Paper Review

Orhan Ozguner

May 5, 2012

Constrained Cartesian Motion Control for Teleoperated Surgical Robots

Janes Funda, Russel H. Taylor, *Fellow, IEEE*, Benjamin Eldridge, Stephen Gomory, and Kreg G. Gruben, *Member , IEEE*

J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "*Constrained Cartesian motion control for teleoperated surgical robots*," IEEE Transactions on Robotics and Automation, vol. 12, pp. 453-466, 1996.

Summary:

In this paper, authors discussed the optimal motion control of teleoperated surgical robots in a restricted workspace such as laparoscopic surgeries. The main focus is to optimally determine which degrees of freedom at hand. In this method the goal motion is defined as separate tasks in different coordinate frames respect to additional linear constraints. Therefore, the main problem proposed in this paper is minimizing some objective function using quadratic optimization techniques subject to some constraints.

The motivation of the problem, laparoscopic surgery, is given as an example since it has a very restricted workspace. In this kind of surgeries, known as minimally invasive surgery in the robotics world, the surgeon penetrates some tools into the part of the body through a single hole to perform the surgery. As seen on the figure-1, the procedure is the best example of the Remote Center of Motion (RCM) mode of operation.

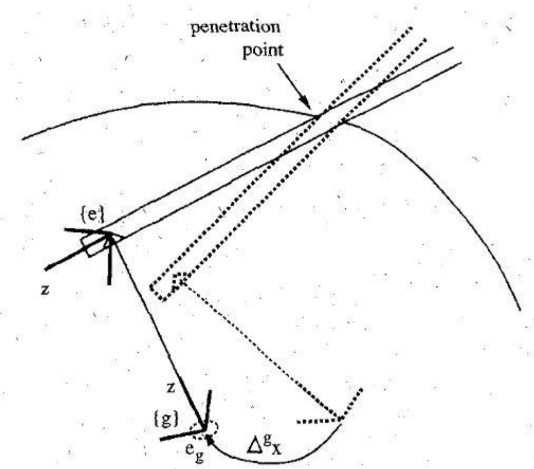


Figure-1

The paper gives an example of RCM procedure in the form of optimization function and the constraint matrices. In the given example, the problem is given the desired Cartesian motion of the gaze frame $\Delta^{g}x_d = [x, y, z, 0,0,0]^T$, which way is the appropriate robot motions that can move the gaze center to the new desired location while enforcing a number of constraints. In order to solve this problem, authors described a mathematical approach below.

1) Specifying Task Frame Objective Function and Constraint

Given a tolerance, ϵ_g, to reach the desired location such that:

$$\left\| \begin{bmatrix} \Delta^g \mathbf{x}[1] \\ \Delta^g \mathbf{x}[2] \end{bmatrix} - \begin{bmatrix} \Delta^g \mathbf{x}_d[1] \\ \Delta^g \mathbf{x}_d[2] \end{bmatrix} \right\| \leq \epsilon_g \tag{1}$$

Where $\Delta^g x_d$ and $\Delta^g x$ are the goal and actual gaze displacements.

(1) can be replaced as a family of linear equations of the form:

$$[\cos(\theta_k), \sin(\theta_k), 0, 0, 0, 0]^T \cdot (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d) \leq \epsilon_g, \quad k = 1, \cdots, n \tag{2}$$

and also this linear equations can be rewritten in the form:

$$\mathbf{H}_g \Delta^g \mathbf{x} \geq \mathbf{h}_g \tag{3}$$

If we want to minimize the error on the rotation about z-axis which is the surgeon's viewing axis, we need to minimize $||\Delta^g x_d[6] - \Delta^g x[6]||$ subject to constraints above. A weighting diagonal matrix is formed such that the coefficient matrix (H) correspond to each actuator will limit the actuator motion proportional to these coefficients. Therefore, we will have another function to optimize such that:

$$||\mathbf{W}_g(\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)|| \tag{4}$$

where W_g is a diagonal matrix that specifies relative importance of each actuator.

In this paper, end effector constraints (H_e) and joint limit constraints (H_j) are defined using same way. So that there will be two more diagonal weighting matrices, W_e, W_j that represent minimization of the end effector motion and minimization of the joint motion in the form of objective function matrix.

2) Putting All Together:

When all defined weighting matrices are combines in a single equation, the objective function and the constraint equations can be represented as follows:

$$\left\| \begin{bmatrix} \mathbf{W}_g & & \\ & \mathbf{W}_e & \\ & & \mathbf{W}_j \end{bmatrix} \left(\begin{bmatrix} {}^g\mathbf{J}(\mathbf{q}) \\ {}^e\mathbf{J}(\mathbf{q}) \\ \mathbf{I} \end{bmatrix} \Delta\mathbf{q} - \begin{bmatrix} \Delta^g\mathbf{x}_d \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \right\| \quad (5)$$

also we can form this as follows:

$$\begin{bmatrix} \mathbf{H}_g & & \\ & \mathbf{H}_e & \\ & & \mathbf{H}_j \end{bmatrix} \begin{bmatrix} {}^g\mathbf{J}(\mathbf{q}) \\ {}^e\mathbf{J}(\mathbf{q}) \\ \mathbf{I} \end{bmatrix} \Delta\mathbf{q} \geq \begin{bmatrix} \mathbf{h}_g \\ \mathbf{h}_e \\ \mathbf{h}_j \end{bmatrix}. \quad (6)$$

which can be represented as the final form of

$$\text{minimize } \|\mathbf{A}\ \Delta\mathbf{q} - \mathbf{b}\|, \text{ subject to } \mathbf{C}\ \Delta\mathbf{q} \geq \mathbf{d} \quad (7)$$

This final form of the equation can be solved using least square method.

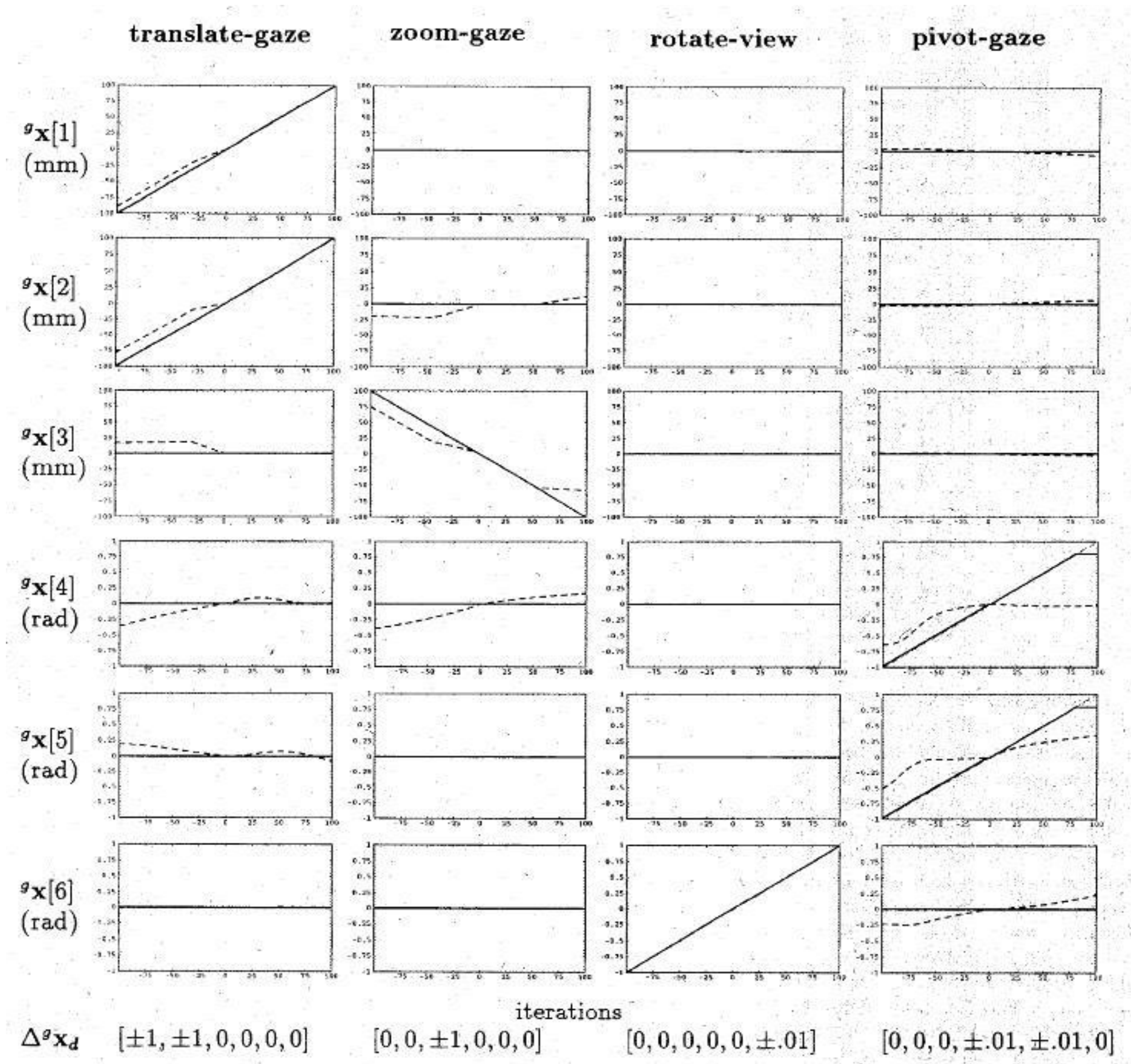
3) Assignment of Optimization Weights:

For a specific purpose of an optimization process, assigning weighing coefficients for each actuator is very important. In this part, the weighting factors can be separated into two parts such that:

$$\mathbf{w}_f[i] = \mathbf{u}_f[i] \cdot \mathbf{v}_f[i] \quad (8)$$

In this equation, $u_f[i]$ represents relative importance of minimizing the objection function for a particular degree of freedom and $v_f[i]$, which can find the difference between the translational and rotational errors, represents scaling factor.

In the paper, authors give results for an experimental set up which the above expressions are replaced with a real operation robot having redundant degree of freedom (the number of robot degree of freedom is greater that the number of task degree of freedom).



- Results for 4 different motion types listed at the top of the table.
- Dashed lines are for task-deficient system whereas the solid lines are for task-redundant system and dotted lines for desired motion.

Figure-2

From figure-2, it is seen that results are satisfactory except for the pivot-gaze trajectories in which the degree of freedom is not enough for that particular goal. However, this exception can show us that the system will either find a solution or will not be able to find one. This can tell us that the system is safe for such a task that requires extreme safety.

In the next part of the paper, authors explain how they implemented their algorithm for software tools. In my project, I implemented the algorithm for my own purpose.

Critique:

1) This algorithm is very useful for surgical systems that has the robot motion is slow compared to its internal loop so that a person can anticipate the incremental joint motion as joint velocities in that period of time.
2) In the paper, there is an example that shows how RCM process works in real time. This makes the paper very strong and reader can easily understand the algorithm.
3) Authors show a case that their algorithm fails. This is very useful for the users to avoid such cases.
4) In this method, one needs a very good initial guess for the system start position of the robot. This is not clearly mentioned in the paper.
5) In some cases, the method can find a wrong solution and also cannot find a solution. The reason for that is in some cases desired position is not reachable with current system. However, algorithm cannot make this separation. It assumes that there is always a solution.
6) In the paper design, authors need more experimental results and graphic in order not to confuse reader.

Relevance:

In our project, we are using Steady-Hand Eye Robot for telemanipulation. The Steady-Hand Eye robot has 5 degrees of freedom. However, during the surgery the surgeon can only use 3 degrees of freedom and also has very restricted area. In order not to damage penetration point, there has to be an RCM.

In the project, we have translation only and bilateral modes. In translation only mode, we want the robot moves only in X, Y, Z directions. There are few ways to solve this problem. One method is to disable joint velocity on the rotations. Another is to save the rotation of the robot when user clutch in and use the same rotation. So that robot will always stay on the same rotation.

However, these solutions are not functional and also they are not generic. To address these problems, I implemented the method described in this paper. In the optimization algorithm, I used higher weights for rotational joints and lower weights for translational joints. In this case, algorithm tends to choose X, Y, Z movements in order to reach desired position.