2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

# A Constrained Optimization Approach to Virtual Fixtures for Multi-Handed Tasks

Ankur Kapoor and Russell H. Taylor

*Abstract*— **In this work, we have extended the concept of constrained motion control of robots to surgical tasks that require multiple robots. We present virtual fixtures to guide the motion of multiple robots such that spatial and temporal relationship is maintained between them. At the same time, our algorithm keeps the surgeon in the control loop. Moreover, we show that these virtual fixtures allow bimanual tasks to be completed using input for a single robot. That is, the user requires only one hand to cooperatively control multiple robots. This reduces the cognitive load on the surgeon and makes multiple-robot setup for surgery more relevant. We demonstrate this architecture by using an example of manipulating a surgical knot to position it at a target point. Significant improvement is observed in the accuracy when bimanual virtual fixture assistance is provided. Moreover, the accuracy when using a single input from user is similar to the accuracy obtained from bimanual assistance.**

The benefits of Minimally Invasive Surgery (MIS) over conventional open surgery are well known, however, the surgeon faces the challenge of limited and constrained motion as well as loss of direct visualization. Over the last decade, several surgical assistant systems have been developed [1]–[4] to assist in MIS and some are becoming common place in hospitals [5]. However, in the various surgical systems described above, the surgical procedures are still performed by the surgeon; the robotic device merely follows the human commands.

Planning a minimally invasive intervention of soft tissue is not a trivial task. The organs move around and thus, the preplanned motion based on pre-operative images is of little use. Unlike bone machining, which requires minimally surgical intervention during the process, surgeons must make decisions and evaluations during procedures. Nevertheless, many researches have automated parts of surgical intervention. For example, researchers [6]–[8] have focused on the knot tying aspects of interventions. Mayer *et al.* [8] use a supervised machine learning algorithm on trajectories recorded during actual trials by surgeons. Thus they can generate a semi-automated procedure that can be "played back" by the robot at a later time, thus allowing automatic task completion.

But the uncertain and varying nature of surgical procedures makes building robust controllers a difficult problem. Therefore, we support the alternative approach of "surgeon-in-the-loop" controller that allows reduction in the cognitive load of the surgeon while keeping the surgeon in command at all times. Recently, Lin *et al.* [9] presented an algorithm that captures experts (surgeons) performing the task and builds a language for surgery. Our approach is to develop methods to transform these human-understandable surgical primitives obtained from such methods into assistive motion control algorithms. To this effect, we had presented a library of task primitives [10], [11] that can be combined using different operators to provide assistance for complex surgical tasks. The main contribution of this paper is to extend this prior work for tasks that require multiple hands.

The motivation for this extension is twofold. First, most surgical tasks require two or more arms. A predetermined spatial and temporal relationship exists between the arms to accomplish a desired task. Thus, a multi-robot virtual fixture must not only guide the individual robots but also assist in maintaining this relationship. Next consider a multi-arm robot setup such as the commercial daVinci system that can have up to four patient-side arms. If one arm holds the laparoscope, it still leaves three arms free for surgical manipulation. It would be beneficial if the surgeon could perform some critical task with his/her dominant hand, while his/her non-dominant hand is performing a routine task such as knot tying and the "intelligent" controller is moving the third arm to assist in completing the knot task. This is the other motivation to consider multiple robot virtual fixtures.

Several researchers have applied virtual fixtures for different surgical situations; References [12]–[17] list a small selection. To the best of our knowledge, this is the first application of surgical assistance that provides spatial and temporal guidance in performing a task using multiple robots sharing the same workspace.

In section II, we present a task that involves placing a knot using a dual robot arm. This bimanual task demonstrates our algorithm's ability to handle complex spatial and temporal relations between two or more arms. Moreover, we show that there is an improvement in task performance when such motion constraints are used to provide assistance. In this preliminary study, we show feasibility of our approach and demonstrate the promise of cooperative assistance for complex bimanual tasks such as knot placement.

## I. Constraint Control for Multiple Robots

### A. Surgical Assistant Architecture

The heart of the surgical assistant architecture is the constraint optimization block, which gathers the real-time inputs from different sources and computes a desired incremental motion for the robot based on constraints and objectives. In our earlier work [10], [11], we had shown methods to generate basic virtual fixtures for a single robot using the optimization block. Multiple robot control is different from bilateral control for a teleoperator where a single "slave" robot mimics the "master" manipulator. By multiple robot

control, we imply that there are two or more robots that may or may not be constrained to perform some motion relative to each other within a shared workspace. A common example is to have two cooperatively controlled robots, such as the Steady Hand [19], assisting the surgeon. Another example is the daVinci system, where a master-slave pair is associated with each of surgeon's hands. In this example, there are four robots but only two pairs that can be related to each other with a complex relation. Teleoperator controller determines the relationship between the robots in each pair. Thus, in multiple robot control we incorporate complex virtual fixtures that assist the surgeon to maintain complex spatial and temporal relationships between different robots involved in the task.

### B. Architectures for Multiple Robots

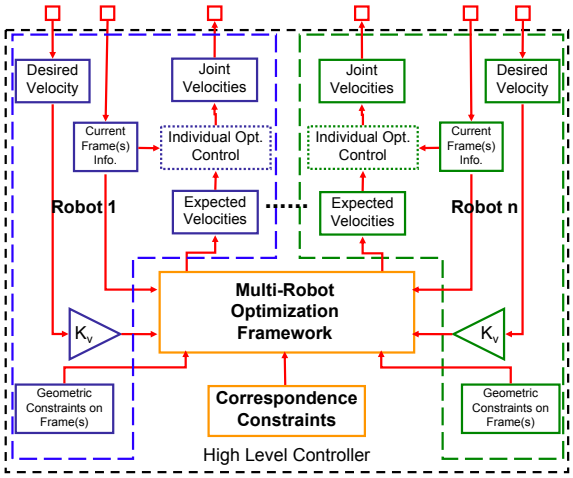


Fig. 1. The architecture for a distributed multiple robot constrained optimization block.

The architecture for the more general case of a distributed algorithm for applying the constrained control algorithm is shown in Figure 1. In the distributed algorithm, the control is divided into $n+1$ optimization problems. The $n$ optimization problems are for individual robot control and there is one combined problem to be solved for all the $n$ robots. The combined problem has constraints that relate the different robots under consideration, besides the task based constraints that can be applied to frames on individual robots. The output of the multirobot optimization framework is the expected incremental Cartesian motions for the individual robots. The input for the combined problem comes from the desired incremental Cartesian motions for the individual robots. Assume that we denote the desired and expected incremental Cartesian motion of the $i^{th}$ robot by ${\Delta\mathbf{x}_i}^d$ and ${\Delta\mathbf{x}_i}^e$, respectively. Then, the combined problem can be written as in (1). In (1), $C_i$ is a cost function for the $i^{th}$ robot that depends on the expected and desired incremental Cartesian motions of that robot. $A_j$ is the $j^{th}$ constraint that relates the motion of the different robots. It could also be a task based constraint on the $i^{th}$ individual robot, in which case it would depend only on ${\Delta\mathbf{x}_i}^e$. $\mathbf{W}_i$ is a diagonal matrix that relates the relative importance of minimizing the objective function cost between different robots.

$$\begin{bmatrix} {\Delta\mathbf{x}_1}^e \\ \vdots \\ {\Delta\mathbf{x}_n}^e \end{bmatrix}^* = \begin{array}{l} \arg\min \sum_{i=1}^{n} \mathbf{W}_i C_i({\Delta\mathbf{x}_i}^e, {\Delta\mathbf{x}_i}^d) \\ \text{s.t.} \quad A_j({\Delta\mathbf{x}_1}^e, \cdots, {\Delta\mathbf{x}_n}^e) \leq b_j \\ \quad\quad j \in [1, m] \end{array} \tag{1}$$

The expected Cartesian incremental motions are inputs for the individual optimization blocks. The incremental joint motions are the outputs of the individual optimization blocks, which are the result of an optimization problem that can have robot specific constraints. Of course, one may also introduce additional geometrical constraints in the individual optimization problem. The individual optimization problem for the $i^{th}$ robot is the similar to that of a single robot as presented in [10], [11].

The steps of our algorithm for high-level control loop are summarized below:

1) Check for updates from the constraint generation block of the constraints and the objectives for the main as well as individual robot optimization problems.
2) Compute the desired incremental Cartesian motions of the robots, ${\Delta\mathbf{x}_i}^d$, $i \in [1, n]$.
3) Solve the optimization problem (1), and pass the expected incremental Cartesian motions, ${\Delta\mathbf{x}_i}^e$, $i \in [1, n]$ to the individual robots.
4) Solve the $n$ optimization problems for the individual robots.
5) Numerically integrate the incremental joint motion to arrive at a new joint position.
6) Repeat step 1.

## II. Application: Knot Positioning

In this section the capabilities of our controller to perform a bimanual task are demonstrated. We have selected the task of positioning a knot as an application example. Once the suture is passed through the tissue and the free ends of suture are looped around each other, the knot is ready to be positioned on the tissue. The task involves controlling the magnitude and the direction of motion of the two free ends of the thread appropriately. The nomenclature used in the section to describe parameters of the knot are shown in Figure 2(a).

It will be shown that with the help of motion constraints it is possible to obtain reasonable positioning of the knot, even with single hand. This ability could be used to decrease the cognitive load on the surgeon and at the same time allow surgeon's to have a "spare" arm, thus making a multi-arm setup more useful.

### A. Modeling the Kinematics of Knot Placement

The multi-robot controller requires knowledge of the relationship that must be maintained between the two robots. In case of the knot position task this is in the form of a relationship between the incremental motion of the knot and the motion of the free ends of the thread. Because the placement of a knot is dictated by the complex friction between

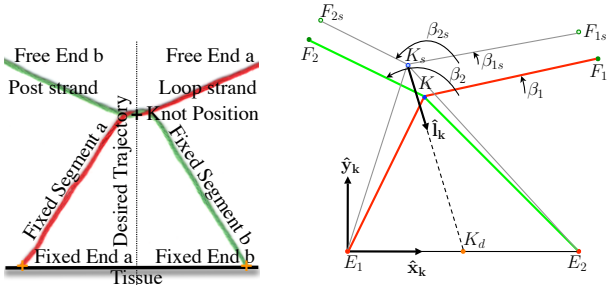


Fig. 2. (Left) Nomenclature used to describe knot parameters. (Right) A simplified model for placement of knot.

the knot segments, we make some simplifications to obtain a model that can be used in real-time control. Figure 2(a) shows an annotated picture of the half-square knot, whereas, Figure 2(b) shows the simplified model resulting for our assumptions. The assumptions are: 1) The length of the two ends of the thread that loop around each other is negligible. 2) At the moment we begin our motion control algorithm, the threads have negligible slack. 3) The ends labeled "fixed" are fixed with respect to some coordinate frame and their location is known in that frame.

In Figure 2(b), $E_1$ and $E_2$ are the fixed ends $a$ and $b$, respectively. $F_1$ and $F_2$ are the corresponding free ends that are pulled by the user to move the knot position denoted by $K$. The portion of the thread between the knot and the free ends is the "free section" of the thread and the portion of the thread between the knot and the fixed ends is the "fixed section" of the thread. $\beta_1$ and $\beta_2$ are the angles made by the free section of the threads with the horizontal. Subscript $s$ denotes the values at the start of the motion constraints.

Before we can generate motion constraints for the position of the knot, we need to derive two relations: 1) the relation between the motion of the free ends of the thread, $(\dot{F}_1, \dot{F}_2)$, and that of the knot $\dot{K}$ and 2) the relation between the motion of the free ends of the thread and the rate of change of angle, $(\dot{\beta}_1, \dot{\beta}_2)$, made by the free section of the threads with the horizontal.

The first relationship can be obtained by using the length invariant constraint for the total length of the fixed and free segments of the thread. One can write

$$\|F_i - K\| + \|K - E_i\| = c_i; \quad i = 1, 2 \tag{2}$$

We introduce the following substitutions:

$$\begin{aligned} a_i &= \frac{F_{ix} - K_x}{\|F_i - K\|} \quad b_i = \frac{F_{iy} - K_y}{\|F_i - K\|} \\ c_i &= \frac{K_x - E_{ix}}{\|K - E_i\|} \quad d_i = \frac{K_y - E_{iy}}{\|K - E_i\|} \end{aligned} \tag{3}$$

Using the substitutions of (3) and combining the time derivative of the length invariant constraints, i.e. (2), for both threads ans solving the resultant equations for the unknown rate of change of the knot, $\dot{K}$, gives the following

$$\begin{bmatrix} \dot{K}_x \\ \dot{K}_y \end{bmatrix} = J_{kf} \begin{bmatrix} \dot{F}_{1x} & \dot{F}_{1y} & \dot{F}_{2x} & \dot{F}_{2y} \end{bmatrix}^t \tag{4}$$

where

$$J_{kf} = \frac{1}{\Delta} \begin{bmatrix} a_1(b_2-d_2) & b_1(b_2-d_2) & -a_2(b_1-d_1) & -b_2(b_1-d_1) \\ -a_1(a_2-c_2) & -b_1(a_2-c_2) & a_2(a_1-c_1) & b_2(a_1-c_1) \end{bmatrix}$$
$$\Delta = (b_2-d_2)(a_1-c_1) - (b_1-d_1)(a_2-c_2) \tag{5}$$

The relationship between the slope of the free section of the thread, the free end points and the knot point is given by

$$\beta_i = \tan^{-1} \frac{F_{iy} - K_y}{F_{ix} - K_x}; \quad i = 1, 2 \tag{6}$$

Using the following substitutions and (4) we can write the time derivative of (6) as (8)

$$e_i = \frac{F_{ix} - K_x}{\|F_i - K\|^2} \quad f_i = \frac{F_{iy} - K_y}{\|F_i - K\|^2} \tag{7}$$

$$\begin{bmatrix} \dot{\beta}_1 \\ \dot{\beta}_2 \end{bmatrix} = J_{\beta f} \begin{bmatrix} \dot{F}_{1x} & \dot{F}_{1y} & \dot{F}_{2x} & \dot{F}_{2y} \end{bmatrix}^t$$

where

$$J_{\beta f} = \begin{bmatrix} -f_1 & e_1 & 0 & 0 \\ 0 & 0 & -f_2 & e_2 \end{bmatrix} + \begin{bmatrix} f_1 & -e_1 \\ f_2 & -e_2 \end{bmatrix} J_{kf} \tag{8}$$

Thus (4) and (8) can be called the "Knot Jacobian" as they provide a means of estimating the geometry of the knot if the current geometry and incremental motions of the free ends are known.

### B. Knot Positioning Implementation

Before the task of knot positioning is started, we must position the two arms of the robot such that the thread is almost taut. Moreover, we wrap the two threads around each other to form a half-square knot. For this implementation, we use two 3-DoF robots which are described later in section II-C. The free ends of the thread are held by the robot arms and the position of the free ends is given by $\mathbf{F} = [F_{1x}, F_{1y}, F_{1z}, F_{2x}, F_{2y}, F_{2z}]^T \in \Re^6$. The relationship for Jacobian obtained in section II-A is for a planar knot in the XY plane. Thus, we limit the motion of both robots to the plane of the knot. Further, motion constraints are applied to assist the user to move the two robot arms such that the knot moves along the desired direction. These motion constraints are presented in this section.

*1) Maintain sliding condition of the thread:* Jang *et al.* presented a friction model for knot sliding in [20]. One consequence of this model is a range for the thread angles $\beta_i$ that must be maintained for sliding to occur between the two threads. This range $[\beta_L, \beta_U]$, is a function of the coefficient of sliding friction, typically $\beta_L = \mathbf{0}$. For sliding to occur, we want thread angles at time $t + \Delta t$ to be within this range. Using the approximation that $\beta_i(t + \Delta t) \approx \beta_i(t) + \Delta\beta_i$ and (8), this constraint can be written as

$$\beta_L \leq J'_{\beta f} \cdot \Delta\mathbf{F} + \beta \leq \beta_U; \; J'_{\beta f} \in \Re^{2\times 6} \tag{9}$$

In (9), $J'_{\beta f}$ is obtained by introducing two zero column vectors, $\mathbf{0}$, corresponding to components $F_{1z}$ and $F_{2z}$. From (8)

we know that the Jacobian depends on the current value of the knot position. This value is obtained from a stereo vision system with some uncertainty. To avoid this constraint from being violated due to this uncertainty, we use a slack variable to provide some flexibility in the constraint. With the addition of the slack variables, $\mathbf{s}_\beta$, and some rearrangement, this constraint can be written as

$$\begin{bmatrix} -J'_{\beta f} \\ J'_{\beta f} \end{bmatrix} \cdot \Delta\mathbf{F} + \mathbf{s}_\beta \leq \begin{bmatrix} -\beta_L + \beta \\ \beta_U - \beta \end{bmatrix} \tag{10}$$

*2) Move knot along desired direction:* The second constraint that must be maintained is that the knot must move along a desired line. This desired line, $L$, is given by the position of the knot at the start of the motion constraints, $\mathbf{k}_s$, and the desired end position of the knot, $\mathbf{k}_e$. In our previous work [11], we had described a library of task primitives. One of these task primitives was to move a given point along a line. We use this basic constraint to move the knot point along line $L$. Again, we must account for some uncertainty in computation of the knot position by introducing another slack variable, $s_k$, for this constraint.

*3) Do not move knot backwards:* The third constraint deals with the direction of the motion of the knot. It is important that the knot move along the positive direction of the line $L$, towards the desired end position. This is necessary to ensure that the threads are sufficiently taut at all times, which is a necessary condition for our approximations made in section II-A to be valid. If the slack variable for this constraint is $s_d$, then the constraint can be written as

$$-\hat{\mathbf{l}}_{\mathbf{k}}^T \cdot J'_{kf} \cdot \Delta\mathbf{F} - s_d \leq \epsilon_d \tag{11}$$

*4) The optimization objective function:* The objective function is a means to incorporate the intended behavior of the robot. It is obvious to include an objective function that ensures that the actual motion is close to the desired motion in Cartesian coordinates for both robots. We select the L-2 norm as a measure of closeness. To minimize the L-2 norm of the slack variables is an additional objective function. We also incorporate weights to obtain a reasonable trade off between these different objectives when combining them together. The overall objective can be written as:

$$\begin{aligned} &[\Delta\mathbf{x}_l, \Delta\mathbf{x}_r, \mathbf{s}_\beta, s_k, s_d]^* = \\ &\quad \arg\min OBJ_r(\mathbf{W}_l, \{l\}, {\Delta\mathbf{x}_l}^d) + OBJ_r(\mathbf{W}_r, \{r\}, {\Delta\mathbf{x}_r}^d) \\ &\qquad OBJ_s(\mathbf{W}_\beta) + OBJ_s(W_k) + OBJ_s(W_d) \\ &\quad \text{where} \\ &\quad OBJ_r(\mathbf{W}_i, \{i\}, {\Delta\mathbf{x}_i}^d) \equiv \left\| \mathbf{W}_i(\Delta\mathbf{x}_i - {\Delta\mathbf{x}_i}^d) \right\|^2 \\ &\quad OBJ_s(\mathbf{W}_i) \equiv \left\| \mathbf{W}_i s_i \right\|^2 \end{aligned} \tag{12}$$

In (12), $\mathbf{W}_l$ and $\mathbf{W}_r$ are $3 \times 3$ diagonal matrices of weights corresponding to the left and right robots, respectively. The $2 \times 2$ diagonal matrix $\mathbf{W}_\beta$ and scalars $W_k$ and $W_d$ are weights associated with the respective slack variables. We would like to keep the slack variables as small as possible to avoid a large deviation from the desired constraints, but at a value that still allows the constraints to remain feasible due to uncertainty in the knot position obtained from the vision system.

### C. Experimental Results and Discussion

We now present experimental validation of the knot placement controller described earlier.

*1) Setup:* For preliminary investigation of the bimanual knot placement task, we designed a simple dual robot testbed, shown in Figure 3. Each robot manipulator consists of three translational stages that are orthogonal to each other. The two robots are mounted facing each other on a stable table. The end effector is designed to have a hole, at a known location, to pass the suture thread. A peg can be inserted in this hole to firmly hold the thread. This simple mechanism was selected to avoid the complexities of a gripper, and at the same time to simplify locating the free end of the thread. A force sensor is mounted on each end effector to determine the desired user velocity by using an admittance gain as a scale factor. A stereo camera system, with a baseline of 500mm, is mounted over the robots and used to locate the position of the knot.

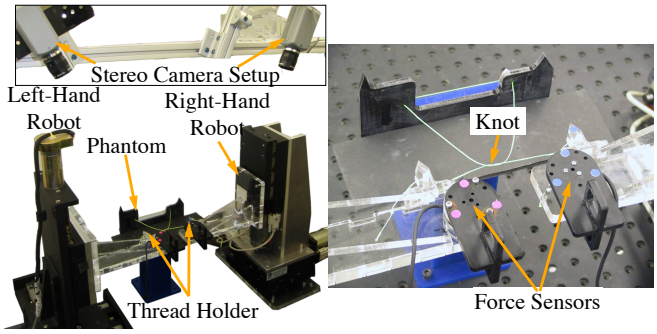


Fig. 3. Experimental setup for suture positioning task.

The CISST Library, along with OpenCV [21], provide the necessary tools for the image processing needs for this experiment. We next describe some of the assumptions and approximations adopted to keep the image processing simple and feasible for real-time application. To reiterate, the aim of the experiment is to illustrate a bimanual task and not to indulge in an accurate and elaborate methodology for segmenting a suture thread in stereo video streams. The first simplification for the knot procedure is the use of a bright colored fishing line instead of a typical dark colored suture thread. This allows us to use existing color segmentation algorithms to segment the thread. The next approximation is to use straight lines to represent the four segments of thread. This approximation keeps the result sufficiently accurate because the thread is mostly taught for the duration of the experiment. As stated in the section II-A, we ignore the length of threads that loop around each other. Thus, the knot position is given by the intersection of the four thread segments. This procedure for obtaining the knot position is repeated for both image streams. The camera calibration is then used to obtain the 3D position from these 2D projections, using a standard technique [22].

*2) Protocol:* Seven subjects participated in the research study to determine the efficacy of using robot assistance for knot positioning. The subjects were varied in experience using an admittance type robot from novice to expert. All subjects were asked to manipulate the robots and perform the task of moving the ends of the thread held by the robot as accurately as possible. Five trials for each of the three different modes listed below were performed.

*a) No assistance mode:* The users held the two force sensors attached to the robots in their corresponding hands. The robot would then move under admittance control without any motion constraints. The positions of the knot and the robots are tracked using the overhead cameras. The user carefully moves the robots such that the knot moves along the desired trajectory.

*b) Dual arm assistance mode:* The users cooperatively manipulate both robots using their corresponding hands. The robots move under admittance control with the constraints mentioned in the previous section.

*c) Follower assistance mode:* In this mode, the user holds the robot that corresponds to his or her dominant hand. The robot moves under admittance control and the controller also has the constraints on the knot position mentioned in the previous section. Due to these constraints, the robot not being controlled by the user follows the robot being held by the user to ensure these constraints are met.

For all modes, we determine the actual position of the knot using the overhead stereo camera system. The desired trajectory of the knot is the line joining the starting position of the knot and the center of the two fixed ends of the thread. The error between the actual knot position and the projection of this position along the desired trajectory is also recorded.

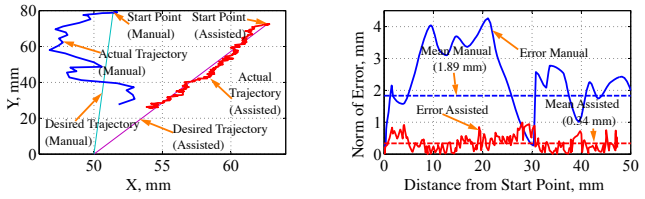


Fig. 4. (Left) The positions of the knot for manual and assisted modes for a typical trial. (Right) The error between the actual position of the knot and its projection on the desired trajectory.

*3) Results:* The positions of the knot for a trial with and without assistance are shown in Figure 4(a). As seen in the figure for this typical case, there is a large deviation from the actual path for the manual case, whereas there is much less deviation for the assisted case. Also in the manual case, one can see the large change in the direction of motion or "swings" about the direction perpendicular to the desired trajectory. The user notices a deviation from the path and tries to correct it, but as the relationship between the motion of the thread ends and knot ends is not trivial, he or she is not able to precisely manipulate the ends to achieve the desired correction, and hence these "overshoots". On the other hand, in the assisted mode, the camera and the controller modify the input velocity of the user thereby limiting these "overshoots" to a smaller value.

Figure 4(b) shows the errors between the actual position of the knot, as measured by the stereo camera, and its projection on the desired trajectory. This figure uses the same trial as Figure 4(a) and shows the error with and without assistance. The average of the L-2 norm of the error was computed for each trial. This is defined as the ratio of the sum of the L-2 norm of error to the total number of samples, $T$, collected in that trial. That is, the average of the L-2 norm of error for the $i^{th}$ trial using $k^{th}$ mode, $\bar{e}_{i,k}$, is given by $\bar{e}_{i,k} = \frac{\sum_{t=1}^{T} \|e_t\|}{T}; k = 1, 2, 3$. For the trial shown in Figure 4(b), the average error for a travel of 50 mm along the desired trajectory is 1.89 mm and 0.34 mm for manual and assisted mode, respectively. The standard deviation for the manual and assisted modes are 1.28 mm and 0.25 mm, respectively.

As shown in Figure 5(a), errors in both the robot-assisted modes (Mean, that is, $\frac{1}{N}\sum_{i=1}^{N} \bar{e}_{i,1} = 0.4217$ mm; N = 35 in dual arm assistance mode and Mean = 0.4379 mm in the follower assistance mode) are significantly better than in the no assistance mode (Mean = 1.9666 mm). Because the average errors for the no assistance mode fail the Shapiro-Wilk parametric hypothesis test of composite normality (p = 0.0357, $\alpha$ = 0.05), we use the Wilcoxon signed rank test as an alternative to the paired Student's t-test. Our null hypothesis is that the difference between the medians of the average errors for the dual arm assistance mode and the no assistance mode is zero. The p-value for this test is $2.7 \times 10^{-7}$ with an $\alpha$ value of 0.05. There is no significant difference between the two assisted modes (p = 0.3421, $\alpha$ = 0.05).

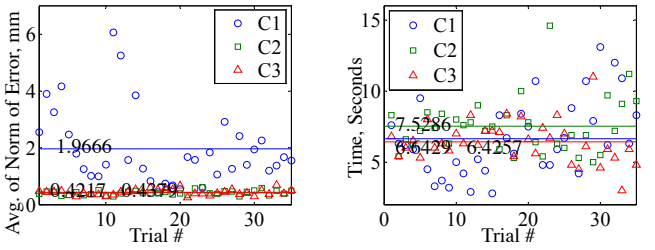


Fig. 5. (Left) Average of norm of error. (Right) Time required to complete the knot positioning task. C1 - No Assistance; C2 - Dual Arm Assistance; C3 - Follower Assistance (7 subjects × 5; total of 35 trials)

Figure 5(b) shows the time required to complete the task for each of the trials. The total time is measured from the moment the user starts to move the knot to the moment the user has completed moving the knot by a fixed distance. The average time of all the trials for the three modes are 6.6429 ms, 7.5286 ms and 6.4257 ms, respectively. A pair-wise t-test between average time for no assistance mode and dual arm assistance mode indicates that there is no statistical difference between them ($\alpha$ = 0.01, p-value = 0.14). Similar result is obtain for t-test between dual arm assistance and automatic follower modes ($\alpha$ = 0.01, p-value = 0.012). Anecdotally, the users felt that they were slower when any one of the assistance modes were enabled. Though the introduction of image processing required for the constraints did impose an additional delay in the system. This did not significantly effect the overall performance as seen from the time required for task completion.

The acceptable accuracy of surgical assistant systems depends on the application task. The motivation behind this application was to demonstrate the capability of the motion constraint algorithm in bimanual tasks. In spite of several assumptions regarding determination of knot point using vision system, we have achieved an average accuracy that is within the tolerance bound specified for this task (0.5 mm). The maximum error over all the trials for both the assistance modes is 0.62 mm. The main cause of this error appears to be in determination of the knot point. We have assumed that this point is given by the intersection of line segments. But we observed variations in the tautness of the thread introduced error in estimation of knot point. Unfortunately, it is difficult to establish a ground truth for the knot position and quantify this error in determination of knot point. For these experiments we also obtained the mean error of determining the tip of the robot in the vision system to be 0.2465 mm and 0.1349 mm for the left and right robots, respectively. The standard deviation for this measurement was 0.1092 mm and 0.11 mm, respectively. The error has two main sources, first arising from the determination of the transform between robot frame and camera frame and the second arising from determination of the 3D location tip in the vision system. Again we believe that accuracy and robustness of the system can be increased by enhancing the vision component of the system.

## III. Conclusions

Endoscopic surgery presents a constrained working environment for surgeons and they must deal with the realities of long instruments and awkward angles. We believe that robot assistance can improve accuracy, especially in a constrained environment such as that of endoscopic surgery. Cooperative control that combines surgeon's input with assistive motion control can be helpful especially when multiple robots are being controlled.

We analyzed the bimanual task of placing a half-square knot at a given target location. The instantaneous kinematics of the knot, that is the relationship between the incremental motion of the knot with respect to the free ends of the thread was developed. These were validated using an experimental testbed consisting of two robots and a stereo camera head. User trials with a number of subjects were performed on this setup. We observed a significant reduction in the average error from the desired trajectory when robot assistance was provided. We also observed that in bimanual tasks that require significant coordination between the two arms, it becomes difficult for human subjects to get an estimate of the motion that is required in each arm to keep the target (knot) on the desired trajectory. This results in large "overshoots" and "swings" about the desired trajectory. The bimanual virtual fixture assistance corrected this by adjusting the velocities of the robots associated with the two arms, resulting in a reduction in magnitude of such overshoots. We believe that assistance in bimanual tasks can be extended to tasks that require collaboration between more than one user. Also, there is merit in considering the follower assistance mode with the human user controlling the arms and a third robot performing a complex role that depends on the current state of the task.

## Acknowledgements

Partial funding of this research was provided by the National Science Foundation under grants EEC9731748 (CISST ERC), IIS0205318, and JHU internal funds.

## References

[1] M. C. Cavusoglu, I. Villanueva, and F. Tendick, "Workspace analysis of robotic manipulators for a teleoperated suturing task," in IROS, vol. 4, 2001, pp. 2234–2239 vol.4.

[2] M. J. H. Lum, J. Rosen, M. N. Sinanan, and B. Hannaford, "Optimization of a spherical mechanism for a minimally invasive surgical robot: theoretical and experimental approaches," IEEE Trans. Biomed. Eng., vol. 53, no. 7, pp. 1440–1445, 2006.

[3] A. J. Madhani, G. Niemeyer, and J. Salisbury, J. K., "The black falcon: a teleoperated surgical instrument for minimally invasive surgery," in IROS, vol. 2, 1998, pp. 936–944 vol.2.

[4] M. Mitsuishi, J. Arata, K. Tanaka, M. Miyamoto, T. Yoshidome, S. Iwata, M. Hashizume, and S. Warisawa, "Development of a remote minimally-invasive surgical system with operational environment transmission capability," in ICRA, vol. 2, 2003, pp. 2663–2670 vol.2.

[5] Intuitive Surgical, Inc., "davinci surgical system," http://www.intuitivesurgical.com/index.aspx.

[6] K. Hyosig and J. T. Wen, "Robotic knot tying in minimally invasive surgeries," in IROS, vol. 2, 2002, pp. 1421– 1426.

[7] I. Nagy, H. Mayer, A. Knoll, E. U. Schirmbeck, and R. Bauernschmitt, "Endo[PA]R: An open evaluation system for minimally invasive robotic surgery," in Mechatronics and Robotics, 2004, p. ??

[8] 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," in IROS, 2006, pp. 543–548.

[9] H. C. Lin, I. Shafran, T. E. Murphy, A. M. Okamura, D. D. Yuh, and G. D. Hager, "Automatic detection and segmentation of robot-assisted surgical motions," in MICCAI, 2005, pp. 802–810.

[10] M. Li, A. Kapoor, and R. H. Taylor, "A constrained optimization approach to virtual fixtures," in IROS, 2005, pp. 1408–1413.

[11] A. Kapoor, M. Li, and R. H. Taylor, "Constrained control for surgical assistant robots," in ICRA, 2006, pp. 231–236.

[12] N. Turro, O. Khatib, and E. Coste-Maniere, "Haptically augmented teleoperation," in ICRA, vol. 1, 2001, pp. 386–392 vol.1.

[13] S. Park, R. D. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in MICCAI, vol. 2208, 2001, pp. 1419–1420.

[14] A. Bettini, S. Lang, A. M. Okamura, and G. D. Hager, "Vision assisted control for manipulation using virtual fixtures," in IROS, 2001, pp. 1171–1176.

[15] P. Marayong, M. Li, A. M. Allison, and G. D. Hager, "Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures," in ICRA, 2003, pp. 1954–1959.

[16] M. Dewan, P. Marayong, A. M. Okamura, and D. H. Gregory, "Vision-Based Assistance for Ophthalmic Micro-Surgery," in Medical Image Computing and Computer Assisted Intervention (MICCAI), 2004.

[17] M. Matinfar, C. Baird, A. Batouli, R. Clatterbuck, and P. Kazanzides, "Robot-assisted skull base surgery," in IROS, 2007, p. To be published.

[18] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," IEEE Trans. Robot., vol. 23, no. 1, pp. 4–19, 2007.

[19] R. H. Taylor, P. Jensen, L. L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi, "A steady-hand robotic system for microsurgical augmentation," Intl. J. Robot. Res., vol. 18, no. 12, pp. 1201–1210, 1999.

[20] K. Hyosig, "Robotic assisted suturing in minimally invasive surgery," Ph.D. dissertation, Rensselaer Polytechnic Institute, 2002.

[21] B. Davies and *et al.*, "Open computer vision library," http://sourceforge.net/projects/opencvlibrary/.

[22] E. Trucco and A. Verri, Introductory Techniques for 3-D Computer Vision. Prentice-Hall, 1998.