

Optimum Robot Control for 3D Virtual Fixture in Constrained ENT Surgery

Ming Li and Russell H. Taylor

Department of Computer Science Department
NSF Engineering Research Center for
Computer Integrated Surgical Systems and Technology
The Johns Hopkins University, Baltimore, Maryland, USA
3400 North Charles Street - NEB B26, Baltimore, MD 21218
{liming,rht}@cs.jhu.edu

Abstract. In ENT surgery, the operating volume is very limited. This is especially true in sinus surgery, when the instrument passes through the sinus cavity to reach the pathological area. The sinus bones impose geometric constraints on the work volume. During the surgery, the surgeon needs to take care of the motion of the instrument tip to accomplish some delicate procedure; meanwhile he/she needs to avoid hitting anatomic constraints. In this paper, we present a method to assist the path following task in a constrained area. We project instrument tip motion, boundary information on the joint displacement via robot kinematics, and then use a constrained quadratic optimization algorithm to compute the optimal set of corresponding joint displacements. In the preliminary study, we show that the optimization constrained control can avoid collision with geometric constraints on the work volume, while keeping the precision of the tool motion.

1 Introduction

In sinus surgery, medical instruments or an endoscope camera are inserted into the sinus cavity through the nose. The instruments or endoscope have some degree of translational and rotational freedom but their motion is constrained by anatomic structure. During surgery, the instruments or the camera should avoid collisions or excessive force on delicate anatomy while still moving in the desired manner to accomplish the intended task. This paper presents a method to control a cooperatively controlled robot to achieve desired motions, such as following a specified path, subject to anatomic constraints in ENT surgery.

The goal of human-machine collaborative systems (HMCS) research is to create mechanisms that selectively provide cooperative assistance to a surgeon, while allowing the surgeon to retain ultimate control of the procedure. Kumar [1] developed an augmentation system for fine manipulation. Based on this system, recent research on motion constraints [2], [3], [4] has focused on simple techniques for “guidance virtual fixtures”. This prior works focused on 2D geometric guidance motion of the tool tip or camera and assumed that the tool or camera itself did not have any other environmental constraints.

Constrained robot control has been previously discussed in both telemanipulation and cooperative manipulation contexts. Funda and Taylor [5] formulated desired motions as sets of task goals in any number of coordinate frames relevant to the task, optionally subject to additional linear constraints in each of the task frames for redundant and deficient robots.

This work extends [5] by applying the method to surgical environment in which motion constraints are automatically derived from registered pre-operative models created from 3D images.

Figure 1. shows the concept of relationship between the instrument, 3D path and aperture to cavity in our task. The surgical instrument in this case is a sharp-tipped pointer held either by a robot or free-hand. In other cases it might be a surgical endoscope or a grasping instrument. We use the term “tip frame” to refer to a coordinate system whose origin is at the tip of the pointer and whose orientation is parallel to the tool holder of the robot. The “tool boundary frame” is a coordinate system whose origin corresponds to the point on the centerline of the tool that is closest to the surrounding anatomy and whose orientation is again parallel to the tool holder.

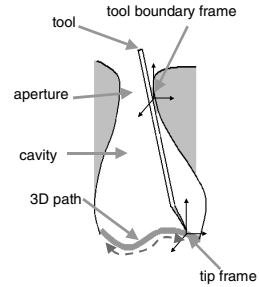


Fig. 1. The relationship of the instrument, 3D path and aperture into cavity in ENT surgery

2 Constrained Control Algorithm Overview

It is important to be able to place absolute bounds on the motion of the instrument in the constrained working environment. Within these bounds, the controller should try to place the instrument tip as closely to the desired position as possible. The basic control loop may be summarized as follows:

- Step 0:** We assume that the robot is holding the surgical instrument, and that a model of the patient’s anatomy has been obtained and registered to the coordinate system of the robot.
- Step 1:** Describe a desired incremental motion of the surgical instrument, based upon (a) surgeon inputs, such as may be obtained from a joystick or hands-on cooperative force control; (b) an *a priori* surgical task description; (c) real time sensor feedback, such as might be obtained from a vision sensor. This description may include both an objective function describing desired outcomes (e.g., move as close as possible to a target) and motion constraints (e.g., avoid collisions, do not exceed robot joint limits, do not permit position errors to exceed specified limits, restricting tip motion to remain within a desired envelope, etc.). The desired incremental motion is described as the solution to a constrained optimization problem.
- Step 2:** Use the robot and task kinematic equations to produce a new linearized optimization problem, in which instrument motion variables and other task variables have been projected onto incremental joint variables. This problem has the general form:

$$\begin{aligned} \min & \|W \cdot \Delta x - f\| \\ & H \cdot \Delta x \geq h \\ & \Delta x = J \cdot \Delta q \end{aligned} \quad (1)$$

where Δq are the desired incremental motions of the joint variables and Δx is an arbitrary vector of task variables. Different components of the optimization function may be assigned different relative weights, so that the errors of critical motion elements are close to zero, while errors in other non-critical motions simply stay as low as possible within tolerances allowed by the constraint set.

Step 3: Use known numerical methods [7] to compute incremental joint motions Δq , and use these results to move the robot.

Step 4: Go back to Step 1.

3 Algorithm Implementation

Here, we discuss implementation of this method for the task of aiming the tool tip within an anatomical cavity. We assume that a target has been specified, that its position relative to the current tip frame location has been computed, and that we know the boundary constraint configuration for the tool boundary frame. Given the desired Cartesian displacement of the tip frame

$$\Delta P_{tip_d} = (x_{tip_d}, y_{tip_d}, z_{tip_d}, 0, 0, 0)^T$$

we need to compute the appropriate robot motions (joint displacement vector) that will move the tool tip to the new location, while enforcing constraints such as boundary collision avoidance and joint limits. We identify 3 classes of requirements: in the tip frame, tool boundary frame, and in joint spaces.

- Tip Frame

We require that an incremental tool tip motion be as close as possible to some desired value. We express this as:

$$\min \| \Delta P_{tip} - \Delta P_{tip_d} \|, \quad \text{subject to } \Delta P_{tip_d}^T \cdot \Delta P_{tip} \geq 1 - \varepsilon \quad (2)$$

where ΔP_{tip_d} , ΔP_{tip} denote the 6-vectors of desired and actual tip frame displacements respectively. ε is a positive value approaching to 0. We project the tip frame motion to robot joint displacement Δq via the Jacobian relationship, $\Delta P_{tip} = J_{tip}(q)\Delta q$, then (2) can be rewritten in the form

$$\min \| W_{tip} \cdot (J_{tip}(q)\Delta q - \Delta P_{tip_d}) \|, \quad \text{subject to } H_{tip} J_{tip}(q)\Delta q \geq h_{tip} \quad (3)$$

where $W_{tip} = diag\{w_{tip}\}$ denotes a diagonal matrix of weighting factors specifying the relative importance of each component of ΔP_{tip} . In this case, we set the three components of w_{tip} associated with tip translation to relatively large values compared to the other weights. $H_{tip} = \Delta P_{tip,d}^T$, $h_{tip} = 1 - \varepsilon$.

- **Boundary Constraints**

Since the instrument is inserted into a cavity, we want to ensure that the instrument itself will not collide with the cavity boundary as a result of the motion¹. This requirement implies that

$$n^T \cdot (P_{tool} + \Delta P_{tool} - P_b) \geq d \quad (4)$$

where n denotes the normal direction of the boundary surface. ΔP_{tool} , P_{tool} , P_b denote 6-vectors with last 3 entries as 0, respectively. The first 3 entries of ΔP_{tool} represent the next translational motion of the tool boundary frame; the first 3 entries of P_{tool} denote the position of the point on the center line of the tool which is closest to the boundary (the origin of our tool boundary frame in robot space); the first 3 entries of P_b indicate the position of the boundary point. d represents the angle relationship between vector n and $P_{tool} + \Delta P_{tool} - P_b$.

We invoke minimizing the objective function to minimize extraneous motion of the tool,

$$\|W_{tool} \cdot \Delta P_{tool}\| \quad (5)$$

where W_{tool} gives the relative importance of minimizing displacements in the individual DOF of the tool boundary frame. All the components are set to relatively low weights in our task. By projecting the tool boundary frame motion to robot joint displacement via Jacobian relationship, $\Delta P_{tool} = J_{tool}(q)\Delta q$, (4) and (5) can be rewritten in the form

$$\text{Min } \|W_{tool} \cdot J_{tool}(q)\Delta q\|, \text{ subject to } H_{tool} J_{tool}(q)\Delta q \geq h_{tool} \quad (6)$$

where $H_{tool} = n^T$, $h_{tool} = -n^T(P_{tool} - P_b) + d$.

- **Joint Limitation**

Finally, we want to ensure that none of the joint limits are exceeded as a result of the motion. This requirement can be stated as

¹ For simplicity, we consider only single-point collisions with the cavity boundary. The extensions for multiple potential collision points are straightforward.

$$q_{\min} - q \leq \Delta q \leq q_{\max} - q$$

where q is the vector of the current values of the joint variables, and q_{\min} and q_{\max} denote the vectors of lower and upper bounds on the joint variables respectively. This can be rewritten in the form

$$H_{joint} \Delta q \geq h_{joint} \quad (7)$$

where

$$H_{joint} = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad h_{joint} = \begin{bmatrix} q_{\min} - q \\ -(q_{\max} - q) \end{bmatrix}$$

We also want to minimize the total motion of the joints by adding an appropriate objective function

$$\|W_{joint} \cdot \Delta q\| \quad (8)$$

where W_{joint} can be assigned to favor minimizing the motion of joints whose kinematic range of motion is small. All the components are set to relatively low weights here.

We combine all the task constraints and objective functions, then obtain the objective function to be minimized, which is:

$$\left\| \begin{bmatrix} W_{tip} \\ W_{tool} \\ W_{joint} \end{bmatrix} \cdot \left(\begin{bmatrix} J_{tip}(q) \\ J_{tool}(q) \\ I \end{bmatrix} \Delta q - \begin{bmatrix} \Delta P_{tip,d} \\ 0 \\ 0 \end{bmatrix} \right) \right\| \quad (9)$$

$$\text{subject to } \left\| \begin{bmatrix} H_{tip} \\ H_{tool} \\ H_{joint} \end{bmatrix} \cdot \begin{bmatrix} J_{tip}(q) \\ J_{tool}(q) \\ I \end{bmatrix} (\Delta q) \geq \begin{bmatrix} h_{tip} \\ h_{tool} \\ h_{joint} \end{bmatrix} \right\| \quad (10)$$

which can be solved numerically using the method of Lawson and Hanson [7] for the set of joint displacements Δq , satisfying the constraint (10) and minimizing the error norm of (9). Since we weigh a 1 mm error in translational displacement of the Cartesian task frame equal to a 1° error in rotational displacement about any of the frame's axes, we provide a scaling factor $\pi/180$ additional to relative importance of minimizing the objective function error in a certain frame.

4 Experiment Setup and Results

We performed preliminary experiments to evaluate the applicability of our optimum constraint robot control algorithm. Figure 2. shows the experimental setup. We built

a phantom with a plastic skull; a sewing wire attached at the bottom of the nasal cavity served as the target path. The sample task was to trace this path with the tip of a sharp pointer without colliding with the walls of the cavity. The pointer (Figure 2., right) had a bent tip in order to increase the volume that could be reached by the point. Five small spherical fiducials were attached to the skull, which was then CT scanned. A surface model of the skull was created from the CT images, as shown in Figure 3., and the positions of the fiducials in CT coordinates were determined via standard methods.

Our current implementation uses the JHU Steady-hand robot [1]. Steady-hand robotic mechanisms are coupled with computation to enhance human capabilities at highly skilled, interactive tasks. It is a 7 DOF remote-center-of-motion (RCM) manipulator with high position resolution ($<10\mu m$).

The surgical pointer was placed in the end-effector of the robot so that its shaft passed through the RCM point. We performed registration of the robot coordinate system to the pre-operative CT space in a straightforward manner using a Northern Digital OPTOTRAK[®] 3D tracking system.

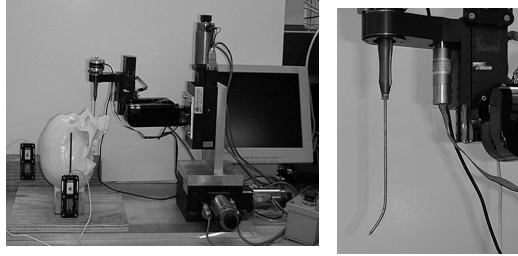


Fig. 2. (left) Experimental setup; (right) closeup of bent pointer in robot end-effector

4.1 Tip Path and Boundary Definition

In our path following experiment, we defined the target path with respect to CT space by tracing the sewing wire with the tip of an Optotrak pointer. We gathered multiple sample points along the path and then interpolate B-Spline curve to fit these sample points. The degree of our B-Spline model is 5.

The current tool tip position with respect to the robot can be determined from the robot joint encoders and the forward kinematics of Steady-hand robot. First we manually choose a point on the curve close to the current tip position, and guide the tool tip to the curve by using our control algorithm. Then for each iteration of our control loop, we read the current tool tip position P_{cur} , increase or decrease our B-Spline parameter t by 0.02 to obtain the next desired tip position P_{next} , and compute the desired Cartesian displacement of the tip frame $\Delta P_{tip_d} = P_{next} - P_{cur}$.

3D-SLICER's [6] built-in segmentation functionality was used to threshold, segment and render the skull CT dataset to create a triangulated model of the skull sur-

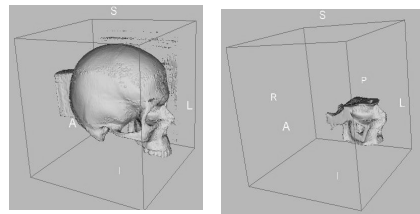


Fig. 3. 3D-SLICER [6] surface models. (left) whole skull surface model and (right) the model of nasal cavity portion used to generate constraints

face. For the current experiment, we only used the nose and sinus portion of the resulting skull model. The 3D-SLICER skull surface model and the model of the sinus portion are shown in Figure 3. There are about 99,000 vertices and 182,000 triangles in this surface model, all of which were transformed to robot coordinates after registration. The tool can be simplified as a cylinder; in the current version we only work on the collision of the tool itself, and ignore the bent part. In our work, we use a covariance tree data structure [8] to search for the closest point on the surface to the tool.

We used the same parameters and methodology to control the Steady-hand simulator (which theoretically simulates the robot motion) and the Steady-hand robot manipulation. As a comparison, we also gathered freehand data. A user held the tool that has an Optotrak rigid body affixed, and moved the tool through the sinus cavity to follow the sewing wire.

Table 1. The error of 5 trials for robot simulator, autonomous Steady-hand robot manipulation and freehand motion

| | Simulator (mm) | Autonomous Steady-hand Manipulation (mm) | Freehand (mm) |
|----------------|----------------|--|---------------|
| <i>Trial 1</i> | 0.000137 | 0.12 | 1.03 |
| <i>Trial 2</i> | 0.000520 | 0.08 | 0.95 |
| <i>Trial 3</i> | 0.000278 | 0.11 | 1.37 |
| <i>Trial 4</i> | 0.000303 | 0.17 | 1.53 |
| <i>Trial 5</i> | 0.000262 | 0.10 | 1.02 |
| <i>avg</i> | 0.0003 | 0.12 | 1.18 |

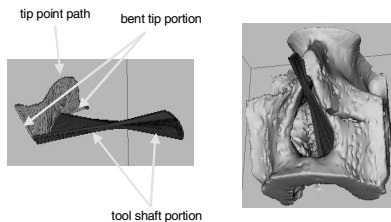


Fig. 4. Trajectories of the tool during the path following procedure. (left) the swept volume of the tool path, (right) the relative position between the tool and the nasal cavity

4.2 Results

The instrument tip position is gathered by robot encoders and transformed back to CT space. The average errors of 5 trials are presented in Table 1. The result of our simulator shows that the robot can exactly track the desired tip motion. The error of autonomous Steady-hand manipulation is 0.12mm. During the path following task, the tool itself did not hit the bone. Figure 4. shows the trajectories of the tool during the procedure and the relative position of the tool with the nasal cavity model. Figure 5. shows the complete error of the system. In our task, the size of Δq , $H \cdot J$, $W \cdot J$ in (1) is 7×1 , 19×7 ,

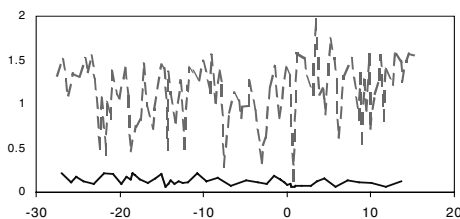


Fig. 5. Magnitude of position error using Steady-hand robot (solid line) and freehand (dashed line). x-axis(mm): the measured position along y-axis of the CT coordinate system, y-axis(mm): the magnitude of the position error

21×7 respectively. With our 1GHz Pentium IV PC, the average time in one control loop for the boundary search was 32.40ms and for optimization problem solving was 3.37ms.

5 Conclusion

ENT surgery presents a constrained working environment for both surgeon and mechanical devices designed to assist them. The control algorithms for the medical design of the devices must reflect these constraints. In this paper we outlined and implemented the constrained control for 3D path following for sinus surgery.

Optimal robot control that incorporates the environment constraints can provide a cooperative robot that assists with skilled manipulation tasks, while maintaining desirable prosperities such as collision avoidance and safety. The result of our experiments suggest that in sinus surgery the constrained optimization robot control can release the surgeon's tension on avoiding collision of the instrument to the anatomic structure during precision instrument motion. This control strategy not only can be used in ENT surgery, but also can be used in any other robot assistance surgery that requires precise motions in a constrained environment.

In the future, we would want the surgeon to be able to control the progress of the tool along the constrained path (e.g. by varying input forces) rather than having the robot follow the path at a preprogrammed speed. In addition, for real-time control, a more efficient method to search the boundary constraint is required.

References

- [1] R. Kumar, T.M. Goradia, A. Barnes, P. Jensen, L.M. Auer, L. L. Whitcomb, D. Stoianovici and R. H. Taylor. *Performance of robotic augmentation in microsurgery-scale motions*. In Proceeding of Medical Image Computing and Computer Assisted Intervention, volume 1679 of Lecture Notes in Computer Science, pages 1108–1115. Springer-Verlag, 1999.
- [2] A. Bettini, S. Lang, A. Okamura and G Hager. *Vision assisted control for manipulation using virtual fixtures*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1171–1176, 2001.
- [3] P. Marayong, M. Li, A. Okamura, G. Hager *Spatial Motion Constraints: Theory and demonstrations for robot guidance using virtual fixture*. In Proc. IEEE International Conference on Robotics & Automation, to be appeared, 2003.
- [4] M. Li, A. Okamura *Recognition of Operator Motions for real-time assistance using virtual fixture*. In 11th Int'l. Symp. On Haptic Interfaces for Virtual Environment and Teleoperator Systems, pages 125–131, 2003.
- [5] J. Funda, R. Taylor, S. Gomory and K. G. Gruben. *Constrained Cartesian Motion Control for Teleoperated Surgical Robots*. In IEEE Transactions on Robotics and Automation, Vol. 12, No. 3 pages 453–465, 1996.
- [6] <http://www.slicer.org>
- [7] C. Lawson and R. Hanson. *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
- [8] Williams, J.P., R.H. Taylor, and L.B. Wolff. *Augmented k-d techniques for accelerated registration and distance measurement of surfaces*. In Computer Aided Surgery: Computer-Integrated Surgery of the Head and Spine. Linz, Austria, September, pages 01–21, 1997.