

N90-29889

An Alternative Control Structure for Telerobotics

P.T. Boissiere and R.W. Harrigan
Sandia National Laboratories *
Albuquerque, N.M. 87185

Abstract

This paper discusses a new telerobotic control concept which couples human supervisory commands with computer reasoning. The control system is responsive and accomplishes an operator's commands while providing obstacle avoidance and stable controlled interactions with the environment in the presence of communication time delays. This provides a system which not only assists the operator in accomplishing tasks but modifies inappropriate operator commands which can result in safety hazards and/or equipment damage. Research and development of this concept is being carried out in the Telerobotics Research Laboratory at Sandia National Laboratories.

1 Introduction

This paper describes the KRITIC (Knowledge based Review and Intervention To Impose Constraints) controller, a multiprocessor layered control architecture for telerobotic systems. In this concept, a computer local to the robot serves as an intelligent agent to monitor operator commands and to perturb those commands, if needed for safe operation, based upon environmental constraints. This provides automatic obstacle avoidance and controlled interactions with the environment even in the case where significant time delays are present in the communication link between the operator and robot. Such time delays exist, for example, in earth-based control of robots in space. Responsive, yet safe operation is achieved by adjusting the robot's speed to the computational capabilities of the controller. Thus, the robot moves slowly in cluttered environments where extensive obstacle avoidance computations may be necessary and quickly in obstacle-free environments. While operating in an obstacle-free environment, the KRITIC controller monitors inputs from force and proximity sensors in order to respond to obstacles not represented in the world model. Implementation of this control concept in a system containing a PUMA 560 robot is discussed.

This work is motivated by the difficulty most operators experience while controlling a robot using either a teach pendant or joystick. The difficulty arises from several sources. Most robot manipulators can be operated in any of several coordinate frames. The *best* coordinate frame depends strongly on the task to be executed and skill of the operator. Coordinate frames can, in fact, change during execution of a task. In addition, most commercial robot manipulator systems do not provide sensory feedback to the robot operator. Tasks which involve, for example robot contact with the environment can prove to be especially difficult in the absence of force feedback to the operator. Even operations involving visual feedback (especially in the form of a conventional television monitor as in a teleoperated system) can be difficult to execute even for highly trained operators [7].

*Work Supported by the U.S. Department of Energy at Sandia National Laboratories under Contract DE-AC04-76DP00789

Mohr [8] discusses recently initiated research programs to develop the concept of *telepresence* in which sensory feedback to a human operator would be of such quality that the operator would feel as if he or she were co-located with the robot experiencing the same interactions with the environment. Research has been undertaken on the influence of robot environmental and status displays on operator performance [10]. Bejczy and Handlykken [2] discuss the incorporation of force reflection to the operator in a master/slave manipulator system. Of particular concern in space-based teleoperated robotic systems is the impact of communication delays on system control stability when operators controlling the robots are on earth [1]. Much of this work to compensate for time delays has centered around the development of predictive displays which show the operator what the results of a robot command will be before the robot actually executes the command [5]. An additional problem, in the case of a robot in space, is that the communication bandwidth necessary to transmit the sensory feedback to the operator may not be feasible.

The KRITIC control concept integrates computer with operator control. Other approaches to such integrated control of robot manipulators have been referred to as *human supervisory control* by Sheridan [11] and *telerobotics* by others [9]. The term *telerobotics* will be used here. In the telerobotic control architecture discussed in this paper the sensory feedback to the operator can be limited to visual information. Other feedback from the sensory subsystems (*i.e.*, force and torque, robot position, and proximity) is used locally by the KRITIC controller to stabilize the robot system while interacting with the environment, accomplishing object avoidance, and avoiding robot and workspace limits.

Advantages of the KRITIC controller include reduction in the amount of information which must be transmitted to the operator and system stability in the presence of operator/robot communication delays. These attributes are very desirable in environments such as space and underwater exploration where communication bandwidths are limited. The integration of human and computer control described in this work not only aids the operator in the presence of poor sensory feedback and communication delays, but provides a redundancy check on commanded robot motions. This significantly increases robot safety independent of the distance between robot and operator.

2 An Architecture for Telerobotic Control

The basic function of the KRITIC controller is to monitor operator commands, evaluate the impact of those commands with respect to the robot's operating environment, and modify the commands if necessary to achieve the *intention* of the operator in a safe manner. Determining the *intention* of the operator is, of course, a difficult task in general but, with respect to KRITIC, the basic assumption is that the operator is serving in a supervisory position. However, the operator may not know all the constraints which may prevent the completion of a commanded operation. Thus, the operator commands are assumed to be basically valid but may require perturbations to account for environmental and robot constraints.

The KRITIC controller must possess several characteristics. It must provide for smooth transitions from one constraint region to another without direct operator intervention, it must provide for both model-driven control and real-time servo control, and it must achieve the intent of the operator in a natural manner that produces no surprises. Otherwise the operator will not trust the system. The operator's input to the system is evaluated with respect to an approximate world model and sensory information. The KRITIC controller determines what modifications, if any, to the commanded motions are appropriate. These perturbed motions are then communicated to the robot controller. The world model is constructed by combining *a priori* knowledge (usually a mathematical description of the robot and its work environment) with sensory information (*e.g.*, vision) and then

associating this knowledge with constraints defined within the world model. Environmental regions are modeled as *exclusionary* (the robot must not enter a particular region of the environment under any circumstances) or *cautionary* (the robot may enter a particular region of the environment but satisfy predetermined conditions when doing so). Environmental constraints associated with a region typically depend on the task. A cautionary region in one task may become an exclusionary region in another. Nonenvironmental constraints such as robot joint limits are independent of task.

Exclusionary regions include obstacles within the environment which the robot must not contact. The robot must either avoid or stop before reaching an exclusionary region. There should be no way for the robot to enter the region defined as exclusionary. Cautionary regions include objects with which the robot may interact (*e.g.*, touch) but must do so in a controlled manner. For example, a table-top may represent a cautionary region in that the robot may place something down on the table; however, the robot must place the object down on the table *gently*. Thus, associated with cautionary regions are constraints which define how the robot should interact with the environment. As illustrated in the example of placing an object flat on a table top, the constraints are the limits to the forces and torques of interaction. These constraints are implemented by adjusting the forward path gains and setpoints for the control law which allows the robot to comply with the table top as it makes contact. In this case, the robot would be constrained to servo about zero torques and a predefined normal force. Therefore, the constraints on robot motion are zero setpoints for the three torques and, for example, a one pound setpoint for the robot's world Z -axis force and the forward path gains which allow the robot to approach and contact the table top in a stable manner.

Constraint free regions contain no constraints except for those governing the stable control of the robot. Characteristics of constraint free regions of the environment include direct pass through of operator commands to the robot, and a willingness to accept relatively high contact forces should the robot inadvertently collide with an object in the environment. Basically, a constraint free region is one where there is high certainty that an arbitrary operator command will not result in damage.

A layered parallel control structure was developed to achieve the above characteristics [3]. Figure 1 shows the basic structure of the KRITIC controller and its interactions. The basic concept behind the KRITIC controller architecture is that complex system behaviors can be developed through linear combinations of simpler responses to sensory or mapped information. The relative contribution of each control layer to the overall system behavior is determined by adjusting the weighting coefficients (k_i 's in Figure 1) associated with each layer. Thus, as the task varies, the relative weighting of the k_i 's can be adjusted to develop the desired system response.

Perhaps the most important feature of the KRITIC architecture is that the model of the environment and the sensory information pervade the control system at all levels. Since the KRITIC controller is layered there is no single computational bottleneck through which each command must pass. Each layer runs all the time with all control layers individually contributing perturbations to the original operator command. The magnitude of the perturbation varies with time depending upon the robot's perceived location within the environment or on sensory feedback. Each layer of the KRITIC controller can be rather complex and, in fact, layered as well.

This architecture differs from the subsumption architecture of Brooks [4] in which a given layer of control suppresses, or subsumes, the behavior of other control layers. Only one control layer at a time actively influences the system's overall behavior. The KRITIC controller's architecture allows all control layers to influence the overall system behavior at all times. This allows a range of system behaviors to be derived from a limited number of basic response actions.

When an operator command is received by the KRITIC controller it is directed to all control levels simultaneously. The basic idea of the controller is that as soon as an operator's command is received, the *Pass Through* (Figure 1) layer communicates this command to the robot controller. This continues as long as the operator directs the robot to move in a given direction. At the same time, however, the

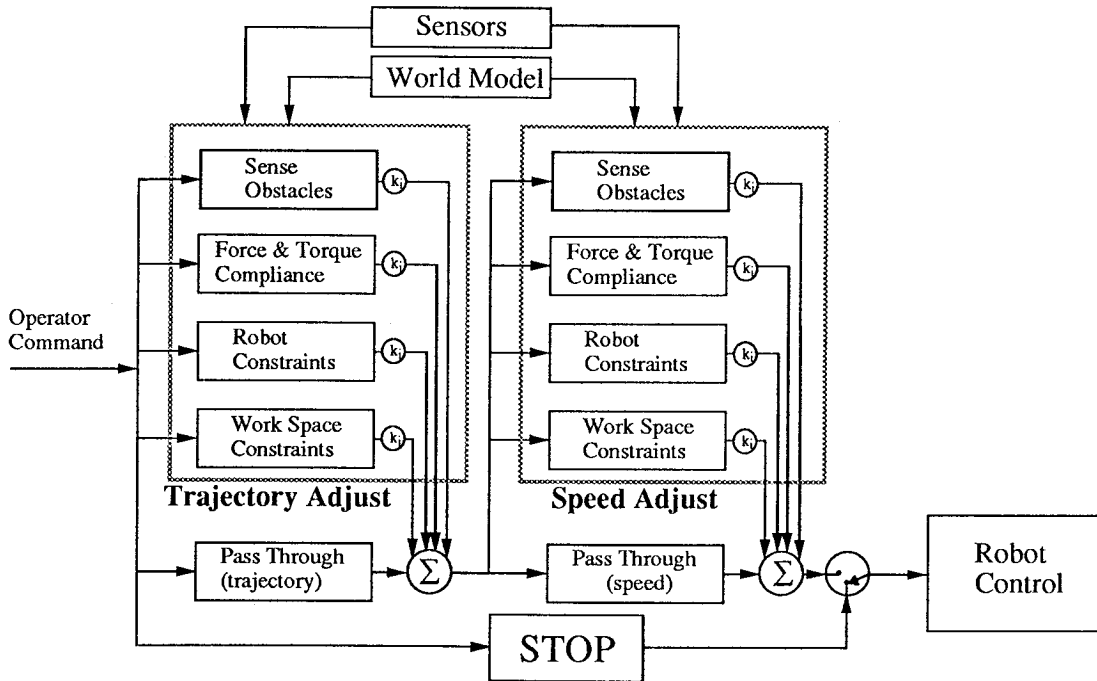


Figure 1: KRITIC Control Architecture

other control layers receive the same command and develop perturbations to the original command based upon inputs from the world model or real-time sensors. These trajectory perturbations are applied to the operator's command (exiting from the *Pass Through* layer) to generate a new robot trajectory. Since the complexity of the control algorithms within the different layers of the KRITIC controller vary from one layer to the next, the outputs from the different layers of the constraint analyzer are not synchronized. Thus, all perturbation computations must be done quickly relative to the speed of the robot. When the robot enters a region where the world model indicates obstacles and thus significant perturbations to the commanded motion are likely, the *Speed Adjust* portion of the constraint analyzer reduces the robot speed. The amount of speed reduction must be balanced to the computational requirements of the constraint analyzer to provide a responsive yet stable control system.

Thus, within the control layers of Figure 1, for example, the *Workspace Constraints* layer checks for the location of known obstacles and limits to the work space within the environment with respect to the robot's location and produces perturbations to the robot's motion to avoid the obstacle or limit. Similarly, the *Robot Constraints* layer monitors the robot's approach to conditions such as joint limits and singularities and provides perturbations to avoid these situations. Sensing of the environment is also provided. Force and torque compliance layers are shown explicitly since this control layer allows interactions with the environment. Other sensing modalities, such as ultrasonic proximity sensing, are included as well, as indicated by the *Sense Obstacles* layer.

As the robot approaches the vicinity of either a sensed or known obstacle (*i.e.*, exclusionary region), the speed of the robot is gradually reduced. The intent is to reduce the robot's speed to allow time for the robot's controller to respond without applying excessive force in the event the robot strikes something. Eventually, if the robot approaches an obstacle closely enough and cannot

go around the obstacle, its speed is reduced to zero.

Thus, the robot's direction is first computed followed by computation of an appropriate speed depending on the presence of obstacles. An additional control layer of the constraint analyzer is the *STOP* layer which, rather than perturb the operator command, overrides it with an emergency stop. The robot should immediately cease all motion when the operator stops commanding the robot to move. In the architecture shown in Figure 1, this is accomplished by activating the *Stop* layer whenever there is no operator input. The *Stop* control layer is unperturbed and communicates directly with the robot controller. As such, *Stop* serves the role of a software emergency stop. Thus, any delayed perturbations passing out of KRITIC which might result in residual robot motion after cessation of an operator command, are intercepted and not passed on to the robot controller.

3 Implementation and Experimentation

As currently configured, the KRITIC controller models the joint limits and singularities of a PUMA-560 robot manipulator (*Robot Constraints*), its work environment (*Workspace Constraints*), which consist of a table top and objects (located by a vision system) on the table top. In addition, force and torque information is provided by a six axis force sensor and proximity sensing is available at the robot gripper in the tool Z-axis direction. An adaptation of the artificial potential field approach discussed by Khatib [6] was implemented to represent physical objects within the environment. In this representation, the robot manipulator is assumed to move in a two dimensional field of repulsive (obstacles) and attractive (goals) forces. A detailed description of this representation and its implementation is given by Boissiere and Harrigan [3]. Figure 2 shows the movement of the robot in the presence of obstacles which results from the KRITIC controller. All deviations from straight line motion were generated by KRITIC as perturbations to the operators command.

Robot joint limits are not modeled as repulsive fields. Instead, the current joint positions of the robot are computed using the inverse kinematics and compared to the stored limits for each joint. If the robot approaches a joint limit, the robot is slowed and stopped before reaching the limit. If the robot approaches a singularity, small joint angle increments are automatically added to the affected joints allowing the robot to pass around the singularity and continue along the path smoothly.

The detection of unexpected objects within a constraint free region results in placement of an exclusionary region within the world model. Once the nature of the new obstacle is determined, the operator may reclassify the region and constraints associated with the object if desired to allow the robot to interact with the object. This may be necessary to identify or map the extent of the object.

During contact with an object input from three different control layers might be used to allow the robot to comply with the environment. The first and most important layer is the *Force & Torque* control layer. For example, if a block in the robot gripper is placed on a table, and the contact surface of the block is not parallel with the surface of the table top, torques will be generated as the block and table make contact. The control algorithm used for force and torque compliance computes the perturbations required and the robot actively complies to establish zero torque. The result is that the block is automatically placed flat on the table with the required contact force. The second control layer to have an effect during this task is the *Workspace Constraints* layer. Here the distance between the robot and the contact surface is computed from the current robot position and information contained within the world model. This distance is used to compute a perturbation which slows the robot as it approaches an obstacle in a cautionary region. The third control layer which might add a perturbation is the *Sensed Obstacle* layer. This layer complements the *Workspace Constraints* layer by providing accurate distance measurements from a proximity sensor. The sum of the perturbations from these three control layers allows the robot to contact objects in a controlled

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

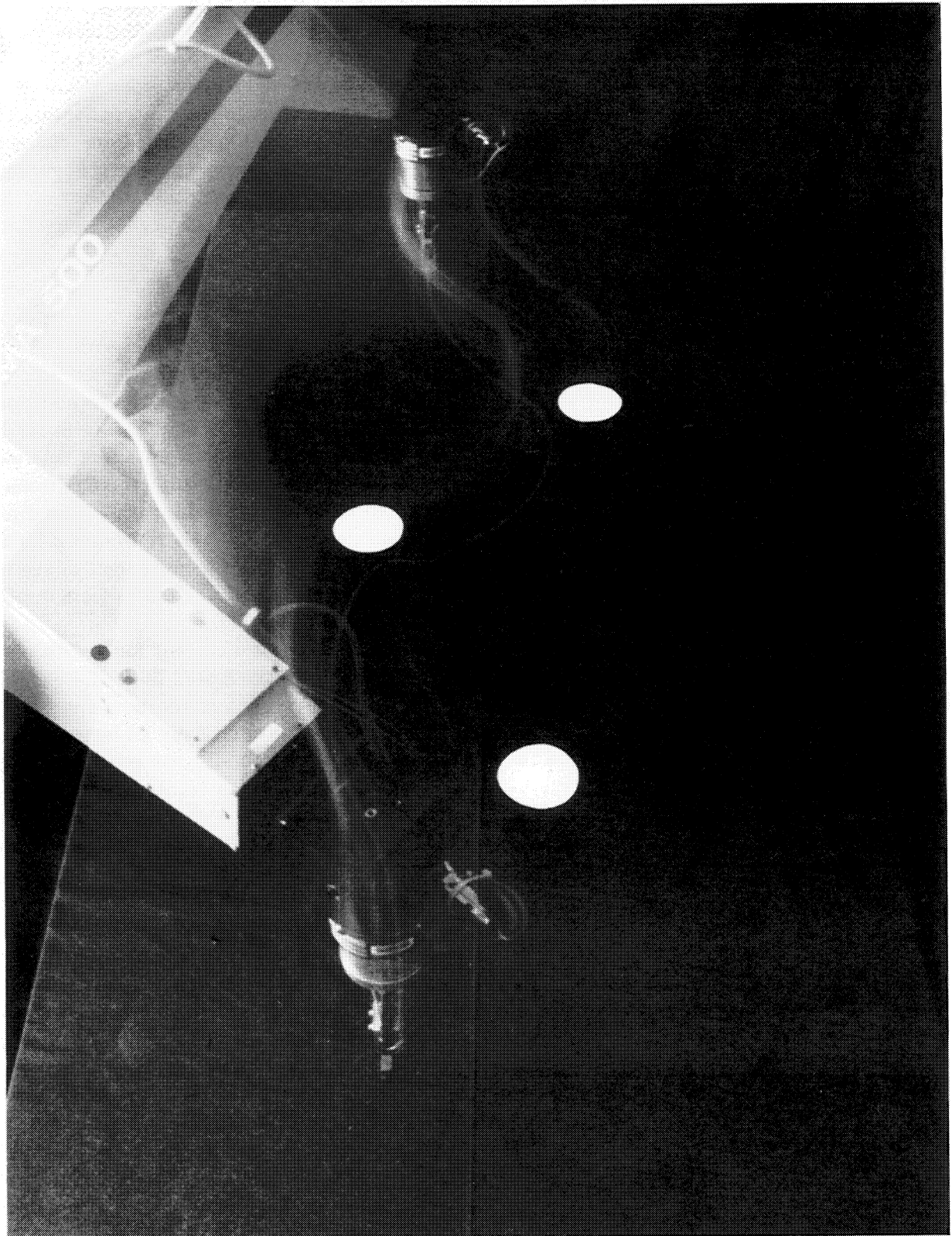


Figure 2: Object Avoidance in Telerobotic Testbed

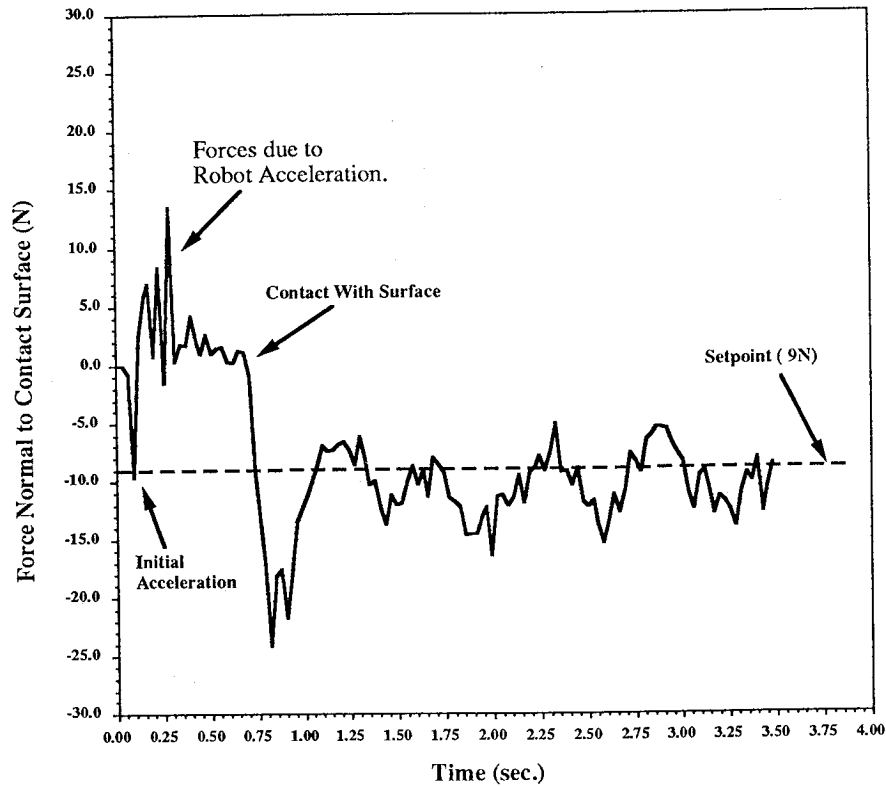


Figure 3: Force History During Contact With Table Top.

and stable manner.

Figure 3 shows the forces generated when a rectangular object held in the robot's gripper is placed on a table. The robot starts 18 inches above the table, places the object flat on the table, and then slides the object along the the surface with approximately 9N of force. The existence of the table top and the constraints associated with placing an object on the table top are obtained from the world model. This information is used to set the controller gains so that the robot can approach and contact the table top. In Figure 3 there is an initial overshoot of 15N during contact. This is generated from the interplay between all the control layers. Each control layer is designed to generate critically damped behavior, and when each layer is used individually there is no force overshoot generated. But when all control layers are in operation some underdamped behavior is exhibited. This can be controlled by manipulation of the weighting coefficients in the KRITIC controller during a specific task.

Without the proximity sensing, the robot's approach to the table top must be considerably slower since the information in the world model is typically only approximate and the exact height of the table is not known. The accurate distance information provided by the proximity sensor allows rapid approach to the table top. As illustrated in Figure 3, the time required to move to the table and establish stable contact is approximately 1.0 second. Note that, during the initial acceleration toward the table, forces are generated due to the mass of the gripper mounted on the force sensor. While under normal circumstances these forces would generate perturbations which would affect the robot's trajectory, information from the proximity sensor can be used to determine whether the sensed forces are forces generated by the acceleration of the robot or actual contact forces.

In other telerobotic control architectures delays in communication to and from manually controlled remote robot manipulators can cause instability in control. Anderson and Spong [1] have investigated

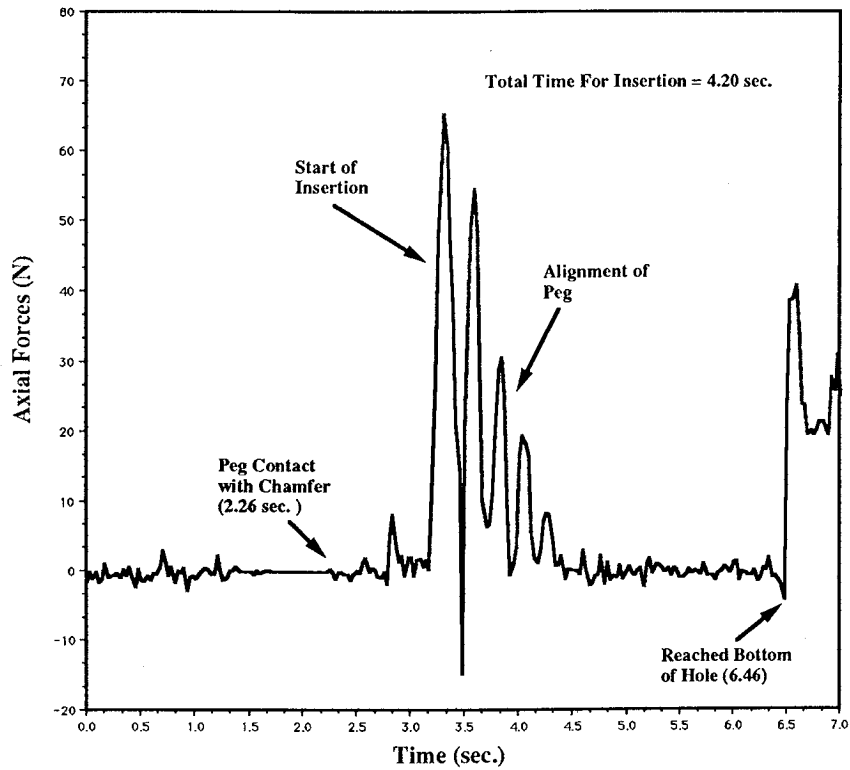


Figure 4: Peg Insertion Five Second Time Delay

the impact of time delays on control stability in manipulators in which forces generated by the robot's interaction with its environment are electronically reflected to the operator's joystick. In the KRITIC controller, communication delays do not affect the stability of the robot system since servo control of the robot is handled locally by KRITIC. In fact, time delays have the effect of placing the operator in a more supervisory role. Once the command from the operator arrives at the KRITIC controller, the control layers generate the appropriate perturbations to insure safe operation. The only effect of communication time delay is the introduction of an equivalent time delay to the execution of the next operator's command. No instabilities have been introduced into the system because stability of the system is handled locally by the KRITIC controller.

Figure 4 shows the force history for the complex telerobotic task of inserting a round peg in a hole with 0.5 mm clearance and a five second communication delay between operator and KRITIC. The operator approximately aligned the peg with the hole and directed the robot to move in the negative world Z-axis direction. All forces and torques are computed with respect to the tip of the peg. As the peg is inserted into the hole the forces and torques generated due to contact are used to satisfy the constraints for this task. The constraints associated with a successful peg insertion include minimizing the torques generated about the tip of the peg [12]. Note that, at the point of initial contact the axial force does not sharply increase. However, forces in the plane of the hole are generated as the peg encounters the chamfer at the opening of the hole. The KRITIC controller responds to these forces by moving the peg in the direction of the center of the hole. As the peg is centered with the opening of the hole the operators command to move in the negative Z-axis direction drives the peg into the hole. This causes the peg to bind and axial forces start to build as shown. At this point KRITIC uses the torques generated to align the center line of the peg with that of the hole. Once this has been accomplished the axial force decreases and the insertion task continues

until the bottom of the hole is encountered or the operator stops the command. If the bottom of the hole is encountered while the operator is still executing the command to move in the negative Z -axis direction the KRITIC will stop the robot at the bottom of the hole. Using the KRITIC controller an operator can accomplish the peg insertion task in 4–5 seconds after initial contact is made. It was found that the time required to insert a peg from the same starting position and orientation was independent of the amount of communication time delay. As expected, the communication time delay had no effect on the performance of the KRITIC controller.

Notice that rather sophisticated behaviors are generated through the linear combination of relatively simple primitive behaviors. In the case of inserting a round peg into a hole, the *Force Compliance* layer of the constraint analyzer slows movement in the vertical direction to prevent jamming while the *Torque Compliance* layer simultaneously generates robot motions to eliminate any impressed torques on the peg. Finally, the *Pass Through* layer continuously tries to move the peg downward in response to operator inputs. The combined effect of the three primitive behaviors is rapid successful insertion of the peg into the hole under operator control.

4 Conclusions and Future Work

The KRITIC control architecture offers a feasible approach to enhancing the flexibility of robot manipulators while providing operator interfaces which reduce the impact of operator error and communication time delays. As a result, teleoperated robot manipulators incorporating a KRITIC controller can accomplish many tasks (especially those requiring interactions with the environment) faster and more reliably than standard master/slave manipulators. The use of a perturbation-based layered control concept provides an environment for real-time model based trajectory and speed control of the robot manipulator. Complex behaviors are generated by combining simple responses of the robot to sensory and model-based information. This concept naturally leads to modular control constructs and allows smooth transitions from one control mode to the next.

The specific implementation of the layered telerobotic control concept discussed in this paper will be expanded in the future. Initially the KRITIC control architecture was implemented on a PDP-11/73 [3]. Currently the KRITIC architecture is being implemented on a VME-based multiprocessor environment. The use of a VME-based computing environment will facilitate experimentation into the distribution of the KRITIC over many individual processors. The increase in computing capabilities will benefit the KRITIC in several ways. For example, as the implementation of this architecture is moved into more complex environments the World Model will also increase in complexity. In order to handle the increase in computational requirements the control layers which use the world model may themselves be distributed over several processors to maintain a responsive system.

Work on prototype telerobotic systems for robots other than the PUMA-560 is also underway. In particular, a telerobotic control structure for a CIMCORP XR6100 gantry robot is being developed for use in a project to remotely inspect and manipulate spent nuclear fuel shipping casks and other nuclear waste containers. Since this is a three dimensional environment, the repulsive potential fields used in this work will be extended and alternative approaches to object modeling will be investigated. The modular nature of the KRITIC architecture allows easy application and even mixing of various concepts. An important application area for the KRITIC control concept is hazardous environments. Telerobotic control which decreases handling times while minimizing the impact of operator error is very important. KRITIC can not only be used to constrain the robot's interaction with the environment but to enforce procedures to ensure proper sequencing of operations as well.

Acknowledgments

The authors would like to thank the following contributors for making the implementation of the KRITIC a reality: William M. Davidson for developing many of the interrupt service routines and overall software architecture, William D. Drotning and Pablo Garcia for hardware and software support for the proximity sensor.

References

- [1] Anderson, Robert J. and Mark W. Spong, "Bilateral Control of Teleoperators with Time Delay", Proceedings of 1988 IEEE Conference on Decision and Control, Austin, TX, 1988.
- [2] Bejczy, A. K. and M. Handlykken, "Generalization of Bilateral Force-Reflecting Control of Manipulators", Proceedings of the 4th Ro-Man-Sy, Warsaw, 1981, pp. 242-255.
- [3] Boissiere, P.T. and R.W. Harrigan, "Telerobotic Control of Conventional Robot Manipulators", Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, 1988.
- [4] Brooks R.A., "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, vol. RA-2, no. 1, pp. 14-23, 1986.
- [5] Conway, Lynn, Richard Valz, and Michel Walker, "Tele-Autonomous Systems: Methods and Architecture for Intermingling Autonomous and Telerobotic Technology", Proceedings of the 1987 IEEE Conference on Robotics and Automation, San Francisco, Ca., 1987
- [6] Khatib O., "Real-Time Obstacle Avoidance for Manipulators and and Mobile Robots", International Journal of Robotics Research, vol. 2, no. 1, pp. 90-98, 1986.
- [7] McGovern, Douglas E., "Human Interfaces in Remote Driving", SAND88-0562, Sandia National Laboratories, Albuquerque, New Mexico, March 1988.
- [8] Mohr G.C., "Robotic Telepresence", 1987 IEEE Proceedings: Annual Reliability and Maintainability Symposium, January 1987, Philadelphia, pp. 25-30.
- [9] Pennington J.E., "Space Telerobotics: A Few More Hurdles", Proceedings of the 1986 IEEE International Conference on Robotics and Automation, vol. 2, pp. 813-816, 1986.
- [10] Stark L., *et al*, "Telerobotics: Display, Control and Communication Problems", IEEE Journal of Robotics and Automation, vol. 3, no. 1, February 1987, pp. 67-75.
- [11] Sheridan T.B., "Human Supervisory Control of Robot Systems", Proceedings of the 1986 IEEE International Conference on Robotics and Automation, vol. 2, pp. 808-812, 1986.
- [12] Whitney, D.E., "Quasi-static Assembly of Compliantly Supported Rigid Parts", Journal of Dynamic Systems, Measurement and Control, vol. 104, no. 1, March 1982.