

# Constrained Cartesian Motion Control for Teleoperated Surgical Robots

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

**Abstract**—This paper addresses the problem of optimal motion control for teleoperated surgical robots, which must maneuver in constrained workspaces, often through a narrow entry portal into the patient's body. The control problem is determining how best to use the available degrees of freedom of a surgical robot to accomplish a particular task, while respecting geometric constraints on the work volume, robot mechanism, and the specific task requirements. We present a method of formulating desired motions as sets of task goals in any number of coordinate frames (task frames) relevant to the task, optionally subject to additional linear constraints in each of the task frames. Mathematically, the kinematic control problem is posed as a constrained quadratic optimization problem and is shown to be computable in real time on a PC. We will present experimental results of the application of this control methodology to both kinematically deficient and kinematically redundant robots. Specifically, we will discuss the control issues within the context of a representative set of tasks in robot-assisted laparoscopy, which includes (but is not limited to) teleoperated navigation of a laparoscopic camera attached to a surgical robot. A system based on this control formalism has been used in preclinical *in vivo* studies at the Johns Hopkins University Medical Center and the early experience with the system will be summarized.

## I. INTRODUCTION

THIS paper is concerned with control of robotic systems in constrained working volumes. We will develop and demonstrate an analytical and computational formalism which will allow us to treat the control of both kinematically redundant and kinematically deficient robotic systems in a uniform manner. While the techniques presented in this paper are completely general, we will specifically address the control issues as they pertain to robotically assisted minimally invasive surgery. These procedures are characterized by restricted access to the working volume as well as constrained maneuvering once inside the volume. Together with the requirement for utmost safety, this poses a challenging set of requirements on the design of the control strategy for such surgical robots. In this paper we develop a methodology for incorporating the constraints imposed by restricted workspace, safety considerations, and other geometric restrictions into the robot control strategy, thus offering a safer, more reliable, and more predictable control than classical robot control approaches.

Manuscript received October 18, 1994; revised May 15, 1995. This paper was recommended for publication by Associate Editor B. Hannaford and Editor A. J. Koivo upon evaluation of reviewers' comments.

J. Funda, R. H. Taylor, B. Eldridge, and S. Gomory are with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA.

K. G. Gruben is with the School of Medicine, Johns Hopkins University, Baltimore, MD 21205 USA.

Publisher Item Identifier S 1042-296X(96)00490-9.

While restrictions and requirements pertaining to the dynamic behavior of the manipulator could also be included in the formulation, this is generally not necessary with surgical robots whose motions are relatively slow for safety reasons. The specific application providing the context in which to evaluate the proposed control strategy will be that of teleoperated navigation of a laparoscopic camera held by a robot.

The use of robotic devices in surgical applications has been growing in recent years. Surgical robots have been successfully used for a number of orthopaedic applications [1], [2], neurosurgical procedures [3], [4], microsurgery [5], and are beginning to find applications in minimally invasive surgical procedures. Davies *et al.* [6] developed and clinically tested a special purpose "robot" for transurethral prostatectomies. Green [7] is developing a dexterous force-reflecting telesurgical system for performing minimally invasive surgery remotely in battlefield situations. A robotic system (AE-SOP) for positioning a medical telescope during laparoscopic surgery, developed by [8], has already gained regulatory approval and is being offered commercially. Our own work [9] has concentrated on providing the surgeon with a computer-controlled robotic "third hand" to work alongside the surgeon in a number of surgical scenarios, with the system capable of operating in a teleoperated as well as a semi-autonomous mode (see Section VI).

The control issues pertaining to many of the above applications of robotics to surgery are similar. In all cases the ability to accommodate unique, special purpose mechanical designs (kinematically redundant or deficient) is important, as is the ability to restrict the motions of the end-effector as well as various other parts of the mechanism to within strict safety motion envelopes to avoid inadvertent contact with surrounding patient anatomy. The control methodology described in this paper has been designed specifically to address these issues and should therefore apply to most of the above surgical applications as well as to a host of applications outside the domain of computer-assisted surgery.

Various methods of controlling redundant robot mechanisms have been proposed in the published literature. Broadly, three categories of techniques can be distinguished. The first category relies on pseudoinverse based gradient projection techniques, first introduced by [10] and illustrated by (1)

$$\Delta \mathbf{q} = \mathbf{J}^+ \Delta \mathbf{x} + (\mathbf{I} - \mathbf{J}^+ \mathbf{J}) \mathbf{z} \quad (1)$$

where  $\mathbf{x}$  is a 6-vector of current Cartesian coordinates of the robot's end effector and  $\mathbf{q}$  denotes an  $n$ -vector of gen-

eralized joint variables, with  $n > 6$ . In (1)  $J^+$  denotes the Moore-Penrose pseudoinverse of the Jacobian matrix and  $z$  is an arbitrary vector in the null space of the Jacobian. Setting the null-space vector to correspond to the gradient of various optimization functions  $\Phi(q)$ , i.e.,  $z = \alpha \nabla \Phi(q)$ , allows specification of a number of secondary performance criteria, such as joint limit avoidance [11], obstacle avoidance [12], [13], joint torque minimization [14], joint acceleration optimization [15], and maximization of various end-effector dexterity measures [16], [17]. While this approach to redundancy resolution is versatile and commonly used, it does not allow specification of absolute constraints imposed by the robot or by the task.

The second category of techniques was pioneered by [18] and are generally termed "extended Jacobian" techniques. The general approach here is to introduce a set of  $k$  (where  $k$  is the degree of redundancy) additional constraint functions of the form  $h(q) = 0$ , and augment the differential kinematic relationship with extra rows to obtain a square, uniquely determined system of equations, i.e.,

$$\begin{bmatrix} J \\ \frac{\partial h}{\partial q} \end{bmatrix} \Delta q = \begin{bmatrix} \Delta x \\ 0 \end{bmatrix} \quad (2)$$

where  $\frac{\partial h}{\partial q}$  represents the Jacobian of the added constraint functions  $h(q)$ . The square system of equations of (2) is uniquely determined and can be inverted using standard techniques. Again, various secondary optimization objectives can be formulated in terms of the set of linear equations  $h(q) = 0$ . To ensure that the secondary objective functions are not in conflict with the primary task goal, [19] proposes a strategy for prioritizing different task requirements. Seraji *et al.* [20] describe a modification of the extended Jacobian technique, called configuration control method, where the manipulator redundancy is resolved by augmenting the manipulator forward kinematic equations rather than the differential Jacobian relationship. The extended Jacobian approach of (2) is shown to be a special case of the configuration control method. While the extended Jacobian techniques offer a powerful redundancy resolution tool, they unfortunately limit the number of additional constraint equations (secondary task goals) to equal the degree of redundancy.

A number of alternative approaches to controlling redundant mechanisms, not easily classifiable into either of the above two categories, have also been proposed. The use of neural networks for solving kinematics of redundant robots has recently received much attention [21], [22], but sufficient accuracy and computational efficiency remains beyond the reach of these techniques. Chirikjian and Burdick [23] use parameterizations of "backbone curves" to capture macroscopic geometries of hyper-redundant serial manipulators and uses calculus of variations to compute optimal backbone deformations to accommodate mechanism and task constraints. This approach is applicable only to hyper-redundant snake-like kinematic structures. Parker *et al.* [24] propose the use of search procedures based on the natural mechanisms of genetics (reproduction, crossover, mutation) to solve the inverse kinematics of a redundant robot. These genetic algorithms make no assumptions about the problem and proceed blindly, guided only by a set of genetic operators. While interesting,

genetic algorithms are computationally prohibitive and not sufficiently accurate or efficient for practical, real-time control of redundant systems.

Very few of the control strategies reported in the literature are concerned with enforcing absolute constraints imposed by the physical limitations of the mechanism, geometry of the workspace, or particular requirements of the task. Among these [25] describes an interactive interpreted programming environment for specifying motion tasks for highly redundant articulated bodies (e.g., graphical simulation of a human figure). Motion properties and goals are specified as sets of time-varying assertions (inequality constraints) on the various portions of the articulated body. The system appears to have been applied to simulated mechanisms only and it is not clear whether it is suitable for real-time control of an actual robot mechanism. Cheng *et al.* [26] proposes the use of a modified (compact) quadratic programming method to solve the redundant differential kinematic problem subject to joint range and joint rate constraints. The authors report real-time performance for simple three-link simulation examples running on a SPARC II workstation.

The method presented in this paper is a generalization of the extended Jacobian technique, where any number of secondary task objectives can be specified in multiple task frames (including the joint space), the resulting extended Jacobian matrix is not necessarily square, and explicit absolute constraints on the motion envelope of any number of task frames can be specified in addition to the primary task goal. The method can accommodate both task-deficient and task-redundant mechanisms within the same formalism and is shown to be suitable for real-time implementation on a personal computer. Moreover, the control formulation allowed us to implement a very modular, task-oriented Application Programming Interface (API), which can be used to build descriptions of complex manipulation tasks from simple elements in a straightforward and manipulator-independent way.

The paper is organized as follows. Section II and III describe the motivation for this work and provide the necessary definitions and nomenclature for later discussion. Section IV describes the constrained Cartesian control formulation via an example task. In Section V we present results of applying this control strategy to both kinematically redundant and deficient robots. Section VI describes early experience with a surgical robot for laparoscopy which is currently in preclinical evaluations at the Johns Hopkins University Medical Center and which incorporates the constrained Cartesian control as described in this paper. Finally, Section VII summarizes the results and contributions of this work.

## II. MOTIVATION AND PROBLEM STATEMENT

Laparoscopic as well as other minimally invasive surgeries present a constrained working environment, where all access to the patient's internal organs is restricted to small skin incisions and long (usually) rigid instruments. Because of the restriction that the instrument (held by the robot) not be allowed to translate laterally relative to the port of entry into the patient, the mobility of the instrument (and thus the robot) is restricted.

In particular, a rigid instrument with no distal articulation (i.e., no degrees of freedom (DOF) distal to the point of insertion) has at most four DOF of motion about the port of entry—three rotational and one translational. Such an instrument cannot be positioned arbitrarily in all six DOF and is said to be kinematically deficient for tasks which require full six DOF mobility. Examples of such tasks are camera positioning, aiming energy sources, reaching appropriate approach direction for biopsies, etc. In these situations certain task motions are not achievable, and a control methodology is needed which will compute the optimal approximation to the desired motion, given the existing constraints.

Alternatively, if instruments with distal articulations are available (e.g., telescopes with adjustable angle-of-view and/or zoom), then the available DOF may exceed the number of task DOF and the robotic system is kinematically redundant for the task. In this case, the same task goal may be achievable in a variety of ways (i.e., via different robot motions) and a control methodology for specifying additional constraints on the motion of the robot and the instruments is needed.

It is important to consider both kinematically deficient and redundant systems since medical telescopes with adjustable optics may not be available for all types of procedures where a robotic system could be used effectively. Furthermore, extracorporeal redundant DOF can still be used to align the natural intra-corporeal DOF optimally for different tasks. We therefore wish to investigate control issues pertaining to both types of surgical manipulators. The central problem is determining the optimal use of the available DOF of a surgical robot and its instruments to accomplish a particular task, specified by the surgeon. We seek an analytical and computational formalism which will allow us to treat both cases in a uniform manner.

### III. DEFINITIONS AND NOMENCLATURE

In this section we define the basic terminology used throughout the paper.

We will define a *surgical robot* to consist of an actively controlled  $n$ -axis mechanical linkage (with known kinematics) and a set of attached instruments, such as cameras, energy sources, biopsy needles, etc. Associated with each instrument may be one or more Cartesian coordinate frames or *task frames* describing the location of the instrument or its parts relative to the robot. In particular, for the laparoscopic camera navigation task, we will denote the coordinate frame of the robot's end-effector (e.g., tip of the laparoscope) as the *end-effector frame* ( $\{e\}$ ). We will also define a *camera frame* ( $\{c\}$ ) with its origin centered at the optical center of projection of the laparoscope's optics and its  $z$ -axis along the view direction. The relationship between  $\{c\}$  and  $\{e\}$  is presumed fixed and can be obtained via standard extrinsic camera calibration, e.g., [27]. For convenience, we will also define *gaze frame* ( $\{g\}$ ) as the coordinate frame, whose origin (*gaze center*) is coincident with the 3-D point on the patient's anatomy appearing in the center of the 2-D camera image, and whose  $z$ -axis points from the gaze center back to the camera frame center at the end of the laparoscope. Rotational displacement of the gaze frame with respect to the camera frame about the  $z$ -axis is arbitrary

but assumed to be known. The separation between the camera and gaze frames will be referred to as the *gaze distance*. Other task frames can be defined as required by the task goals under consideration. The manipulator's joint space will be considered as a special "task frame"  $\{j\}$ .

We will refer to the 3-D operating volume of the robot as its *task space* and the desired Cartesian displacements within this volume as *task goals*.

Given this nomenclature, any desired change of the viewing direction (task goal) can now be expressed in terms of displacements with respect to one or more of the task frames defined above. A typical task goal may be to reposition the camera so as to center a particular anatomical feature within the camera's field of view while minimizing the translational motion of the camera tip (to avoid damaging surrounding tissue). Or the surgeon may wish to observe the current feature of interest from a different vantage point, i.e., the task is to rotate the camera about the gaze center, keeping the feature of interest centered in the camera's field of view.

### IV. CONSTRAINED CARTESIAN MOTION CONTROL

In considering control strategies for a surgical robot, two requirements are of overriding importance: safety and precision. In certain situations, even slight undesired motions of the instrument tip in certain directions (task DOF) can cause damage to the patient. It is therefore important to be able to place absolute bounds on the motion of the instrument tip or any other relevant part of the mechanism distal to the point of insertion. Within these bounds, the controller should try to place the instrument as closely to the desired position as possible, given the available DOF of the mechanism.

In the following we present a methodology for analyzing an arbitrary task goal (e.g., displacement of the instrument) and formulate the problem of computing the optimal set of corresponding joint displacements as a constrained quadratic optimization problem. The optimization function is defined as the two-norm of the motion error with respect to any subset of the task frames, including the robot's joint space. An optional set of constraints on the motion, expressed in any of these frames, ensures that errors in critical DOF remain strictly bounded. These constraints can be used to force the solution vector to satisfy certain critical requirements, such as restricting the motion of a particular part of the mechanism within a strict motion envelope, ensuring that joint limits are not exceeded, etc. Constraints can also be used to restrict the (possibly infinite) family of solutions in the case of a kinematically redundant system. Different components of the task space motion error (optimization function) can be assigned different relative importance or weight. This can be used so that within the solution space allowed by the constraint set, the critical errors are driven (close) to zero, while errors in other noncritical DOF are allowed greater tolerances.

The combination of a weighted objective function and an additional set of task or joint space constraints allows us to exploit the geometry of a particular task space motion and effectively trade off the various performance criteria. Finally, such a formulation allows us to treat both overdetermined

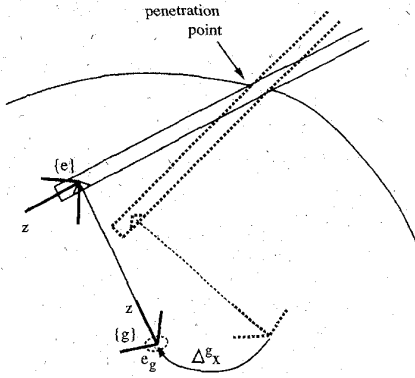


Fig. 1. Example task goal: Aiming a laser beam.  $\Delta^g \mathbf{x}$  denotes the necessary Cartesian displacement of the gaze frame to place the laser beam at the target.

(i.e., kinematically deficient surgical manipulator) and under-determined (i.e., kinematically redundant surgical manipulator) control problems using a uniform mathematical treatment.

#### A. Example Task Analysis

Consider, for example, the task of aiming a laser beam bore-sighted with the laparoscope (Fig. 1). We will assume that a target has been specified and its position relative to the current gaze frame location computed. Given this desired Cartesian displacement of the gaze frame  $\Delta^g \mathbf{x}_d = [x, y, z, 0, 0, 0]^T$ , we need to compute the appropriate robot motions (joint displacement vector  $\Delta \mathbf{q}$ ) which will move the gaze center to the new location, while enforcing a number of constraints as specified below. Analyzing the requirements of the task, we may identify three classes of requirements: those best expressed in the gaze frame, those most naturally expressed in the end-effector frame, and those which can be most naturally described in terms of joint variables.

1) *Specifying Task Frame Objective Functions and Constraints:* It is imperative that the beam hit the target location within a desired tolerance  $\epsilon_g$  in the plane perpendicular to the beam direction. This requirement implies that

$$\left\| \begin{bmatrix} \Delta^g \mathbf{x}[1] \\ \Delta^g \mathbf{x}[2] \end{bmatrix} - \begin{bmatrix} \Delta^g \mathbf{x}_d[1] \\ \Delta^g \mathbf{x}_d[2] \end{bmatrix} \right\| \leq \epsilon_g \quad (3)$$

where  $\Delta^g \mathbf{x}_d$  and  $\Delta^g \mathbf{x}$  denote the 6-vectors of desired and actual gaze frame displacements, respectively. The notation  $\mathbf{x}[i]$  is used here and throughout this paper to denote the  $i$ th Cartesian component of the displacement vector  $\mathbf{x}$ , with  $\mathbf{x}[1..3]$  and  $\mathbf{x}[4..6]$  denoting the translational and rotational components of the displacement, respectively. Equation (3) may be approximated by a family of  $n$  linear inequalities of the form

$$[\cos(\theta_k), \sin(\theta_k), 0, 0, 0, 0]^T \cdot (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d) \leq \epsilon_g, \quad k = 1, \dots, n \quad (4)$$

where  $\theta_k = k(2\pi/n)$ . This can be rewritten in the form

$$\mathbf{H}_g \Delta^g \mathbf{x} \geq \mathbf{h}_g \quad (5)$$

where  $\dim\{\mathbf{H}_g\} = n \times 6$ ,  $\dim\{\Delta^g \mathbf{x}\} = 6 \times 1$ , and  $\dim\{\mathbf{h}_g\} = n \times 1$ .

We would also like to minimize the rotational error about the viewing axis ( $z$ ), so as not to confuse the surgeon who is monitoring the procedure, i.e., we wish to minimize  $\|\Delta^g \mathbf{x}[6] - \Delta^g \mathbf{x}_d[6]\|$ , subject to the above constraints. More generally, we can trade off errors in different task DOF. Thus we want to minimize

$$\|\mathbf{W}_g(\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)\| \quad (6)$$

subject to the above task constraints. Here  $\mathbf{W}_g = \text{diag}\{w_g\}$  denotes a diagonal matrix of weighting factors, specifying the relative importance that the objective function be minimized in a particular DOF. In view of the above we would want to make  $w[6]$  large compared to the other weights. We will say more about the assignment of weights in Section IV-A-3.

When positioning the laparoscope to aim the beam, we must also ensure that it not damage any surrounding tissue. We can express this requirement by a family of linear constraints on the motion of the end-effector frame, i.e.,

$$\mathbf{H}_e \Delta^e \mathbf{x} \geq \mathbf{h}_e \quad (7)$$

where the constraint coefficient matrix  $\mathbf{H}_e$  and the constraint vector  $\mathbf{h}_e$  define the allowable envelope of motion for the laparoscope tip. Specifically, we can set

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{h}_e = \begin{bmatrix} \underline{\Delta^e \mathbf{x}} \\ -\overline{\Delta^e \mathbf{x}} \end{bmatrix} \quad (8)$$

where  $\underline{\Delta^e \mathbf{x}}$  and  $\overline{\Delta^e \mathbf{x}}$  denote the vectors of explicit lower and upper motion bounds, respectively, and  $\mathbf{I}$  denotes an identity matrix. In this case we are primarily concerned with translational motion bounds, which yields a total of 6 linear inequalities.

If we further desire to minimize extraneous motion of the instrument, we can add an appropriate objective function of the form

$$\|\mathbf{W}_e \Delta^e \mathbf{x}\| \quad (9)$$

where the diagonal matrix  $\mathbf{W}_e$  gives the relative importance (weight) of minimizing displacements in the individual DOF of the end-effector frame. See Section IV-A-3 for a discussion of weight assignment.

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

$$\underline{\mathbf{q}} - \mathbf{q} \leq \Delta \mathbf{q} \leq \overline{\mathbf{q}} - \mathbf{q} \quad (10)$$

where  $\mathbf{q}$  is the vector of the current values of the joint variables, and  $\underline{\mathbf{q}}$  and  $\overline{\mathbf{q}}$  denote the vectors of lower and upper bounds on the joint variables, respectively. Rewriting this in the form

$$\mathbf{H}_j \Delta \mathbf{q} \geq \mathbf{h}_j \quad (11)$$

gives

$$\mathbf{H}_j = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{h}_j = \begin{bmatrix} \underline{\mathbf{q}} - \mathbf{q} \\ -(\overline{\mathbf{q}} - \mathbf{q}) \end{bmatrix} \quad (12)$$

For safety and efficiency, we may also want to minimize the total motion of the joints of the surgical manipulator. We can express this as minimizing the objective function

$$\|\mathbf{W}_j \Delta \mathbf{q}\| \quad (13)$$

subject to the above constraints. The weights  $w_j[i]$  can be assigned so as to favor minimizing the motion of joints whose kinematic range of motion is small (see Section IV-A-3).

2) *Putting It All Together*: Having specified the motion requirements for the task, we reduce both the objective functions and the task constraints expressed in the various task frames to equivalent equations involving only the desired (unknown) robot's joint displacements  $\Delta \mathbf{q}$  via appropriate Jacobian relationships. For reasons of computational efficiency, only the end-effector Jacobian ( ${}^e \mathbf{J}$ ) need be computed explicitly, and the Jacobian matrices relating Cartesian and joint velocities for any other task frame  $\{f\}$  can be computed from

$${}^f \mathbf{J} = {}^* \mathbf{J} \cdot {}^e \mathbf{J} \quad (14)$$

where

$${}^* \mathbf{J} = \begin{bmatrix} \mathbf{R}^T & \mathbf{R}^T[\mathbf{p}] \\ \mathbf{0}_{3 \times 3} & \mathbf{R}^T \end{bmatrix} \quad \text{and} \quad {}^e T_f = [\mathbf{R} \mid \mathbf{p}] \quad (15)$$

where  $[\mathbf{p}]$  denotes the right-handed cross-product tensor, e.g.,  $[\mathbf{p}]\mathbf{v} = \mathbf{v} \times \mathbf{p}$ . In particular, if  ${}^g \mathbf{J}$  denotes the gaze-frame Jacobian matrix as obtained from (14), then the objective function of (6) can be rewritten as

$$\|\mathbf{W}_g ({}^g \mathbf{J}(\mathbf{q}) \Delta \mathbf{q} - \Delta^g \mathbf{x}_d)\|. \quad (16)$$

Similarly, the constraints of (5) can be rewritten in terms of the joint displacements as

$$\mathbf{H}_g {}^g \mathbf{J}(\mathbf{q}) \Delta \mathbf{q} \geq \mathbf{h}_g \quad (17)$$

where  $\mathbf{H}_g$  is a matrix of (constant) constraint coefficients and  $\mathbf{h}_g$  is a vector of known (constant) constraint bounds.

Applying similar transformations to all task constraints and objective functions, we can then mathematically formulate the solution to the problem as a constrained (weighted) least-squares problem, where the objective function to be minimized is

$$\left\| \begin{bmatrix} \mathbf{W}_g & & \\ & \mathbf{W}_e & \\ & & \mathbf{W}_j \end{bmatrix} \left( \begin{bmatrix} {}^g \mathbf{J}(\mathbf{q}) \\ {}^e \mathbf{J}(\mathbf{q}) \\ \mathbf{I} \end{bmatrix} \Delta \mathbf{q} - \begin{bmatrix} \Delta^g \mathbf{x}_d \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \right\| \quad (18)$$

subject to the constraints

$$\begin{bmatrix} \mathbf{H}_g & & \\ & \mathbf{H}_e & \\ & & \mathbf{H}_j \end{bmatrix} \begin{bmatrix} {}^g \mathbf{J}(\mathbf{q}) \\ {}^e \mathbf{J}(\mathbf{q}) \\ \mathbf{I} \end{bmatrix} \Delta \mathbf{q} \geq \begin{bmatrix} \mathbf{h}_g \\ \mathbf{h}_e \\ \mathbf{h}_j \end{bmatrix}. \quad (19)$$

This is a problem of the form

$$\text{minimize } \|\mathbf{A} \Delta \mathbf{q} - \mathbf{b}\|, \quad \text{subject to } \mathbf{C} \Delta \mathbf{q} \geq \mathbf{d} \quad (20)$$

which can be solved numerically for the set of joint displacements  $\Delta \mathbf{q}$ , satisfying the constraint equations (19), and minimizing the error norm of (18). In order to achieve the best possible real-time control behavior we implemented our own constrained quadratic optimization package, based on the

algorithms described in [28]. The resulting optimization algorithm executes in real-time on a 33-MHz PC. See Section V for details.

Note that the above approach to Cartesian manipulator control assumes that the elementary task motions are sufficiently small that the equation  $\mathbf{J}(\mathbf{q}) \Delta \mathbf{q} = \Delta \mathbf{x}$  represents a good approximation to the relationship between task-frame displacements ( $\Delta \mathbf{x}$ ) and joint space displacements ( $\Delta \mathbf{q}$ ). In a teleoperated surgical environment, where the surgeon's commands are sampled at a high bandwidth, this assumption should be easily satisfied.

3) *Assignment of Optimization Weights*: Equations (6), (9), and (13) all refer to a diagonal matrix  $\mathbf{W}_f$  of optimization weights  $w_f[i]$ , giving the relative importance of minimizing the objective function error in each Cartesian (see (6) and (9)) or joint space (see (13)) DOF of the respective task frame  $\{f\}$ . However, in order for the optimization problem to be set up properly, we must also ensure that the optimization weights pertaining to different DOF within each task frame are scaled properly relative to each other. Otherwise certain Cartesian or joint space components of the objective function may inadvertently dominate the optimization process, skewing the resulting joint angle displacement vector  $\Delta \mathbf{q}$  and causing poor or incorrect control behavior.

Correspondingly, we will distinguish two components of the weighting factor  $w_f[i]$ , i.e.,

$$w_f[i] = \mathbf{u}_f[i] \cdot \mathbf{v}_f[i] \quad (21)$$

where  $\mathbf{u}_f[i]$  corresponds to the relative importance of minimizing the objective function error in this particular DOF, and  $\mathbf{v}_f[i]$  provides a scaling factor to adjust for dimensionality inconsistency between the various DOF within a given task frame. The relative importance factor  $\mathbf{u}_f[i]$  is a floating point value in the range [0..1], with 1 corresponding to extremely high importance and 0 corresponding to no importance in minimizing the objective function error in the particular DOF.

The dimensionality scaling factor  $\mathbf{v}_f[i]$ , on the other hand, accounts for the fact that different components of the task frame error vector  ${}^f \mathbf{e} = (\Delta^f \mathbf{x} - \Delta^f \mathbf{x}_d)$  carry different units. For a Cartesian task frame  $\{f\}$ , the task frame error vector components  ${}^f \mathbf{e}[1..3]$  carry units of length (i.e., mm), while the components  ${}^f \mathbf{e}[4..6]$  carry units of angular displacement (i.e., radians or degrees). The choice of dimensionality scaling factor  $\mathbf{v}_f[i]$  must reflect this dimensional discrepancy and ensure that comparable rotational and translational errors are weighed comparably.

For example, assume that we are measuring joint variables  $\mathbf{q}$  (and their incremental displacements  $\Delta \mathbf{q}$ ) in terms of millimeters (mm) and radians (rad) for prismatic and revolute joints, respectively. Similarly, assume that an incremental displacement of a Cartesian task frame  $\{f\}$  is expressed as a 6-vector of positional/orientational change along/about each of its Cartesian axes (in mm/rad). Then the Jacobian matrix  ${}^f \mathbf{J}(\mathbf{q})$  (which is a function of the joint variables  $\mathbf{q}$  and thus carries the same dimensionality) will map joint displacements  $\Delta \mathbf{q}$  expressed in terms of mm and rad to the corresponding Cartesian displacements  $\Delta^f \mathbf{x}$  expressed in terms of mm and

rad according to the relation

$$\Delta^f \mathbf{x} = {}^f \mathbf{J}(\mathbf{q}) \Delta \mathbf{q}. \quad (22)$$

Therefore, if we wish to weigh a 1 mm error in translational displacement of the Cartesian task frame  $\{f\}$  equally to, say, a  $1^\circ$  error in rotational displacement about any of the frame's axes, we should set the weights  $\mathbf{v}_f[i]$  so that  $\mathbf{v}_f[1..3] = (\pi/180)\mathbf{v}_f[4..6]$ .

For the case where the task frame  $\{f\}$  denotes the joint space of the manipulator, the Jacobian matrix reduces to the identity matrix and the weights  $\mathbf{v}_f[i]$  must reflect the difference between an incremental displacement of a prismatic versus rotary joint. Again, if the joint variables  $\mathbf{q}$  are measured in terms of mm and rad, then in order to have prismatic joint errors of 1 mm generate the same optimization penalty as 1deg rotational joint error, we must set  $\mathbf{v}_f[p] = (\pi/180)\mathbf{v}_f[r]$ , where the indices  $p$  and  $r$  range over the prismatic and rotary joints of the mechanism, respectively. Note that objective function errors in the joint space are weighed on par with errors in Cartesian task frames. If a different relative scaling is desired, the joint space scaling factors  $\mathbf{v}_f[i]$  should be adjusted accordingly.

Finally, the optimization weights can be adjusted dynamically (at run-time) if the nature of the task is such that weight adjustments can be made intelligently and automatically by the controller during task execution. This may be the case when a task is broken up into a sequence of discrete stages, each of which requires a different weight assignment, or when external sensory information can be used to adaptively modify the weights for optimal task performance.

## V. EXPERIMENTAL EVALUATION AND RESULTS

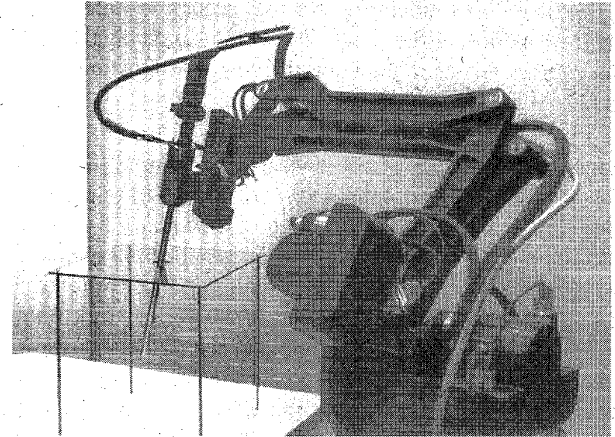
### A. The PLRCM Surgical Robot

The performance of the constrained Cartesian controller will be illustrated using the Parallel Linkage Remote-Center-of-Motion (PLRCM) surgical robot, designed and built at IBM. The PLRCM robot consists of a 3-axis linear XYZ stage ( $d_1, d_2, d_3$ ), a 2-axis parallel four-bar linkage assembly ( $\theta_4, \theta_5$ ), a 2-axis instrument carrier ( $d_6, \theta_7$ ), and a motorized camera rotation stage ( $\theta_8$ ). The appearance and the kinematics of the PLRCM robot are illustrated in Fig. 2. The instrument carrier provides for translation along and rotation about the instrument axis and the camera rotation DOF facilitates control of view orientation when the instrument is an off-axis (AOV  $\neq 0^\circ$ ) medical telescope.

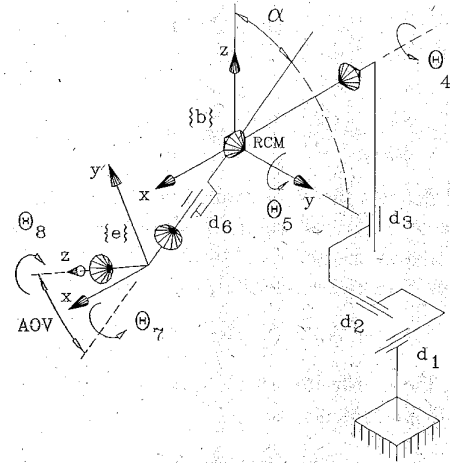
The parallel linkage structure of the distal portion of the mechanism provides a remote center of motion (RCM), distal from the linkage itself, about which all distal axes ( $\theta_4, \dots, \theta_8$ ) are decoupled. This design feature guarantees that the spatial location of the mechanism's RCM will be unaffected by the motion of the distal 5 axes, which provides an inherent safety feature for applications such as laparoscopic surgery and greatly simplifies the control of the mechanism.

The kinematic equation for PLRCM robot is as follows:

$${}^b T_e = T(x, d_1)T(y, d_2)T(z, d_3)R(x, \theta_4)R(y, \theta_5) \\ \times R(x, -\alpha)T(z, d_6)R(z, \theta_7) {}^e T_c R(z, \theta_8) \quad (23)$$



(a)



(b)

Fig. 2. The PLRCM surgical robot and its kinematics (at  $\mathbf{q} = \mathbf{0}$ ). AOV denotes angle of view of the telescope held by the robot.

where  $\alpha = 31^\circ$  and  ${}^e T_c$  is the robot-to-camera coordinate transformation as obtained by the extrinsic camera calibration procedure (e.g., [27]) for the particular laparoscope being used.

### B. The Experimental Protocol

Four representative view adjustment tasks will be considered to illustrate the performance of the constrained Cartesian optimizing controller. Each of the tasks will be executed using a task-deficient and a task-redundant surgical robot and the resulting motion trajectories will be plotted against the desired trajectories.

The four control tasks to be considered are:

- **translate-gaze:** translate the gaze center in the plane containing the current gaze center and orthogonal to the current gaze direction;
- **zoom-gaze:** zoom in and out along the current gaze (view) direction;
- **rotate-view:** rotate the current view, i.e., rotate about the  $z$ -axis of the current gaze frame; and

TABLE I

frm	r,d	c/o	translate-gaze
{c}	-	-	-
{g}	r	c	$\ (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)[1, 2]\  \leq 0.001 * \ \Delta^g \mathbf{x}_d\ $
	r	c	$\ (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)[4, 5, 6]\  \leq 0.001$
	r,d	o	$\ \mathbf{W}_g(\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)\ , \mathbf{w}_g = [HHLLH]$
{j}	r,d	c	$\mathbf{q} - \mathbf{q} \leq \Delta \mathbf{q} \leq \bar{\mathbf{q}} - \mathbf{q}$
	r,d	o	$\ \mathbf{W}_j \Delta \mathbf{q}\ , \mathbf{w}_j = [L \dots L]$

frm	r,d	c/o	zoom-gaze
{c}	r,d	o	$\ \mathbf{W}_c(\Delta^c \mathbf{x} - \Delta^c \mathbf{x}_d)\ , \mathbf{w}_c = [LLHLLH]$
{g}	r	c	$\ (\Delta^g \mathbf{x} + \Delta^c \mathbf{x}_d)[1, 2, 3]\  \leq 0.001 * \ \Delta^c \mathbf{x}_d\ $
	r	c	$\ (\Delta^g \mathbf{x} + \Delta^c \mathbf{x}_d)[4, 5, 6]\  \leq 0.001$
	r,d	o	$\ \mathbf{W}_g(\Delta^g \mathbf{x} + \Delta^c \mathbf{x}_d)\ , \mathbf{w}_g = [LLHLLH]$
{j}	r,d	c	$\mathbf{q} - \mathbf{q} \leq \Delta \mathbf{q} \leq \bar{\mathbf{q}} - \mathbf{q}$
	r,d	o	$\ \mathbf{W}_j \Delta \mathbf{q}\ , \mathbf{w}_j = [L \dots L]$

frm	r,d	c/o	rotate-gaze
{c}	-	-	-
{g}	r	c	$\ (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)[1, 2, 3]\  \leq 0.01$
	r	c	$\ (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)[4, 5, 6]\  \leq 0.001 * \ \Delta^g \mathbf{x}_d\ $
	r,d	o	$\ \mathbf{W}_g(\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)\ , \mathbf{w}_g = [LLLHHH]$
{j}	r,d	c	$\mathbf{q} - \mathbf{q} \leq \Delta \mathbf{q} \leq \bar{\mathbf{q}} - \mathbf{q}$
	r,d	o	$\ \mathbf{W}_j \Delta \mathbf{q}\ , \mathbf{w}_j = [L \dots L]$

frm	r,d	c/o	pivot-gaze
{c}	-	-	-
{g}	r	c	$\ (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)[1, 2, 3]\  \leq 0.01$
	r	c	$\ (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)[4, 5, 6]\  \leq 0.001 * \ \Delta^g \mathbf{x}_d\ $
	r,d	o	$\ \mathbf{W}_g(\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)\ , \mathbf{w}_g = [LLLHHH]$
{j}	r,d	c	$\mathbf{q} - \mathbf{q} \leq \Delta \mathbf{q} \leq \bar{\mathbf{q}} - \mathbf{q}$
	r,d	o	$\ \mathbf{W}_j \Delta \mathbf{q}\ , \mathbf{w}_j = [L \dots L]$

- **pivot-gaze**: rotate about the current gaze center, i.e., move the camera along the surface of an imaginary sphere centered at the gaze center.

The formulations of the above four tasks in terms of the constrained Cartesian control formalism of Section IV are given in Table I. For each of the four control tasks, the tables list the set of objectives (o) and constraints (c) with respect to the various task frames (frm), which are relevant to the task. The diagonal weight matrices  $\mathbf{W}_f$  are given as vectors  $\mathbf{w}_f$  of diagonal elements, where  $H, M$ , and  $L$  denote high, medium, and low weight, respectively. The  $r, d$  column indicates whether the corresponding objective or constraint is used with task-redundant robots (r), task-deficient robots (d), or both (r, d).

Note that for all tasks, with the exception of *zoom-gaze*, the desired motion is specified in terms of the motion of the gaze frame, whereas the task of zooming is more naturally expressed with respect to the camera frame. Also note that in the case of *zoom-gaze* task, the displacement of the gaze frame is the negative of the camera frame displacement  $\Delta^c \mathbf{x} = [0, 0, dz, 0, 0, 0]^T = -\Delta^g \mathbf{x}$ , since the  $z$ -axes of the camera and gaze frames are collinear but opposite (see Section III). For each objective or constraint the  $r, d$  column indicates whether this objective/constraint is used with task-redundant/deficient robots. Note that gaze- and camera-frame constraints are only used with task-redundant robots. Task-deficient robots may not be able to track the desired trajectories at all and so absolute accuracy constraints are not used. The joint-space constraints are specified for both task-redundant and task-deficient robots, since they represent mechanical constraints and must be respected in all cases. In addition, a low-priority (weight) objective function, tending to minimize the total joint motion, is specified in all situations.

For a given task, each iteration of the Cartesian control loop proceeds as follows: The system reads the user-supplied input (i.e., joystick) to obtain  $\Delta^g \mathbf{x}_d$ , and then sets up and solves a constrained optimization problem of the form of (18) and (19). The output of the optimization solver is a set of differential joint displacements  $\Delta \mathbf{q}$ , which are sent to the low-level servo controller of the robot to accomplish the desired view adjustments in task space. Although experiments have shown that servo-level tracking of the joint trajectories is very good, we will in the following show the trajectories as computed by the controller rather than those actually achieved by the surgical robot mechanism. This will be done to exclude any effects of the servo-level behavior, which is not of interest in this exposition, in favor of more clearly illustrating the behavior of the Cartesian level control. Also, since joystick input is not easily repeatable, we will use artificial uniform motion command steps  $\Delta^g \mathbf{x}_d$  at each iteration of the Cartesian control loop.

The performance of both a task-deficient and a task-redundant surgical robot executing the above set of tasks will be illustrated. Both robots are subsets of the 8 DOF PLRCM surgical robot described in Section V-A. The task-deficient robot will consist of the distal 5 axes of PLRCM and will thus not be allowed to use the base axes in achieving the commanded Cartesian goals. The task-redundant robot, on the other hand, will be the full 8-axis PLRCM robot, including the base axes.<sup>1</sup> Note that during laparoscopic surgery the mechanism's RCM must remain coincident with the port of entry into the patient. This implies that the 5-axis task-deficient robot is in fact the one used in a clinical setting. The use of the full 8-axis PLRCM robot in executing the above view adjustment tasks is included only for the purposes of illustrating the behavior of the constrained Cartesian controller with task-redundant robots and does not represent a clinically realistic scenario for the particular task of camera navigation in laparoscopic surgery.

All experiments reported below were performed using a clinical laparoscope with an angle of view of 30°. The 30° laparoscope was used to demonstrate independent control of instrument rotation and camera rotation axes (which would have been coincident for a straight or 0° laparoscope). Extrinsic calibration of the telescope was performed and the following robot-to-camera transformation was used in all experiments.

$${}^e T_c = T(z, -165.464 \text{ mm})R(x, 150^\circ). \quad (24)$$

A gaze distance of 30 mm was assumed in all cases, which gives the camera-to-gaze transformation as

$${}^e T_g = T(z, 30 \text{ mm})R(y, 180^\circ). \quad (25)$$

The starting configuration of the robot for all tasks is the robot's "home position," i.e.,  $\mathbf{q} = \mathbf{0}$ . Finally, note that all of the tasks illustrated in Section V-C exercise a substantial

<sup>1</sup>In fact, the 8-axis PLRCM robot is used for both sets of task. The task-deficient robot is realized by constraining the motion of the base axes to not exceed  $\pm 0.0001$  mm, again using the facilities of the constrained Cartesian control formalism.

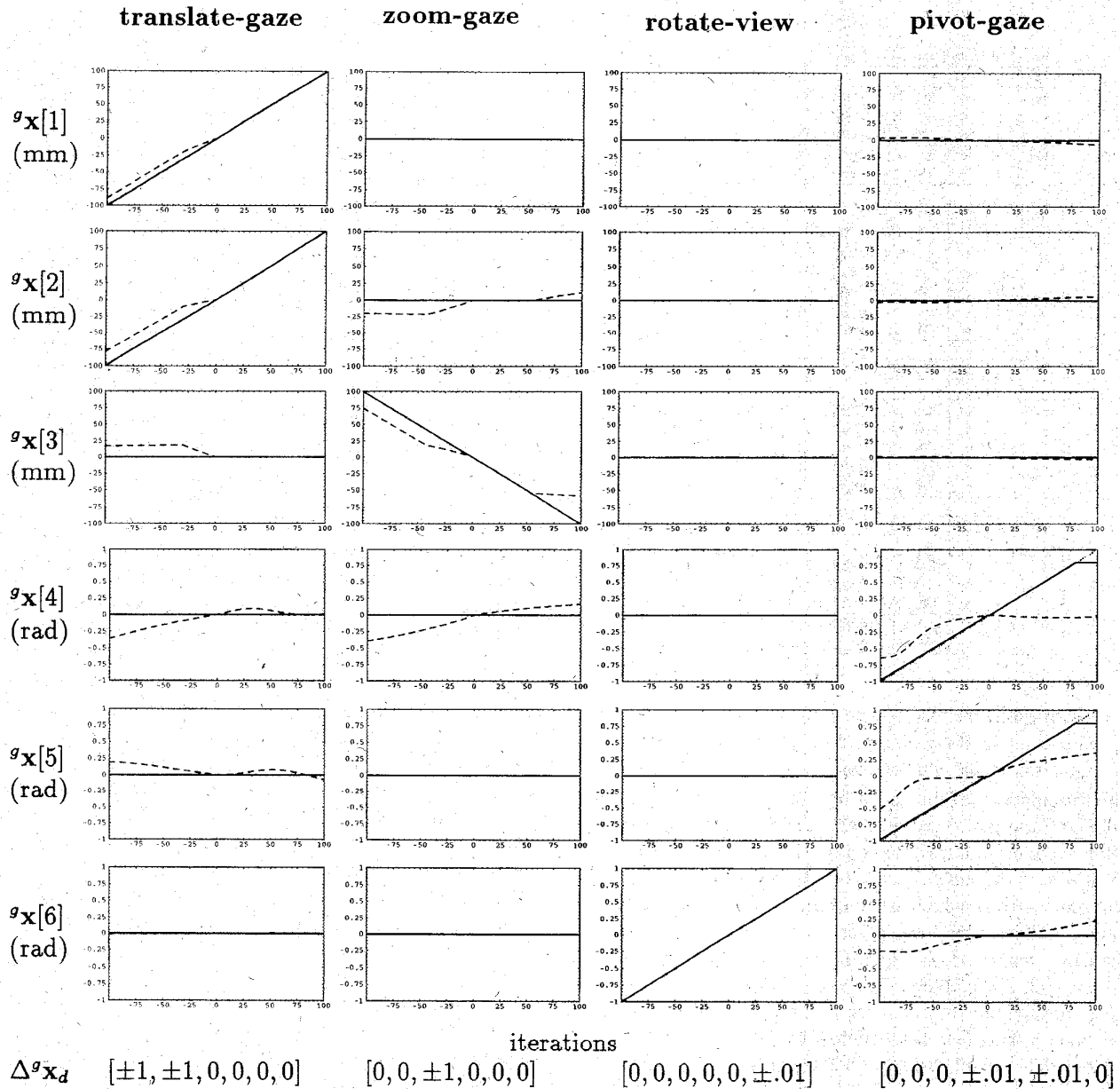


Fig. 3. Experimental results of the four view control tasks using the PLRCM surgical robot. The *dotted* lines correspond to the desired motion trajectories, *dashed* lines to trajectories achieved by the task-deficient PLRCM robot, and the *solid* lines to trajectories attained by the task-redundant PLRCM robot. Note that the dotted lines are generally obscured by the solid lines except for parts of the trajectories in the *pivot-gaze* task.  $\Delta^g \mathbf{x}_d$  gives the commanded incremental motion (mm, rad) at each step of the iteration.

portion of the robot's workspace ( $\pm 100$  mm in translation and  $\pm 1$  rad  $\approx \pm 57^\circ$  in rotation).

### C. Results and Discussion

Fig. 3 illustrates the tracking performance of the task-deficient (*dashed lines*) and task-redundant (*solid lines*) PLRCM robots for the set of four view adjustment tasks described in Section V-B. The desired trajectories are indicated with *dotted lines* (note that the dotted lines are generally obscured by the solid lines except for portions of the *pivot-gaze* trajectories).

As can be seen from Fig. 3, the task-redundant robot tracks the desired gaze-frame trajectories exactly for all four control tasks. This is to be expected as the robot is kinematically redundant for all tasks and, given sufficiently large joint ranges, should be able to position and orient the gaze frame arbitrarily within the task space. The only deviation of the task-redundant robot from the desired trajectory can be seen during execution of the *pivot-gaze* task, where tracking is aborted in the last portion of the trajectory. This is due to the fact that joint limits on base axes  $d_1$  and  $d_3$  were reached in the 80th iteration step of this particular motion, and no feasible solution satisfying the motion command and the constraints



on the accuracy of the resulting motion could be found. This illustrates one of the key features of this controller, which guarantees that either the commanded motion will be executed exactly or not attempted at all. This guarantee is very important in a surgical setting where robot motion errors, however small, may not be acceptable for some applications.

The task-deficient robot, on the other hand, can be seen to be kinematically deficient for most of the view adjustment tasks. The only task which it can execute without error is the *rotate-view* task, which can be accomplished by simply rotating the camera on the eyepiece of the telescope. The controller is thus able to find an exact solution and tracking is exact. For the remaining three tasks the robot is clearly kinematically deficient and can only approximate the desired gaze frame motion. This is particularly apparent in *pivot-gaze* task, which requires task degrees of freedom that are effectively unavailable to the task-deficient 5-axis PLRCM robot and so large errors are incurred in the critical degrees of freedom (rotations about gaze frame  $x$  and  $y$  axes). However, it is important to note that the computed motions are optimal in the sense that they represent the best possible approximation to the commanded motions, given the kinematics of the mechanism and the objectives and constraints imposed by the task.

In all of the above experiments the constrained Cartesian controller was executing on a 33-MHz PS/2 host computer communicating with the joint controller PC/AT via dual-ported shared memory. The overall control rate depends on the size of the optimization problem being solved at each step, i.e., the number of joint axes and the number of active objectives and constraints. Table II gives the average control bandwidth (in hertz) for the four tasks discussed above and two different processors.

A prototype system, based on a 33-MHz PS/2 host computer, has been in use at the Johns Hopkins University since April 1994. The surgeons who have used the system have reported that the control rates for all view adjustment tasks were sufficient for stable and responsive teleoperation of the surgical robot. The surgeons have also reported no perceptible time lag between the joystick commands and resulting robot motion and were generally very pleased with the responsiveness and smoothness of the control. While not strictly necessary in view of the above experimental results and user feedback, higher Cartesian control bandwidth would clearly further improve the control response and behavior of the system. As indicated by Table II, upgrading the host PC to a 66-MHz computer would result in an approximately 30% increase in the control update rates, bringing the teleoperation bandwidth well within the stable region [29].

## VI. SYSTEM EXPERIENCE

A robotic system for computer-assisted laparoscopy, which we call Laparoscopic Assistant Robot System or LARS, and which incorporates the constrained Cartesian controller described above, is currently in the process of preclinical evaluation at the Johns Hopkins University Medical Center (Fig. 4). The functionality of LARS includes teleoperated navigation

TABLE II  
CONTROL RATES FOR CONSTRAINED CARTESIAN MOTION CONTROLLER

task	33 MHz 486, PS/2 mod 95		66 MHz 486, PC/AT	
	5 axes (Hz)	8 axes (Hz)	5 axes (Hz)	8 axes (Hz)
translate-gaze	18.4	17.2	23.9	22.3
zoom-gaze	12.0	11.1	15.7	14.4
rotate-view	19.7	17.2	25.1	22.4
pivot-gaze	18.4	15.1	24.0	19.5

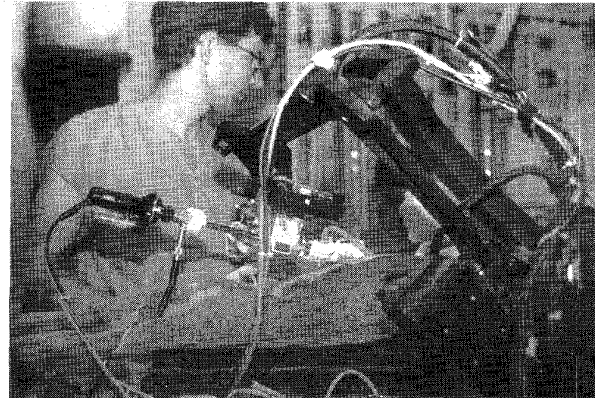


Fig. 4. The IBM laparoscopic assistant robot (LARS) during surgery.

of a laparoscopic camera (including the four view adjustment modes described above), as well as simple autonomous image-guided instrument navigation, based on on-line processing of the acquired laparoscopic images. A detailed description of the system, its functionality and user interface can be found in [30] and [32]. In this section we will summarize our experience with the kinematic control formulation as described above within the context of this fully functional prototype system for laparoscopic surgery.

Since June 1994, a number of *in vivo* studies have been conducted at Johns Hopkins University to evaluate the functionality, reliability, safety, and ergonomics of the system as a whole. Our collaborating surgeons, Drs. M. Talamini and L. Kavoussi, have successfully used the system to manipulate the laparoscopic camera in performing both cholecystectomies and nephrectomies. The constrained Cartesian controller has operated without failure through all of the animal studies to date. The experience with the controller has been so positive, in fact, that the current implementation of the system uses and actively controls all 8 axes of the surgical robot at all times, relying entirely on the constrained Cartesian controller to keep the motion of the base axes constrained within strict bounds while the instrument is inside the patient. Active control of the base axes allows a small amount of motion of the instrument relative to the port of entry into the patient, thus allowing the system to comply with the port of entry and maintain its coincidence with the mechanism's RCM at all times.<sup>2</sup>

The software interface to the constrained Cartesian controller has been designed to allow the application designer to describe surgical motion tasks in a straightforward and

<sup>2</sup>Currently, this compliance must be initiated by the surgeon, but the process could easily be automated, as the forces experienced by the instrument are already being monitored for other purposes.

intuitive manner. The details of the mathematical formulation of the constrained Cartesian controller are hidden from the application developer by means of a concise application programming interface (API). The API encapsulates all information pertaining to a particular surgical robot (number of joints, kinematics) and surgical task (task frames, objectives, constraints) within a `motion_context` data structure. A single statement suffices to establish an objective or constraint of any type in a particular task frame. For instance, the implementations of the example tasks of Section V require about 10 to 15 lines of API code each (see Appendix). The convenience and robustness of this API has allowed us to rapidly prototype and test various parameterizations of particular surgical tasks in terms of the number and types of goals, and objectives/constraints necessary to successfully accomplish the task. The kinematic control formulation has been also shown to be sufficiently general and flexible to allow us to describe complex surgical tasks (such as precise placement of time-release drug capsules in a prespecified 3-D pattern around a pathology) in a clear and straightforward fashion. Finally and perhaps most importantly, the modularity of the API, and in particular its independence of any particular robot kinematics, has proved to be extremely valuable in porting this control strategy between successive hardware implementations

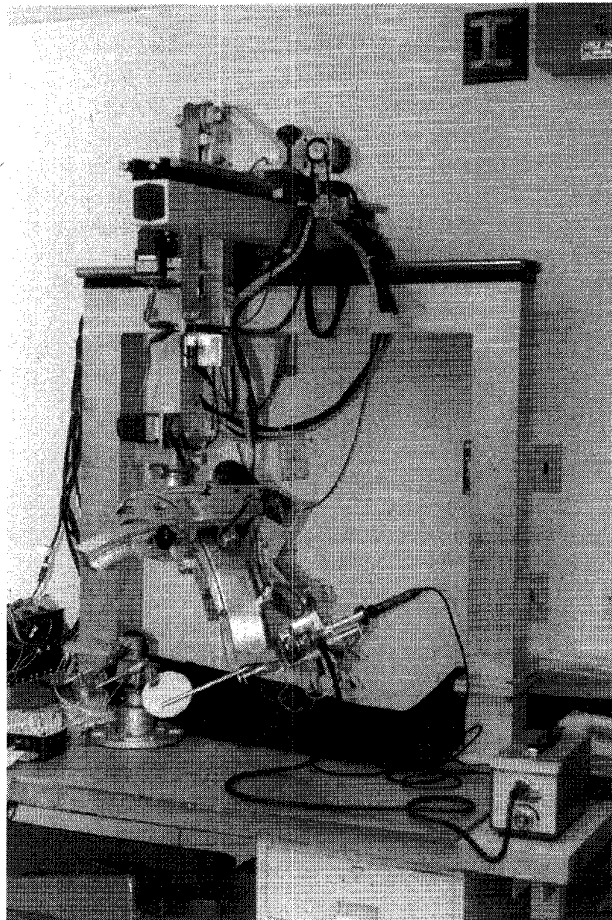
of our laparoscopic surgical assistant. By substituting different kinematic modules (within the `motion_context` of the API) we have been able to successfully use this control formulation to control three physically (and kinematically) very different surgical robots. Besides the PLRCM robot, the trio includes the two mechanisms illustrated in Fig. 5. No other modifications to the control architecture were required in moving from one surgical robot to the next.

Over the course of the last year we have used the constrained Cartesian controller extensively in both task-deficient and task-redundant circumstances. We have found that the trade-offs made by the optimization algorithm in allocating available DOF to best accomplish the commanded motion were reliable and predictable. For example, when confronted with a 0° laparoscope, where instrument rotation is coaxial with (and thus mathematically equivalent to) the camera rotation, the optimizing controller divides the effort of rotating the view approximately equally between the two joint axes. However, if either of the two axes reaches a joint limit, the controller compensates by assigning the entire displacement to the remaining axis, as expected. Similarly, when the robot mechanism is kinematically deficient for a given task, the optimizing constrained Cartesian controller trades off the available DOF of motion in a predictable and intuitive fashion.

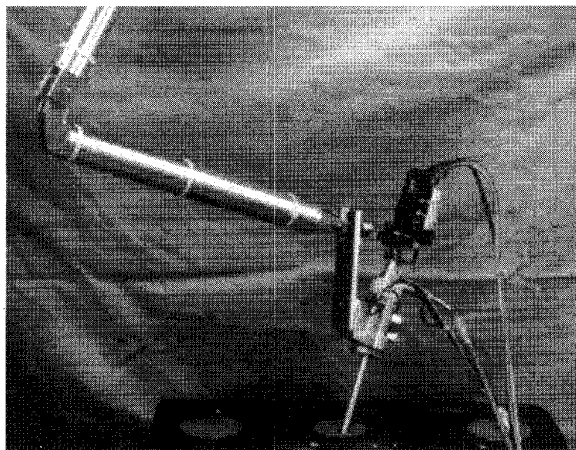
```

/*-----*/
/* translate gaze frame {g} by the incremental Cartesian displacement
 * dx0[] with dx0[i] = 0.0 for i=2..5 (XY translational motion only)
 *
 * Input: ptr to motion context struct
 *        desired Cartesian displacement dx0[] (mm, rad)
 *        motion mode (RC_UPDATE or RC_RECOMPUTE)
 *
 * Return value: 0 if ok, -1 if error
 *
 * Note: RC_WHI = 1.0, RC_MED = 0.5, RC_WLO = 0.1 (predefined weights)
 *        RC_SCL = M_PI / 180.0 (dimensionality scale factor)
 */
1 int rc_translate_gaze(motion_context *mc, double dx0[], int mode){
2   register int i;
3   double wg[6];
4   double wj[RC_MAX_JOINTS];
5   double eps_g = 0.01 * vector_mag(3, dx0); /* must get within 1% of target */
6
7   wg[0] = wg[1] = wg[2] = RC_WHI*RC_SCL;
8   wg[3] = wg[4] = RC_WLO; wg[5] = RC_WHI;
9
10  for (i=0; i < mc->robot->njoints; i++)
11   wj[i] = ((mc->robot->jtype[i] == RC_ROTARY_AXIS) ?
12           RC_WLO : RC_WLO*RC_SCL);
13
14  rc_robot_get_jpos(mc->robot);
15  rc_new_constraint_context(mc->constraints, mode);
16
17  rc_add_objective(mc, GAZE_FM, wg, dx0, mode);
18  rc_add_objective(mc, JNT_FM, wj, NULL, mode);
19
20  rc_add_constraint(mc, JNT_FM, RC_INTERVAL_CST, RC_ALL_JOINTS, NULL,
21                  RC_JOINT_LIMITS, mode);
22
23  if (RC_REDUNDANT(mc->robot)) {
24   rc_add_constraint(mc, GAZE_FM, RC_INTERVAL_CST, RC_TX|RC_TY|RC_TZ,
25                   dx0, eps_g, mode);
26  rc_add_constraint(mc, GAZE_FM, RC_INTERVAL_CST, RC_RX|RC_RY|RC_RZ,
27                   NULL, 0.0001, mode);
28  }
29  return rc_issue_constrained_motion(mc, mode);
30 }

```



(a)



(b)

Fig. 5. Two other surgical robots using the constrained Cartesian controller. (a) The 10-axis crossed-goniometer remote-center-of-motion robot (GRCM). (b) The 7-axis frame-suspended passive-wrist surgical robot (HISAR).

This can be seen from the plots of the trajectories achieved by the task-deficient robot for the four view adjustment tasks of Section V-C. The tracking errors in the various degrees of freedom are distributed in accordance with the penalty weights

associated with each of the DOF. We have found the flexibility of the control formulation in allowing adjustment of these weights in each individual DOF very convenient and have found that the controller responds to changes in the assignment of weights in a predictable manner. This allows easy “tuning” of the behavior of the system in task-deficient situations.

The flexibility of the Cartesian controller has been extremely valuable in allowing us to make on-the-fly changes to task formulations in response to surgeon feedback during the *in vivo* trials at Johns Hopkins. For example, during early experiments the surgeons requested that the global camera orientation remain fixed (“upright”) during all camera motion modes. We were able to accommodate this with a straightforward modification of the gaze-frame objective function (by taking into account the projected angle between the “upright” direction and the gaze-frame  $y$ -axis), and resume the trials.

We have recently extended the constrained Cartesian control formulation to accommodate robotic mechanisms consisting of both active and passive joints. We have successfully tested this extended controller on a 7-axis surgical manipulator for laparoscopy (HISAR), whose mechanical structure consists of an active positioning mechanism and a two-DOF passive wrist for passive compliance with the port of entry into the patient ([31], [32]). A number of such “hybrid” mechanisms have been recently applied to the task of camera navigation in laparoscopy, among them the AESOP system [8] and systems developed by [33]–[35]. The necessary extensions, which pertain primarily to methods of dynamically updating the nature and location of environmental constraints on the mechanism, were readily accommodated within the framework of the constrained Cartesian control formulation and will be the subject of a future publication. Preliminary results have shown that the time-varying “fulcrum constraint” of the hybrid HISAR mechanism can be accommodated in a straightforward and intuitive fashion and we believe that the constrained Cartesian control formulation is well suited for control of arbitrary hybrid mechanisms.

## VII. CONCLUSION

Minimally invasive surgery presents a constrained working environment for both surgeons and mechanical devices designed to assist them. Both the mechanical design of the devices as well as their control algorithms must reflect these constraints. In this paper we have outlined a methodology for safe and precise teleoperated control of a surgical robot, by allowing the specification of task-dependent control modes, where hard constraints can be placed on the motion of any part of the robot and relative importance of tracking the desired motion in various task DOF can be specified as part of the control law. The ability to respect absolute motion constraints and allow flexible control over the tracking error trade-offs in the various degrees of freedom is particularly important in many surgical applications where errors in certain critical DOF must be strictly bounded. Constraints and requirements pertaining to the dynamic behavior of the manipulator could also be accommodated by the formulation, but this is generally not necessary in surgical applications, where safety considerations dictate that the robot motions be relatively slow.

We have applied this control strategy to both kinematically deficient and kinematically redundant surgical robots within the context of medical camera navigation. In the case of a redundant system, the results have confirmed the expected accuracy of tracking commanded motions, whereas in the case of the kinematically deficient system the controller was shown to allow flexibility in trading off errors in critical and noncritical DOF. We have found that in task-deficient situations the system made predictable and intuitive trade-offs in exploiting the available DOF of motion to best accommodate the commanded motions. A flexible software interface to the constrained Cartesian controller has allowed us to rapidly prototype and evaluate various parameterizations of a specific surgical motion task, and quickly converge to a satisfactory formulation. The constrained Cartesian control formulation is independent of the kinematic structure of any particular mechanism and is easily portable from mechanism to mechanism. A robotic system for computer-assisted laparoscopy (LARS), incorporating this control strategy, has performed well in preclinical *in vivo* evaluations at the Johns Hopkins Medical Center.

In summary, we believe that the constrained Cartesian control formulation is a broadly applicable method of describing complex motion tasks in the presence of constraints. The same properties of the formulation (naturalness and ease of formulating task goals and motion constraints, ability to adjust trajectory tracking trade-offs, etc.), which make it convenient in the surgical domain, would make it equally applicable to a number of other application domains, characterized by precise, constrained manipulation. Examples of such domains include precise part assembly and industrial subsystem (e.g., aircraft engine or nuclear reactor) inspections. In the future we plan to exploit this formulation for precise quantitative targeting of surgical lesions for applications in spinal surgery and similar applications.

#### APPENDIX

##### SAMPLE APPLICATION PROGRAMMING INTERFACE CODE

An example implementation of the *translate\_gaze* view adjustment task (Section V) in terms of the Applications Programming Interface to the constrained Cartesian robot control described in Section IV is shown at the bottom of the previous page.

The routine begins by initializing the weights for the gaze frame and joint space objective functions (lines 7–12) in accordance with the discussion of optimization weight assignment of Section IV-A-3 and the tables of Section V-B. The current positions of all joints of the mechanism are then obtained (line 14) and the "constraint context" data structure is initialized (line 15). The constraint context data structure maintains all necessary storage and control information pertaining to the currently active objectives and constraints specified relative to the various task frames. If the incoming `mode` parameter is set to `RC_UPDATE`, the routine assumes that the constraint context structure is of the correct shape and size (i.e., this is *not* the first control cycle of the current task) and causes the context information to be updated only (to save execu-

tion time). Otherwise (i.e., `mode = RC_RECOMPUTE`) the constraint context data structure is reallocated and built up from scratch. The objectives and constraints in the various frames are then added to the constraint context in lines 17 through 27 in accordance with the specification of the *translate\_gaze* task as described in Section V-B. Finally, the `rc_issue_constrained_motion()` routine at line 29 takes the updated motion (and constraint) context, computes the corresponding incremental joint displacements as described in Section IV-A-2, and issues the resulting joint level command to the robot's servo controller.

The code shown at the bottom of the previous page is taken directly from the actual LARS software control library with only cosmetic modifications.

#### REFERENCES

- [1] H. Paul, B. Mittlestadt, W. Bargar, B. Musits, R. Taylor, P. Kazanides, J. Zuhars, B. Williamson, and W. Hanson, "A surgical robot for total hip replacement surgery," in *Proc. IEEE Int. Conf. Robotics and Automation*, Nice, France, May 1992, pp. 606–611.
- [2] T. C. Kienzle, S. D. Stulberg, M. A. Peshkin, A. Quaid, and C. H. Wu, "An integrated CAD-robotics system for total knee replacement surgery," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, May 1993, pp. 889–894.
- [3] P. J. Kelly, B. A. Kall, and S. Goerss, "Computer-assisted stereotaxic laser resection of intra-axial brain neoplasms," *J. Neurosurgery*, pp. 427–439, Mar. 1986.
- [4] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for CT-guided stereotactic brain surgery," in *IEEE Trans. Biomed. Eng.*, vol. 35, no. 2, pp. 153–160, Feb. 1988.
- [5] T. Salcudean, "A force-reflecting, motion-scaling teleoperation system for microsurgery," presented at the NSF Workshop Computer-Assisted Surgery, Washington, DC, Feb. 1993.
- [6] B. L. Davies, R. D. Hibberd, A. Timoney, and J. Wickham, "A surgeon robot for prostatectomies," in *Proc. 5th Int. Conf. Advanced Robotics*, June 1991, pp. 871–875.
- [7] P. S. Green, "Telepresence surgery," presented at the NSF Workshop Computer-Assisted Surgery, Washington, DC, Feb. 1993.
- [8] Y. Wang and J. Sackier, MD FACS, "Robotically enhanced surgery," in *Proc. Medicine Meets Virtual Reality II*, San Diego, CA, Jan. 1994.
- [9] R. Taylor, J. Funda, B. Eldridge, S. Gomory, K. Gruben, D. LaRoque, M. Talamini, L. Kavoussi, and J. Anderson, "A telerobotic assistant for laparoscopic surgery," *IEEE Eng. Medicine Biology*, vol. 14, no. 3, pp. 279–288, 1995.
- [10] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. MMS-10, pp. 47–53, 1969.
- [11] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 868–871, 1977.
- [12] M. Kirčanski and M. Vukobratović, "Trajectory planning for redundant manipulators in the presence of obstacles," *5-th CISM-IFTOMM Symp. Theory and Practice of Robots and Manipulators*, Udine, Italy, June 1984, pp. 43–50.
- [13] A. Maciejewski and C. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1987.
- [14] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," in *Proc. IEEE Int. Conf. Robotics and Automation*, St. Louis, MO, Mar. 1985, pp. 1016–102.
- [15] A. A. Mohamed and C. Chevallereau, "Resolution of robot redundancy in the Cartesian space by criteria optimization," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, May 1993, pp. 646–651.
- [16] T. Joshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics Research: The First International Symposium*, M. Brady and R. Paul, Eds. Cambridge, MA: MIT Press, 1984, pp. 735–748.
- [17] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 72–83, 1987.

[18] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, St. Louis, MO, Mar. 1985, pp. 722-728.

[19] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 410-425, 1991.

[20] H. Seraji, M. K. Long, and T. S. Lee, "Motion control of 7-DOF arms: The configuration control approach," *IEEE Trans. Robot. Automat.*, vol. 9, no. 2, pp. 125-139, 1993.

[21] F. Pourboghraat and J. Shiao, "Neural networks for learning inverse kinematics of redundant manipulators," presented at the Int. Joint Conf. Neural Networks, Seattle, WA, 1992.

[22] E. Hou and W. Utama, "Artificial neural network for redundant manipulator inverse kinematics computation," in *Proc. SPIE Conf. Intelligent Robots and Computer Vision*, Boston, MA, 1991.

[23] G. Chirikjian and J. Burdick, "Kinematically optimal hyper-redundant manipulator configurations," in *Proc. IEEE Int. Conf. Robotics and Automation*, Nice, France, 1992, pp. 415-420.

[24] J. Parker, A. Khoogar, and D. Goldberg, "Inverse kinematics of redundant robots using genetic algorithms," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1989, pp. 271-276.

[25] D. K. Pai, "Least constraint: A framework for the control of complex mechanical systems," in *Proc. American Control Conf.*, Boston, MA, 1991, pp. 1615-1621.

[26] F. Cheng, T. Chen, and Y. Sun, "Resolving manipulator redundancy under inequality constraints," *IEEE Trans. Robot. Automat.*, vol. 10, no. 1, pp. 65-71, 1992.

[27] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Trans. Robot. Automat.*, vol. RA-3, no. 4, pp. 323-344, 1987.

[28] C. Lason and R. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[29] L. W. Stark, W. S. Kim, and F. Tendick, "Cooperative control in telerobotics," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1988, vol. 1, pp. 593-595.

[30] J. Funda, R. Taylor, S. Gomory, B. Eldridge, K. Gruben, and M. Talamini, "An experimental user interface for an interactive surgical robot," in *Proc. 1st Int. Symp. Medical Robotics and Computer Assisted Surgery*, Pittsburgh, PA, Sept. 1994, pp. 196-203.

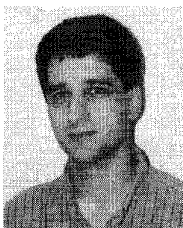
[31] J. Funda, B. Eldridge, K. Gruben, S. Gomory, and R. Taylor, "Comparison of two manipulator designs for laparoscopic surgery," in *Proc. SPIE Telemanipulator and Telepresence Technologies*, Boston, MA, Oct. 1994, vol. 2351, pp. 172-183.

[32] J. Funda, K. Gruben, B. Eldridge, S. Gomory, and R. Taylor, "Control and evaluation of a 7-axis surgical robot for laparoscopy," in *Proc. IEEE Int. Conf. Robotics and Automation*, Nagoya, Japan, May 1995, pp. 1477-1484.

[33] J. A. McEwen, "Robots in surgery," in *Proc. Emerging Surgical Technologies Conf.*, New Orleans, LA, 1986, pp. 51-513.

[34] J. B. Petelin, "Computer assisted surgical instrument control," in *Proc. Medicine Meets Virtual Reality II.*, San Diego, CA, Jan. 1994, pp. 170-173.

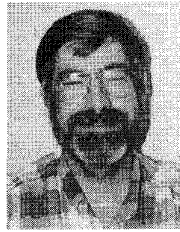
[35] R. Hurteau, S. DeSantis, E. Begin, and M. Garner, "Laparoscopic surgery assisted by a robotic cameraman: Concept and experimental results," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Diego, CA, May 1994, pp. 2286-2289.



**Janez Funda** received the B.A. degree in computer science and mathematics from Macalester College, St. Paul, MN, in 1986, and the Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia, in 1991.

He joined the Computer Assisted Surgery group of IBM Research in 1991. His research interests include robot systems, telemanipulation, man-machine interfaces, and virtual reality. His current research focuses on the use of robotic, sensing, and image processing technologies to assist in performing surgical procedures. He has published over 30 scientific papers and holds two U.S. and international patents.

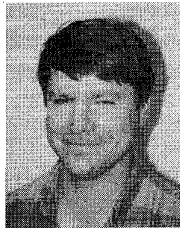
He has published over 30 scientific papers and holds two U.S. and international patents.



**Russell H. Taylor** (S'68-M'77-SM'87-F'94) received the B.E.S. degree from Johns Hopkins University, Baltimore, MD, in 1970 and the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1976.

He joined IBM Research in 1976, where he developed the AML language. Following a two-year assignment in Boca Raton, FL, he managed robotic research activities at IBM Research from 1982 until returning to full time technical work in late 1988. Since March 1990, he has been Manager of Computer-Integrated Surgery. His research interests include robot systems, programming languages, model-based planning, and (most recently) the use of imaging, model-based planning, and robotic systems to augment human performance in surgical procedures.

Dr. Taylor is Editor Emeritus of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and a member of various honorary societies, panels, program committees, and advisory boards.



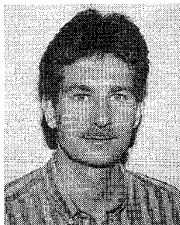
**Benjamin Eldridge** received the M.S. degree in physics from Rensselaer Polytechnic Institute, Troy, NY.

He was a member of the Computer Assisted Surgery Group at IBM Research during 1993-1994, where he worked on electromechanical and electronic design and implementation of robotic devices of surgery. His interests are in instrumentation development and integration. He has published 19 papers and holds 1 patent.



**Stephen Gomory** received the B.A. degree in architecture from Columbia College, Columbia, CA, in 1988, and the M.S. degree in computer science from New York University, New York, in 1995.

He is currently with the IBM T. J. Watson Research Center, Hawthorne, NY, as a Staff Programmer, where he is with the Computer Integrated Surgery group. His computer programming work there has centered around image processing, robotics, and system integration.



**Kreg G. Gruben** (S'89-M'91) received the Ph.D. degree in biomedical engineering from the School of Medicine, Johns Hopkins University, Baltimore, MD, in 1993.

From 1993 to 1994, he was a Postdoctoral Fellow in the Johns Hopkins Department of Radiology where, in collaboration with IBM Research, he worked on the development and testing of surgical robots. He is currently with the University of Wisconsin-Madison as an Assistant Professor of Kinesiology and Mechanical Engineering. His research interests include the biomechanics of human motion.

His research interests include the biomechanics of human motion.