electrical Engineering, University of Washington, Tech. Rep.(uweetr-2012-0006)*, 2016.

[26] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.