FORCE-REFLECTIVE TELEOPERATED SYSTEM WITH SHARED AND COMPLIANT CONTROL CAPABILITIES

Z. Szakaly, W. S. Kim, A. K. Bejczy

Jet Propulsion Laboratory, California Institute of Technology,
Pasadena, California

ABSTRACT

The force-reflecting teleoperator breadboard described in this paper is the first system among available R&D systems with the following combined capabilities: (a) The master input device is not a replica of the slave arm. It is a general purpose device which can be applied to the control of different robot arms through proper mathematical transformations. (b) Force reflection generated in the master hand controller is referenced to forces and moments measured by a six-d.o.f. force-moment sensor at the base of the robot hand. (c) The system permits a smooth spectrum of operations between full manual, shared manual and automatic, and full automatic (called traded) control. (d) The system can be operated with variable compliance or stiffness in force-reflecting control. Some of the key points of the system are the data handling and computing architecture, the communication method and the handling of mathematical transformations. The architecture is a fully synchronized pipeline. The communication method achieves optimal use of a parallel communication channel between the "local" and "remote" computing nodes. A time delay box is also implemented in this communication channel permitting experiments with up to 8 sec. time delay. The mathematical transformations are computed faster than 1 msec so that control at each node can be operated at 1 kHz servo rate without interpolation. This results in an overall force-reflecting loop rate of 200 Hz.

I. INTRODUCTION

Force-reflecting master-slave manipulator systems are widely used in the industry in high radiation or in other dangerous environments. The major advantage of these systems is threefold. (i) The comparatively direct type of control of a six-degree-of-freedom device since control coordination of six joints in position control mode is inherent to these systems. The master arm movements provided by the operator through the hand grip and the movements of the slave arm fully agree in position, direction and velocity, and are synchronized. (ii) The genuine impression of forces and torques transmitted to the operator's hand with respect to the forces exercised or received. That is, force reflection which kinesthetically connects the operator to the working slave unit. (iii) The relatively high working speed resulting from the direct type of motion control.

The industrial master-slave force-reflecting (or bilateral) manipulator systems have two important characteristics. (i) The master arm is a duplicate (possibly a scaled-down duplicate) of the slave arm. (ii) Force-reflection in the servo-type master-slave systems is implemented through a bilateral-type position control, possibly with some current and differential velocity loops added to it. This means that the basic source of force feedback at the master arm is the position error between master and slave joints and not a genuinely sensed force at the slave.

Evolving capabilities in the technology of advanced robot control and intelligent interaction with remote robots are based on sensing and computing intelligence leading to flexible automation and flexible man-robot interaction. Following the principles of this modern technical approach, a laboratory research system has been developed at the Jet Propulsion Laboratory (JPL) for advanced force-reflecting teleoperation. Force reflection in this system is referenced to forces and torques sensed by a six-d.o.f. force-moment sensor at the base of the robot hand. The master device is a general purpose Force-Reflecting Hand Controller (FRHC), not a replica of any slave arm. It can be applied to the control of different robot arms through the proper kinematic transformations. The mechanism of FRHC is described in [1].

The JPL advanced force-reflective teleoperation system permits a spectrum of operations between full manual, shared manual and automatic, and full automatic (called traded) control, and can be operated with variable active compliance referenced to force-torque sensor in force-reflecting manual control. Shared manual and automatic control is implemented by freezing the data output of the master controller in some task space coordinates which are selectable by the operator from a menu. Motion in the frozen task space coordinates can then be controlled by a computer algorithm which can be referenced to force-moment or proximity sensor information. Variable compliance control is implemented through a low pass software filter in the hybrid position-force control loop. This permits the operator to control a "springy" or less stiff robot. Active compliance with damping can be varied by changing the filter parameters in the software menu. Setting the spring parameter to zero in the low pass filter will reduce it to a pure damper which results in a high stiffness in the hybrid position-force control loop.

First we briefly describe the overall system, its electronics architecture with related software development, and present capabilities. In the second part of the paper we discuss active compliance, communication time delay, experimental results and future development plans.

2. OVERALL SYSTEM

The advanced force-reflecting teleoperation system currently consists of (i) a six degree-of-freedom (d.o.f.) PUMA 560 robot arm, (ii) a smart robot hand on the robot arm equipped with a six-d.o.f. force-moment sensor, grasp force sensors and local processing and control electronics, (iii) a six-d.o.f. generalized Force-Reflecting Hand Controller (FRHC), (iv) two computing nodes for control and information display, one at the robot side and one at the FRHC (control station) site, and (v) computer graphics terminal at the control station site. Each computing and control node is built on a MULTIBUS using NS32016 microprocessors. The communication between the two nodes is on a parallel line. Integrated with each computing node is a compact, computerized Universal Motion Control (UMC) system developed at JPL providing rich motor and state sensing, control, safety and self-test capabilities. The computer graphics terminal utilizes (i) a PARALLAX graphics board to generate a real-time graphics display of force-moment and grasp force information and (ii) an IRIS graphics workstation to generate a real-time perspective graphics image of robot arm motion. Figure 1 shows the schematics of the overall system.

The UMC architecture and capabilities, developed at JPL, have been described in several earlier publications ([2] and [3]), where they can be found in more detail. In short, the UMC electronics consists of PWM power amplifiers for up to 1 kW motors and provides sensing of motion parameters at servo rates 1000 Hz. The communication from the motor control elements to the joint processor is a private bus called the BLX bus that makes the joint motion parameters memory mapped. It is notable that with the UMC up to 16 joints can be controlled by a single joint servo processor. The processor currently used is the NS 32016. There is a large number of processors from which we could choose. The NS 32000 family has proven to be a very good candidate for our task. The family has a number of processors with a wide performance range and object level compatibility between the members. Its assembly language has proven to be powerful as well as easy to use. The UMC electronics, thanks to the NASA Technology Utilization program, is now available commercially for up to 10 kW motors either brushed or brushless [4].

2.1 Electronics Architecture

To save development time we used the DB32000 development board which comes with a MULTIBUS interface. This forced us to use MULTIBUS for interprocessor communication. This is a lower bandwidth bus than more recent 32-bit busses, but the available bandwidth is more than enough for our application so the use of MULTIBUS did not hamper the performance of our system. With the upcoming development of new processor boards (still using the 32000 family) a new proprietary bus (the ZBUS) will be introduced that is optimized for high bandwidth shared memory applications.

The internode communication is done via a parallel port that carries one byte periodically at every 125 microseconds in each direction. The narrow bandwidth and periodic use of the communication channel are important parameters. If the usage of the channel is not periodic that means that the bandwidth has to be higher than the number of bytes transmitted per second. This is a waste of the channel bandwidth. This 125 μsec byte transfer rate is also used to synchronize the remote node to the local one. Eight bytes are transmitted in every servo loop from the local to the remote node. The first one is the header byte that is used to determine which byte belongs to which degree of freedom. This is followed by the position change of the X, Y, Z, pitch, yaw, roll degrees of freedom. The communication is done in relative Cartesian coordinates. In every servo loop a change in the range of -7 to +7 is transmitted. These changes are added by the receiver to the robot Cartesian position setpoint number. This method has a number of merits: (i) Small communication bandwidth used. (ii) Error tolerance in communication. (iii) Velocity limiting. (iv) Easy method of indexing the robot. It should be noted that this communication method does not cause any granularity in robot speed whatsoever. It simply limits the granularity of the robot position to 1/10th of a mm. The robot could not be positioned more accurately than that anyway.

The parallel internode communication cable in the future will be replaced by a fiber optic link with a much higher bandwidth, but the principle of communication between the two sides will remain the same.

Artificial time delay between the "local" and "remote" computing nodes has also been implemented to allow the experimental man-in-the-loop study of the effect of communication time delay on control performance. The time-delay MULTIBUS cardcage between the "local" and "remote" computing nodes contains two processors performing the time delay function between 2 ms and 8 sec.

In summary, the "local" node cardcage contains:

- Two joint interface cards (part of local UMC)
- PWM amplifiers for 8 motors (part of local UMC)
- Joint processor (part of local UMC)
- Kinematic transformation processor
- Communication processor with user interface
- Graphics processor
- Parallax graphics card

The "remote" node cardcage contains:

- Remote node UMC (3 cards and power amplifiers)
- Communication processor
- Inverse kinematics processor
- Forward kinematics processor

The communication from the smart end effector to the "remote" node and from the "remote" node to the IRIS graphics robot simulator is via fiber optic RS232 lines at 9600 baud rate.

Figure 2 shows the block diagram of the system and interconnections. Figure 3 indicates the timing of events and the sequence of computations. All computations are carried out at a 1000 Hz servo rate. The force feedback signal is currently received at a 125 Hz rate due to the limitation of the RS232 communication channel used between the Smart End Effector and the communication processor. The total round trip time delay is 5 ms for the position error-based force feedback and it is around 10 msecs for the sensor-based force feedback.

2.2 Software System and Development

The programming language used was the assembly of the NS 32016 itself since this promised the most performance and the fastest results. It has to be noted that the most convenient development environment such as a C cross compiler and UNIX operating system does not necessarily produce the fastest result and the best program performance. Compilers have the tendency to mask the real world of a processor from the programmer making it harder to generate complex interrupt hierarchies and hardware interfaces. We used a development system that one of us (Szakaly) wrote for the IBM-PC. This system makes it possible to edit and store the assembly source programs in the PC as well as up and download object files. In the current version an integrated assembler, developed at JPL, is used which runs on the IBM-PC. Portions of the system such as the force torque graphic display were developed in C using the SYS 32/20 development system marketed by National Semiconductor.

The motor control algorithm is a simple PD control loop. The servo rate is 1000 Hz overall allowing high gains to be used with the associated high tracking fidelity. The position gains are about 0.1 V/encoder unit. The UMC code generator program is used in the joint level controller. This program assures safe robot control by automatically generating the servo code that controls the joints. There is a set of parameters that have to be specified once for every robot. These parameters are stored in an electrically erasable EEPROM chip. When the program is activated it generates servo code and executes it. There is no possibility of breaking the robot due to human error in the coding.

The code generator is very flexible, it can control any number of motors up to 16, with any combination of hardware elements such as encoders, pots, temperature sensors, motors, brakes. All polarities are menu items so, for example, instead of having to switch the two encoder wires the user changes the encoder polarity from 'POS' to 'NEG' in the menu. The code generator will use a SUB instruction in place of an ADD in the servo code to accommodate the negative encoder hookup. The motor, the pot, the index and brake polarities can similarly be changed from the menu. The motor control processor interfaces to the rest of the system via the shared memory. More on the UMC software can be found in [5].

Since the remote node receives Cartesian position set points the inverse kinematic transformation is needed to calculate the robot joint position setpoints. This is carried out by one of the processors on the robot side. This transformation was implemented in integer arithmetic and takes around 700 µsecs to execute. Force feedback to the HC is based on robot position error as well as on force-torque sensor data so the robot end effector Cartesian position has to be computed as well. This is done by computing the robot forward kinematics.

The user has a large number of options available through the user interface. Every parameter can be changed on a degree of freedom basis. It is possible to activate a software spring for rate control on any degree of freedom that pulls the user's hand back to a center position. Any DOF may be in position or rate mode or it may be turned off. Any degree of freedom can have arbitrary force compliance with a zero or non-zero force setpoint. For example, orientation compliance with zero torque setpoint amounts to automatic peg alignment when performing peg insertion into a hole. An X compliance with non-zero force setpoint will press the end effector against the task board and will maintain contact force. Rate mode is useful when motion over large displacements is desired or when slow, constant velocity motion is the requirement.

Extensive experiments have been conducted to evaluate the usefulness of these operating modes and force feedback. The data shows that force feedback brings an improvement in terms of execution time as well as total force required. The shared control routines also bring about additional improvements. The experiments and results are described in detail in [6-8]. The ongoing experiments are concentrated on time-delayed operations using active compliance.

3. ACTIVE COMPLIANCE CONTROL

Variable active compliance has been implemented as a new feature to the current force-reflecting telerobot system. In a conventional telerobot system, each joint is controlled by a very stiff position servo, and thus the human operator has to control a stiff telerobot hand. A stiff telerobot hand tends to hit or bump into objects or walls hard. The implementation of active compliance, which emulates a programmable mechanical passive spring by computer software, allows the human operator to control a compliant or springy telerobot hand, not a stiff one. The compliant hand tends to touch objects or walls softly without exerting much force. It is also compliant to the environmental constraint, facilitating telemanipulation task performance. For example, in the peg-in-hole task, the compliant hand adjusts itself in accordance with the hole structure.

In order to implement active compliance on the telerobot hand, the force/torque signal sensed by the force/torque sensor (FTS) is first low pass filtered by computer software, and then fed back to the position/orientation output command signal (Figure 4). The force/torque sensor consisting of 8 pairs of strain gauges furnishes 3 force components (x, y, z) and 3 torque components (roll, pitch, yaw). Each of these 6 components, after low pass filtered, is individually fed back to the corresponding position (x, y, z) or orientation (roll, pitch, yaw) command input which comes from the hand controller controlled by the human operator. The mechanical equivalent of the above implementation consists of a spring connected in parallel with a damper (Figure 5). There are two parameters to control: compliance (or its inverse, stiffness) and damping (friction). The compliance of the active spring is proportional to the force feedback gain K. The damping (friction) of the active damper is proportional to T/K, where T is the time constant of the first-order low pass filter. In general, higher force feedback gain results in more compliance (less stiffness), but requires more damping (more sluggishness) to stabilize the system. If a pure gain is used instead of the low pass filter, a spring with no damper is realized. But, this turns out to be unstable. If an integrator is used instead of the low pass filter, a damper with no spring is implemented. A damper-alone system has a saturation problem due to the lack of a spring which allows "return-to-center" or enables the system to come back to the normal operating region when there is no force sensed. After a few runs of telemanipulation tasks, the damper tends to saturate and the damping effect disappears in the saturated direction.

Since a compliant telerobot hand is now implemented and available, the following two human-telerobot shared control schemes are suggested for efficient telemanipulation, depending upon the time delay. When the time delay is less than 1 second, approximately, both force reflection (long loop between the human operator and the telemanipulator) and active compliance (telerobot autonomous loop) can be used (Figure 6A). It is observed that a compliant telerobot tends to stabilize the force reflection long loop, and thus force reflection can be still useful even when the time delay is longer than 0.5 second. This also implies that the fidelity of the force reflection from the telerobot hand to the force-reflecting hand controller can be improved.

When the time delay is greater than 1 second approximately, active compliance alone without force reflection can be used (Figure 6B). This active compliance scheme turns out to be extremely useful when there is a long time delay. For example, a peg-in-hole task was successfully accomplished by a human operator even with 8 seconds time delay. It was not possible without active compliance. It took about 0.5 to 1 minute to complete the task with no time delay, 3 minutes with 4 seconds time delay, and 7 minutes with 8 seconds time delay. This proves the significance of the use of active compliance, since a conventional force-reflecting telemanipulator without active compliance cannot be used beyond about 0.5 seconds time delay due to the stability problem. The use of a compliant telerobot hand is also important for safety reasons, because the compliant hand tends to touch a wall softly without exerting much force or bumping into it hard.

4. FUTURE PLANS

To support a long list of man-machine interaction research topics (dual and redundant arm control, dexterous end effector control, coordinated manipulator and visual system control, etc.), the future developments in control electronics and control computing include: (a) An advanced bus architecture to eliminate the bottlenecks of commercial bus systems. (b) New processor cards using two NS 32016 processors or the NS 32332 processor within the advanced bus. (c) 5 Mbit and 15 Mbyte fiber optic links. (d) New smart hand electronics featuring very high (10 kHz) data rates with 12 bit A/D and with fiber optic link. (d) A new assembler to provide an environment similar to Turbo Pascal and other integrated development systems. After some experience with the new assembler improvements will be made to the syntax such that the usage will have the appearance of a high level language. This will provide many of the benefits of high level languages without the associated performance and control loss. This will facilitate to upgrade and expand the control software with new performance capabilities in supervisory control.

ACKNOWLEDGEMENT

This work was carried out at the Jet Propulsion Laboratory of the California Institute of Technology under contract to the National Aeronautics and Space Administration. The computing architecture, electronic hardware and software design and coding was done by Z. Szakaly. Active compliance control was implemented by W. S. Kim.

REFERENCES

[1] Bejczy, A. K., Salisbury, J. K. Jr., Controlling Remote Manipulators Through Kinesthetic Coupling. Computers in Mechanical Engineering (CIME), Vol. 2, No. 1., July 1983, pp. 48-60.

[2] Bejczy, A. K., Szakaly, Z. F., Universal Computer Control System (UCCS) for Space Telerobots. Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, NC, March 30 - April 3, 1987, pp. 318-324.

[3] Bejczy, A. K., Szakaly, Z. F., A Synchronized Computational Architecture for Generalized Bilateral Control of Robot Arms. Proceedings of the Conference on Advances in Intelligent Robotic Systems, SPIE and the International Society for Optical Engineering, Cambridge, MA, November 1-6, 1987, SPIE Vol. 851, pp. 123-134.

[4] Motion Tek, Box 9, Lord Ave., Brunswick, NY 12180.

[5] Szakaly, Z., and Bokor, E., Teleoperator Subsystem/Teleoperator Demonstrator: Hand Controller/Universal Controller Software Document. JPL D-5173, (Internal Document) January 1988.

[6] Bejczy, A. K., Hannaford, B., and Szakaly, Z., Multi-Mode Manual Control in Telerobotics. Proceedings of ROMANSY'88, Udine, Italy, September 1988.

[7] Bejczy, A. K., and Hannaford, B., Man-Machine Interaction in Space Telerobotics, Proceedings of the International Symposium on Teleoperation and Control, Bristol, England, July 1988, IFS Publ., Bedford, U.K., pp. 135-149.

[8] Hannaford, B., and Wood, L., Performance Evaluation of a 6-Axis High Fidelity Generalized Force-Reflecting Teleoperator. Proceedings of NASA Conference on Space Telerobotics, Pasadena, CA, January 1989. (See elsewhere in this proceedings.)

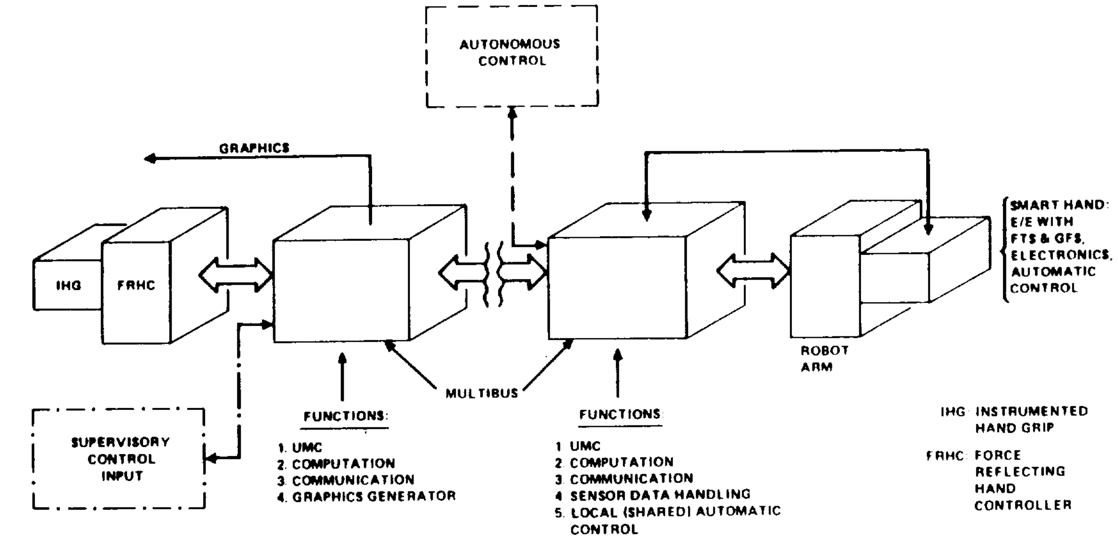

Figure 1. Overall Advanced Teleoperation System with Distributed Computing.

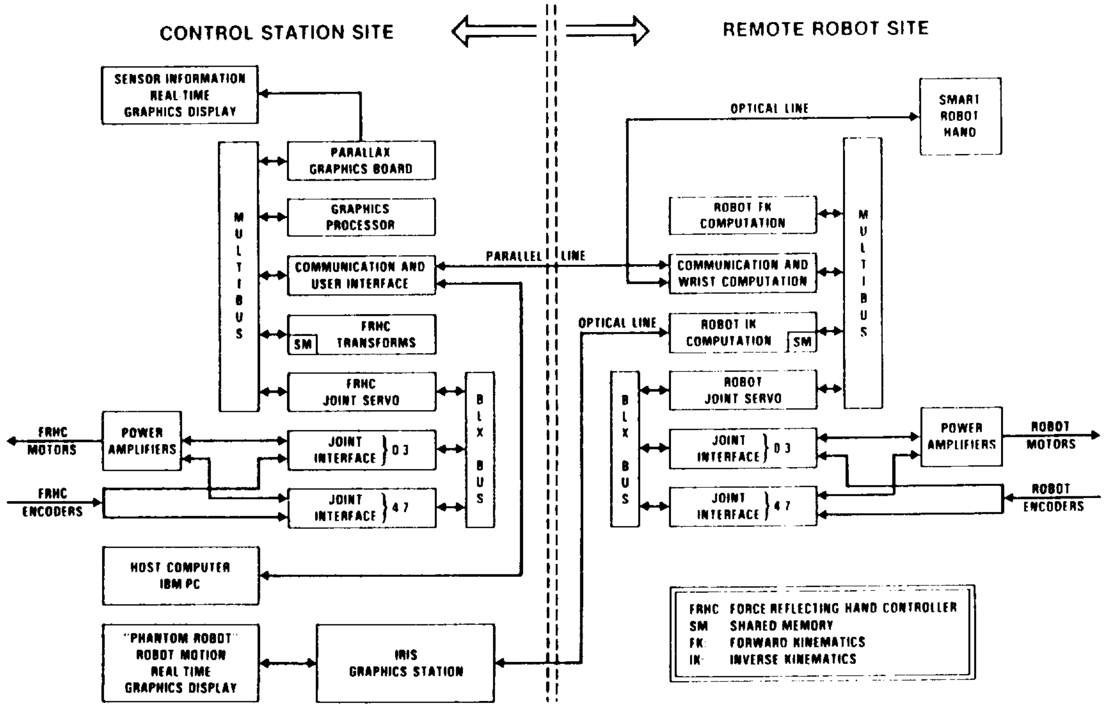

Figure 2. Electronic and Computation System Block Diagram.

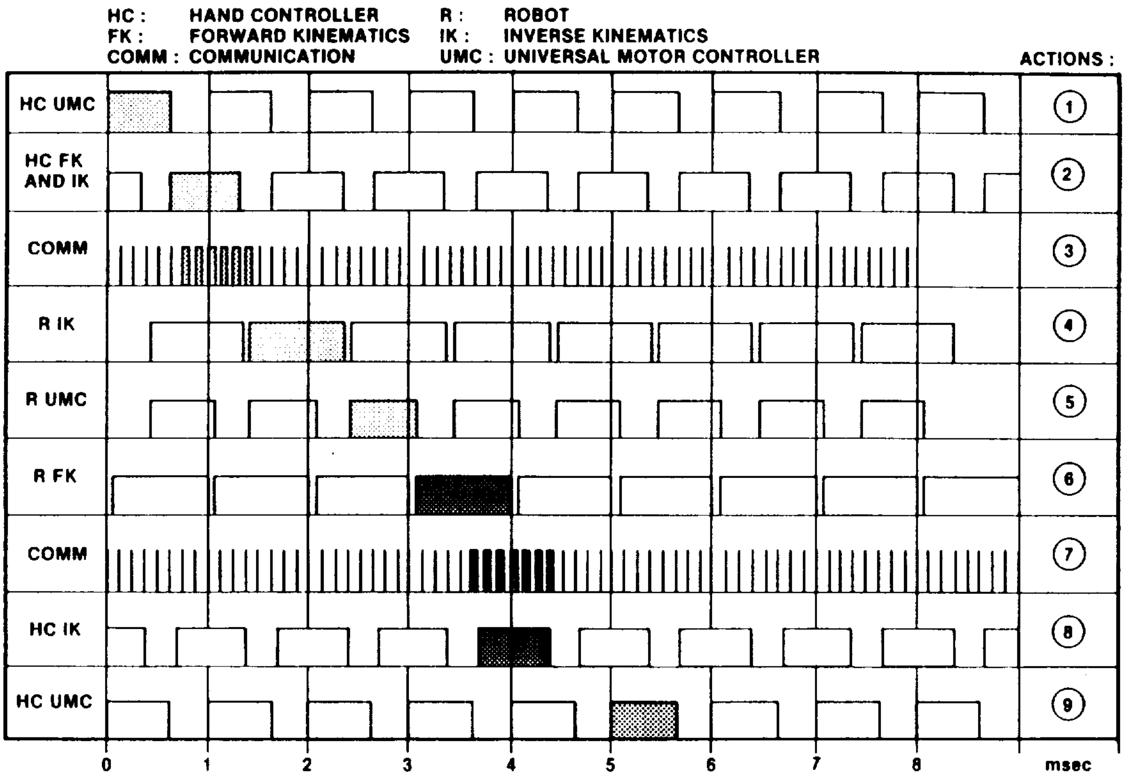

Figure 3. Synchronized Computational Event Sequence Block Diagram.

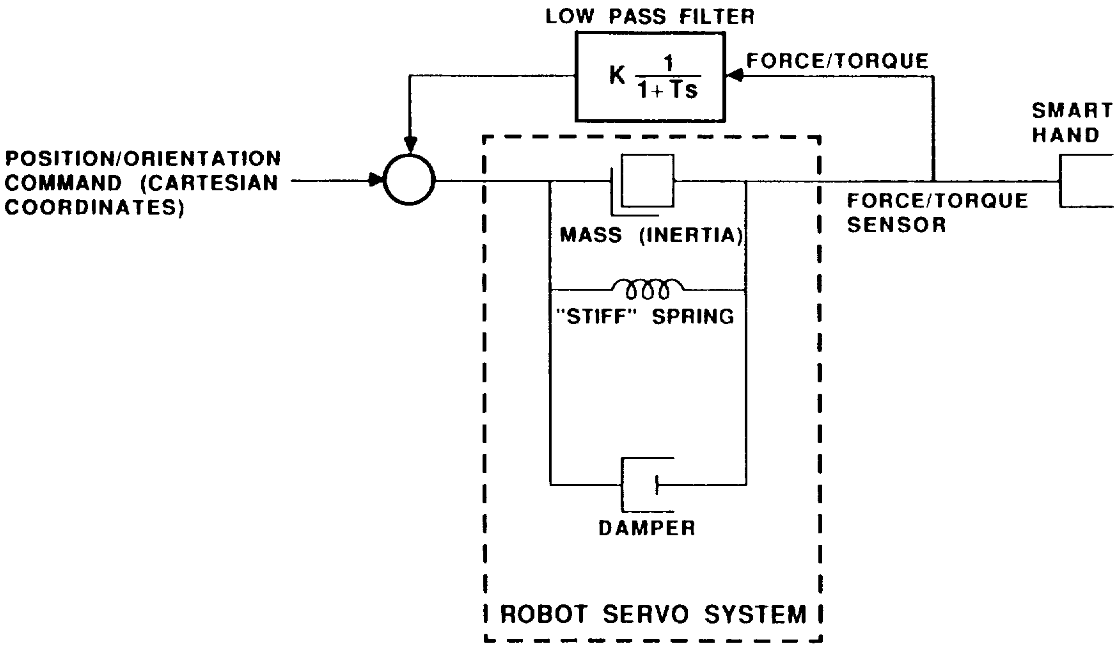

Figure 4. Implementation of Active Compliance for Shared Control.

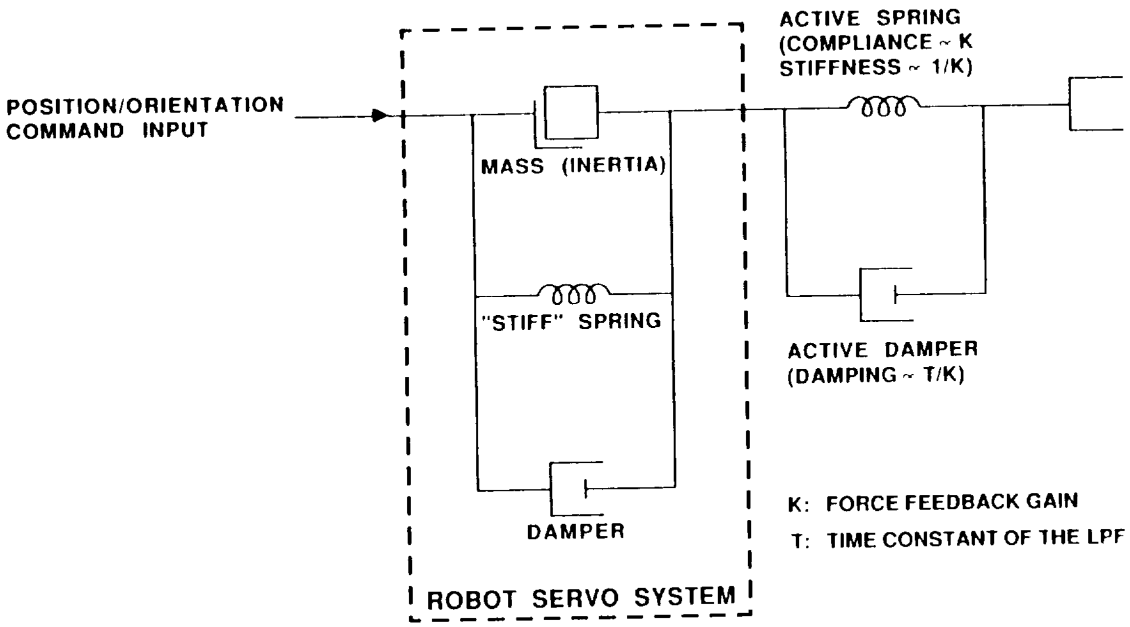
POSITION/ORIENTATION COMMAND INPUT
MASS (INERTIA)
"STIFF" SPRING
DAMPER
ROBOT SERVO SYSTEM
ACTIVE SPRING (COMPLIANCE ~ K STIFFNESS ~ 1/K)
ACTIVE DAMPER (DAMPING ~ T/K)
K: FORCE FEEDBACK GAIN
T: TIME CONSTANT OF THE LPF

Figure 5. Mechanical Equivalent of Active Compliance Implementation.

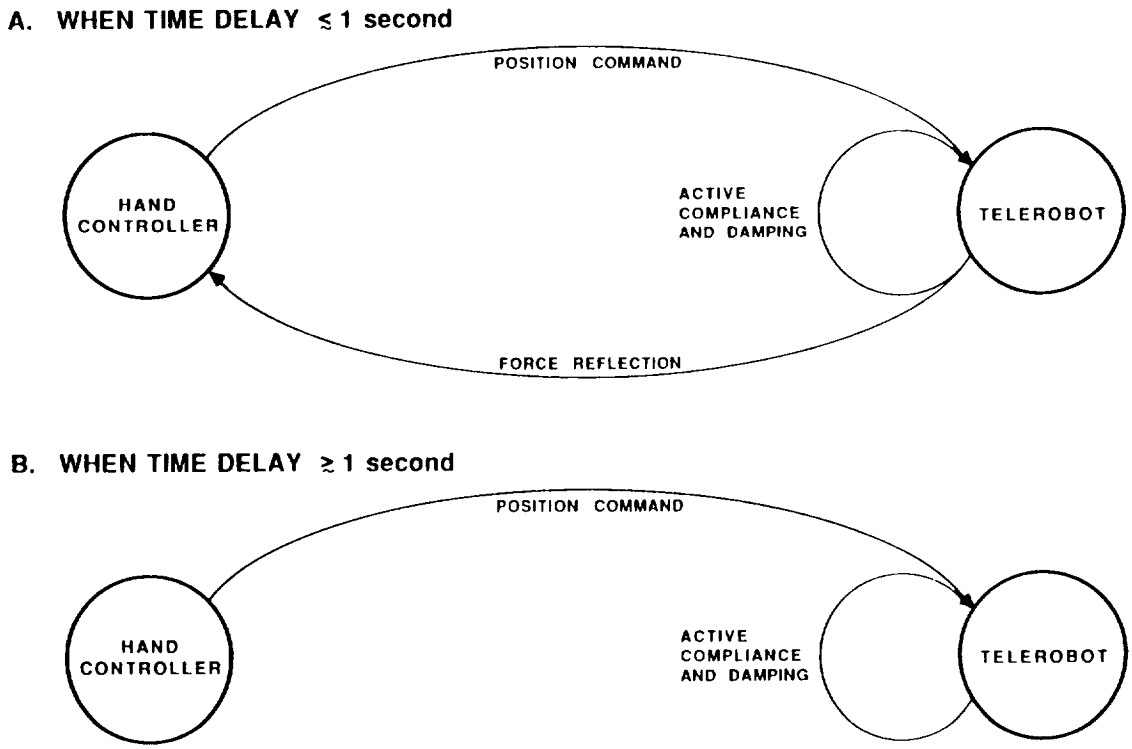
A. WHEN TIME DELAY ≤ 1 second
POSITION COMMAND
HAND CONTROLLER
ACTIVE COMPLIANCE AND DAMPING
TELEROBOT
FORCE REFLECTION
B. WHEN TIME DELAY ≥ 1 second
POSITION COMMAND
HAND CONTROLLER
ACTIVE COMPLIANCE AND DAMPING
TELEROBOT

Figure 6. Control Schemes for Using Active Compliance Control.