Shared Control of Multiple-Manipulator, Sensor-Based Telerobotic Systems

Robert L. Williams II
Ohio University
Athens, Ohio

F. Wallace Harrison
NASA Langley Research Center
Hampton, VA

Donald I. Soloway
NASA Ames Research Center
Moffett Field, CA

Proceedings of the
1997 IEEE International Conference on Robotics and Automation

Albuquerque, NM

April 20-25, 1997

CORRESPONDING AUTHOR:
Robert L. Williams II
Department of Mechanical Engineering
257 Stocker Center
Ohio University
Athens, OH 45701-2979
Phone: (740) 593-1096
Fax: (740) 593-0476
E-mail: bobw@bobcat.ent.ohiou.edu

Shared Control of Multiple-Manipulator, Sensor-Based Telerobotic Systems

Robert L. Williams II
Ohio University
Athens, OH 45701

F. Wallace Harrison
NASA Langley Research Center
Hampton, VA 23681

Donald I. Soloway
NASA Ames Research Center
Moffett Field, CA 94035

ABSTRACT

A control architecture is presented for real-time, sensor-based, shared control of remote, multiple-manipulator telerobotic systems. The system allows teleoperation, autonomy, or a combination (shared, telerobotic control). The rate-based system accepts control inputs from a variety of sources (joystick position or velocity, automated path planner position or velocity, machine vision, force/moment) simultaneously for all Cartesian axes. The system has been experimentally implemented and has proven effective in laboratory simulations of remote space tasks.

1. INTRODUCTION

Remote robotic operations in the space, nuclear, and undersea environments present challenges not normally present in manufacturing industries where the environment may be controlled. Remote operations in harsh environments require control algorithms capable of adapting in real-time to unexpected events in the workspace. Pre-planned, model-based control is not sufficient; the manipulator systems must be sensor-rich for safe, effective, dexterous operations. Human operator involvement should be natural and minimal. In these demanding environments, safety, reliability, adaptability, and low contact forces are generally more important than minimal time motion. Also, highly repetitive motions are not expected; human-like reaction to uncertain circumstances are more important. Many tasks require multiple cooperating manipulators.

Authors have recently presented results in sensor-fusion control [1]; shared control [2]; and coordination of multiple manipulators in the same task [3,4,5].

The current paper proposes a general architecture for shared, real-time, sensor-based Cartesian control of remote, multiple manipulator telerobotic systems. The system has been developed and experimentally implemented during the past ten years at NASA Langley Research Center. Shared (telerobotic) control indicates a combination of human control (teleoperation), autonomous sensor-based control (robotic). The rate-based system allows multiple manipulators operating in closed chains by grasping the same payload, using only one set of Cartesian inputs. Multiple, sensor-based input sources are allowed including joystick, position controller, machine vision controller, and force/torque controller. These multiple sources can potentially operate simultaneously, for all Cartesian axes. The proposed system forms the basis for control of dexterous, task-reflexive remote telerobotic systems.

2. CONTROL ARCHITECTURE FOR SINGLE-MANIPULATOR SYSTEMS

The proposed shared-control, sensor-based control architecture is shown in Fig. 1 for single manipulator systems. Control inputs can be generated simultaneously from various components including joystick (possibly force-reflecting), position controller, vision controller, and force controller. These components are discussed in ensuing subsections. The resolved-rate algorithm is used for motion control so all control inputs must be converted to rate commands. Cartesian rate inputs from all active components are added to form the overall Cartesian rate input $\dot{X}_{MRF}$ to the resolved-rate algorithm. This linear summation is a benefit of controlling in the rate domain.

Figure 2 shows the coordinate frame definitions for a single-manipulator system. For clarity, dextral Cartesian coordinate frames are represented by solid dots in Fig. 2. The *Base* frame is attached to the manipulator before the first moving joint. The *Wrist* frame is attached to the last moving link at its joint. The *World* frame is an inertially-fixed reference frame. The *MRF (Moving Reference Frame)* is a user-defined frame which is being controlled. The *MRF* can be placed anywhere as long as it is rigidly attached to the last manipulator link (such as on a grasped payload or even off the physical link). The *CRF (Control Reference Frame)* is a user-defined frame with respect to which the *MRF* is controlled . Cartesian velocities may be commanded in the coordinates of any frame, but all motion relates the *MRF* to the *CRF*.

The control frames in Fig. 2 are defined for generality. The *CRF* can be moving and the *Base* can also be moving independently with respect to the *World*. For a static *Base* frame, it can be defined identical to the *World*. The *MRF* can be changed during tasks and is defined to facilitate task completion. (For example, the *MRF* is defined on the beam node in a beam assembly problem. In this case the

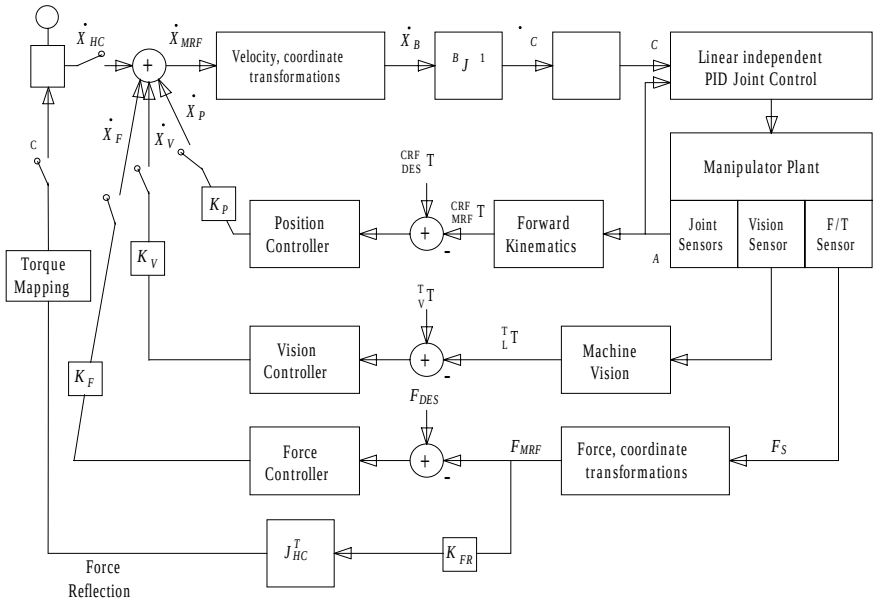

Figure 1. Shared, Sensor-Based Telerobotic Control Architecture for Single Manipulator Systems

CRF would be the target connecting node location.) The inclusion of the *MRF* and *CRF* is intended to decouple the task (including a human operator) from the manipulator. The frames *L* and *S* are the camera lens and F/T sensor frames; both are rigidly attached to the *Wrist* and *MRF*.

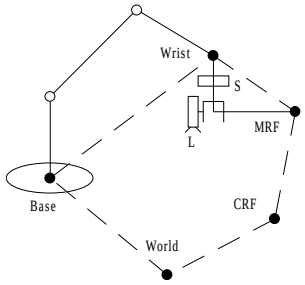

Figure 2. Single Manipulator Coordinate Frames

2.1 Resolved-Rate Control. The resolved-rate control algorithm is used for motion control from all input sources: joystick, position, vision, and force controllers. The algorithm implemented is based on Whitney's method [6]. This section assumes a static *Base* and *CRF*; the method can be extended to handle moving *Base* and *CRF* frames for dynamic tasks. The time-varying manipulator Jacobian matrix maps joint rates to Cartesian rates of the *Wrist*: ${}^k\dot{X}_W = {}^kJ\dot{\theta}$. The Cartesian rates ${}^k\dot{X}_W = {}^k\left\{\underline{v}_W \quad \underline{\omega}_W\right\}^T$ express the translational and rotational velocities of the *Wrist* with respect to the *Base*, expressed in the coordinates of any frame $\{k\}$. Common choices are *k=Wrist* or *k=Base*; simplest symbolic terms for the Jacobian matrix result when *k* is the frame midway between the *Base* and *Wrist*, often the *Elbow* frame. The equation ${}^k\dot{X}_W = {}^kJ\dot{\theta}$ must be inverted (or, more efficiently, solved by Gaussian elimination) at each control step. First, however, the input *MRF* Cartesian rates $\dot{X}_{MRF}$ (the sum of all control inputs, expressed in *MRF*) must be converted to the resolved rate input ${}^k\dot{X}_W$. If a leading superscript is omitted, it is assumed that the frame of expression matches the frame in the subscript.

The first step is find the equivalent Cartesian velocities of the *Wrist* frame to produce $\dot{X}_{MRF}$. This is obtained using the rigid-body velocity transformation in Eq. 1 [7]; the frame of expression is *Wrist*.

$$ {}^W\dot{X}_W = \begin{Bmatrix} \underline{v}_W \\ \underline{\omega}_W \end{Bmatrix} = \begin{bmatrix} {}^W_{MRF}R & {}^WP_{MRF} \times {}^W_{MRF}R \\ 0 & {}^W_{MRF}R \end{bmatrix} \begin{Bmatrix} \underline{v}_{MRF} \\ \underline{\omega}_{MRF} \end{Bmatrix} \quad (1) $$

Next the coordinate transformation of Eq. 2 is used to express the input in frame *k* of the Jacobian matrix.

$$ {}^k\dot{X}_W = {}^k\begin{Bmatrix} \underline{v}_W \\ \underline{\omega}_W \end{Bmatrix} = \begin{bmatrix} {}^k_WR & 0 \\ 0 & {}^k_WR \end{bmatrix} \begin{Bmatrix} \underline{v}_W \\ \underline{\omega}_W \end{Bmatrix} \quad (2) $$

Now the rate equation may be inverted to calculate the instantaneous joint rates necessary to obtain the commanded ${}^k\dot{X}_W$:

$$ \dot{\Theta}_C = {}^kJ^{-1}\,{}^k\dot{X}_W \quad (3) $$

The commanded joint rates are numerically integrated to commanded joint angles Θ_C. These angles are commanded to the manipulator and achieved using linear independent PID control laws. Joint encoder feedback Θ_A is used to form the errors for servo control.

This algorithm is sensitive to singularities, where the manipulator loses freedom to move in one or more Cartesian direction. In the neighborhood of singularities, extremely high joint rates are theoretically required to satisfy a finite Cartesian command. To deal with this problem, the determinant of the Jacobian matrix ${}^{k}J$ must be monitored. When the determinant approaches zero, the matrix inverse (or Gaussian elimination) in Eq. 3 is replaced by a matrix pseudoinverse based on the Singular Value Decomposition (SVD). Near singularities, the exact Cartesian command ${}^{k}\dot{X}_W$ cannot be satisfied, but the SVD will yield bounded joint rates which will move the manipulator through the singular neighborhood until Eq. 3 can take over again.

Compared to an inverse position algorithm, the resolved-rate algorithm is attractive because it is a linearized, unique solution (assuming full rank for the Jacobian matrix). Also, control inputs from various sources are summed linearly to form the final command.

2.2 Teleoperation Joysticks. Joysticks (or hand controllers) are the input device to telerobotic systems for teleoperation by a human. Two three-dof hand controllers or one six-dof joystick can be used to input rates $\dot{X}_{HC}$ to the *MRF* (shown in Fig. 3a) with coordinates in any frame *k*. Vector gains K_{HC} are used to scale the joystick output and convert it to proper units. Alternatively, poses (positions and orientations, ${}^{CRF}_{DES}T$, see Section 2.3) can be commanded to the manipulator. Joysticks need not be kinematically similar to the manipulator because the interface between the controllers and manipulator is in Cartesian space. Joysticks with force-reflection capability are discussed in Section 2.5.

2.3 Position Control. Resolved-rate control may be used to command manipulator poses (positions and orientations) by including a position loop around the rate system. An error vector $\underline{E} = \underline{R} - \underline{C}$ must be calculated as shown in Fig. 3b, giving the difference between the commanded (${}^{CRF}_{DES}T$) and current (${}^{CRF}_{MRF}T$) manipulator pose. The current pose is found from the forward kinematics transformation of joint encoder feedback Θ_A and other known transformation matrices.

$$ {}^{CRF}_{MRF}T = {}^{CRF}_{WO}T\,{}^{WO}_{B}T\,{}^{B}_{W}T(\Theta_A)\,{}^{W}_{MRF}T \qquad (4) $$

The error vector is found by algebraic subtraction of the position vectors. However, because the orientation cannot be represented by vectors ($\underline{E} = \underline{R} - \underline{C}$ was used above for conceptual convenience), the angular velocity error is calculated using a rotation matrix "difference".

$$ {}^{MRF}_{DES}R = {}^{CRF}_{MRF}R^{-1}\,{}^{CRF}_{DES}R = {}^{CRF}_{MRF}R^{T}\,{}^{CRF}_{DES}R \qquad (5) $$

Choosing a rotation convention (e.g. 3-2-1 Euler, [7]), three rotation numbers are extracted from ${}^{MRF}_{DES}R$. Taking these three numbers as both the Euler angles and respective rates, the commanded angular velocity error vector can be calculated using the appropriate rotational kinematic differential equations (e.g. see [8], App. II).

The position and orientation error vector $\dot{X}_P$ is added into the summing junction as seen in Fig. 1, after applying the vector gain K_P (with translational units 1/*s* and unitless rotational components).

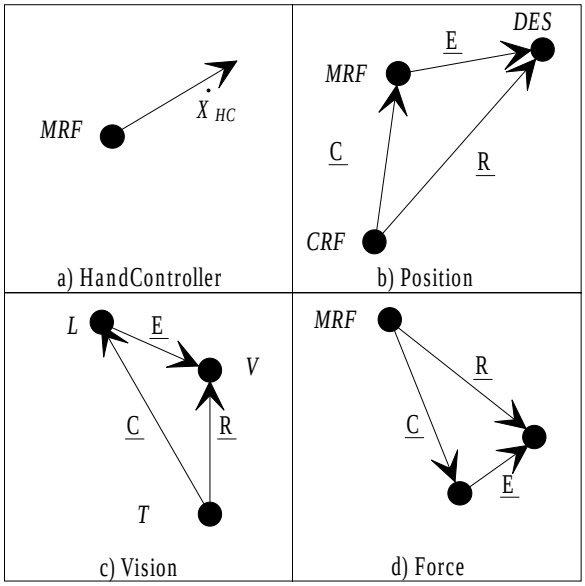

Figure 3. Error Vectors for the Control Algorithms

2.4 Vision Control. A machine vision controller [9] is used to move the manipulator from the current state $\underline{C}$ (sensed using a camera) to the desired state $\underline{R}$ represented by an image stored in memory. An error vector $\underline{E} = \underline{R} - \underline{C}$ is calculated as shown in Fig. 3c, where *L* is the *Lens* frame, *T* the *Target*, and *V* the desired manipulator pose. The vision algorithms use homogeneous transformation matrices to represent the important poses and so the subtraction is handled in a similar way to that discussed for the position controller. The current pose sensed by the camera is ${}^{L}_{T}T$ but the pose represented conceptually by $\underline{C}$ is its inverse ${}^{T}_{L}T$ and the desired pose derived from the stored image is ${}^{V}_{T}T$, but the pose represented conceptually by $\underline{R}$ is ${}^{T}_{V}T$. This is

similar to the position controller, where *Lens* plays the part of *MRF*, *V* is *DES* and *Target* is the *CRF*.

The error vector is multiplied by the vector gain K_V (same units as K_P) to yield an input rate command $\dot{X}_V$. This command is first transformed from *Lens* to *MRF* using rigid-body velocity transformations and coordinate rotations similar to Eqs. 1 and 2 before being sent to the summing junction in Fig. 1.

2.5 Force Control. An active force controller has been implemented in the resolved-rate scheme to command forces to the environment with the manipulator. This active force controller is basically a general impedance controller [10] with only the damping term. A six-dof force/torque sensor (with frame *S*) mounted after the last joint reads the contact wrench $F_s = \left\{ \underline{f}_S \quad \underline{m}_S \right\}^T$ (force and moment vectors). The weight and gravity-moment of the end-effector mounted outboard of the force/torque sensor (transformed to *S*) must be subtracted from the sensor reading. This modified sensor reading in *S* must be transformed by rigid body transformations and coordinate rotations [7] to the equivalent *MRF* wrench:

$$F_{MRF} = \begin{Bmatrix} \underline{f}_{MRF} \\ \underline{m}_{MRF} \end{Bmatrix} = \begin{bmatrix} {}^{MRF}_{S}R & 0 \\ {}^{MRF}P_S \times {}^{MRF}_{S}R & {}^{MRF}_{S}R \end{bmatrix} \begin{Bmatrix} \underline{f}_S \\ m_S \end{Bmatrix} \quad (6)$$

An error vector $\underline{E} = \underline{R} - \underline{C}$ must be calculated as shown in Fig. 3d, where $\underline{R}$ is the commanded wrench F_{DES} and $\underline{C}$ is the sensed wrench F_{MRF}, both in the *MRF*. Since both force and moment are vector quantities, algebraic subtraction applies. The wrench error $\underline{E}$ is converted to a rate $\dot{X}_F$ (via the vector gain K_F with force units *m*/*Ns* and moment units 1/*Ns*), which is sent to the summing junction in Fig. 1. The force controller drives manipulator motion so the desired wrench is achieved continuously.

If zero force is commanded and manipulator grasps an object, the motion will automatically align the gripper for minimal contact wrench and misalignments. This is called force-moment accommodation (FMA).

If a force-reflecting joystick is available, the sensed *MRF* wrench can also be sent to the operator's hand so she can feel the task forces and moments exerted by the manipulator. The required transformation is [7]:

$$\tau = J_{HC}^T F_{MRF}, \quad (7)$$

where τ is the vector of joystick joint torques/forces, J_{HC} is the joystick Jacobian matrix, and F_{MRF} is from above. If the joystick Jacobian is derived for the wrist relative to the base, a rigid body wrench transformation to the hand-grip similar to Eq. 6 is required. F_{MRF} is scaled by matrix gain K_{FR} to get the F_{MRF} shown in Eq. 7. The joint torques are achieved by torque mapping to the force-reflecting hand controller.

2.6 Simultaneous Control. In the control architecture of Fig. 1, all input sources (i.e. joystick, position, vision, and force) can be enabled simultaneously for all Cartesian axes. In most other experimental telerobotics systems the authors are acquainted with, only one input source is enabled at any one time and changing between sources requires artificial software or hardware switches.

Often different input sources will result in competing goals (e.g. different poses commanded to the position and vision controllers). Therefore, software switches are included (set by script file keyboard input) to enable or disable each input source during the execution of tasks. Also, zero values in the vector gains K_{HC}, K_P, K_V, and K_F can be used to disable some or all Cartesian axes from the input sources.

Raibert and Craig [11] present a method for hybrid position/force control where certain Cartesian axes are chosen for position control and the remaining ones for force control. The current system can achieve this by proper placement of zeros in K_P and K_F. However, the authors achieved excellent control characteristics in transitioning between unconstrained motion to manipulator/workspace contact by combining rate control and force-moment accommodation (FMA) on all axes simultaneously. This is termed Naturally-Transitioning Rate-to-Force Control and is presented in [12].

A limitation of the proposed control architecture is that the gains are tuned heuristically. Gain scheduling is allowed but there is no theoretical basis for computing the gains. As the manual gain selection is necessarily conservative to achieve stability, it is likely that suboptimal performance is obtained.

2.7 Shared Control. The proposed control architecture allows shared control, which is control by a human operator (teleoperation), autonomous sensor-based control (robotic), or a combination of both modes (telerobotic). In this system the human controls the system via the joystick or through keyboard inputs. The joystick input is integrated seamlessly. For instance, if it appears the automated system will drive the end-effector into an obstacle the operator can modify the trajectory in real-time by using the joystick. After the danger is past and the joystick input is zero the original target pose is still reached by the manipulator. Even though the system allows shared control, the emphasis is on increasing sensor-based autonomy, allowing the operator to assume a supervisory role.

2.8 Kinematically-Redundant Manipulators. The development above assumes the manipulator has the same number of degrees-of-freedom (dof) as the task requirement. For instance, for general three-dimensional motion, a six-dof manipulator is required. Kinematically-redundant manipulators (which have more dof than the task requirement) have been used to satisfy motion trajectories and optimize performance with the extra dof. The proposed control architecture can be adapted for use with a kinematically-redundant manipulator by replacing Eq. 3 with:

$$\dot{\Theta}_C = {}^k J^{+\,k}\dot{X}_W + \left(I - {}^k J^{+\,k} J\right) z \qquad (8)$$

In Eq. 8, the first term ${}^k J^{+\,k}\dot{X}_W$ is the particular solution which satisfies the input Cartesian rates ${}^k\dot{X}_W$. The pseudoinverse of the manipulator Jacobian matrix is used, ${}^k J^{+} = {}^k J^T \left({}^k J\, {}^k J^T\right)^{-1}$, because the Jacobian is underconstrained (more columns than rows). In the vicinity of manipulator singularities, ${}^k J^{+}$ can be computed using SVD. The second term in Eq. 8, $\left(I - {}^k J^{+\,k} J\right) z$, projects a vector *z* into the null space of the Jacobian matrix. This vector is chosen to be $z = c\nabla H(\Theta)$ in order to minimize or maximize an objective function $H(\Theta)$, where *c* is a gain constant. Objective functions can avoid singularities, avoid joint limits, avoid obstacles, and minimize energy, among others. The proposed control architecture has been applied to kinematically-redundant manipulators [13].

3. CONTROL ARCHITECTURE FOR MULTIPLE-MANIPULATOR SYSTEMS

Section 2 presents the control architecture for shared, multiple-sensor input control of a single manipulator system. The current section extends this development to allow multiple manipulators cooperating to manipulate the same payload. Figure 4 shows two manipulators grasping the same beam. A closed kinematic chain is formed. The general algorithm handles *n* manipulators, with *n Base* frames, *n Wrist* frames, but only one *World*, *CRF*, and *MRF* frame. There are also *n* force/torque sensor frames *S* and as many lens frames *L* as there are cameras (not shown for clarity). As before, all control input sources add to yield a Cartesian rate $\dot{X}_{MRF}$, where the *MRF* is attached to the grasped payload in a convenient location. This task rate is commanded to each of the *n* manipulators independently to obtain the desired coordination.

In many early multiple manipulator systems (e.g. [14], [15]), one manipulator is chosen as the master and the others (slaves) follow the motion of the master by way of force feedback. In the proposed system (see [16] for more information) this is not necessary; commands are naturally and independently distributed to the *n* manipulators by using the common *MRF*. Individual manipulators do not need to know what commands are sent to the other manipulators. As in the single manipulator case, control of the *MRF* relative to the *CRF* is intended to decouple the operator from the specific manipulators and instead focus on the task. Also, only one set of Cartesian inputs need be specified (via joystick or path planner) rather than one set for each manipulator.

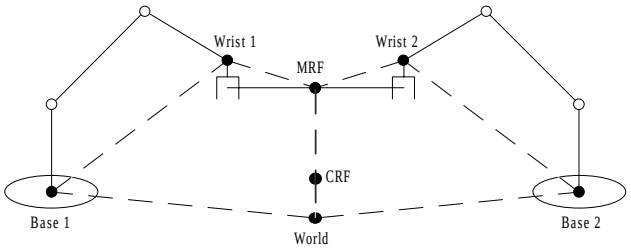

Figure 4. Dual Manipulator Frames

In Fig. 4 it can be imagined that errors in calibration of DH parameters and other uncertainties can lead to misalignment, antagonism, and reduced performance if not task failure. This problem can be overcome by enabling the force-moment accommodation (FMA, Section 2.5) for all manipulators at all times. In this way, the errors due to uncertainties will be automatically minimized while the task is underway. Also, the *MRF* definition is periodically updated when sensor readings and forward kinematics indicate the *MRF* as seen by different manipulators has diverged.

A method has been implemented under the proposed multiple manipulator control architecture for automatically balancing the loads from a jointly-manipulated payload [17]. The method requires a wrist-mounted force/torque sensor for each of the *n* manipulators. Distribution of loads among the *n* manipulators is handled by minimizing a quadratic cost function in task-space wrench. In this way, manipulators cooperating on the same task can exert unequal loads on the mutually-grasped payload.

In most multiple-manipulator tasks there is a combination of jointly-manipulated motions and separate motions. For the jointly-manipulated motions, the discussion in this section applies, relying on the methods of Section 2. For separate manipulator motions, there are *n* independently-controlled *MRFs* and the methods of Section 2 apply. Currently, collision avoidance in separate and coordinated motions is performed heuristically by task design.

4. EXAMPLE

A brief example is given to demonstrate task-level operation of the proposed control architecture for a dual-manipulator system performing insertion of a beam simultaneously into two nodes for assembly. Figure 4 shows the scenario for this example. Force-moment accommodation (FMA) is enabled at all times for both manipulators. For free motion of the beam, the MRF can be attached to the grappled beam near its midpoint. In this way rotations of the beam will be very convenient. Assume the beam has been grasped by both manipulators. The first step is to use the position controller (alternatively, teleoperation via the joystick) to command the *MRF* at the midpoint to a pose in the vicinity of the assembled configuration. Next the *MRF* is alternated between beam ends while the vision controllers of first one manipulator then the other acquire the targets (mounted on the beam receptacle nodes). These targets are designed to place the beam in the proper pose some distance above the insertion points. After some iteration the force controller takes over to move the beam (alternatively, teleoperation via the joystick) until a sufficient force is sensed in the insertion direction (with all other forces/moments zero to allow smooth alignment).

In this simple example, primarily one control input is enabled at a time. However, FMA operates in parallel with each one, for all Cartesian axes simultaneously. Also, the operator may assist the trajectory at any time via the joystick.

5. CONCLUSION

This paper has presented a unique rate-based architecture for shared, simultaneous sensor-input, real-time, task-reactive control for remote, multiple-manipulator telerobotic systems. The control is shared: it allows a combination of teleoperation, telerobotic, and autonomous modes. Multiple input sources (joystick, position, vision, and force/moment) can potentially control the system at the same time, for all Cartesian axes. Sometimes these sources can be competing, such as two different poses commanded to the position and vision algorithms simultaneously, which is to be avoided by proper software switches and/or gains. Other combinations of control input result in a sum which is greater than the parts, such as when free velocity commands from the joystick naturally transition to force commands in contact by enabling force moment accommodation. Practical implementation has shown the system to be effective in accomplishing simulated remote space tasks. The system was designed to perform well despite uncertainties in system parameters, uncertainties in the task, and imperfect manipulator and environment models.

6. REFERENCES

[1] Neira, J., et. al., "Multisensor Mobile Robot Localization", *IEEE International Conference on Robotics & Automation*, Minneapolis, MN, April, 1996.

[2] Anderson, R.J., "Autonomous, Teleoperated, and Shared Control of Robot Systems", *IEEE International Conference on Robotics & Automation*, Minneapolis, MN, April, 1996.

[3] Brady, K.J., et. al., "Modular Controller Architecture for Multi-Arm Telerobotic Systems", *IEEE International Conference on Robotics & Automation*, Minneapolis, MN, April, 1996.

[4] Lewis, C.L., "Trajectory Generation for 2 Robots Cooperating to Perform a Task", *IEEE International Conference on Robotics & Automation*, Minneapolis, MN, April, 1996.

[5] Zhao, C.S., et. al., "Collision-free Path Planning for a Robot with Two Arms Cooperating in the 3-D Workspace", *IEEE International Conference on Robotics & Automation*, Minneapolis, MN, April, 1996.

[6] Whitney, D.E., "Resolved Motion Rate Control of Manipulators and Human Prostheses", **IEEE Transactions on Man-Machine Systems**, June 1969.

[7] Craig, J.J., **Introduction to Robotics: Mechanics and Control**, Addison-Wesley, 1989.

[8] Kane, T.R., Likins, P.W., and Levinson, D.A., **Spacecraft Dynamics**, McGraw-Hill, 1983.

[9] Cornils, K., and Goode, P.W., "Location of Planar Targets in Three Space from Monocular Images", *NASA Goddard Conference on Space Applications of AI and Robotics*, 1987.

[10] Hogan, N., "Impedance Control: An Approach to Manipulation", **ASME Journal of Dynamic Systems, Measurement, and Control**, March, 1985.

[11] Raibert, M., and Craig, J., "Hybrid Position/Force Control of Manipulators", **ASME Journal of Dynamic Systems, Measurement, and Control**, June, 1981.

[12] Williams, R.L., II, Harrison, F.W., and Soloway, D.I., "Naturally-Transitioning Rate-to-Force Controller for Manipulators", *IEEE International Conference on Robotics & Automation*, Minneapolis, MN, April, 1996.

[13] Williams, R.L., "Local Performance Optimization for a Class of Eight DOF Manipulators", *IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994.

[14] Ishida, T., "Force Control in Coordination of Two Arms", *Proc. Fifth Int. Conf. AI*, August 1977, pp. 717-722.

[15] Alford, C.O., and Belyu, S.M., "Coordinated Control of Two Robot Arms", *IEEE International Conference on Robotics and Automation*, March 1984, pp. 468-473.

[16] Barker, I.K., and Soloway, D.I., "Coordination of Multiple Robot Arms", *Workshop on Space Telerobotics*, January, 1987.

[17] Alberts, T.E., and Soloway, D.I., "Force Control of a Multi-Arm Robot System", *IEEE International Conference on Robotics and Automation*, April, 1988.