Teleprogramming: Remote Site Research Issues

MS-CIS-92-01
GRASP LAB 298

Thomas Lindsay

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

January 1992

Teleprogramming: Remote Site Research Issues

Dissertation Proposal[†]

Thomas Lindsay

Abstract

This document proposes the development of the remote site workcell for teleoperation with significant communication delays (on the order of one to 20 seconds). In these situations, direct teleoperation becomes difficult to impossible due to the delays in visual and force feedback. Teleprogramming has been developed in order to overcome this problem. In teleprogramming, the human operator interacts in real time with a graphical model of the remote site, which provides for real time visual and force feedback. The master system automatically generates symbolic commands based on the motions of the master arm and the manipulator/model interactions, given predefined criteria of what types of motions are to be expected. These commands are then sent via a communication link, which may delay the signals, to the remote site. Based upon a remote world model, predefined and possibly refined as more information is obtained, the slave carries out commanded operations in the remote world and decides whether each step has been executed correctly.

The remote site receives commands sent via the delayed communication link. These commands must be parsed and translated into the local robot control language, which includes insertion of dynamic parameters that are not generated by the master system. The commands are then executed by the hybrid position/force controller, and the resulting motions monitored for errors.

This proposal addresses the following remote site issues: low level manipulator control using an instrumented compliant wrist for sensory feedback, higher level command execution implementing dynamic parameters, and remote manipulator tool usage and control.

1 Introduction

The first automatic electric-powered teleoperator was developed in the 1940s in order to manipulate radioactive material [8]. Teleoperators are still used today for tasks in hazardous environments. For certain environments, such as shallow space and subsea applications using an unmanned, untethered submersible, significant communication delays occur between the master and slave sites.

These communication delays make real-time feedback from the remote site difficult or impossible to use. One solution to this problem is *teleprogramming*, which will be described herein, and in [7, 16].

[†] This material is based upon work supported by the National Science Foundation under Grant No. BCS-89-01352, "Model-Based Teleoperation in the Presence of Delay." Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1.1 Problem Statement

The remote site for the teleprogramming system evokes some pertinent research issues. The first is the development of a hybrid control scheme that can act semi-autonomously and interact with an unknown or partially known environment. The controller will use an instrumented compliant wrist for sensory input. The second research issue is command execution with the inclusion of dynamic parameters and robust collision detection. A final issue is manipulator tool usage.

The major research goal is to develop a working remote site system for teleprogramming, and to study the behavior of this system, which may present improvements to both the remote system and the local master system.

1.2 Outline of Proposal

The remainder of this proposal is organized as follows. Section 2 reviews the current state of research in telerobotics and time-delayed teleoperation. Section 3 provides an overview of teleprogramming and our experimental setup. Section 4 describes the instrumented compliant wrist, and how it relates to the robot control. Command execution including command parsing is studied in section 5. Tool usage by the remote manipulator is discussed in section 6. Finally, conclusions and contributions are discussed in section 7.

2 Related Work

The following section presents a brief review of related work in the fields addressed by this paper: teleoperation with time delays, low level hybrid control, higher level command execution, and tool usage.

2.1 Teleoperation with Time Delays

Time delayed teleoperation has been a large research issue for many years. Ferrell notes that given a time-delayed teleoperation task with only visual feedback, a human operator tends to adopt a *move-and-wait* strategy [5]. For a given size of task and time delay, this may not be practical. Because of this, Ferrell and Sheridan formalized supervisory control [6], a broad strategy for balancing the workload between the human operator and a semi-autonomous remote system. Supervisory control addresses most of the issues involved with time-delayed teleoperation. However, as formalized by Ferrell and Sheridan it has some drawbacks. First, there is still the need for the human operator to acknowledge the completion of a command. As stated, "In the region of combined man-and-machine control, the operator is able to extend his open-loop 'moves' so that he gives fewer but more comprehensive commands. With fewer commands there are correspondingly fewer waits for correct feedback." Second, the example of implementation uses a command language that depends on branching. Using branching for error recovery creates a large and at times error-prone programming task for the human operator. Finally, the intercommunication needed to recover from an error need not be as intensive as Ferrell and Sheridan imply.

Niemeyer and Slotine have shown, using a two-ported model, that direct force feedback is possible and stable with delayed feedback loops [12]. However, it is noted that performance still degrades as the time delay increases. Furthermore, the human operator and the environment are not included in the formulation. In order for the system to work in a manner that the human operator can understand, prediction algorithms are needed.

Prediction algorithms are used extensively by Hirzinger [9]. However, the remote site and the delay lines must be accurately modeled. This approach does not appear suitable for unknown, unstructured environments.

Predictive displays have been used in many delayed and non-delayed teleoperation projects. These displays have in all cases shown to improve the performance of the system. Hirzinger displays the complete predicted world for the operator to work on [9]. ETL overlays cad models on a view of the slave environment to enhance the model [13]. For free-space motion, JPL uses a "ghost manipulator" overlayed on a view of the actual remote manipulator, to show its predicted position to the operator's inputs [3].

Recently Paul, Funda, et. al. [7, 16] have developed the teleoperation system outlined in section 3. The research presented in this paper is directly associated with this system.

2.2 Low Level Hybrid Control

Most manipulation tasks require the implementation of a hybrid position/force control scheme.

Whitney [22] showed the usefulness of RCC devices for peg-insertion tasks. The RCC device passively causes the peg to correctly compensate for forces that are caused by misalignment in the insertion process. However, the RCC device is not ideal for generalized tasks [26, 10], and positional accuracy is lost with a passive compliant device [20]. Xu developed an all-purpose 6 DOF instrumented compliant wrist to overcome these problems [24]. Three main benefits arise from using an instrumented compliant wrist. The first is the ability to track surfaces, allowing the task frame to become dynamic with the end effector trajectory [19]. Second, using the wrist as a force/torque sensor allows for more responsive force control [18, 21]. Finally, the transition from unconstrained to constrained motions are facilitated, and impact energy is absorbed [24, 20].

Hybrid control is formalized by Raibert and Craig [17], and has roots in compliance control [15, 11]. Natural and artificial constraints of a task dictate orthogonal axes that should be either position controlled or force controlled. Xu adapted the hybrid control scheme for controlling a manipulator with the instrumented compliant wrist as the force/torque sensor. Position feedback is recovered directly from the wrist. Force feedback is obtained implicitly from the stiffness matrix of the wrist.

Low level hybrid control using the instrumented compliant wrist is further generalized in this research by allowing control task frames to be arbitrarily defined. The effect of defining a task frame remote from the wrist sensor is examined.

2.3 Higher Level Command Execution

The concept of the guarded move was evident in Ferrell and Sheridan's supervisory control: "At the primitive level, the language is constructed around a basic action: a movement in a given direction terminated on the achievement of specified sensor states, and/or a given distance moved." [6], and this is implemented in the MANTRAN language [2]. Peter Will coined the phrase "guarded move" as "a move until some expected sensory event occurs" [23]. It is important to note that unexpected sensory states must also be "expected". In both these cases, the occurance of a sensory state leads to branch statements. Because of the impossibility of recognizing every sensor state and creating a branch for each, and the fact that each branch may also rely on a guarded move, the idea of branching has been eliminated in the research proposed here. Instead, the guarded move needs to be robust. If a guarded move succeeds (desired sensor states are reached), a programmed post motion command sequence is executed. If a guarded move fails (undesired sensor states are reached), the robot stops all motion and sends information about the error state back to the master.

The command execution stage also requires some adaptation to the remote environment. As Ferrell and Sheridan state, "Simply because predictions often are not borne out and environments change, such feedback as there is [at the remote site] must be able to modify the internal representation of the environment. In at least this elementary sense, the system must be able to learn from experience" [6]. This is applicable to the research proposed here, in that the master system does not have any dynamic or real-world information about the remote site. The remote model of surface properties, for example, must be modified as information about the surface is gathered.

2.4 Tool Usage

Bolles and Paul used an electric screwdriver in their programmable assembly task [4]. Using the WAVE language [14], the tool could be controlled by forces monitored by the manipulator. In a sense, the tool was controlled by a guarded move.

Although many industrial robots use tools, most are position controlled. The tool is treated as a separate device, and the only use of the manipulator is to move the tool into a required position. One notable exception is with deburring robots, where the robot needs a sense of force from the deburring task in order to determine the feed rate for the robot motion [1].

In this research, a more generalized approach will be presented. Tools can be treated as extra degrees of freedom in a robot-tool system. If the natural degree of freedom for the tool is treated as one of the system degrees of freedom, the system uses the tool to control that degree of freedom naturally, and guarded moves can be applied to that degree of freedom to control the tool.

3 Teleoperation With Time Delays: Teleprogramming

Direct teleoperation becomes difficult to impossible when significant time delays occur in position and force feedback. As noted in section 2, many solutions have been presented to overcome this problem, and these solutions each have their limitations. In this section, the teleprogramming paradigm is presented, and the current teleprogramming implementation is presented.

3.1 Overview

The teleprogramming concept bypasses the problems associated with delayed feedback from the remote site. Control loops are closed locally for both locations; the human operator receives direct visual and kinesthetic feedback from a local model of the remote site, and the remote manipulator autonomously compensates for inaccuracies in position and force. Based on the input from the human operator, the master system automatically generates a set of commands that are sent via the delayed communication link to the remote site. The remote site executes these commands, and acknowledges successful execution or reports errors back to the master. Error recovery is left to the human operator.

Using teleprogramming, teleoperation becomes delay-invariant: increases in the communication delay do not affect task performance.

3.2 Types of Operation

The subsea environment is the main focus of application for the teleprogramming system. Manned submersibles are used for operations that are too deep for divers. They are expensive to maintain and operate, and because the humans on board must stay inside the submersible, there is little advantage to using them. Unmanned, tethered submersibles have been used extensively for subsea operations at various depths. They have great advantages in extending our subsea abilities. Disadvantage to a tethered submersible include the cost of maintaining the surface support for an operation, dependence on fair weather for operations, and problems associated with operating two or more tethered submersibles on the same location at the same time. An unmanned, untethered submersible vehicle can communicate to the surface via an acoustic link. This will introduce significant communication delays between the master and remote sites. The submersible can be used for exploration, retrieval, repair, or any other subsea task.

The other main application for teleprogramming is shallow space. The cost of human activities, especially EVA, is extremely high. However, delays in communication between earth and shallow space can easily be as much as 18 seconds, considering earth and satellite based relay stations. Teleprogramming systems in space could be used for satellite repair and maintenance, greatly reducing the amount of human intervention time in space.

3.3 Specifications of the Remote System

For this research, specifications have been imposed on the remote system. First, because the master site model will be inaccurate, the remote manipulator must be able to perform adequately while working with distance parameters that may differ from actual values by a tolerance ϵ. Second, the dynamic parameters of the remote environment are not explicitly given to the remote system. Third, the remote system is constrained to follow instructions at the same rate as the master. Thus, a move that takes the master 5 seconds to complete should also be completed by the slave in 5 seconds. The basic system at the remote site has no extra time for exploratory procedures.

3.4 Experimental Setup

Any teleoperation system can be visualized as a string of four elements: the human operator, the master system, the remote system, and the environment. The master and slave systems are connected by a communication link. The implementation of these elements and the interaction between these elements in the teleprogramming system are shown in figure 1 and discussed below.

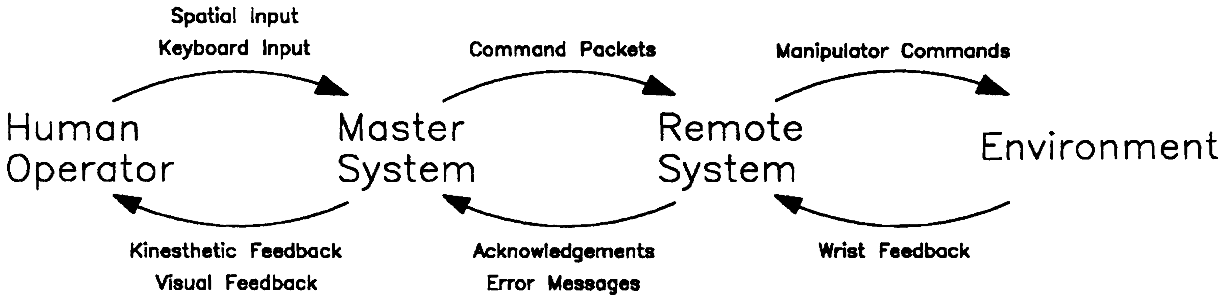

Figure 1: Basic Teleprogramming Schematic

3.4.1 Human Operator

The human operator is the main decision making element in the teleprogramming system. The operator is responsible for path planning and error recovery. The operator uses a Puma 250 manipulator as a spatial positioning device. The Puma also provides the operator with kinesthetic feedback generated from the master model, as explained below. Visual feedback is provided by a Silicon Graphics (SGI; Mountain View, CA) Iris workstation.

3.4.2 Master System

The main tasks of the master system are to convert inputs from the human operator into commands that can be sent to the remote site, and to feed back information to the operator to ease the input task.

The master system supplies real time kinesthetic and visual feedback based upon the operator's interaction with a graphical model of the remote site. The model is built from remote site data as the first step in a teleprogramming session.

Based on the input from the human operator and *a priori* information about the task, the master system automatically generates commands that are sent to the remote site. These commands are generated once every second unless a specific event is detected, such as a collision. In this case, a command is immediately generated.

When errors occur at the remote site, a message is sent back to the master system, and the graphics model is reset to correspond to the remote site error state.

3.4.3 Communication Link

The communication system links the master and slave systems together. For subsea applications, using an unmanned, untethered submersible communicating via an acoustic link, this communication system may introduce large delays between the master and slave sites. Earth-based teleoperation in shallow space is also subject to significant time delays.

Communication is limited to command packets, called execution environments, sent from the master to the slave at least once every second, and an acknowledgment or an error message sent back to the master at the completion of every command. There is a programmable communication time delay in the current implementation.

3.4.4 Remote System

The main tasks of the slave system are the parsing of commands, including environment-specific parameters, control of the remote site manipulator, and monitoring the remote process, which includes error detection, reporting to the master system, and error recovery.

The remote manipulator, a Puma 560 robot, uses a hybrid position/force control algorithm. Position and force feedback is supplied by an instrumented compliant wrist.

3.4.5 Environment

Initially, the remote task environment is assumed to be unknown. For many tasks, there may be knowledge about objects in the environment, including blueprints, CAD models, etc. However, the subsea environment is very dynamic, and may impart changes upon the known environment, such as corrosion, silt, and marine growth. Therefore, the environment is usually unknown, unstructured, and possibly hostile. One of the major benefits of the teleprogramming system is the ability to work with unknown and unstructured environments.

3.5 Summary

This section has described the teleprogramming paradigm and the physical structure of the teleprogramming system. In the next three sections, the remote system will be described in greater detail, starting with the servo-level hybrid control of the remote manipulator, continuing with the higher level command execution, and concluding with control and use of tools in the remote world.

4 Robot Control Using an Instrumented Compliant Wrist

The remote manipulator uses a low level PD controller based on a hybrid position/force algorithm.

The slave manipulator is fitted with an instrumented compliant wrist (see figure 2). Compliance in a robot wrist is desire to reduce the effect of impacts between the robot and the environment, and to create a more responsive force control. However, a compliant wrist by itself has a limits the effective stiffness of the manipulator in position control, and the exact position of the end of the wrist (and thus the environment, when in contact) is lost [20]. By instrumenting the wrist, though, both these problems are overcome. Active control can increase the stiffness of the system, and the position transform of the wrist is known. Using the instrumented wrist as a compliant force/torque sensor leads to more responsive force control than with a stiff sensor [18], and more accurate position control than with a compliant wrist [25].

4.1 Hardware

The two components of the wrist are the compliant structure, and the sensing linkage. The compliant structure is composed of rubber elements with known stiffness, connecting the upper and lower plates of the wrist. The sensing linkage is composed of six links, with

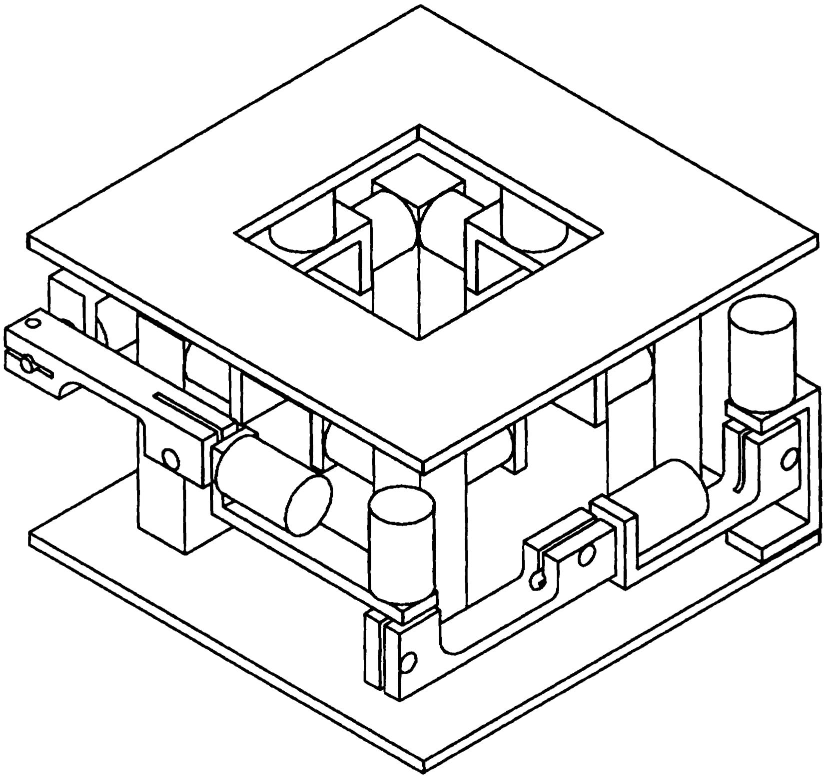

Figure 2: Instrumented Compliant Wrist

potentiometers at their connections. By reading the change in voltage across the potentiometers, the angle of each joint can be determined. Using a simple forward kinematic formulation, the relative position of the top plate with respect to the bottom plate can be found.

4.2 Hybrid Control

The remote manipulator uses a hybrid force/position controller. Displacements from the wrist sensor are used directly for position feedback. Force feedback is determined from the wrist stiffness matrix.

Any cartesian frame can be partitioned into force and position directions for the hybrid control. Two obvious choices for the frame are the tool coordinate frame (T6), and a world coordinate frame with origin at the end of the manipulator. However, control is not limited to these two frames. Any frame relative to the T6 frame or the world coordinate frame can be used. It is important to note, though, that the wrist sensor is located at the end of the robot. Control about a frame with an origin that is far from the end of the robot can become unstable due to the amplification of sensor noise. Also, small motions in the task frame are transformed into large motions at the end of the manipulator. The combination of sensor noise amplification and large motions of the robot may present a natural limit to the distance that the task frame can be located from the end of the manipulator.

4.3 Partial Results

Currently, an instrumented compliant wrist has been built and tested on several research projects. The hybrid control algorithm has been tested. At this point in time, the controller appears to be stable in all manipulator configurations. The control frame can be any frame defined relative to the end of the manipulator. Frames defined relative to the world coordinates require extra computation, which was beyond the capability of the current controlling computer (a MicroVax II) using a 28ms servo interrupt rate. However, the manipulator will soon be controlled by a Sun Microsystems Sparc Station 2, which should be able to handle the computational load. Other improvements will also be possible. Preliminary studies of task frames remote from the end of the manipulator agree with theory, in that as the task frame is moved away from the wrist point, the performance degrades and finally becomes unstable.

5 Command Execution

The language parser and interpreter convert commands generated by the master system into instructions which the slave manipulator can execute.

5.1 Command Types

The current command set is presented in table 5.1. The commands are sorted into four categories. The task frame management commands allow arbitrary frames to be defined for use as the hybrid control frame. The force control commands are used for assigning the force/position directions for the hybrid control, to assign actual forces that the manipulator is to maintain, and to set the guards for guarded moves. These guards can be either force guards, where the manipulator moves until it senses the given force, or velocity guards, where the manipulator stops if velocity limits are violated. Above these guards are maximum force and velocity limits that the manipulator cannot violate.

Motion commands are used to move the manipulator around. When used in combination with a force or velocity guard, the motion commands imply also a tolerance limit. If the guard is violated before the tolerance is reached, an error message is generated. Further, if a tolerance is passed without violation of the guard limits, an error message is generated.

Finally, the special commands are used to control tools. The UseTool command identifies the tool being used, which may change the function of Grasp and Release. Further, the use of certain tools may imply that one or more degree of freedom of the manipulator/tool system is to be controlled by the tool. This will be discussed in section 6.

5.2 Execution Model Steps

Each task can be broken down into infinitely smaller sub-tasks. For this work, the first level of sub-tasks, such as removing a bolt or opening a hatch cover, are defined as execution models. Each execution model, in turn, is broken up into execution model steps, which are small motion commands. These steps are composed of free-space motion, motion to contact a surface, motion in contact with a surface, etc. The execution model steps are generated as packets at the master site, and executed autonomously at the remote

Task Frame Management
DefineVector(*name;* $< v_x, v_y, v_z >$ *: ref-frame*)
DefineTaskFrame(*name : ref-frame; origin; x-axis; y-axis; z-axis*)
UseFrame(*frame*)
Force Control
AssignMode(X,X,X,X,X,X) where X $\in$ (F,P)
Force($< v_x, v_y, v_z >; < \tau_x, \tau_y, \tau_z >$)
GuardForce($< v_x, v_y, v_z >; < \tau_x, \tau_y, \tau_z >$)
GuardVelocity($< v_x, v_y, v_z >; < \omega_x, \omega_y, \omega_z >$)
Motion Commands
Move(t ; $< p_x, p_y, p_z >; < \phi_x, \phi_y, \phi_z >$)
Slide(t ; $< p_x, p_y, p_z >$)
Pivot(t ; $< \phi_x, \phi_y, \phi_z >$)
Special Commands
UseTool(*toolname*)
Grasp()
Release()

Table 1: The Teleprogramming Command Language

site. Execution model steps are also known as execution environments, as they contain information about the working task frame, and the contact states of the manipulator. Furthermore, they contain information about the hybrid modes and forces to be applied after a motion is successfully completed.

5.3 Double Buffering

In order to eliminate pauses between execution of commands, it is important that the information needed for a step has been completely parsed before the previous step has completed execution. Therefore, as the manipulator moves through one execution model environment, the next environment is being parsed. It is imperative that the parsing of the next step is completed before the previous step has finished.

5.3.1 Task Frames

Motion commands outlined in section 5.1 define movement with respect to the current task frame TF. TF can be defined as a static frame (referenced to the kinematic base frame KB), or as a dynamic frame (moving with and defined relative to the end effector frame EE).

Because the next execution environment often relies upon the final position of the previous step, it is important to predict the EE frame at the end of a motion, or be able to quickly update frames at the end of a motion.

5.4 Real World Parameters

A close examination of the commands presented in section 5.1 reveals that there is no reference to masses, friction, or any other real-world parameters. The master operator works in a purely kinematic model of the world, and has no real knowledge of information such as surface roughness at the remote site. The remote system, however, must be able to interact with the remote environment and thus must be able to compensate for these parameters.

Commands such as Force and GuardForce need real-world parameters as they are parsed. The GuardForce command is very sensitive to surface conditions. As the manipulator slides along a surface, more information about the surface can be gathered, and GuardForce and other commands that need these parameters can be modified. This is possibly the most important aspect of the command execution level: the ability of the system to learn from the tasks currently executed in order to respond more intelligently to the sensor inputs.

5.5 Tolerance

The human operator works in a model of the remote world, but the model is only accurate to some tolerance, ϵ. Because the model may be built from sonar or video scans of the remote environment, the tolerance may be quite large.

Motion commands Move, Slide, and Pivot must all be able to compensate for tolerance errors. As noted before, the tolerance level is crucial for the effectiveness of the error message generation. However, there is a correlation between the effectiveness of tolerance limits and the generated motion distance. If the expected limit of the motion is the same magnitude or smaller than the tolerance, problems will result. These situations are under investigation.

5.6 Robust Collision Detection

As the remote manipulator moves around its environment, its most important task is to detect collisions. A collision may signal the correct termination of a command, or may be an unwanted interaction with the environment that may damage the manipulator. In the first case, it is important that all mode changes and contact forces are quickly implemented, and the next command is started immediately. In the second case, motion should be immediately terminated, and an error message sent to the master.

5.6.1 Detecting Contacts vs. Friction

When the manipulator is in free space, the sensor noise is relatively small compared with the sensor readings for a collision. When the manipulator is sliding in contact with a surface, however, the noise associated with the dynamic friction is quite large. Detecting contact above this friction noise has proved to be a problem. Robust methods of detecting collisions and stopping the robot must be developed.

If a model of the frictional forces can be built while the manipulator is in contact with the environment, this information can be beneficial in determining when a collision occurs.

5.7 Partial Results

Basic implementation of command execution, as well as the hybrid control, has been demonstrated to be effective. The language parser and interpreter are continually modified and improved, and a faster controlling computer will make more improvements possible. Robust stopping conditions using a standard deviation model of the friction have been simulated with actual data from the manipulator, and this algorithm can be implemented with the new computer system.

6 Tool Usage

As with humans, robots use tools to extend their abilities. A robot manipulator is greatly limited in its strength to accuracy ratio. In order for a relatively low-strength robot to do tasks that require more strength, we are experimenting with using high-powered tools in conjunction with the robot. Specifically, we are using an impact wrench, which can deliver more torque than the robot itself, and a winch that can lift more than the payload capability of the robot. These tools are sensorless, so the control of them requires the sensing of the robot. Thus, the precision of the low-power robot is coupled with the strength of less-precise heavy tools.

Some of the tools that the robot uses will create extra degrees of freedom in the robot-tool system. If the tool has a natural axis of rotation or translation, it is often desirable to have the tool itself control this degree of freedom, and make the corresponding degree of freedom of the manipulator passive. In this way, control of the tool will not create an extra burden for the human operator. Control of the tool will be as simple as the control of a degree of freedom of the manipulator. The control of the robot/tool system with added degrees of freedom will be studied.

6.1 Impact Wrench

An impact wrench has advantages over a conventional wrench used by the robot. First, it can deliver greater torque than the robot can by itself. Second, the impact wrench is easier to use than the conventional wrench, because no complex movements are needed. Drawbacks to using the impact wrench include vibrations caused by tool use, and the need for a power source.

The impact wrench used for this research will be a 3/8" drive pneumatic wrench. Mounted internally on the wrist (see figure 3), it creates an extra degree of freedom with the same axis of rotation as the terminal joint of the robot. By specifying the winch in the UseTool() command, rotation about the Z axis of the end effector frame (T6) would imply control of the wrench. Guarded moves can be employed to stop the wrench when a large torque is encountered, indicating that a bolt has been properly seated. Appropriate error conditions must also be sensed. The corresponding degree of freedom in the manipulator is position controlled with no specified motion.

6.2 Winch

A winch can increase the load-carrying capacity of the robot manipulator. The need for power to offset the force of gravity on payloads, which usually accounts for a large

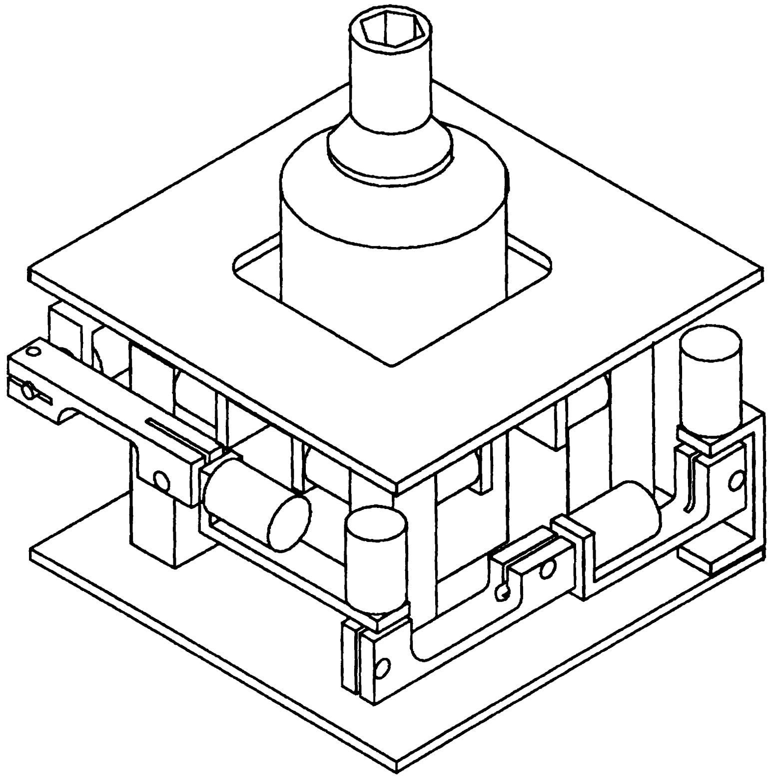

Figure 3: Wrist Outfitted With Impact Wrench

fraction of the total power of the robot, is eliminated. Because the payload capacity of a manipulator is usually inversely proportional to its accuracy, a much more accurate robot can be used for manipulation of heavy objects when a winch is used.

The winch is sensorless, and thus relies on the robot for sensing. The winch also adds a degree of freedom to the system, in essence adding a prismatic joint in the world z-coordinate. Motion in this direction is naturally controlled by the winch when the winch is specified with the UseTool() command. The corresponding degree of freedom in the manipulator is force controlled with a force preload to keep the winch cable taut.

The winch used for this research (see figure 4) is small, and not much more powerful than the robot itself. However, knowledge gained about using the winch in conjunction with the robot can be applied to much more powerful winches. The winch operates at a fixed speed, and therefore the motions at the operator's station must be constrained to this motion when the operator is using the winch.

6.3 Partial Results

The winch hardware is in place, and experiments using the winch have been performed, with moderate success. Implementation of control by the winch of a degree of freedom of the system has been demonstrated, but is not yet an integral part of the system. The impact wrench is functional, but works only under manual control until an interface with the computer can be built. However, preliminary tests of the winch attached to the manipulator have been successful.

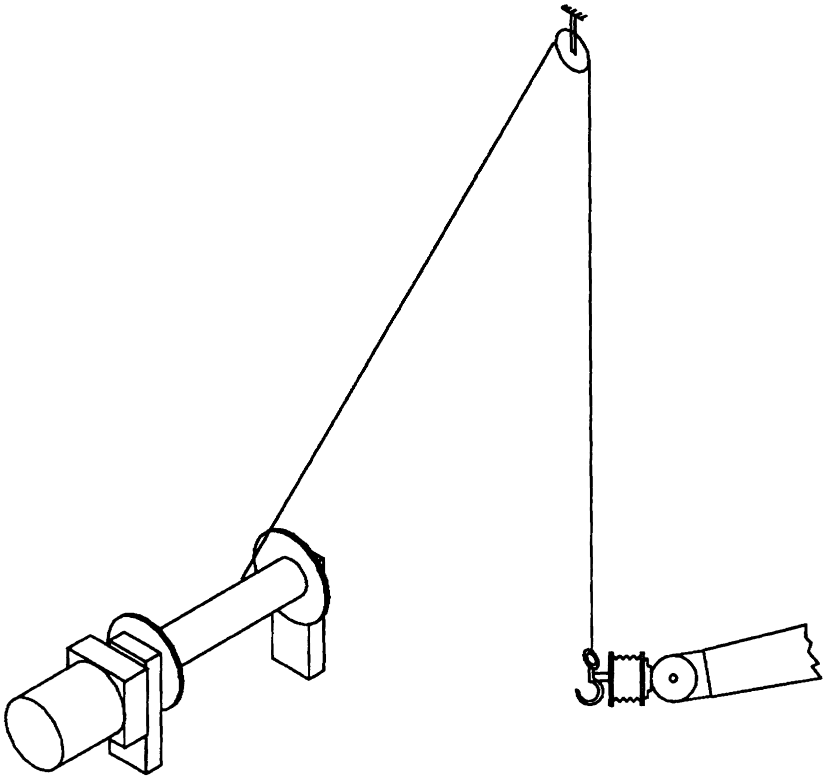

Figure 4: System Controlled Winch

7 Conclusions and Contributions

A remote teleprogramming system capable of dealing with uncertainty and unspecified real-world parameters will be developed. The low level control must be generalized to the point that the robot can be controlled in an arbitrary task frame. The system will need robust collision sensing, and must have no hybrid control mode switching problems. In developing this system, research into tool usage will be conducted, including the modification of the manipulator control to allow the tools to control certain degrees of freedom.

The control structure used for teleprogramming will also be available for direct programming of the slave robot. This is a useful tool for debugging of the teleprogramming system, and also for directly controlling the robot for other purposes.

References

[1] Haruhiko Asada and Sheng Liu. Experimental verification of human skill transfer to deburring robots. In *Proceedings of the Second International Symposium on Experimental Robotics*, Toulouse, France, 1991.

[2] D. J. Barber. *MANTRAN: A Symbolic Language for Supervisory Control of a Remote Manipulator*. Master's thesis, Massachusetts Institute of Technology, 1967. S. M. Thesis.

[3] A. K. Bejczy, W. S. Kim, and S. C. Venema. The phantom robot: predictive displays for teleoperation with time delays. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 546–551, 1990.

[4] R. Bolles and R. Paul. *The Use of Sensory Feedback in a Programmable Assembly System.* Technical Report, Stanford Artificial Intelligence Lab., October 1973. Memo 220.

[5] William R. Ferrell. Remote manipulation with transmission delay. *IEEE Transactions on Human Factors in Electronics*, HFE-6(1):24–32, 1965.

[6] William R. Ferrell and Thomas B. Sheridan. Supervisory control of remote manipulation. *IEEE Spectrum*, 81–88, October 1967.

[7] Janez Funda. *Teleprogramming: Towards Delay-Invariant Remote Manipulation.* PhD thesis, University of Pennsylvania, 1991.

[8] R. C. Goertz. Manipulators used for handling radioactive materials. In E.M.Bennet, editor, *Human Factors in Technology*, chapter 27, McGraw-Hill, 1963.

[9] G. Hirzinger, J. Heindl, and K. Landzettel. Predictive and knowledge-based telerobotic control concepts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1768–1771, 1989.

[10] Thomas Lindsay, Janez Funda, and Richard P. Paul. Contact operations using an instrumented compliant wrist. In *Proceedings of the Second International Symposium on Experimental Robotics*, Toulouse, France, June 1991.

[11] Matthew T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(6):418–432, June 1981.

[12] G. Niemeyer and J-J. E. Slotine. Stable adaptive teleoperation. In *Proceedings of the Second International Symposium on Experimental Robotics*, Toulouse, France, June 1991.

[13] T. Ogasawara, K. Kitagaki, T. Suehiro, T. Hasegawa, and K. Takase. Model based implementation of manipulation systems with artificial skills. In *Proceedings of the Second International Symposium on Experimental Robotics*, Toulouse, France, June 1991.

[14] R. P. C. Paul. Wave: a model based language for manipulator control. *The Industrial Robot*, 4(1):10–17, 1977.

[15] Richard Paul and Bruce Shimano. Compliance and control. In *Joint Automatic Control Conference*, 1976.

[16] Richard P. Paul, Janez Funda, Thierry Simeon, and Thomas Lindsay. Teleprogramming for autonomous underwater manipulation systems. In *Intervention '90*, pages 91–95, The Marine Technology Society, June 1990.

[17] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[18] Randall K. Roberts, R. P. Paul, and Benjamin M. Hillberry. The effect of wrist force sensor stiffness on the control of robot manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 269–274, April 1985.

[19] J. De Shutter and H. Van Brussel. Compliant robot motion. *International Journal of Robotics Research*, 7(4), August 1988.

[20] Richard Volpe and Pradeep Khosla. Experimental verification of a strategy for impact control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1854–1860, 1991.

[21] D. E. Whitney. Historical perspective and state of the art in robot force control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 262–268, 1985.

[22] Daniel E. Whitney. Quasi-static assembly for compliantly supported rigid parts. *ASME Journal of Dynamic Systems, Measurement, and Control*, 104:65–77, March 1982.

[23] Peter M. Will and David D. Grossman. An experimental system for computer controlled mechanical assembly. *IEEE Transactions on Computers*, C-24(9):879–888, 1975.

[24] Y. Xu and R. Paul. On position compensation and force control stability of a robot with a compliant wrist. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1173–1178, 1988.

[25] Yangsheng Xu. *Compliant wrist design and hybrid position/force control of robot manipulators.* PhD thesis, University of Pennsylvania, 1989.

[26] Tsuneo Yoshikawa. *Foundations of Robotics.* MIT Press, 1990.