A Haptic Guidance System for Computer-Assisted Surgical Training Using Virtual Fixtures

Minsik Hong[1] and Jerzy W. Rozenblit[1, 2]
[1]Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona, USA
[2]Department of Surgery, University of Arizona, Tucson, Arizona, USA
mshong@email.arizona.edu, jr@ece.arizona.edu

***Abstract*— A simple and robust fuzzy logic controller with a forbidden region virtual fixture is proposed for a haptic guidance system developed to assist in simulation-based training of laparoscopic surgery. A surgical training system that involves human in the control loop is discussed. It uses a controller which adjusts control outputs for different trainees based on their skill level. The system involves virtual fixtures to improve human manipulation tasks inherent in this problem domain. The proposed controller cooperates with a heuristic-based scaling factor modifier to make such adjustments. The experimental results illustrate the feasibility and effectiveness of the proposed haptic guidance system.**

Keywords – Surgical Training, Laparoscopy, Haptic Guidance, Virtual fixtures, Fuzzy Logic Controller

I. Introduction

As an effective medical procedure, laparoscopic surgery has several benefits such as minimizing operative blood loss, easing post-operative pain, and reducing the recovery time. However, the procedure is more challenging than conventional open surgery due to the use of long and thin surgical instruments with an endoscope camera, and the lack of depth perception when using a monitor to view the operating field. Therefore, it is necessary to practice such surgical skills using laparoscopic simulators [1].

Many training devices have been developed to aid in laparoscopic surgery training. Such trainers range from inexpensive training kits [2][3] to large-scale, expensive Virtual Reality Simulators (VRSs) [4][5].

In our research, we are developing the state-of-the-art surgical system called Computer-Assisted Surgical Trainer (CAST) [7]. We incorporate many features from both the simple kits and VRSs. One of the advantages of VRSs is providing haptic feedback for a trainee to sense force when the trainee touches an object. However, this *does not guide the trainee* to learn how to execute a particular surgical movement due to the absence of haptic-based navigation. Unlike VRSs, the goal of CAST is to provide both realistic training environments and <u>haptic (force) as well as visual guidance to trainees.</u>

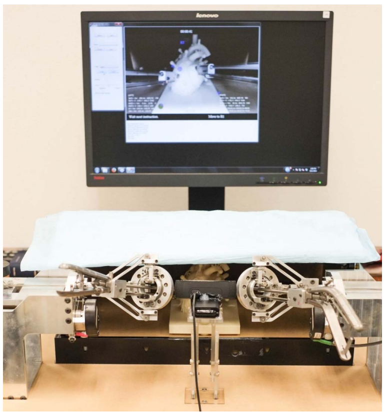


Figure 1. Computer-Assisted Surgical Trainer (CAST)

In this paper, we propose a haptic guidance system based on the concept of virtual fixtures [10]. As active constraints, the virtual fixtures were first introduced by Rosenberg to assist a human operator in teleoperation [6]. The active constraints have been widely used in teleoperation tasks, robotic surgery, and several haptic devices. A good analogy of the virtual fixtures is using a ruler when drawing a line. If we use a ruler, we can easily draw a line without any difficulties. Likewise, if the CAST system provided an active constraint, this would be very helpful in surgical training. (Note that we do not propose to research the virtual fixture concept further. Rather, we draw from it as an inspiration and foundation for refining the controller.)

The significance of the work presented here is the ability to flexibly adjust force-based guidance for learners with different skill levels, who must practice fundamental skills of laparoscopy in a virtual training environment.

The remainder of this paper is organized as follows: the overview of CAST is given in Section II. In Section III, the proposed haptic guidance system is described. Experimental results are presented in Section IV. Discussion and conclusion are given in Section V.

II. Overview of Computer Assisted Surgical Traininer

We have developed the foundations for CAST in

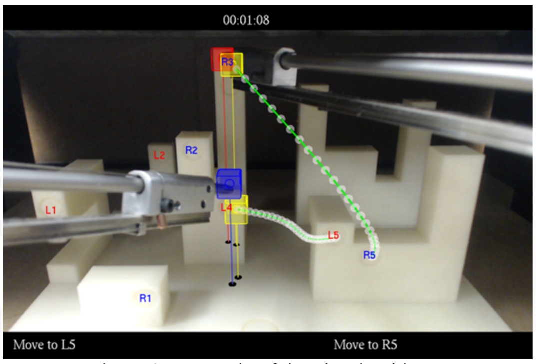


Figure 2. Example of the visual guidance

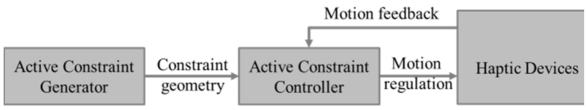


Figure 3. Generalized implementation framework for virtual fixtures [10]

previous work [7]-[9]. Fig. 1 illustrates the overall setup. CAST's hardware consists of two fixtures to hold surgical instruments, a camera to imitate an endoscope, an exchangeable cassette to support a variety of practice scenarios, motors to provide haptic feedback and navigation, and electronics to support the various control, sensing, and data collection functions. The instrument gimbal allows four degrees of movement freedom – three rotational movements (i.e., yaw, pitch, and roll) and one linear insertion movement.

CAST software consists of four modules – *opt*MIS, *opt*Assessment, *opt*Viz, and *opt*Guide. The *opt*MIS module generates collision free and shortest paths for laparoscopic instrument movements in the rigid training environment [8]. The *opt*Assessment quantifies competency objectively in minimally invasive surgical skills [9]. The *opt*Viz is responsible for the visualization of guidance information [7]. (Fig. 2 illustrates an example of visual guidance.) *opt*Guide is the haptic guidance module.

The training process consists in generating a recommended path for a surgical instrument movement in a 3D operating space, guiding the user in how to most effectively steer the instrument in this space, and collecting performance metrics such as task completion time, economy of movement, and accuracy [9]. In this paper, we focus on a new technique to provide a better haptic guidance.

III. Haptic Guidance System

In [10], the authors proposed a generalized implementation framework for virtual fixtures. It is shown in Fig. 3. The active constraint generator provides constraint geometry. The active constraint controller is responsible for assisting a user by motion regulation. Based on this framework, a haptic guidance system is designed and implemented for our *opt*Guide module.

A. Active Constraint Generator

For the haptic guidance system, a *forbidden region virtual fixture* (FRVF) is used to restrict motions such as deviation from a desired path. Fig. 4-(a) depicts an example of FRVF. The gray area represents a forbidden region. Also, an *attractive type constraint* (Fig. 4-(b)) is used to minimize the deviation from the desired path.

To construct constraint geometry, the following

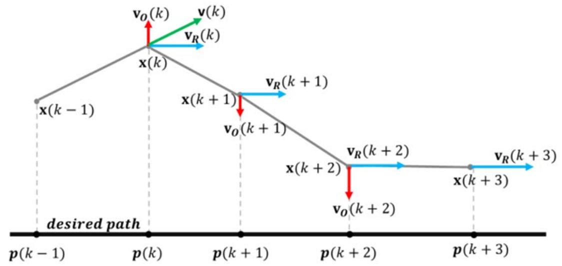


Figure 5. Examples of decomposed velocity vectors

information is used.

- Discrete time domain: k
- Sampling time : ΔT
- A desired path that consists of a set of three-dimensional discrete points:
  $\mathbf{P} = \{p_1, \cdots, p_n\}, where\ p_i \in \boldsymbol{R}^3, i = 1, \cdots, n$
- Instrument tip position: $\mathbf{x}(k) \in \boldsymbol{R}^3$
- Instrument tip velocity: $\mathbf{v}(k) \in \boldsymbol{R}^3$,
  $\mathbf{v}(k) = \big(\mathbf{x}(k) - \mathbf{x}(k-1)\big)/\Delta T$

The instrument tip velocity is decomposed as follows:

$$\mathbf{v}(k) = \mathbf{v}_R(k) + \mathbf{v}_O(k) \qquad (1)$$

where $\mathbf{v}_R(k)$ is a velocity vector along the reference direction and $\mathbf{v}_O(k)$ is a velocity vector orthogonal to the reference direction. Fig. 5 illustrates examples of decomposed velocity vectors.

Let us introduce a control parameter ($0 \le c_O(k) \le 1$) to adjust the non-preferred velocity component ($\mathbf{v}_O(k)$). Equation (1) is then written as

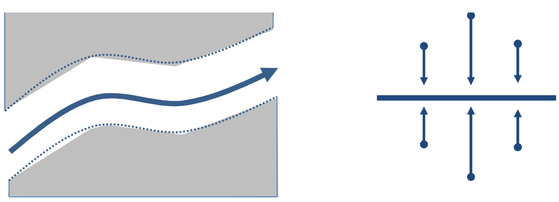
$$\mathbf{v}(k) \triangleq \mathbf{v}_R(k) + c_O(k)\mathbf{v}_O(k).$$

By varying $c_O(k)$, we can control the instrument tip movement. For instance, if we set $c_O(k)$ as 0, then we cannot allow any non-preferred movement. If we set $c_O(k)$ as 1, then a user can move an instrument freely without any restriction.

In [11], various constraint geometries (e.g., curve, tube, and cone) were proposed to implement a visually-assisted control system. For the CAST system, a virtual tube is defined to design FRVF. The virtual tube is described by a tube axis and a tube radius (r_d). A desired path ($\mathbf{P}$) generated by *opt*MIS is used to represent the axis of the virtual tube. (Fig. 6-(a) depicts an example of the virtual tube.) Using this constraint geometry, the control parameter (c_O) is defined as follows:

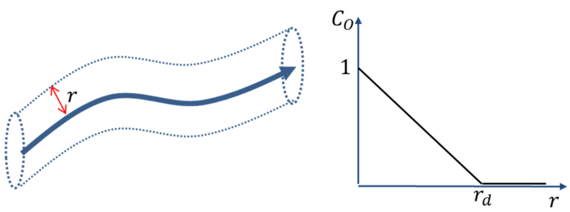
$$c_O(k) = \begin{cases} 0, & r(k) > r_d \\ 1 - r(k)/r_d, & r(k) \le r_d \end{cases}$$

where $r(k)$ is the distance from the instrument tip ($\mathbf{x}(k)$) to a desired path ($\mathbf{P}$). Fig. 6-(b) illustrates $c_O(k)$. The outside area of the virtual tube represents a forbidden



(a) (b)

Figure 4. (a) A forbidden region virtual fixture and (b) an attractive constraint [10]



(a) (b)

Figure 6. (a) A virtual tube for FRVF, (b) the relationship between $\boldsymbol{c_O}$ and $\boldsymbol{r}$

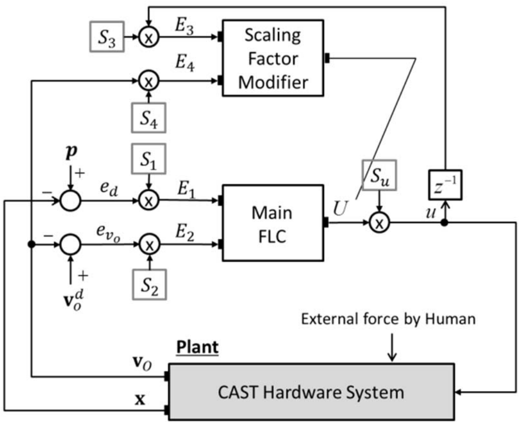


Figure 7. The proposed active constraint controller

region. Any non-preferred movement is not allowed in the forbidden region. Therefore, $c_O(k)$ is set as 0 if $r(k) > r_d$. Within the virtual tube, $c_O(k)$ gradually increases as $r(k)$ decreases in order to prohibit rapid changes of $c_O(k)$.

Based on the defined constraint geometry, an active constraint controller is designed.

B. Active Constraint Controller

CAST involves human in a control loop, unlike an autonomous system such as a manipulator in an assembly line. The main role of *opt*Guide is to provide force guidance to a novice trainee. Existing guidance systems [12],[13] use a PD controller for motor control. Similarly, we have proposed a PD controller and PD-like Fuzzy Logic Controller to provide haptic feedback in [7]. However, these simple PD-type controllers could not adjust the guidance force for different trainees who varied in how they operated the instruments because the controllers don't have adaptive features. To improve the guidance system, here we propose a novel haptic guidance controller that consists of Fuzzy Logic Controller (FLC) and Scaling Factor Modifier (SFM). The block diagram of the proposed controller is shown in Fig. 7.

The active constraint controller (*main*FLC) is a discrete-time FLC that has two inputs and a single output. The inputs of the *main*FLC, the position error ($\mathbf{e}_d$) and the velocity error ($\mathbf{e}_{\mathbf{v}_O}$), are defined as

$$\mathbf{e}_d(k) = \boldsymbol{p}(k) - \mathbf{x}(k)$$
$$\mathbf{e}_{\mathbf{v}_O}(k) = \mathbf{v}_O^d(k) - \mathbf{v}_O(k) = (c_O(k) - 1)\mathbf{v}_O(k)$$

where $\boldsymbol{p}(k)$ is the nearest point on a desired path from the instrument tip position, $\mathbf{x}(k)$ is the instrument tip position, and $\mathbf{v}_O^d(k) = c_O(k)\mathbf{v}_O(k)$. The position error is related to generating attractive force and the velocity error is related to FRPF.

Table 1. Rule-bases for the main FLC

E_1 \ E_2	NB	NM	NS	ZO	PS	PM	PB
NB	-1.0	-0.8	-0.6	-0.3	-0.2	0.2	0.2
NM	-0.8	-0.6	-0.4	-0.2	-0.1	0.2	0.3
NS	-0.4	-0.4	-0.3	-0.1	0.0	0.2	0.3
ZO	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3
PS	-0.3	-0.2	0.0	0.1	0.3	0.4	0.4
PM	-0.3	-0.2	0.1	0.2	0.4	0.6	0.8
PB	-0.2	-0.2	0.2	0.3	0.6	0.8	1.0

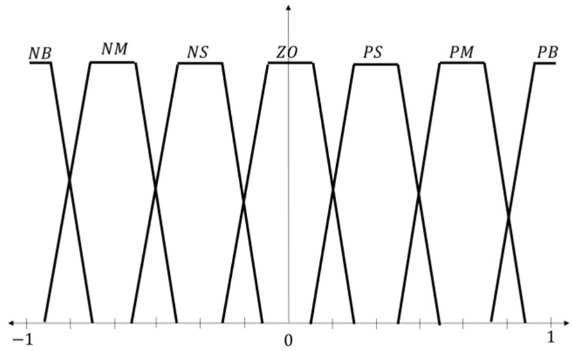


Figure 8. Membership functions for inputs

For input fuzzy variables, trapezoidal membership functions (Fig. 8) are used to characterize the seven fuzzy sets which are negative big (NB), negative medium (NM), negative small (NS), zero (ZO), positive small (PS), positive medium (PM), and positive big (PB).

Two input fuzzy variables are defined as

$$E_1(k) = S_1\,\mathbf{e}_d(k)$$
$$E_2(k) = S_2\,\mathbf{e}_{\mathbf{v}_O}(k)$$

where S_1 and S_2 are constant scaling factors for $\mathbf{e}_d$ and $\mathbf{e}_{\mathbf{v}_O}$, respectively, and $0 \le E_1, E_2 \le 1$. Also, S_1 and S_2 are determined by experimental results.

For an output fuzzy variable (Y), single-valued constant membership functions are used to characterize the output fuzzy sets; Y is restricted to $[-1,\ 1]$.

IF-THEN rules for the main FLC are represented in the following form:

$$R^l\text{: If } E_1 \text{ is } \mathrm{F}_1^l \text{ and } E_2\ is\ F_2^l, then\ Y^l\ is\ G^l$$

where F_1^l and $F_2^l \in$ {NB, NM, NS, ZO, PS, PM, PB}, G^l is a constant output value ($-1 \le G^l \le 1$), and l is the rule number.

Since there are seven fuzzy sets for each antecedent variable of the main FLC, the number of rules is 7^2.

For proper control action, we build the rule-bases as follows: if an instrument's tip deviates far away from a desired path, large attractive control force is applied to pull the instrument tip toward the desired path, whereas small control force is applied when the instrument tip is close to the desired path. If $\mathbf{e}_{\mathbf{v}_O}$ is large, then large control force is applied to restrict non-preferred motion. The rule-bases are shown in Table 1.

To calculate $U(k)$, the *main*FLC uses a singleton fuzzifier, product inference engine, and center average defuzzifier. Consequently, $U(k)$ is obtained from the following equation:

$$U(k) = \frac{\sum_{l=1}^{49} Y^l \mu^l(E_1)\mu^l(E_2)}{\sum_{l=1}^{49} \mu^l(E_1)\mu^l(E_2)}$$

where $\mu^l(E_1)$ and $\mu^l(E_2)$ are the degree of membership functions for E_1 and E_2, respectively, and Y^l is the singleton output value.

The overall output for the *main*FLC is given as follows:

$$u(k) = S_u(k)U(k)$$

where $S_u(k)$ is an output scaling factor. To adjust the control output using $S_u(k)$ for different trainees, we design a Scaling Factor Modifier (SFM) based on a heuristic self-tuning method [14]. For instance, assume a novice trainee holds a surgical instrument for a while when practicing a particular task. Then, he/she moves the instrument toward the reference point. At first, SFM will increase $S_u(k)$ until the instrument moves in the right direction. Then, SFM will adjust $S_u(k)$ based on the trainee's behavior.

Like the *main*FLC, the SFM is a fuzzy inference system that has two inputs and a single output. Two input fuzzy variables are defined as

$$E_3(k) = S_3\, u(k-1)$$
$$E_4(k) = S_4\, \mathbf{v_O}(k)$$

where S_3 and S_4 are constant scaling factors for $u(k-1)$ and $\mathbf{v_O}(k)$, respectively, and $0 \le E_3, E_4 \le 1$. Like the *main*FLC, the SFM uses trapezoidal membership functions and single-valued constant membership functions to characterize input fuzzy variables (E_3, and E_4) and the output fuzzy variable (W), respectively. Also, S_3 and S_4 are determined by experimental results.

IF-THEN rules for the SFM are represented in the following form:

$$R^m\text{: If } E_3 \text{ is } F_3^m \text{ and } E_4 \; is\; F_4^m, then\; W^m \; is\; V^m$$

where F_3^m and $F_4^m \in$ {NB, NM, NS, ZO, PS, PM, PB}, V^m is a constant output value ($-1 \le V^m \le 1$), and m is the rule number. The number of rules for the SFM is 7^2.

The considerations to build the rules of the SFM are listed below:

- When E_3 is ZO (i.e., $u(k-1) \approx 0$), then the SFM preserves the present output scaling factor because there is not enough information to adjust it.
- If the direction of $\mathbf{v_O}$ and the direction of feedback force are the same (e.g., $E_3E_4 > 0$), then the SFM decreases the output scaling factor because the instrument tip moves toward the desired path.
- If the direction of $\mathbf{v_O}$ and the direction of feedback force are different (e.g., $E_3E_4 < 0$), then the SFM increases the output scaling factor because the instrument tip deviates from the desired path (moves to a forbidden region).

Based on these considerations, we construct the rule-bases as shown in Table 2.

To calculate the output (w) of the fuzzy inference system, the proposed SFM uses a singleton fuzzifier, product inference engine, and center average defuzzifier. $w(k)$ is obtained from the following equation:

$$w(k) = \frac{\sum_{m=1}^{49} W^m \mu^m(E_3)\mu^m(E_4)}{\sum_{m=1}^{49} \mu^m(E_3)\mu^m(E_4)}$$

where $\mu^m(E_3)$ and $\mu^m(E_4)$ are the degree of membership functions for E_3 and E_4, respectively, and W^m is the singleton output value.

Table 2. Rule-bases for the scaling factor modifier

E_4 \ E_3	NB	NM	NS	ZO	PS	PM	PB
NB	-1	-1	-1	0	1	1	1
NM	-1	-1	-1	0	1	1	1
NS	-1	0	1	0	1	1	1
ZO	1	1	0	0	0	1	1
PS	1	1	1	0	1	0	-1
PM	1	1	1	0	-1	-1	-1
PB	1	1	1	0	-1	-1	-1

The final $S_u(k)$ is calculated using the following form:

$$S_u(k) = \begin{cases} S_u(k-1) + \alpha(1 - c_O(k))w(k), & w(k) \ge 0 \\ (1 + \beta w(k))S_u(k-1), & w(k) < 0 \end{cases}$$

where α ($\alpha > 0$) is an additive increase gain and β ($0 < \beta < 1$) is a multiplicative decrease gain. Additive Increase and Multiplicative Decrease (AIMD) algorithm is used for network congestion control and it guarantees convergence [15]. To adjust $S_u(k)$, we use the AIMD concept.

IV. Experimental Results

In CAST, each fixture has three motors, each to control yaw, pitch, and insertion axis. A motor is not attached to the roll axis because rotation does not affect the instrument tip position. The instrument tip position is updated every 50ms ($\Delta T = 0.05s$). By using the center of the instrument fixture gimbal as a local frame of reference, the tip position is calculated by the following equation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\emptyset) & -sin(\emptyset) \\ 0 & sin(\emptyset) & cos(\emptyset) \end{pmatrix} \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ INS \\ 0 \end{pmatrix}$$

where θ is a rotation angle around the yaw axis, $\emptyset$ is a rotation angle around the pitch axis, and INS is an insertion length.

To verify the proposed haptic feedback system, we use the right-hand fixture. The ranges of rotation angle θ and $\emptyset$ are $[-82.5^\circ, 0^\circ]$ and $[-180^\circ, 0^\circ]$, respectively. The range of insertion length INS is $[0\ cm, 14.9\ cm]$. The tip position ($\mathbf{x}(k)$), the nearest point ($\boldsymbol{p}(k)$), and $\mathbf{v}_O(k)$ are converted to encoder counter value to control each motor. The converted value ranges for θ, $\emptyset$, and INS are restricted to [0, 7019], [0, 6652], [0, 16781], respectively.

We design a simple hand-eye coordination task (such as the one shown in Fig. 2 where instrument tips must traverse the space along optimal, collision free trajectories) to verify the proposed haptic guidance system. The training scenario is such that a trainee must touch multiple targets (e.g., four targets labeled by R1, R2, R3, and R5) using the right instrument.

For the proposed system, design parameters that are determined from the experimental results are listed below:

- $S_1 = 1/1000$, $S_2 = 1/100$, $S_3 = 1/200$, $S_4 = 1/500$, $\alpha = 125$, and $\beta = 0.2$ for the yaw axis

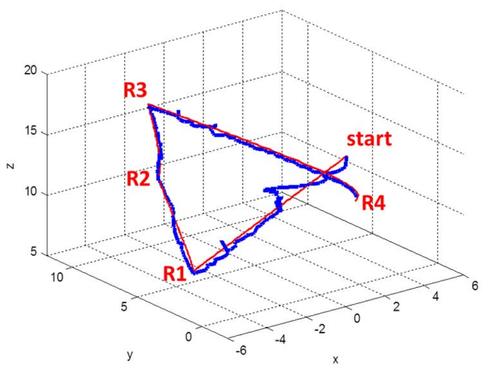


Figure 9. Practice trajectory (red: a desired path, blue: the right instrument tip movement). $\boldsymbol{r_d} = \mathbf{0.5cm}$.

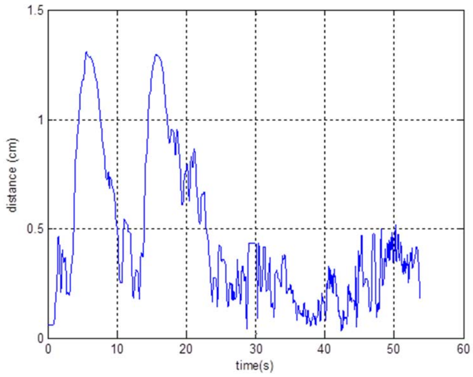


Figure 10. Distance ($\boldsymbol{r}$) profile while performing the hand-eye coordinate task. $\boldsymbol{r_d = 0.5}$**cm**.

controller.

- $S_1 = 1/2200$, $S_2 = 1/440$, $S_3 = 1/500$, $S_4 = 1/500$, $\alpha = 75$, and $\beta = 0.2$ for the insertion axis controller.
- $S_1 = 1/800$, $S_2 = 1/80$, $S_3 = 1/160$, $S_4 = 1/500$, $\alpha = 150$, and $\beta = 0.2$ for the pitch axis controller.
- S_u is restricted to [0, 15000] in order for a safety reason. $S_u^{yaw}(0) = 2500$, $S_u^{ins}(0) = 1500$, $S_u^{pit}(0) = 3000$

To define the virtual tube geometry, we set the tube radius (r_d) as 0.5cm. Fig. 9 illustrates the trajectory of the right instrument tip. The red line and blue line represent a desired path and the actual instrument tip movement, respectively. While doing this simple task, we always try to deviate from the desired path in order to evaluate the overall performance of the proposed feedback system. There is a huge deviation when the instrument traverses from the start position to target R1. Fig. 10 depicts the huge deviation for the first 23 seconds. The main reason for this large deviation is that the SFM is adjusting output scaling factors. Fig. 11 illustrates the behavior of the SFM. In this test, the tuning process is moderately slow. If we use a large value for α, we can speed up the tuning process. Also, if we increase the maximum value of S_u, we can reduce the deviation. After finishing the self-tuning process, the proposed system guides the user to move the instrument within the virtual tube area. The

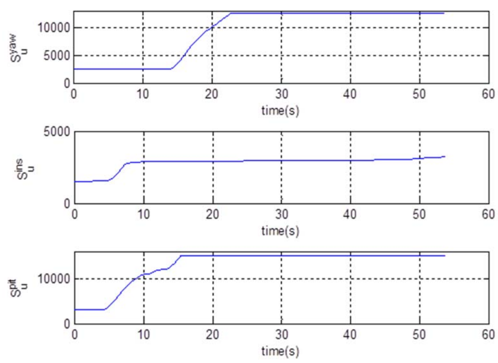


Figure 11. $\boldsymbol{S_u(k)}$ profile while performing the hand-eye coordinate task. $\boldsymbol{r_d = 0.5}$**cm**.

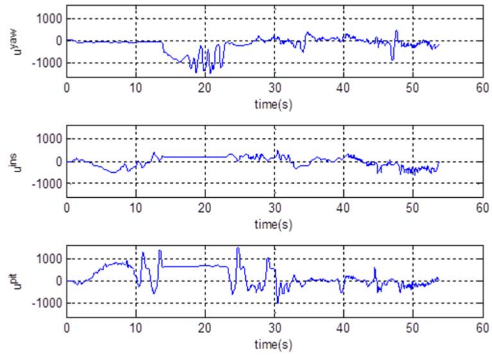


Figure 12. Control outputs while performing the hand-eye coordinate task. $\boldsymbol{r_d = 0.5}$**cm**.

control outputs are illustrated in Fig. 12. Whenever the instrument tip moves toward the forbidden region, the proposed guidance system applies large feedback force to restrict the motion.

Figs.13-16 illustrate the results of $r_d = 2.0$cm**.** As in the previous test case, we frequently try to deviate from the desired path. The results show that the haptic guidance system restricts non-preferred movements to minimize entry into the forbidden region by applying feedback force.

V. DISCUSSION AND CONCLUSION

By varying r_d, we can adjust the guidance level. We may use a trainee's proficiency level as the criterion to do so. For example, if a trainee is a novice, it may be helpful to restrict non-preferred motion actively by using small r_d. However, if a trainee's proficiency level is close to an expert level, the trainee may want to practice without any guidance force in order to prepare for specific surgical certification tests. For this trainee, the system sets r_d as a large value so that the guidance system tries to correct only major mistakes. To implement this adaptive feature, we will design a decision process to adjust r_d dynamically based on the assessment results generated by *opt*Assessment, the module in CAST that collects data on users' performance (e.g., the accuracy, speed, or economy of movement, etc.).

As a future work, we will consider system stability in order to prove that the proposed controller is stable given bounded inputs. Also, more complex geometries will be designed to verify the proposed control system. Finally, a human subjects study will be conducted by enrolling

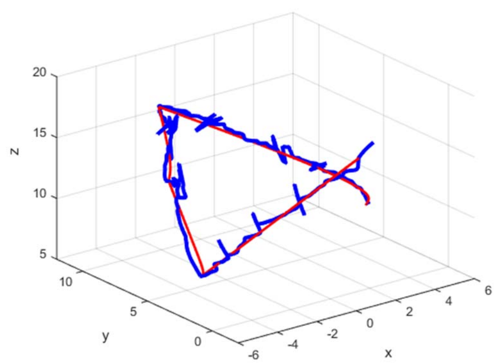


Figure 13. Training trajectory (red: a desired path, blue: the right instrument tip movement). $\boldsymbol{r_d = 2.0}$**cm**.

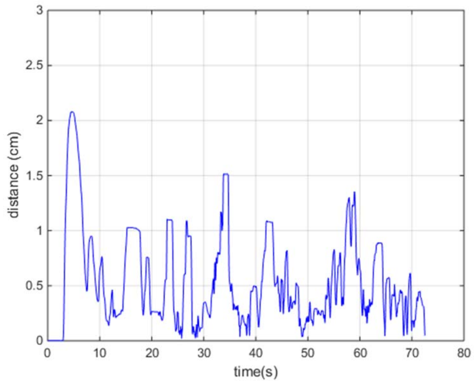


Figure 14. Distance ($\boldsymbol{r}$) profile while performing the hand-eye coordinate task. $\boldsymbol{r_d = 2.0}$**cm**.

volunteers and dividing them into sub-groups in accordance with their expertise in laparoscopic surgery.

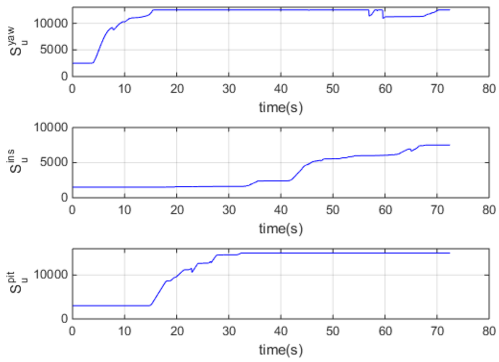


Figure 15. $\boldsymbol{S_u(k)}$ profile while performing the hand-eye coordinate task. $\boldsymbol{r_d = 2.0}$**cm**.

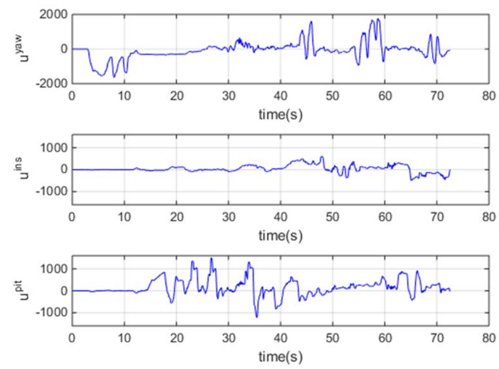


Figure 16. Control outputs while performing the hand-eye coordinate task. $\boldsymbol{r_d = 2.0}$**cm**.

In this paper, we have presented a design method for a haptic guidance controller with "forbidden region" virtual fixtures. Like traditional FLCs, the proposed system does not require high computing power because of using simple multiplications and a reasonable number of rules. The proposed controller enables haptic guidance in a system specifically designed for laparoscopy training. The key contribution is the ability to restrict non-preferred movements using the forbidden region virtual fixture and to adjust control outputs based on a self-tuning scheme. Our experimental results are promising and indicate that the system will be relatively robust. This work can be extended to applications beyond medical training, namely to all human-machine collaboration systems.

REFERENCES

[1] Rozenblit, Jerzy W. "Models and techniques for computer aided surgical training." In Computer Aided Systems Theory–EUROCAST 2011, pp. 233-241, 2011.

[2] eoSurgical, http://www.eosurgical.com.

[3] Jaber, Nidal. "The basket trainer: A homemade laparoscopic trainer attainable to every resident," Journal of minimal access surgery, vol. 6, no. 1, pp. 3-5, 2010.

[4] Simbionix Co., http://www.simbionix.com.

[5] Stylopoulos, N., S. Cotin, S. K. Maithel, M. Ottensmeyer, P. G. Jackson, R. S. Bardsley, P. F. Neumann, D. W. Rattner, and S. L. Dawson. "Computer-enhanced laparoscopic training system (CELTS): bridging the gap," Surgical Endoscopy and Other Interventional Techniques, vol. 18, no. 5, pp. 782-789, 2004.

[6] Rosenberg, Louis B. "Virtual fixtures: Perceptual tools for telerobotic manipulation." In Virtual Reality Annual International Symposium, pp. 76-82, 1993.

[7] Rozenblit JW, Feng C, Riojas M, Napalkova L, Hamilton AJ, Hong M, Berthet-Rayne P, Czapiewski P, Hwang G, Nikodem J, Shankaran A. "The Computer Assisted Surgical Trainer: Design, Models, and Implementation," Proc. of the 2014 Summer Simulation Conference, Modeling and Simulation in Medicine, pp. 211-220, 2014.

[8] Napalkova, Liana, Jerzy W. Rozenblit, George Hwang, Allan J. Hamilton, and Liana Suantak. "An Optimal Motion Planning Method for Computer-Assisted Surgical Training," Applied Soft Computing, vol 24, pp. 889-899, 2014.

[9] Riojas, Mario, Chuan Feng, Allan Hamilton, and Jerzy Rozenblit. "Knowledge elicitation for performance assessment in a computerized surgical training system," Applied Soft Computing, vol. 11, no. 4, pp. 3697-3708, 2011.

[10] Bowyer, Stuart A., Brian L. Davies, and Ferdinando Rodriguez y Baena. "Active constraints/virtual fixtures: A survey." Robotics, IEEE Transactions on, vol. 30, no. 1, pp.138-157, 2014.

[11] Bettini, Alessandro, Panadda Marayong, Samuel Lang, Allison M. Okamura, and Gregory D. Hager. "Vision-assisted control for manipulation using virtual fixtures." Robotics, IEEE Transactions on, vol. 20, no. 6, pp. 953-966, 2004.

[12] Teo, Chee Leong, Etienne Burdet, and H. P. Lim. "A robotic teacher of Chinese handwriting," Proc. of the IEEE Virtual Reality Conference (HAPTICS), pp. 335-341, 2002.

[13] Forsyth, Benjamin AC, and Karon E. MacLean. "Predictive haptic guidance: intelligent user assistance for the control of dynamic tasks," Visualization and Computer Graphics, IEEE Transactions on vol. 12, no. 1 pp. 103-113, 2006.

[14] Chou, Chih-Hsun, and Hung-Ching Lu. "A heuristic self-tuning fuzzy controller." Fuzzy Sets and Systems vol. 61, no. 3, pp. 249-264, 1994.

[15] Chiu, Dah-Ming, and Raj Jain. "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks." Computer Networks and ISDN systems, vol. 17, no. 1, pp. 1-14, 1989.