N89-10092

A Shared Position/Force Control Methodology for Teleoperation

Jin S. Lee

RCA Corporation, Advanced Technology Laboratories
Moorestown, New Jersey 08057

RK 11 36443

ABSTRACT

A flexible and computationally efficient shared position/force control concept and its implementation in the Robot Control C Library (RCCL) are presented from the point of teleoperation. This methodology enables certain degrees of freedom to be position-controlled through realtime manual inputs and the remaining degrees of freedom to be force-controlled by computer. Functionally, it is a hybrid control scheme in that certain degrees of freedom are designated to be under position control, and the remaining degrees of freedom to be under force control. However, the methodology is also a shared control scheme because some degrees of freedom can be put under manual control and the other degrees of freedom put under computer control.

Unlike other hybrid control schemes, which process position and force commands independently, this scheme provides a force control loop built on top of a position control innerloop. This feature minimizes the computational burden and increases disturbance rejection. A simple implementation is achieved partly because the joint control servos that are part of most robots can be used to provide the position control innerloop.

Along with this control scheme, several menus have been implemented for the convenience of the user. As a result, the user can define a center of compliance on any point in the workspace and a selection matrix that assigns certain axes under position control and other axes under force control. Finally, the user can define force gains in the force control strategy to ensure overall system stability and to avoid overshoots and oscillations in the robot motions.

The implemented control scheme has been successfully demonstrated for the tasks of hinged-panel opening and peg-in-hole insertion.

INTRODUCTION

Over the past several years, many compliant control techniques have been proposed and developed to extend robot applications to a wide variety of tasks requiring compliance. Compliance is almost inevitably required when the robot manipulator comes into contact with the environment and its position is constrained. Slight position errors of the robot may produce enormous forces and cause serious damage both to the robot and the workpiece. While compliant motion can be provided by a passive mechanical compliance device such as a remote center compliance (RCC), the development described in this paper is centered around digitally implemented active compliance, which is more flexible and can be reconfigured in realtime.

Most active compliance techniques, however, have been developed for autonomous operations, in which a complete task is executed under computer control. Automated compliant control, while effective for structured tasks, does not provide the operator with direct control of the position at which the operation is performed. As a result, it is not suitable for unstructured tasks. Autonomous control techniques at the present time are neither intelligent nor reliable enough to perform any but the simplest and most routine tasks.

In teleoperation, force reflecting master/slave hand controllers have been used for many years to control remote robots in unstructured hazardous environments [1]. They provide an operator with an accurate experience of forces encountered by the robot and the illusion of doing the task directly. They are widely used in the nuclear and undersea environments, where no restrictions are imposed on size, and no communication time delays exist between control station and robot manipulator. These devices, however, cannot be readily extended to unmanned-orbit servicing tasks because of limitations on the size of the control station and communication time delays between control station and servicer.

Another alternative in teleoperation is the use of the resolved rate joystick. The joystick's compactness and the operator's familiarity with it makes it a logical candidate for use in telerobotic tasks. It is, however, a strict rate-control device without any force feedback and, therefore, not suitable for compliant control tasks.

This paper reports on the extension of compliant control techniques to teleoperated systems and the development of new concepts to accomplish compliant control tasks under joystick control. Developed at RCA's Advanced Technology Laboratories, the control mode presented is termed "shared position/ force control" or simply "shared control". Under shared control, the operator retains realtime control of robot motion while leaving the responsibility for compliance to a computer local to the robot.

An important potential application of shared control is in space telerobotic servicing. When the servicer is controlled from a shuttle or from the space station, the use of a joystick in shared control reduces the physical size of the master-slave hand controller, thereby making it possible to reduce the size of the control station. When the servicer is controlled from a ground station, excessive time delays (estimated at two to five seconds), resulting from space and ground communication links, prohibit realtime control of the servicer.

Shared control may be an important complement to the incremental "move-and-wait" tactic currently demonstrated for motions without force feedback. With an on-board controller, shared control provides for task adaptability at the work site, in which the tool, under local control, adapts to any excessive force. Initial investigations and tests of shared control techniques have opened promising new possibilities for more efficient and safe teleoperation control for both short-term and long-term satellite servicing.

Compliant Control

The behaviour of compliant motion can be systematically described by specifying a center of compliance and its compliance frame. A center of compliance is a point in the workspace in which a force applied to the point causes a motion in the direction of the force. Its compliance frame is an orthogonal coordinate frame, with its origin at the center of compliance. Thus, a compliant task is described in terms of desired position trajectories or force/torque profiles for each of the compliance frame axes. In many cases, the center of compliance is placed at the center of the end-effector.

There are two prevalent approaches in active compliant control: explicit feedback and hybrid control [2]. Explicit feedback specifies a linear relation between sensed forces* and the corresponding positions* accommodations [3, 4]. This is typically modelled by the equation:

$$f = K(p - p0)$$

where:

f = sensed force
p = the current position
$p0$ = the predefined nominal position
K = the gain matrix that relates sensed forces linearly to deviations from the nominal position.

This explicit feedback scheme is built around the joint servos, which process joint set points and drive joint actuators of the robot (Figure 1). It is computationally efficient and simple to implement on the joint servos provided with most robots. However, the force control gains must be appropriately selected for each task to ensure system stability and desirable performance.

The hybrid control approach, on the other hand, processes position and force commands independently through their own control loops (Figure 2) [5, 6]. It first selects certain degrees of freedom to be under position control and the others to be under force control, and then drives each actuator according to the sum of its contributions, whether force or position. The hybrid control scheme is computationally expensive but conceptually elegant because it can process force commands under any environment. It usually involves significant modification of existing joint servos to implement a force servo loop. Because of joint friction/stiction in some robots, force servo loop are harder to design and implement.

---

*In this paper, force implies force and torque and position implies position and orientation.

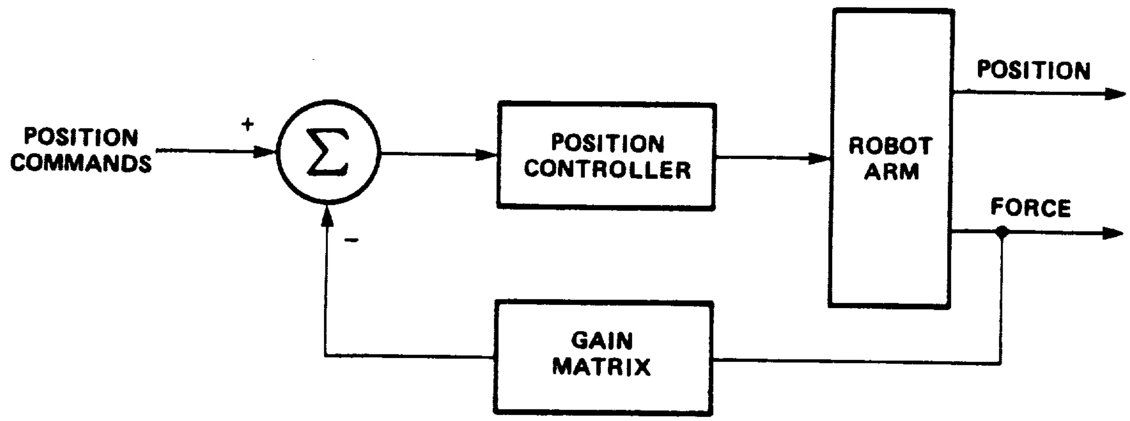

Figure 1. Explicit feedback block diagram.

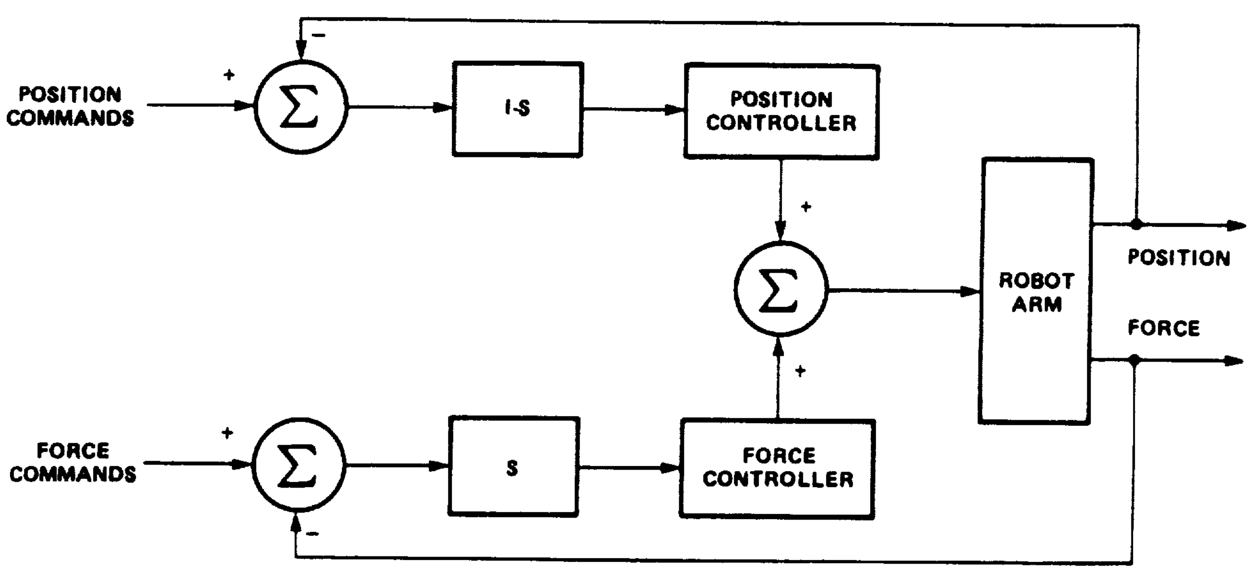

Figure 2. Hybrid control block diagram.

The compliant control algorithm in the shared position/force control is functionally a hybrid control scheme in that certain axes of the compliance frame are assigned for position control (rate control); while the other axes are assigned for force control. But the algorithm's force control loop is built on top of its position innerloop; hence, it is technically an explicit feedback scheme. As in explicit feedback, it is computationally efficient and simple to implement. The fast joint-based position innerloop automatically rejects disturbance torques arising from any source, even gravity and joint frictions/stictions [7].

Shared Position/Force Control

Figure 3 outlines the Cartesian-based shared control scheme currently used at RCA's Advanced Technology Laboratories to control a PUMA 762 robot. The implemented algorithm allows the operator to control selected position axes through realtime manual inputs or through predefined trajectories providing fully automated compliant task functions. The desired position values are derived from the joystick rate commands by integrating them. The control system itself consists of two feedback control loops: the inner PID joint servo loop and the outer force feedback loop. For the innerloop, it uses the standard PUMA 762 PID joint servos with a sampling rate of approximately 500 Hz. For the outer loop, it feeds sensed forces at the wrist sensor back to the position control loop with a sampling rate of approximately 36 Hz.

The position control scheme simply receives six rate values from the joysticks, selecting only a subset of those that correspond to position axes. Position axes are determined with a 6x6 diagonal selection matrix S. Each diagonal element of S, which is Boolean, is associated with each axis of the compliance frame. When its ith diagonal element is 0, the corresponding axis is under position control. When it is 1, the axis is under force control. The selected rate values are appropriately scaled with a 6x6 diagonal matrix K, which determines the desired robot velocity. The scaled rates are combined with the compensatory rates described in the text that follows, to form the combined rates at the compliance frame. These rates are integrated to derive the next desired Cartesian set points, which are then resolved to the robot end frame. These set points are then transformed to the joint set points, which are then input to the inner PID joint servos to drive the joint actuators of the PUMA 762 robot.

The force control loop is implemented around the inner joint PID servo loop, which is shared by the position control loop. The outer force control loop receives the predefined bias forces and drives position changes to produce those forces in the selected directions. It subtracts the sensed forces from the commanded forces and updates the force errors in the selected directions. The sensed forces at the compliance frame are computed from the sensed forces at the wrist force sensor and from the transformation T between the compliance frame and the sensor frame. The force control loop then scales the force errors via a force gain matrix to compute the compensatory rates. As described earlier, these rates are combined with the scaled rates to form the combined rates. The combined rates are then integrated to compute the next Cartesian set points. The gains in the force gain matrix must be

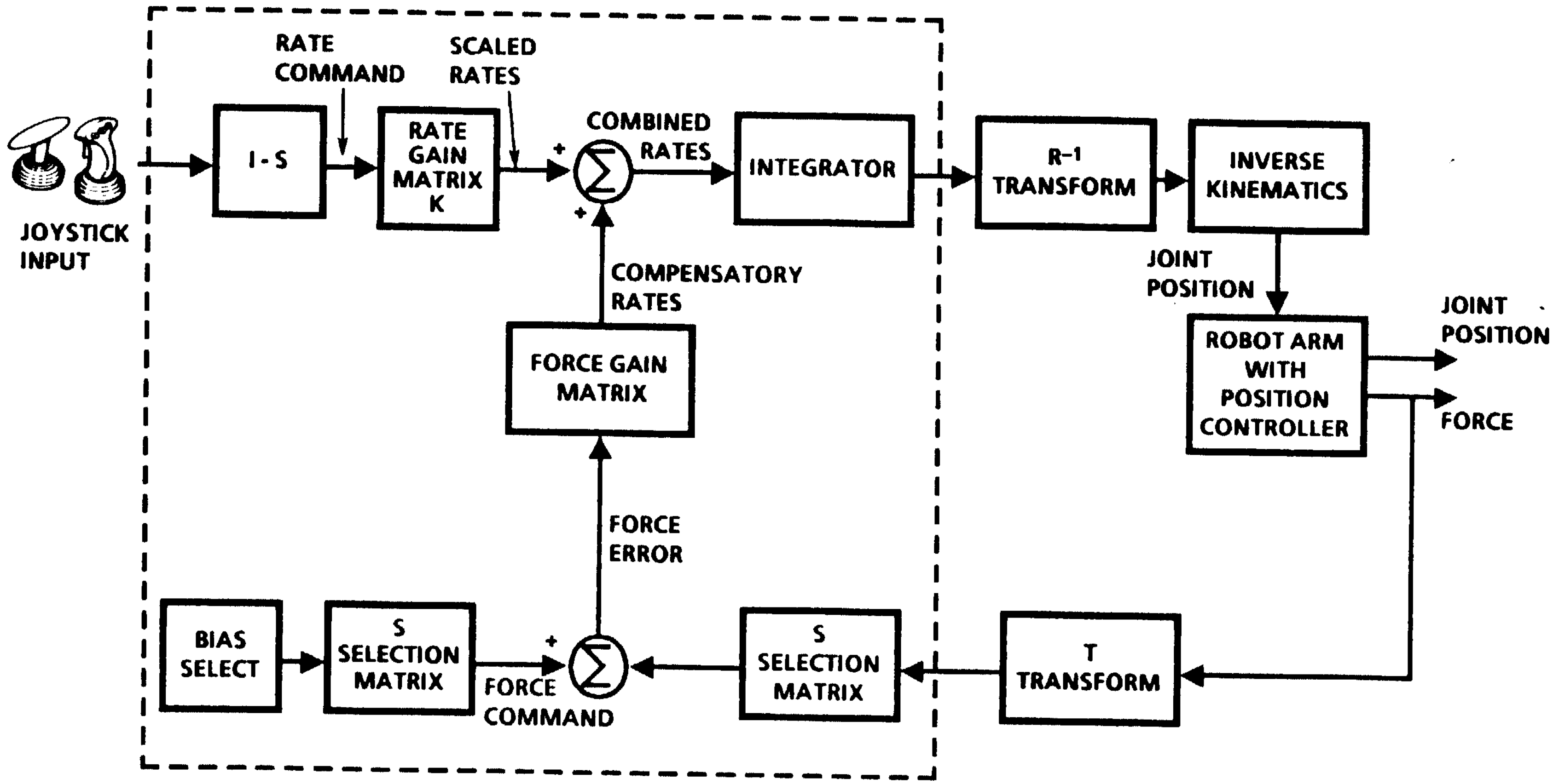

Figure 3. Shared position / force control scheme.

carefully tuned, not only to ensure system stability but also to prevent the closed loop system from overshoot and oscillation.

The inner joint servo loops interpolate between the commanded joint set points and the current joint set points, and they update the joint error actuating signals from the joint-interpolated set points and the joint encoder values. Next, each of these joint errors is regulated by each joint PID servo control algorithm. The gains of the PID control loops are tuned to make the closed loop systems sufficiently overdamped so that small perturbations and/or disturbances do not jeopardize the closed loop system stability.

Implementation in RCCL

RCCL is a general-purpose robot programming system originally developed at Purdue University by V. Hayward under the direction of R. P. Paul [8, 9] and later improved by J. Lloyd at McGill University [10]. Currently at RCA, RCCL is installed on a microVAX II to control a PUMA 762 [11].

RCCL's environment consists of two levels: the planning level and the control level. The control level is a C language software facility under the UNIX operating system that generates realtime Cartesian trajectories. Built on top of this is the planning level, which is a collection of C primitives and data structures that provide robot motion in Cartesian coordinates and joint coordinates. RCCL allows a user to modify the Cartesian trajectory in realtime by means of control level user functions. This facility is used to implement a shared control algorithm in the RCCL.

Figure 4 outlines the RCCL implementation of the shared control algorithm. In the RCCL planning-level user program, the user defines a simple transformation equation, which drives the robot according to external inputs such as joystick commands and sensed forces. The equation is described by T6*E = G, where T6, E, and G are the homogeneous transforms describing, respectively, the robot end, the compliance center from the robot end, and the desired set point of the compliance center. Predefined at the current compliance center position, G is functionally defined and computed in realtime in the control-level user functions. When the execution begins, the control function first collects joystick commands and sensed forces from the LSI 11/73, and then runs the control algorithm to compute the new combined rate V and the corresponding Cartesian set point G from the equation G = G*V. The RCCL trajectory generator then updates a new goal robot end transform T6 according to the new G and computes corresponding joint set points to drive the joint actuators.

Demonstrations

The shared control algorithm has been demonstrated for two compliance tasks: hinged-panel opening and peg-in-hole insertion.

(1) Hinged-Panel Opening

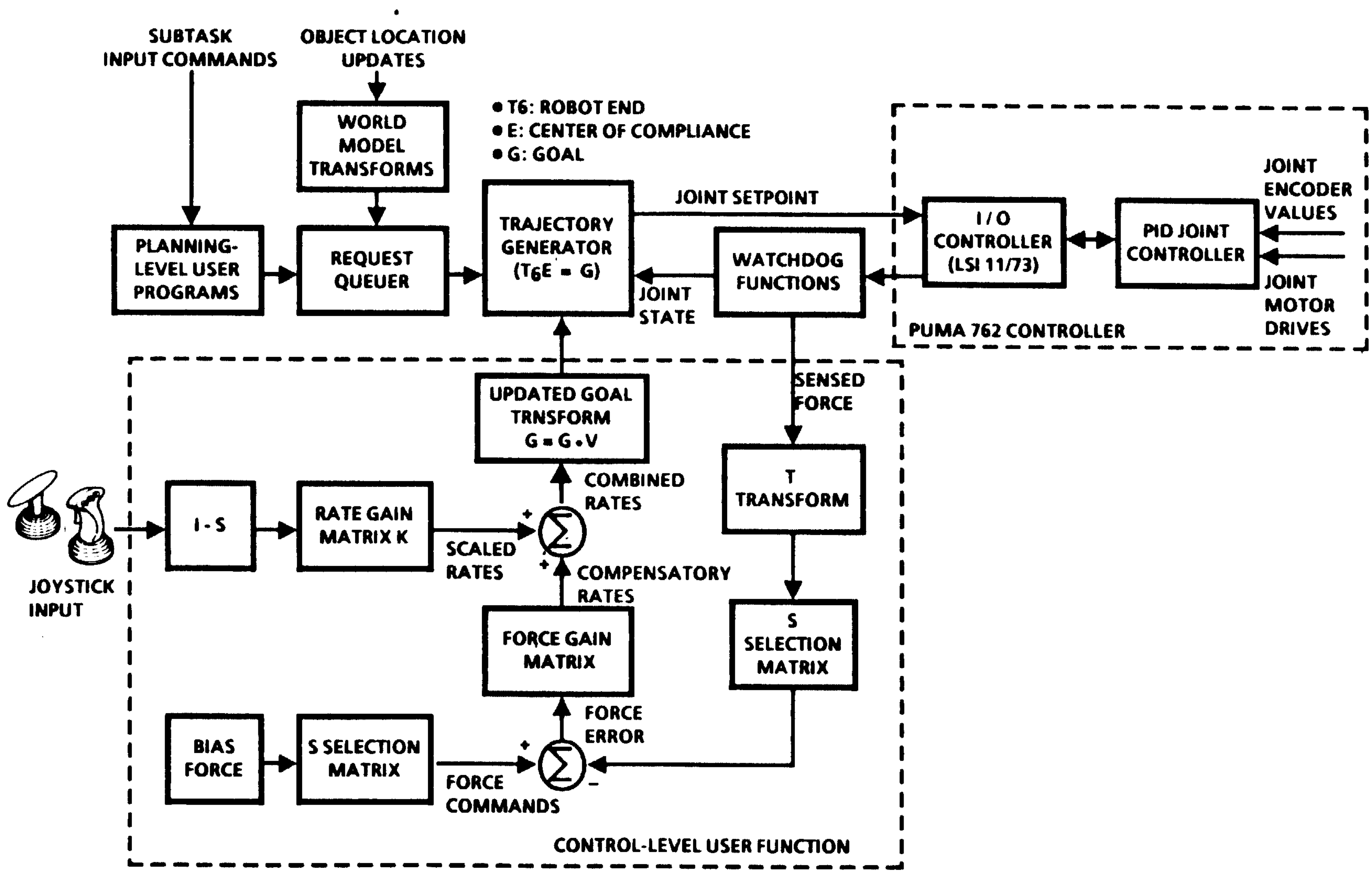

Figure 4. Shared position / force control algorithm implemented in the RCCL.

Figure 5 illustrates a hinged-panel opening task consisting of two subtasks: aligning the gripper to the doorknob and opening the hinged-panel. The compliance frame is defined at the center of the gripper, with the z-axis as the robot approach direction and the y-axis as the parallel gripper open/close direction.

When the robot grips the doorknob, a misalignment always occurs between the gripper and the doorknob. This misalignment is detected by a wrist sensor in terms of nontrivial force values along certain axes. By selecting these axes to be under force control, compensating positions are generated to accommodate to the geometric constraints of the doorknob.

When the robot opens the hinged-panel, the operator is concerned only with how far the hinged panel is open as he turns the doorknob in a direction normal to the current position, that is, along the z-axis of the compliance frame. The other axes are left to comply. A single axis of the joystick is then used to rate control the z-axis motion. Other motion adjustments are performed automatically to relieve forces in realtime. The force gain matrix must be carefully chosen to keep the gripper fully aligned to the doorknob. Otherwise, the misalignment may be increased, causing the system to be unstable.

The operator can arbitrarily stop the hinged-panel at any position and resume motion under manual control at will. The task is performed in realtime, with the natural motions and flexibility inherent in manual control. The same algorithm can be used to close the hinged-panel. This differs from automated compliant control, in which the program must be aborted to stop the motion, and can only be resumed by issuing a restart motion.

(2) Peg-In-Hole Insertion

The classical peg-insertion task is representative of assembly operations likely to be undertaken by robotic servicers. Figure 6 illustrates a peg-in-hole insertion task with a round tapered peg and a round hole with approximately a 1-mil clearance. The task is performed in two phases: the taper-crossing phase and the side contact phase. The compliance frame is defined at the end of the tapered peg, and again the z-axis is the robot approach direction. Here, the z-axis is a natural choice for position control, since the depth of insertion is the primary task parameter.

The operator approach the point of insertion in a purely manual mode. With the initial surface contact having been made, the taper-crossing phase begins. Still in the manual mode, the operator begins insertion with the rate under joystick control. Here, accommodation is made only in x and y-axes to slide the peg into the center of the hole. This phase in shared control mode continues until jamming occurs, which is indicated by excessive forces in the z direction. Then, the side-contact phase begins.

In this phase, angular alignments are the most important adjustments to release the peg from jamming and enable another insertion attempt.

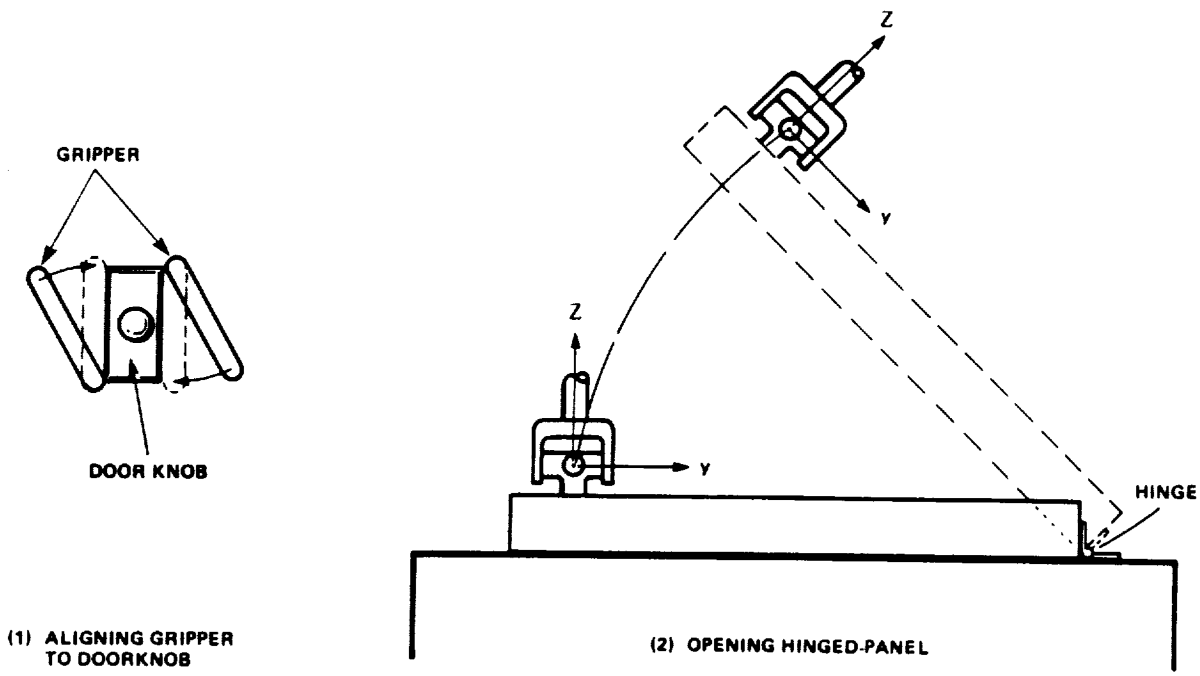

Figure 5. Hinged-panel opening task.

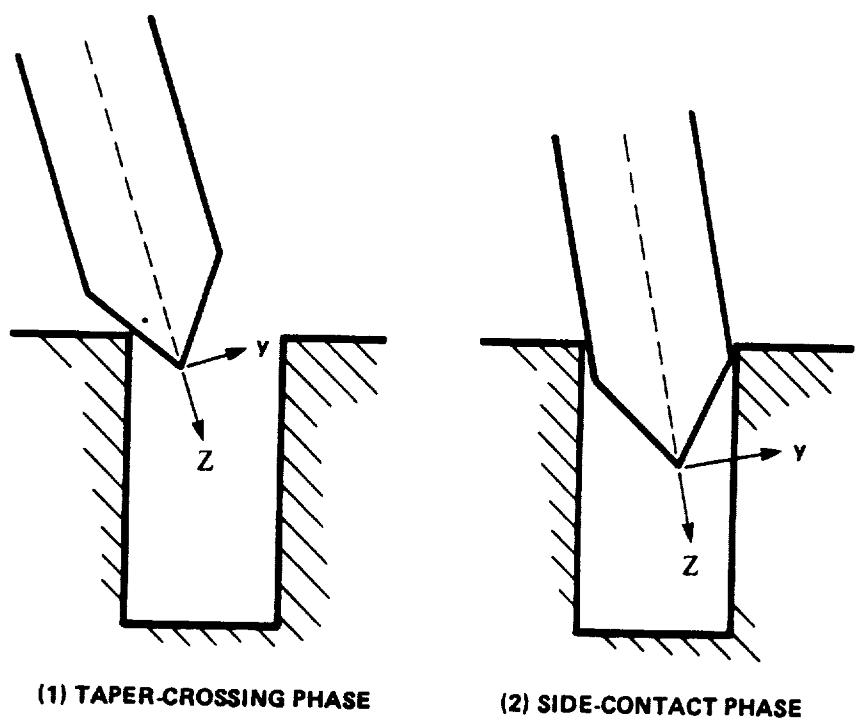

Figure 6. Peg-in-hole insertion task.

Except for the z-axis, which is under manual control, all other axes are selected for force control to provide both angular alignments and positional accommodations. With these arrangements, the operator can continue insertion, with all other axes complying to the geometric constraints of the hole. Because of tight geometric constraints, the force gains can be lowered significantly to boost system stability and performance.

Recommendations for Further Research

Under the current shared control scheme, force gains must be readjusted for each task to ensure stability and the efficient performance of the control system. To improve the current scheme, an adaptive algorithm should be developed to adjust these gains automatically for each task. The force control algorithm in the current shared control scheme can also be replaced by Raibert and Craig's hybrid control [5], where the adjustment of force gains is not necessary. This will involve the design and implementation of a PID-type force control algorithm similar in structure to joint position servos with a 500 Hz sampling rate.

References

1. A. K. Bejczy and J. K. Salisbury. "Kinesthetic Coupling Between Operator and Remote Manipulator," Proceedings of ASME Conference on Computer Technology, Volume 1. San Francisco, CA, August 1980.

2. M. T. Mason. "Compliance and Force Control for Computer Controlled Manipulators," IEEE Transactions on Systems, Man, and Cybernetics SMC-11, June 1981. pp. 418-432.

3. D. E. Whitney. "Force Feedback Control of Manipulator Fine Motions," Journal of Dynamic Systems, Measurement, and Control, June 1977. pp. 91-97.

4. J. K. Salisbury. "Active Stiffness Control of a Manipulator in Cartesian Coordinates," IEEE Conference on Decision and Control, November 1980.

5. M. H. Raibert and J. J. Craig. "Hybrid Position/Force Control of Manipulators," Journal of Dynamic Systems, Measurement and Control, 102, June 1981. pp. 126-133.

6. O. Khatib. "The Operational Space Formulation in Robot Manipulator Control," 15th ISIR, September 1985. Tokyo, Japan.

7. J. A. Maples and J. J. Becker. "Experiments in Force Control of Robotic Manipulators," Proceedings of IEEE Conference on Robotics and Automation, Volume 2, April 1985. San Francisco, CA.

8. V. Hayward and R. P. Paul. "Introduction to RCCL: A Robot Control C Library," IEEE First International Conference on Robotics, June 1984. Atlanta, Georgia.

9. V. Hayward and R. P. Paul. "Robot Manipulator Control under UNIX: RCCL, A Robot Control C Library," International Journal of Robotics Research, 5(3), Fall 1986.

10. J. E. Lloyd. "Implementation of a Robot Control Development Environment," Master of Engineering Thesis, Department of Electrical Engineering, McGill University, December 1985. Montreal, Canada.

11. J. S. Lee, S. Hayati, et al. "Implementation of RCCL, a Robot Control C Library on a MicroVAX II," Proceedings of the SPIE Conference on Advances in Intelligent Robotics Systems, Vol. 726, October 1986. Cambridge, MA.