Implementation of Computer-Guided Navigation in Surgical Training

CONFERENCE PAPER · APRIL 2013

DOI: 10.1109/ECBS.2013.31

2 AUTHORS:

George Hwang
The University of Arizona

Jerzy W Rozenblit
The University of Arizona

Implementation of computer-guided navigation in surgical training

George Hwang and Jerzy W. Rozenblit
Electrical and Computer Engineering Department
University of Arizona
Tucson, AZ 85721
hwangg{jr}@ece.arizona.edu

***Index Terms*—minimal invasive surgery; model-based simulation; method validation; surgical training system**

***Abstract*—Simulation is becoming a standard means of surgical training. This paper describes the development of the haptic element of a simulation training system called Computer Assisted Surgical Trainer (CAST) III. Research is being done to show that the combination of graphical and haptic guidance may be an effective technique in achieving optimal path enumeration in a three-dimensional space. The haptic feedback system used in CAST III provides a tactile dimension of training. The implementation focuses on the design of a multi-axis control system. This implementation provides an element of a full-spectrum navigation system that trains users in optimal path enumeration given a limited perceptual environment.**

I. Introduction

The medical community is going through a quiet crisis. Rising costs and gaps in coverage in the USA are problem areas that threaten care affordability and the quality of medical care for all people in the United States [1]. One acknowledged problem area are some inefficiencies in training medical students. Current medical training utilizes the Halsted paradigm, which is basically a master and apprentice model [2]. This is inefficient and costly. Senior physicians have limited time to mentor students. Students and residents require years of training to become proficient.

One solution that has started to gain attention is the use of Computer-Based simulation to augment the training process. There have been numerous research systems that have tackled this issue. These systems tend to be feature-rich. However, numerous features do not necessarily ensure the quality of training. The work by Stefanidis [3] illustrates this. As such, it is imperative to find systems that balance the availability of features with the efficacy of training.

We envision a system that not only provides more dimensions, but is also well grounded in human factors research. We have taken the approach of utilizing Graphical and Haptic Guidance to aide in training within this domain. Our approach is grounded in a body of research in done by Feygin [4] and validated by Morris [5], which indicates that the use of both graphical and haptic guidance is effective in training the hand-eye coordination in a limited sensory environment. We implemented our system within the domain of laparoscopic surgery. We designed a system that we named CAST III, which utilizes haptic and visual feedback. The augmentation and work that we have done is best described in Figure 1. In this figure, we emphasize the use of feedback, control, and multiple sensory data.

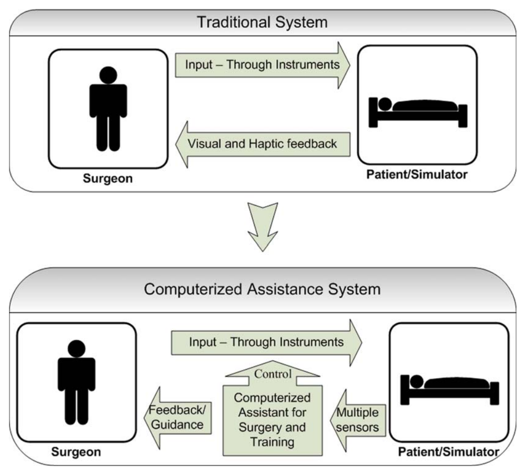

Fig. 1. Laparoscopic Surgery

II. Related Work

This work derives heavily from other research in the fields of control, modeling, and software engineering. References to these works are made throughout this paper. This section gives a short description of their uses as they pertain to CAST III.

This paper is a subset of the work on CAST III by Hwang [6]. CAST III is the total system describing the haptic and visual system. Here, we focus on control and reference points generation. The detailed description of the underlying hardware and visual platform is not within the scope of this paper.

The control scheme used in this work heavily utilizes the work of Dong et al [7], which uses a feedthrough term to synchronize all axes as seen in Equation 1. We modified the equation to Equation 2 for ease of implementation in our system. As such, the basic control scheme is a PD controller with a feedthrough for synchronization. We discuss this result later in the paper.

$$\tau = K_p \cdot E + K_d \dot{E} + (I + \alpha \cdot T)^{-1} \cdot \dot{e} \tag{1}$$

978-0-7695-4991-0/13 $26.00 © 2013 IEEE
DOI 10.1109/ECBS.2013.31

$$y = K_p \cdot e(t) + K_d \cdot \frac{e(t) - e(t-1)}{T_{sample}} \qquad (2)$$

The reference generator relies on our earlier work in Nikodem et al [8]. In that paper, we utilized a state-based approach with different control schemes for different states. The central thrust of that work was the division into zones of areas in 3D space. Each zone had different behaviors depending on the location of the instrument tip. In this work, we decided to maintain the state-based [9] and zone approach for the generation of references, but not to tie it to a control scheme per se. We used three zones centered around one reference on the optimal path. As seen in Figure 3, the zones are essentially a ball around the reference. With each zone, there is an associated state as seen in Figure 3. More information on this subject is given in the the Hardware section.

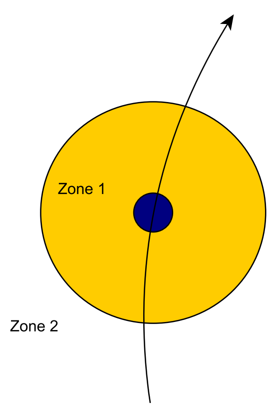

Fig. 2. Zones

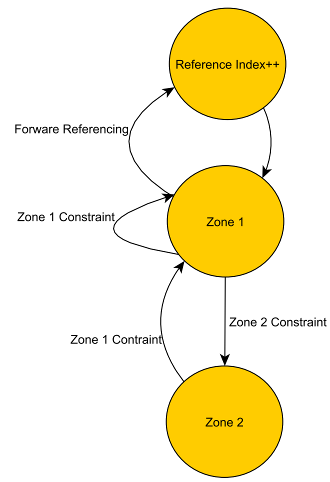

Fig. 3. Zone State Charts

III. System Description

Figure 4 illustrates the overall CAST III system. The intention of CAST III is to assist trainees in a limited perceptual environment for laparoscopic surgery. The overall system accomplishes this by providing both haptic and graphical feedback to the users. One of the most difficult aspects of laparoscopic training is gauging depth of the operating field as views through an endoscopic camera. In addition, hand-eye coordination is difficult because of the limitation of the degrees of freedom compared to a traditional, open cavity surgery. The graphical feedback system provides dimensions that otherwise cannot be seen by a camera alone. The haptic system gives tactile feedback to force the user back onto an optimal path.

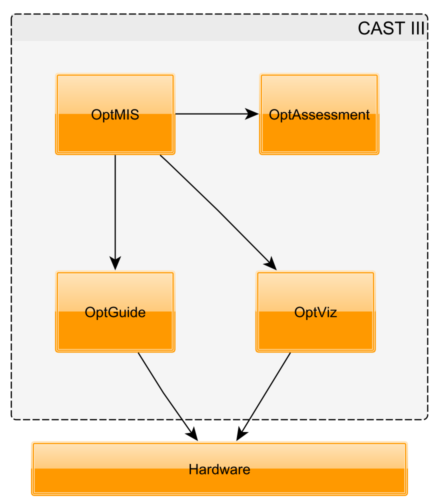

Fig. 4. Architectural Overview of the CAST III system

These qualities are manifested in the elemental blocks of the system. OptMIS is the system that generates Optimal Path data, as described by Napalkova et al [10]. The optimal path is generated from an input of obstacles in a working space. The path is generated offline and fed into both OptViz and OptGuide. OptViz is the graphical guidance system implementation by Hwang [6]. This system augments the user's view by providing a 3D visualization environment with multiple views of a working environment. OptAssessment refers to prior work done by Riojas et al. [11] to classify novice surgeons in relation to experts; Classification is accomplished by a fuzzy logic system. The hardware is the central platform used to measure position and orientation of the laparoscopic instrument. This encompasses motors for the yaw, pitch, insertion, and rotation used for haptics. Hwang [6] and Nikodem et al [8] describe the hardware extensively.

The thrust of this paper covers OptGuide. This is the haptic guidance system used in CAST III, which encompasses both control and software implementations. In addition, this paper analyzes the system and gives the rationale behind our design. The analysis has its roots in control theory and software systems design.

IV. Hardware

We discussed the solution to enumerate through the optimal instrument path in our previous work [8]. Tests of our solution indicated that more work was necessary. We decided to use a simpler method utilizing a PD controller based on the work of Dong et al. [7]. This work has proven to be stable. It also synchronizes each axis so that the tip movement follows a straight line. However, we did modify it for our own needs. For this reason, we also performed analysis of the algorithm for stability prior to using it on CAST III.

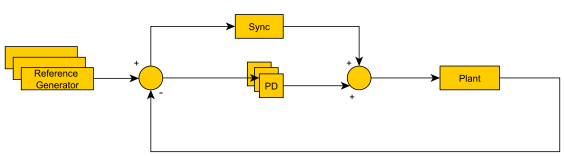

Fig. 5. Controller

As shown in Figure 5, our proposed controller utilizes a PD controller with a synchronization feed-through. We used a PD controller to control each individual axis (yaw, pitch, and insertion). The implementation for the PD controller uses its discrete form, which is shown in Equation 3. Synchronization of the yaw, pitch, and insertion axes uses the feed-through term, represented as "sync" in Figure 5. This term provides a simple way to compensate for gravity and inertia. Each axis has gravity exerted on it differently, based on the position and orientation of the instrument. Inertial parameters also change based on orientation and position. The feed-through term, described by $K_{sync} \cdot \sigma$ in Equation 6, accomplishes this by ensuring that if one axis starts to dominate, it appropriately compensates in the other axes. The term for this type of compensation is Type II Synchronization by Dong et al. [7].

$$y = K_p \cdot e(t) + K_d \cdot \frac{e(t) - e(t-1)}{T_{sample}} \tag{3}$$

$$\sigma = \begin{pmatrix} 2 \cdot yawError - (pitchError + insError) \\ 2 \cdot pitchError - (yawError + insError) \\ 2 \cdot insError - (pitchError + yawError) \end{pmatrix} \tag{4}$$

$$error = \begin{pmatrix} yawError \\ pitchError \\ insError \end{pmatrix} \tag{5}$$

$$gain = K_p \cdot error + K_d \cdot error_d + K_{sync} \cdot \sigma \tag{6}$$

We also implemented a reference generator that uses the optimal path data generated by OptMIS [10]. OptMIS generates a series of discrete sequences of points representing the optimal path. Each of these points can be a reference. Figure 6 illustrates the current reference point as a ball on the optimal path. This ball relates to the zone scheme seen in Figure 2.

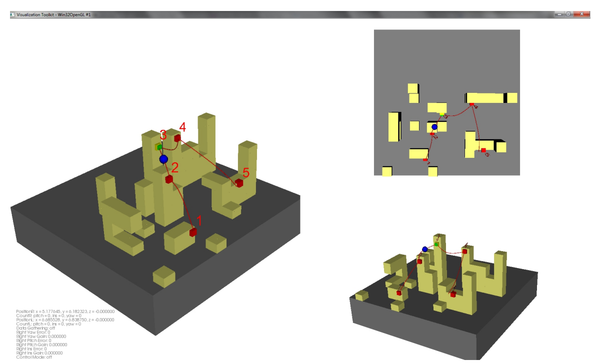

Fig. 6. Reference Ball

Each zone has a certain constraint. Zone 1 surrounds the ball with a radius of 0.5 cm (described in Equation 7). When the instrument is in Zone 1, it is in the Zone 1 state as illustrated in Figure 3. The Zone 2 state is anywhere outside of the Zone 1 state described in Equation 8.

$$\text{Zone 1 Constraint:} \quad OptPath_n - 0.5cm >= position \tag{7}$$

$$\text{Zone 2 Constraint:} \quad OptPath_n - 0.5cm < position \tag{8}$$

The transition between Zone 1 and Zone 2 is also where the reference increments. However, this is based on a narrow window that requires proximity to the optimal path forward of the current reference. We implemented what we call "forward referencing" for this updating. There are multiple constraints that need to be met. First, users need to be no more than

1cm and no less than 0.5cm from the current reference point. Second, users need be within 0.5 mm of a point in which its optimal path index is greater than the current reference point optimal path index. Equation 9 describes these constraints, where $OptPath_n$ is the current position of the optimal path indexed by n and $position$ is the current user position.

$$\text{Forward Referencing:} \quad \begin{aligned} &OptPath_n - 0.5cm <= position && \cap \\ &position <= OptPath_n + 1cm && \cap \\ &k > n && \cap \\ &position <= OptPath_{n+k} + 0.5 && \cap \\ &OptPath_{n+k} - 0.5 <= position && \end{aligned} \tag{9}$$

V. Stability Analysis

We use two methods to validate stability: linearizing at a point followed by analysis and experiments. For both methods, a weight of one (1) for the proportional and 0.5 for the derivative term are initially selected for analysis and validation. We find stability by iteratively modifying each value, and then re-testing. We further adjusted the values to achieve a "feel" performance that is comfortable to the user. Table I illustrates the values we found to be optimal.

TABLE I
Optimal Gains

Axis	Proportional	Derivative
Yaw	0.7	0.1
Pitch	0.9	0.1
Insertion	0.7	0.1
sync	0.2	0

A. Linearization

Stability analysis, based on the selected gains, requires linearizing the feedback model shown in Figure 5. Linearization utilizes the "linmod" command on the feedback system described in the previous section with an input of 0. Equation 10 illustrates the minimized realization of the resulting linear state space for the yaw.

$$\begin{aligned} A &= \begin{pmatrix} -4.167e18 & -4.996e8 & -3.464e11 & 4.167e18 & -8.138e9 \\ 9153 & -1.569 & 0.0007609 & 9153 & -1.788e-5 \\ 0 & 1 & 0 & 0 & 0 \\ 1.582e11 & -18.97 & -1.315e4 & -1.582e11 & -309 \\ 0 & 0 & -3274 & 0 & -100 \end{pmatrix} \\ B &= \begin{pmatrix} 1.058e9 \\ 2.324e6 \\ 0 \\ 40.18 \\ 10 \end{pmatrix} \\ C &= \begin{pmatrix} 0 & 0 & 327.4 & 0 & 0 \end{pmatrix} \\ D &= 0 \end{aligned} \tag{10}$$

When all of the real values of the Eigenvalues of an A matrix are negative for a linear state space, a system is stable [12]. Based on Equation 11, yaw is stable. This validates the yaw gains.

$$eig(A) = \begin{pmatrix} -4.1667\mathrm{e}18 \\ -4.9890\mathrm{e}3 \\ -1.0008\mathrm{e}2 + 3.0975\mathrm{e}2i \\ -1.0008e\mathrm{e}2 - 3.0975\mathrm{e}2i - 21.0232 \end{pmatrix} \tag{11}$$

The pitch uses the same model as the yaw, with the exception that there is a static gravitational torque of 0.44 Nm applied to the DC motor. Using the same analysis as above, but with a gravitational torque of 0.44 Nm, will yield a state space equation with the same A matrix as the yaw. Since the yaw system is stable, the pitch axis is also stable.

The insertion uses the same gravitational torque as the pitch axis. This results in the exact same equation found for the pitch axis. Since the pitch axis is stable, the insertion is too.

B. Experiments

The synchronization feed-through term is partly derived from the model-free cross coupled controller used for position synchronization [7]. However, there were modifications of the control equation for implementation simplicity, as seen in Equation 6. The cross-coupled synchronization controller [7] guarantees asymptotic convergence for any mechanical model. However, the controller architecture of CAST III is different. This requires a re-evaluation of stability with the CAST III architecture. The approach to test for stability utilizes experimentation. This will only prove that a sub-set of scenarios are stable. The designed experiments will cover the possible usage range of CAST III.

Two experiments represent the scenario which CAST III will encounter. As long as users use CAST III within the bounds established by these experiments, we can guarantee stability. The first experiment enumerates through live paths with some deviation. This exercises the haptic feedback system in its normal operation. In this mode, the system will nudge the user back with a force proportional to the distance and the force balanced between different axes. This experiment shows no unstable behavior. The second experiment is conducted by pulling the instrument to its furthest possible distance and then releasing the instrument. The corresponding reference for this test is x=9, y=2.8, and z=6.9. This corresponds to 10,000 for the insertion encoder count, 1,400 for the pitch encoder count, and 7000 for the yaw encoder count. The intended result is for the instrument to return "near" the stated position without any instability. This is roughly 0.5cm from the desired position. Tracking perfectly is not necessary. Figures 7, 8, and 9 are the results for each axis. This data shows that there are no cases of overshoot and that the system is stable.

VI. Software

The entry point for OptGuide is a single class defined as controller. The CAST III software instantiates it. The controller class contains three classes that represents the controller

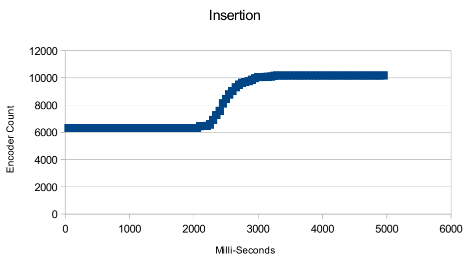

Fig. 7. Insertion Synchronization Measure

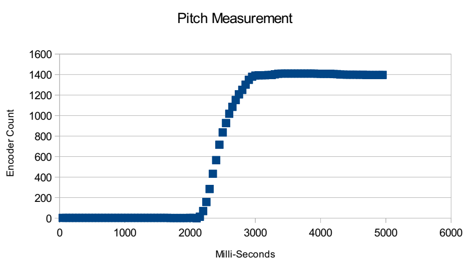

Fig. 8. Pitch Synchronization Measure

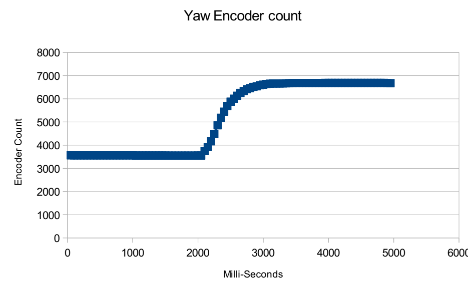

Fig. 9. Yaw Synchronization Measure

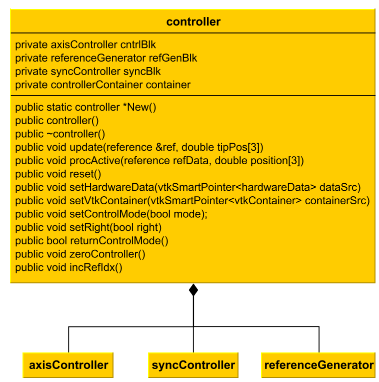

Fig. 10. Reference Generator

blocks illustrated in Figure 5: “sync” block, “PID” block, and “Reference Generator” block. In addition to its role as the OptGuide entry point, it also ties the three contained blocks together. It further provides methods to the stated information in each of its contained blocks. Figure 10 illustrates this. We discuss the details of the “sync”, “PID”, and “Reference Generator” blocks. These are the essential software modules that define this system. Hwang [6] presents further details.

The “PID controller” block implements the PD controller based on Equation 3. The class implementation is for a general PID controller defined in Equation 12. The “set” function sets the gains for each of the PID’s weighted sum, which are private variables, in the PID class. The getGain method is called for each sample point. CAST III samples the axis data every 50 ms, thus getGain is called in that period. There are three instantiations of the PID Controller class in the axisController class. The instantiations represent each of the axes: yaw, pitch, and insertion. Figure 11 illustrates the UML diagram.

The “PID controller” block has several important features for configuration. The maximum gain for P, D, and I is the maximum value of a 32 bit integer. There is a limiter for the range of output gains generated by the PID controller, where the initial value is zero. This limiter allows the user to prevent the output of this block from exceeding an output that might cause damage to the system. This limiter is also utilized to clip the summation of the I module in the PID controller. Any Integral summation greater than the range limit will not occur.

$$\begin{aligned}
\sigma_0 &= K_p \cdot e(t) \\
\sigma_1 &= K_i \cdot T_s \cdot \sum e(t) \\
\sigma_2 &= K_d \cdot \frac{e(t) - e(t-1)}{T_s} \\
y &= \sigma_0 + \sigma_1 + \sigma_2
\end{aligned} \tag{12}$$

The “sync” block implements the behavior described by $K_{sync} \cdot \sigma$ in Equation 6. This block prevents any one axis from dominating by adjusting the gain of the dominant axis. The sync class implements this and is only instantiated once. However, this class is called three times to generate the appropriate gain for each axis. The getGain method is the entry point, where the axis is a function argument. This class has one private variable that defines the gain for the Level II Synchronization Feedthrough [7]. Figure 12 illustrates the UML structure for this class.

The “Reference Generator” block generates references based on data provided from OptMIS [10]. The reference-Generator class implements this block. The controller block instantiates it once. The main entry point for this class is the genGain method, which updates the reference index based on the constraints defined in Equation 9. There are also methods to restart the index from the beginning and to set the current index. The reset and setRefIdx methods implement these respectively. The controller class provides a wrapper function that utilizes these functions as well. The controller

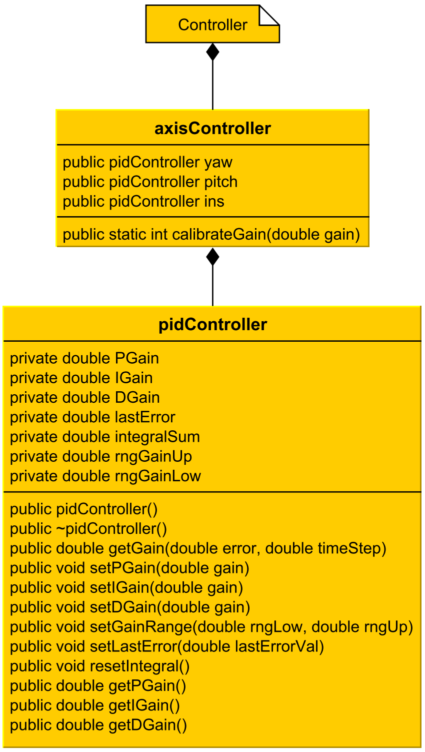

Fig. 11. PID Controller Implementation

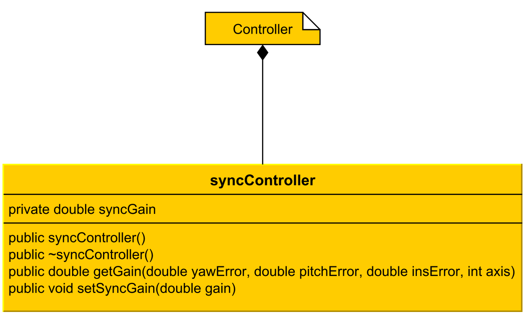

Fig. 12. Synchronization Controller

class method names are the same as the referenceGenerator method. The reference structure stores the current reference point information. Figure 13 illustrates the UML structure of this class.

VII. Conclusions

The haptic system described in this paper has met the our objectives. The only caveat is that the original objective of

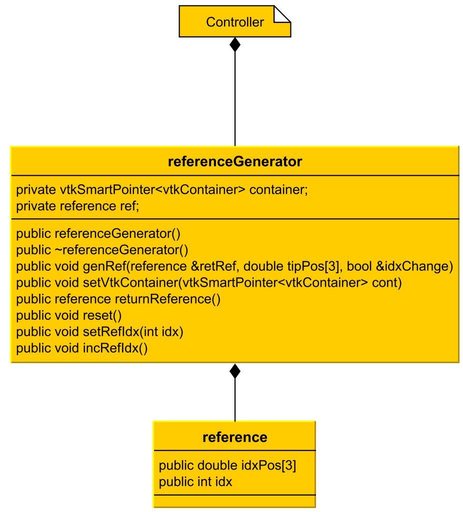

Fig. 13. Reference Generator

stability is limited to a subset of cases. As long as the haptic system in CAST III is within the cases stated, stability is ensured.

A study to confirm the efficacy of computer guided vs. non-computer guided training has not been performed yet. Therefore, we cannot provide statistical data attesting to how good the training outcomes are if CAST's control system has been activated. Such a study will be the subject of future research and a separate publication. However, we have done initial exploration of the system with our development team. We have extensive experience in designing studies that will prove (or disprove) the efficacy of CAST with guided navigation. We will draw from previous work documented in [13] and [14]. Using this experience, we intend to conduct a proper experiment at a later date. Thus, CAST III is now ready for the next step, which is the validation of the graphical and haptic features in a laparoscopic training surgical environment.

References

[1] I. o. M. Committee on Quality of Health Care in America, *Crossing the Quality Chasm: A New Health System for the 21st Century*. The National Academies Press, 2001.

[2] J. Carroll and J. Messenger, "Medical simulation: the new tool for training and skill assessment.," *Perspectives in Biology and Medicine*, vol. 51, no. 1, pp. 47–60, 2008.

[3] D. Stefanidis, J. Korndorffer, R. Sierra, C. Touchard, J. Dunne, and D. Scott, "Skill retention following proficiency-based laparoscopic simulator training.," *Surgery*, vol. 138, no. 2, pp. 165–70, 2005.

[4] D. Feygin, M. Keehner, and R. Tendick, "Haptic guidance: experimental evaluation of a haptic training method for a perceptual motor skill," *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, pp. 40 –47, 2002.

[5] D. Morris, H. Tan, F. Barbagli, T. Chang, and K. Salisbury, "Haptic feedback enhances force skill learning," *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, pp. 21 –26, March 2007.

[6] G. Hwang, "Motion control and guidance in the computer assisted surgical trainer," Master's thesis, University of Arizona, 2012.

[7] D. Sun, X. Shao, and G. Feng, "A model-free cross-coupled control for position synchronization of multi-axis motions: Theory and experiments," *Control Systems Technology, IEEE Transactions on*, vol. 15, pp. 306 –314, March 2007.

[8] J. Nikodem, G. Hwang, J. W. Rozenblit, and L. Napalkova, "Technical realization of the optimal motion planning method for minimally invasive surgery," *Engineering of Computer Based Systems (ECBS), 2012 IEEE 19th International Conference and Workshops*, pp. 195–200, April 2012.

[9] D. Harel, "Statecharts: A visual formalism for complex systems," *Sci. Comput. Program.*, vol. 8, pp. 231–274, June 1987.

[10] L. Napalkova, J. W. Rozenblit, G. Hwang, A. Hamilton, and L. Suantak, "Optimal motion planning method for computer-assisted surgical training system," *Applied Soft Computing*, submitted for publication".

[11] M. Riojas, C. Feng, A. Hamilton, and J. Rozenblit, "Knowledge elicitation for performance assessment in a computerized surgical training system," *Applied Soft Computing*, vol. 11, no. 4, pp. 3697 – 3708, 2011.

[12] G. F. Franklin, D. J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 4th ed., 2001.

[13] M. Riojas, C. Feng, A. Hamilton, and J. Rozenblit, "Knowledge elicitation for performance assessment in a computerized surgical training system," *Appl. Soft Comput.*, vol. 11, pp. 3697–3708, June 2011.

[14] C. Feng, J. Rozenblit, and A. Hamilton, "A computerized assessment to compare the impact of standard, stereoscopic, and high-definition laparoscopic monitor displays on surgical technique," *Surgical Endoscopy*, vol. 24, pp. 2743–2748, 2010.