

**Contact Operations Using An Instrumented  
Compliant Wrist**

**MS-CIS-91-68  
GRASP LAB 275**

**Thomas Lindsay  
Janez Funda  
Richard Paul**

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

**September 1991**

# Contact Operations Using an Instrumented Compliant Wrist<sup>†</sup>

Thomas Lindsay, Janez Funda, and Richard Paul

## Abstract

*Teleprogramming* was developed as a solution to problems of teleoperation systems with significant time delays [5]. 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.

Contact operations involve the remote site manipulator interacting with the environment, including planned and unplanned collisions, and motion with contact with the environment. A hybrid position/force control scheme using an instrumented compliant wrist has been demonstrated to be very effective for these types of operations. In particular, switching between position and force modes (when contacting a surface, for example) does not present problems for the system. A brief introduction of teleprogramming and contact operations is presented, including a model of sliding motions and early experimental results. Problems with these early experiments are presented, and solutions discussed. The criteria for an object to slide rather than tip over are presented, relating to the geometry of the object and the applied forces. Finally, methods are presented to match the experimental results to a simple model, to help the remote manipulator to quickly and robustly sense collisions.

## 1 Introduction

Teleoperation systems are important for the execution of tasks in hazardous and unstructured environments. Hazardous environments range from those extremely dangerous to humans, such as contaminated nuclear power plants and hazardous waste sites, to those such as space and deep sea that can be made safe to humans only for short periods

---

<sup>†</sup>To appear in *Proceedings of the Second International Symposium on Experimental Robotics*, Toulouse, France, June 1991. 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.

at great expense. Completely autonomous activity and manipulation is impractical in unstructured environments with state of the art artificial intelligence.

When delays in excess of one second occur, direct force reflecting teleoperation becomes difficult to impossible [6, 13]. Delays can occur on the order of 2-8 seconds for communication with a remote site orbiting the earth (shallow space), and up to 20 seconds for subsea operations (communicating via acoustic link). In order to solve problems associated with communication delays, we have developed a teleoperation structure called teleprogramming [10]. At the master site, a human operator works with a 6-DOF master manipulator to guide a simulated slave manipulator in a geometric model of the remote site. The model provides for monitoring of contacts, and feeds back information to the master arm to give the operator kinesthetic feedback, lacking in most of the current work involving time delays [2, 8, 12, 13].

The master system generates commands based upon the motions and manipulator/model interactions. This information is sent to the remote site, which interprets and executes these command steps. Each step is executed autonomously, and the resulting motion of the slave manipulator is analyzed as to whether it succeeded or failed. If it succeeds, an acknowledgment is sent to the master and the slave continues with the next command. Commands from the master are sent continuously, so there is no delay between commands at the remote site if they are executed without errors. If a command fails, information about the error state is sent to the master, and then the slave waits for the human operator to send a new set of commands that will correct the error.

At the remote site, the slave interprets small execution model steps that make up individual motions. Each execution model step contains information about how long and how far to move, information about contacts and contact forces, and information about what conditions the slave should expect to terminate the motion. For example, a typical move could command the slave manipulator to slide along a surface, pushing against it with a given force, and stop when a wall is encountered. Errors in this example could include falling off the surface, failing to find the specified wall, and encountering an unexpected obstacle.

We are using an instrumented compliant wrist for sensory feedback [17, 9]. The compliance is extremely beneficial for the interactions (expected and unexpected) between the manipulator and the environment [11]. However, the compliance makes sliding motions more complex. Depending on the surface friction and the applied forces, the object on the surface may tend to tip over instead of sliding. Control and other problems lead to non-constant steady state forces in the normal and tangential directions. Peaks in these forces, which are used to determine expected and unexpected collisions, can cause a false identification of an error state. The level of system 'noise' is partially a function of manipulator configuration and direction of movement, so that constant limits that would work successfully in one direction will not work in another. A more robust method of detecting collisions while performing contact operations is necessary.

The rest of the paper is organized as follows. First, the experimental teleprogramming testbed is presented. Next, contact operations in the task environment are examined, from both a model based and experimental based perspective. Criteria for an object to slide rather than tip over are presented. Finally, methods to relate the model and experimental results are examined, with an emphasis on a more general and robust method for interpreting sensor readings.

## 2 Experimental Setup

The GRASP Lab<sup>1</sup> teleprogramming testbed is shown in schematic form in figure 1. The operator's station and the remote workcell are physically separated. The system can be divided into the master site (operator's station), the remote workcell, the communication link between these sites, and the task environment.

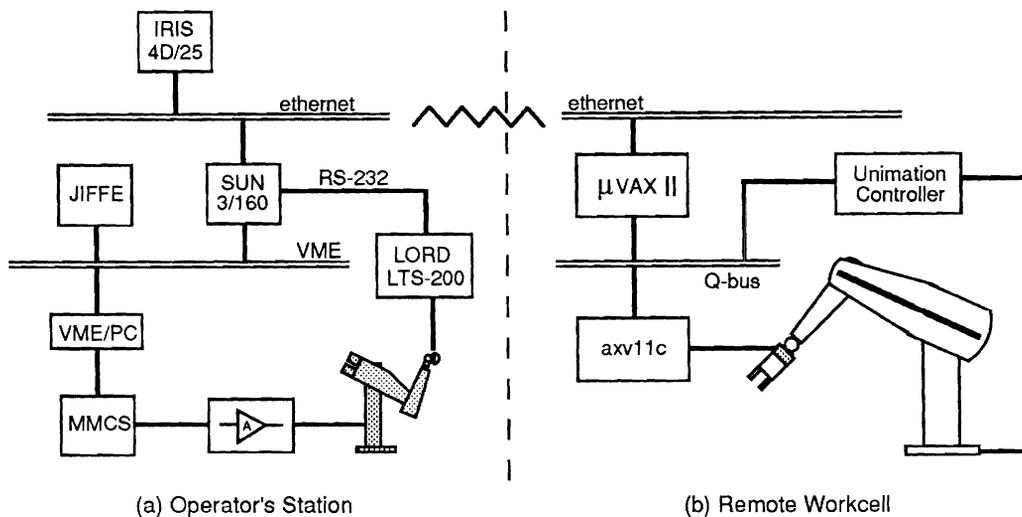


Figure 1: The experimental teleprogramming testbed

### 2.1 Master Site

The master site is composed of a Unimation Puma 250 robot, acting as a 6-DOF backdrivable input device, and several computers. The Puma hardware is controlled by PC-bus based Modular Motor Control System (MMCS) [4]. There is a 6-d.o.f. force/torque sensor (LORD Corp., LTS-200) mounted at the tip of the 250, which measures the directional input from the human operator. Joint and cartesian level control for the master is performed by JIFFE - a 20 Mflop VME-based floating point co-processor [1]. JIFFE communicates with its host (Sun 3/160) via shared memory and with the graphical workstation (Iris 4D/25) via the Sun and ethernet socket connection. The Iris runs a modeling environment for 3-D manipulation of articulated figures, provided by the Computer Graphics Laboratory at the University of Pennsylvania [3]. This software provides the operator with a graphical model of the remote manipulator and its environment. Manipulator/environment interaction is monitored, and is fed back to the master manipulator. This provides the operator with kinesthetic feedback, which is an important part of the teleprogramming system. The master system contains no information about the dynamics or friction at the remote site.

<sup>1</sup>General Robotics and Active Sensory Perception Laboratory, University of Pennsylvania Dept. of Computer and Information Science, Philadelphia, PA. Ruzena Bajcsy, Director.

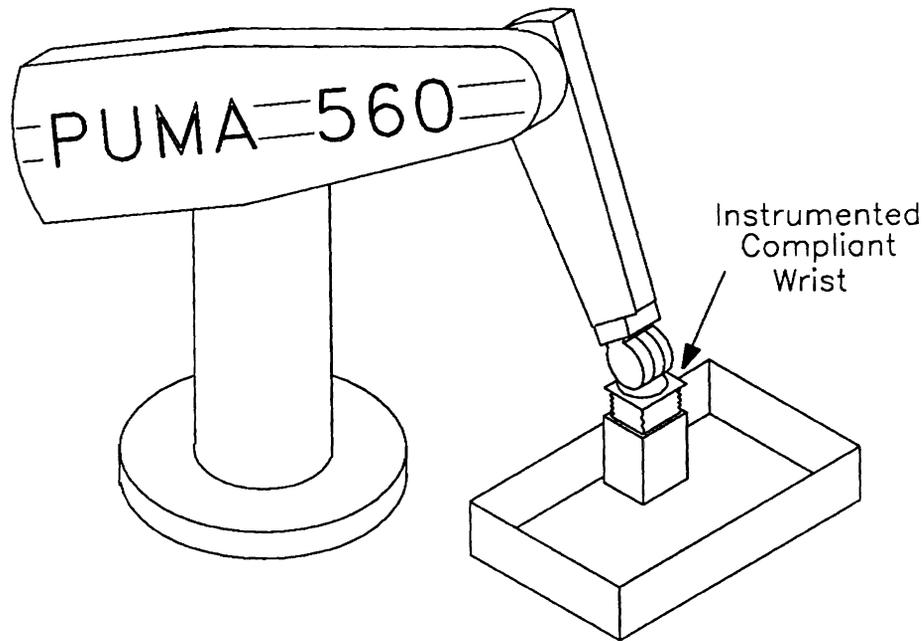


Figure 2: Remote Manipulator and Task Environment

## 2.2 Remote Workcell

The remote manipulator is a Puma 560 robot, linked to a MicroVax II. The robot uses a cartesian-based hybrid position/force controller, built upon the low-level RCI robot interface [7]. The hybrid controller has been shown experimentally to be stable in the operating region we are using, as long as the task frame origin is located relatively close (within 20 cm) to the robot wrist point. We use a 6-DOF instrumented compliant wrist for force/torque measurements.

The compliance of the wrist simplifies interactions between the robot and the environment. This is especially beneficial in dealing with the impact forces generated when the robot makes the transition from free space motion to motion in contact with the environment. Both natural and active damping help absorb the energy of impact [15]. Also, the compliance of the sensor helps to make the force control more responsive [11].

## 2.3 Communication

Communication from master to slave is composed of execution models (EMs), which are automatically generated at the master site. Each EM step can contain information about the working task frame, the hybrid modes, contact forces, and movement information. Information not supplied in a given EM step is assumed to carry over from the previous step, thus communication time is reduced by elimination of repetition of known information. The EM step does not contain information about the dynamics or friction of the environment.

The communication between the robots in the lab, using an ethernet connection, is virtually instantaneous. Therefore, communication delays are emulated by software. Currently, we are using a delay of 3 seconds for our experiments.

## 2.4 Task Environment

We are currently experimenting with very simple contact operations. A small box attached to the manipulator is maneuvered into and around a larger box, as shown in figure 2. Elements of tasks include free-space motion, transitions between free-space (position mode) and constrained space (force mode), and constrained motion. In this environment we can test overall system performance, error detection, and error recovery. Within this task environment, our command language and teleprogramming concept have been shown to be effective. Problems between theory and experimental work have also been examined, and in many cases we have modified how commands are interpreted at the remote site. Complex procedures can be built using the commands we can now generate. Current work includes creating a new task environment which requires more complex motions.

## 3 Contact Operations

Although many tasks include free-space motion, most tasks require interaction with the environment. Free-space motion is a relatively simple operation; there is no need for feedback at the operator's station, and therefore a telerobotic scheme that has only visual feedback (in real time) would be adequate (as with JPL's predictive display). Most of our work concentrates on contact operations, where the manipulator interacts with the environment.

For two reasons, contact operations are executed semi-autonomously. First, the communication delays make force feedback to the operator impossible. Therefore, the remote site must close the feedback loop locally. Second, there may be inaccuracy in the graphical model at the master site. If the geometry of the environment is known only to a tolerance  $\epsilon$ , the remote site must locally deal with this inaccuracy. A fully autonomous system would have to understand all possible problems and deal with them appropriately; this is beyond the scope of modern artificial intelligence. When the remote system runs into a problem that it cannot correct, it simply sends back information to the human operator, who can reason through the problem and create a suitable correction.

Due to slow and often unreliable (esp. with acoustic links) communications, the commands sent to and from the remote site need to be minimal. The remote site receives only information about the kinematics of the system. Dynamics and friction are dealt with locally at the remote site. Further, the remote site must keep pace with the master site, albeit delayed by the communications. The slave therefore has no opportunity to explore the environment beyond the scope of the commanded actions. Thus the system can only gain information about the remote environment, such as friction, while it is also trying to discern expected and unexpected changes from the sensor data. Within these constraints, the remote system must interact with the environment and robustly sense forces, contacts, and collisions.

### 3.1 Contact Model

Motion of the robot/sensor/environment interaction can be simulated for one degree of freedom using a second order model. The second order model is a mass-spring-damper

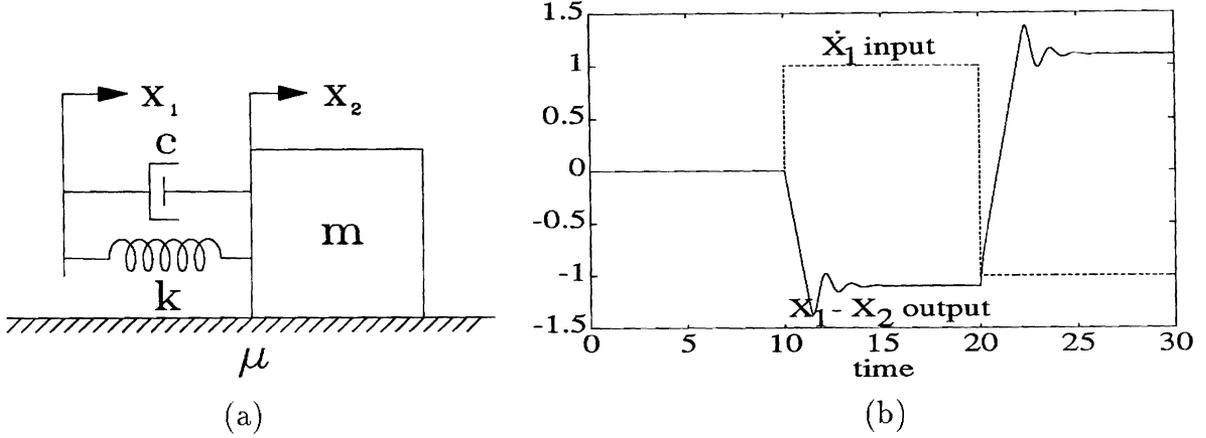


Figure 3: Second Order Model: (a) System; (b) Response.

system with a velocity input and coulomb friction. The equation for this model is shown below:

$$m\ddot{x}_2 + c\dot{x}_2 + k(x_2 - x_1) = f \quad (1)$$

where  $f$  represents the coulomb friction.

$$f = \begin{cases} \mu_{sl}N\left(\frac{\dot{x}_2}{|\dot{x}_2|}\right) & \text{if } |\dot{x}_2| > v_s \text{ and } F > \mu_{sl}N\left(\frac{\dot{x}_2}{|\dot{x}_2|}\right) \\ \mu_{st}N\left(\frac{\dot{x}_2}{|\dot{x}_2|}\right) & \text{if } |\dot{x}_2| < v_s \text{ and } F > \mu_{st}N\left(\frac{\dot{x}_2}{|\dot{x}_2|}\right) \\ F & \text{otherwise} \end{cases} \quad (2)$$

where  $F = c\dot{x}_2 + k(x_2 - x_1)$ , and  $N$  is the surface normal force. The value of  $v_s$  is the cutoff velocity that defines where, for simulation purposes,  $\mu_{st}$  (static friction) no longer applies, and the value  $\mu_{sl}$  (sliding friction) is used. Values for  $m$ ,  $c$ , and  $k$  are selected to model the wrist behavior, but do not represent the exact physical parameters of the wrist.

The spring-damper subsystem models the wrist sensor. The output from the sensor will be the change in spring length,  $\Delta x$ , and can be interpreted explicitly as a position deflection, or implicitly, using Hooke's law  $F = k\Delta x$ , as a force. Figure 3(a) illustrates the second order model. Figure 3(b) displays data from a simulation of the second order model, with the velocity input shown. For a given mass and input velocity, the rise time and the output level are a function of the spring constant and the coulomb friction. Overshoot is a function of the coulomb friction value and the viscous damping term.

### 3.2 Experimental Data

Data from the system has been collected and compared to the simulation model. In this section, some of the data will be presented, along with an introduction to some of the problems that were encountered while using the system. One problem was that the box being moved had a tendency to tip over while being pushed. Also, there were many problems associated with sliding along a surface until a wall was encountered. False interpretation of sensor readings, due to uneven frictional force and noise caused by sensor

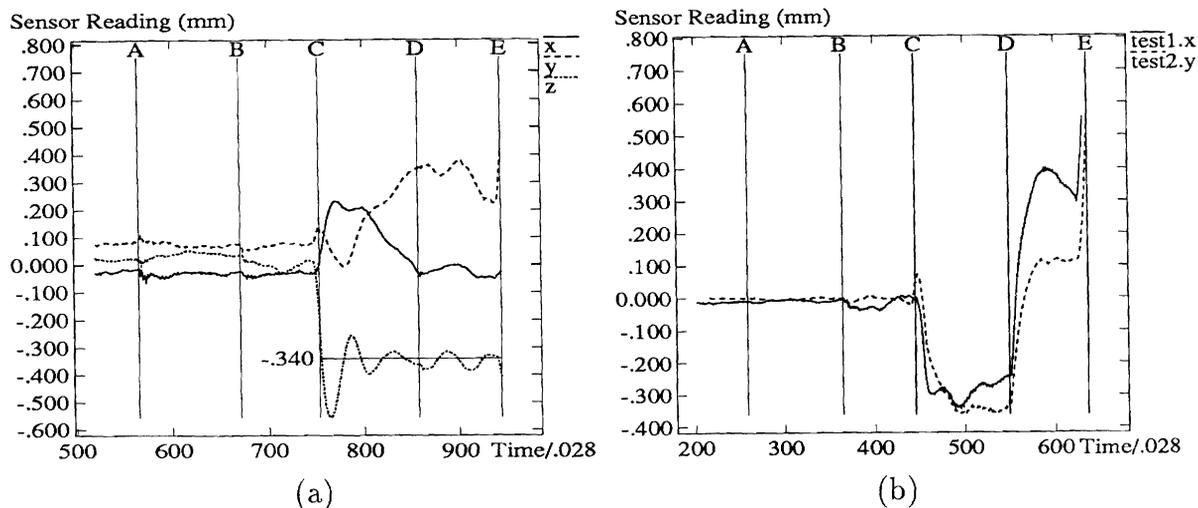


Figure 4: Sensor Readings: (a) Typical Move; (b) Direction Dependent.

electronics and by arm control, cause the system to stop before hitting a surface or to press on the surface with a large force before deciding to stop. Methods to overcome these problems are presented later in this paper.

Figure 4(a) shows the sensor readings, for translational directions, of a typical move. The robot moves at approximately 2 cm/sec. Section A-B is a free-space motion. There is a small amount of noise at the beginning of move A-B, which is caused by the transition from the previous free-space motion. Section B-C is a guarded move. At the end of move B-C, the robot comes into contact with the environment. Here, there is a large change in the z-direction sensor reading (note that the contact is smooth and stable, and the robot never breaks contact with the environment). Move C-D is a standard sliding motion, with the robot in contact with the environment and moving in the negative y-direction. The robot tries to maintain a normal force (z-direction) of approximately 4.2N (.34 mm) while sliding. There are large, unexpected changes in the x and y sensor readings during this section of the motion. Theoretically, there should be no forces in the x-direction, and the y-direction should have a constant frictional force of  $\mu N$ . The sensors, however, show that the tangential force (y-direction) has a minimum below zero, and a maximum of approximately 2.5N. Section D-E is another guarded move, and the robot comes into contact with a wall of the box. The slope of the y-direction sensor reading is high, but the actual value of the reading when the robot touches the wall is not significantly higher than other readings in the D-E section.

Figure 4(b) illustrates one of the inaccuracies of the sensor readings. Data “test1” and “test2” are from similar moves. Section B-C shows the robot coming into contact with the environment. In section C-D, the robot moves slightly away from a wall, and in section D-E the robot moves into contact with the wall. Motion is in the x-direction for the “test1” data, while test2 motion is in the y-direction. Although the normal force for both tests are nearly identical, section D-E in figure 4(b) shows very different tangential forces. The cause of this direction dependent phenomena is unclear, but a method to overcome the problem must be found in order to correctly monitor collisions and contacts.

The data presented above was collected after the tipping problem, presented below, was overcome.

## 4 Sliding vs. Tipping

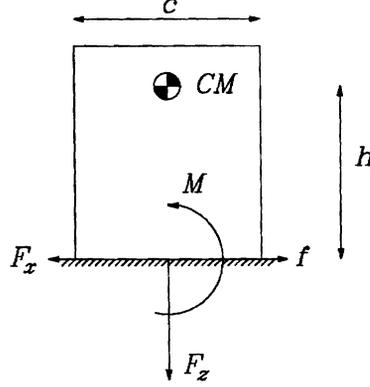


Figure 5: Forces on box in tool tip coordinates

When sliding in contact with the environment, the robot sometimes has the tendency to tip the box over. There are many factors that contribute to this tipping phenomenon. Three factors discussed here are the height to width ratio of the box, the normal to tangential force ratio, and the effect of rotational compliance upon sliding stability.

Expressing the applied forces in tool tip coordinates [16], which for this case will be the bottom of the box, the conditions for the box to tip over in the positive and negative Y-directions are found by summing the moments about the center of mass. The normal force  $N$  will act at the left side of the box if it is tipping about the negative Y-direction (into the page in figure 5), and the criteria for a 2-dimensional box not to tip is:

$$(F_x - f)h + M + N\left(\frac{c}{2}\right) \geq 0 \quad (3)$$

The normal force will act at the right side if it is tipping in the positive Y-direction. The criteria for the box not to tip in this direction is:

$$(F_x - f)h + M - N\left(\frac{c}{2}\right) \leq 0 \quad (4)$$

where  $N = F_z + mg$ , and  $f = \mu N$  (See figure 5). Reorganizing,

$$hF_x - \left(\mu h - \frac{c}{2}\right)F_z - mg\left(\mu h - \frac{c}{2}\right) \geq -M \quad (5)$$

$$hF_x - \left(\mu h + \frac{c}{2}\right)F_z - mg\left(\mu h + \frac{c}{2}\right) \leq -M \quad (6)$$

If we assume that  $mg$  is negligible compared with applied forces and moments,

$$hF_x - \left(\mu h - \frac{c}{2}\right)F_z \geq -M \quad (7)$$

$$hF_x - \left(\mu h + \frac{c}{2}\right)F_z \leq -M \quad (8)$$

In terms of  $h/c$ , these equations become

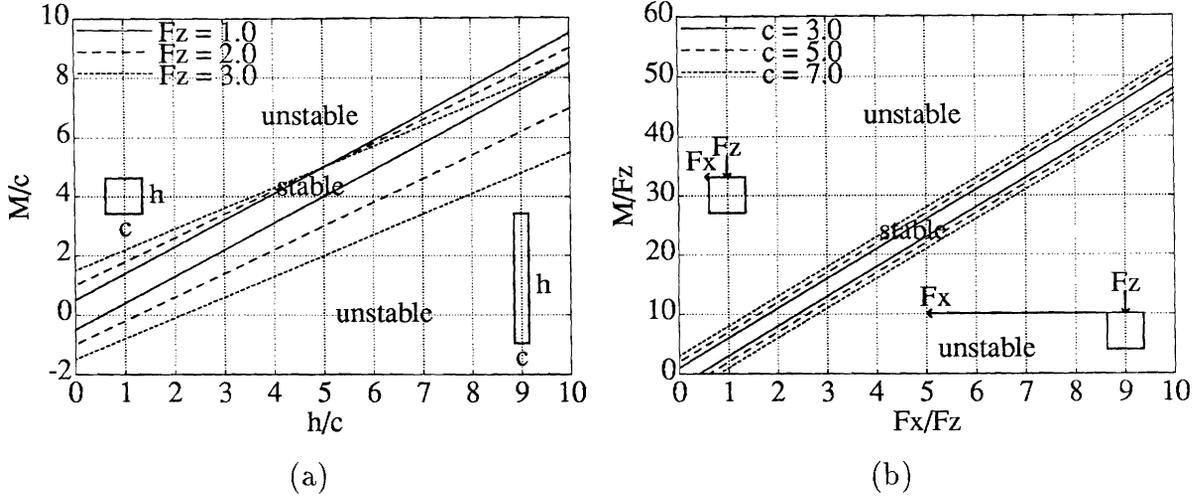


Figure 6: Tipping Criteria: (a)  $(-M/c)$  vs.  $h/c$ ; (b)  $(-M/F_z)$  vs.  $F_x/F_z$ .

$$\frac{h}{c}(F_x - \mu F_z) - \frac{c}{2}F_z \geq -\frac{M}{c} \quad (9)$$

$$\frac{h}{c}(F_x - \mu F_z) + \frac{c}{2}F_z \leq -\frac{M}{c} \quad (10)$$

These equations are plotted in figure 6(a), with parameters:  $F_x = 1.0$ ,  $\mu = .1$ , and  $F_z$  as shown. Note that for a given  $F_z$ , if  $h$  is large compared to  $c$ , then a moment must be applied for the box to remain stable. Also note that as  $F_z$  increases, the box will not tip for a greater range of applied moments. It is therefore more stable.

In terms of  $F_x/F_z$ , equations 9 and 10 become

$$h\frac{F_x}{F_z} - (\mu h - \frac{c}{2}) \geq -\frac{M}{F_z} \quad (11)$$

$$h\frac{F_x}{F_z} - (\mu h + \frac{c}{2}) \leq -\frac{M}{F_z} \quad (12)$$

$$(13)$$

These equations are plotted in figure 6(b) with parameters:  $h = 5.0$ ,  $\mu = .1$ , and  $c$  as shown. For a given value of  $c$ , if  $F_x$  is large compared with  $F_z$ , a moment must be applied for the box to remain stable. As  $c$  increases, the range for the applied moment becomes greater, and the box becomes more stable.

The conditions above are intuitive and easy to compensate for. However, in our experimentation the box still tends to tip. The reason pertains to the rotational compliance, and with transforming the applied forces and moment into the tool tip frame.

To transform the forces and moment, the compliance values of the wrist are needed. There are two parts to the compliance that are important here: the physical compliance and the control compliance. The physical compliance is inherent in the structure of the wrist and its compliant elements. The control compliance is a result of the gains used in the control of the system. A stiff wrist can be made more compliant with higher gains, if

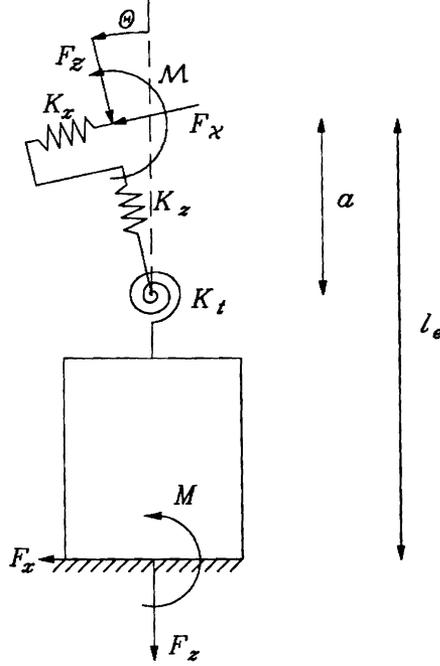


Figure 7: Transformation of forces from application to tool tip coordinates

it remains stable. The important point is that we can change the control compliance to suit our needs.

To transform the applied forces and moment for the two-dimensional wrist, the following equations are needed (see figure 7)

$$F_x = F_\chi \cos(\Delta\theta) - F_Z \sin(\Delta\theta) \quad (14)$$

$$F_z = F_\chi \sin(\Delta\theta) + F_Z \cos(\Delta\theta) \quad (15)$$

$$M = F_x(l_e - \Delta z) + F_z\left(\frac{c}{2} - \Delta x\right) + \mathcal{M} \quad (16)$$

$$\Delta x = F_\chi / K_x \quad (17)$$

$$\Delta z = F_Z / K_z \quad (18)$$

$$\Delta x = (\mathcal{M} + F_\chi a) / K_t \quad (19)$$

Substitution yields:

$$F_x = F_\chi \cos\left(\frac{\mathcal{M} + F_\chi a}{K_t}\right) - F_Z \sin\left(\frac{\mathcal{M} + F_\chi a}{K_t}\right) \quad (20)$$

$$F_z = F_\chi \sin\left(\frac{\mathcal{M} + F_\chi a}{K_t}\right) + F_Z \cos\left(\frac{\mathcal{M} + F_\chi a}{K_t}\right) \quad (21)$$

$$M = F_x\left(l_e - \frac{F_\chi}{K_x}\right) + F_z\left(\frac{c}{2} - \frac{F_\chi}{K_x}\right) + \mathcal{M} \quad (22)$$

$$(23)$$

Equation 20 is plotted in figure 8. The constants in the equation are chosen to approximate the behavior of the wrist:  $K_x = 7.29$  N/mm,  $K_z = 12.36$  N/mm,  $l_e = 25$  cm,  $c = 10$  cm,

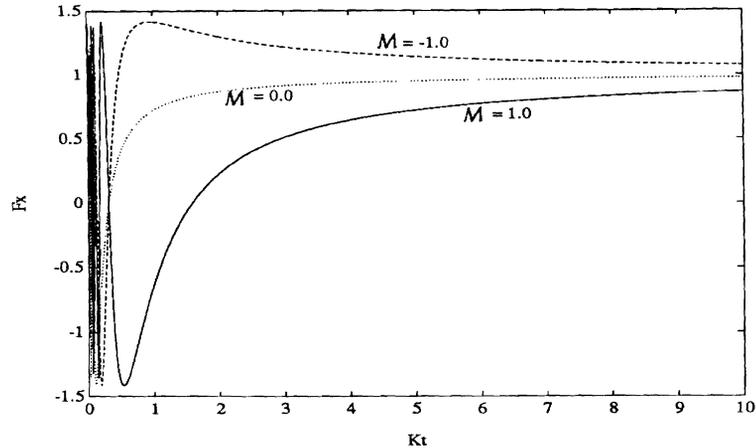


Figure 8:  $F_x$  vs.  $K_t$  for different values of  $M$

$F_x = 1$  N,  $F_z = 1$  N, and  $M$ , in N-m, as shown. Variable  $a$  was chosen to be 25 cm, which would correspond to the case where the center of compliance is at the tool tip (bottom of box, here), although the actual value is smaller. The physical value for  $K_t$  is 6.93 N-m.

The plot shows how the transformed forces and torque vary from those applied to the wrist. It is obvious that small values of  $K_t$  did not yield satisfactory performance. By decreasing the control compliance (increased  $K_t$ ), the box became much more stable. Note here that figure 8 also shows that the tool tip forces are never the same as the applied forces. It is important to the stability of the box that the tool tip forces are controlled accurately, and that the compliance of the wrist must be compensated for in the control. By examining equations 20 and 21, it is seen that the smaller the distance from the applied forces to the center of compliance ( $a$ ), the less effect that the force  $F_x$  has upon changing the values of the peg tip forces. Better results would be obtained for the operation of sliding if the center of compliance coincided with the applied forces. This is much different than the conclusions for peg insertion operations with RCC devices by Whitney [16], for which the center of compliance should be located at the tool tip.

## 5 Robust Stopping Conditions

The data presented in section 3.2 deviates from the predicted second order model behavior of the system. The deviations have many causes, and as a whole will be termed “noise”.

Noise from the sensors is inherent in any system. Experiments suggest noise that may be dependent on complex phenomena that may be difficult or impossible to model. Such phenomena include non-homogeneous friction, static friction, sensor coupling (coupling of compliant directions in the sensor), orientation instabilities (tipping, as presented above), and sensor-based hysteresis. These phenomena are all responsible for sensor “noise”.

As the manipulator slides around the environment, it attempts to maintain a constant normal force. With a constant normal force and velocity, the sliding friction should also be constant, assuming homogeneous surface friction. Contact with a side wall of the box thus could be determined by even a small increase in the tangential force. However, experiments have shown that a small threshold value causes the system to stop on noisy

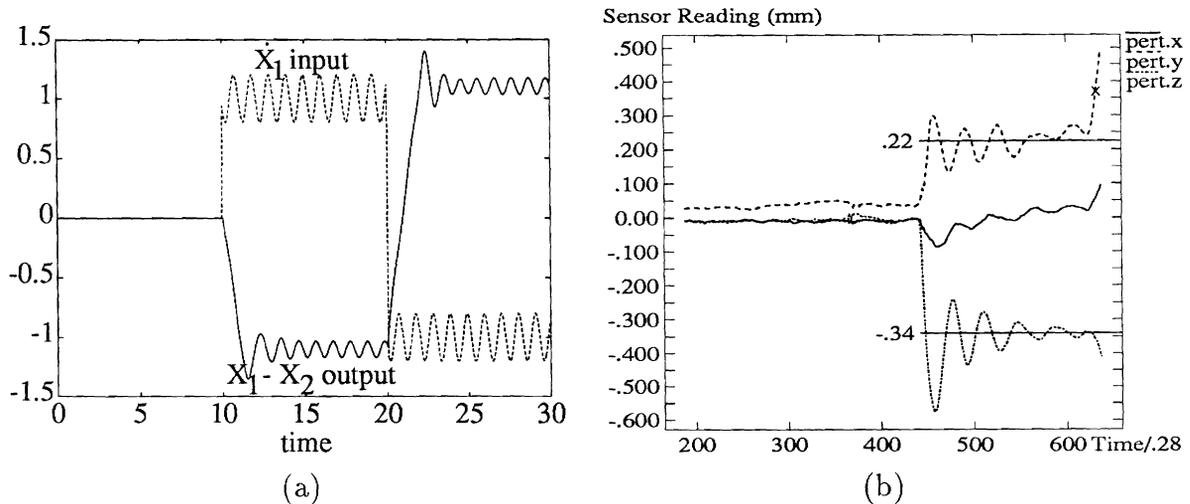


Figure 9: Motion Perturbation: (a) Model Behavior; (b) Typical System Behavior.

data. Using a constant threshold based stopping condition, a high threshold is needed to keep the noisy data from interfering with normal stopping criteria. Too high of a threshold may cause the system to interpret an actual contact with the wall as mere noise. Also, a high threshold causes the box to impact the environment with much more force than is wanted.

The following sections present attempts at developing more robust methods for determining stopping conditions, including ways to reduce the effects of the sensor noise, and to determine stopping criteria under noisy conditions.

## 5.1 Torque Preloads

Some of the control noise could be a result of the box being on the verge of tipping over. In order to reduce this noise, a torque preload could be used to make the box more stable. The preload direction is computed as  $F \times v$ . Using this preload unfortunately does not reduce the control noise, and does not significantly improve the performance of the system. However, it will make the box stable under more adverse conditions, at little computational cost.

## 5.2 Motion Perturbation

By perturbing the motion of the manipulator with small amplitude sine waves, some of the effects of noise phenomena can be actively reduced. Specifically, static friction problems can be overcome.

Figure 9(a) shows the simulated output of the second order model with a velocity input as in figure 3(b), with a superimposed sine wave with an amplitude of 1/5 the constant velocity input. The output is quite similar to that of figure 3(b), superimposed with a very small amplitude sine wave. The sine wave perturbation causes no adverse effects to the output as long as the frequency is not near the natural frequency of the system.

Experimental results from motion perturbation are shown in Figures 9(b) and 10(a). Figure 9(b) can be compared with figure 4(a) to show the improvement of the sensor

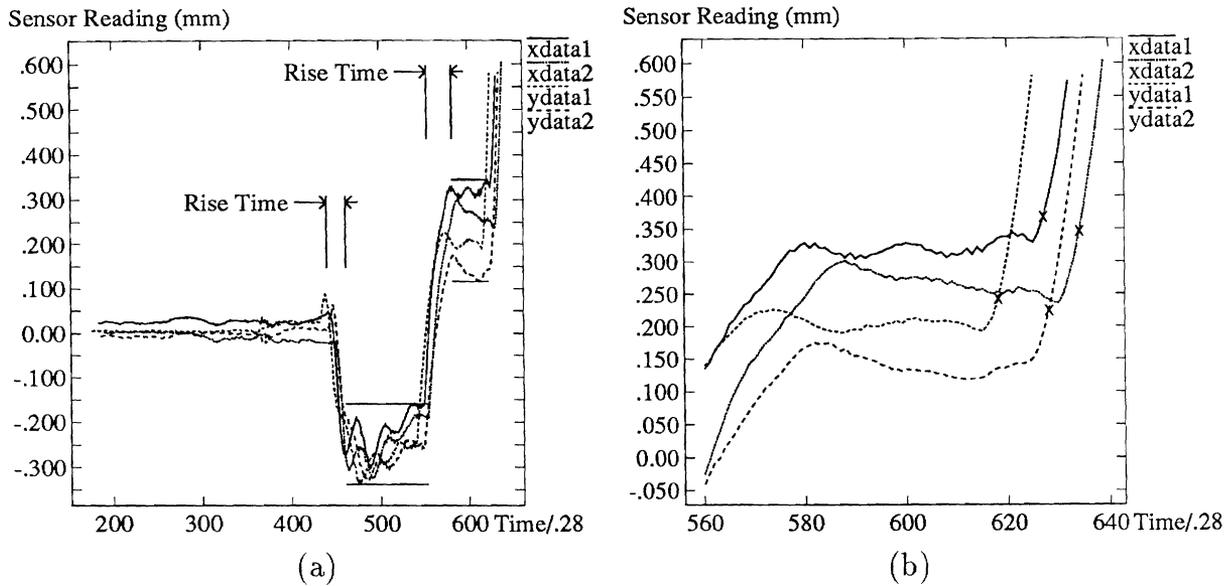


Figure 10: Motion Perturbation: (a) Experimental Data; (b) Magnified View of Collision Data.

output with motion perturbation. The z-direction output is similar for both moves, but in the perturbed motion move, the x-direction (normal to motion) output remains close to zero. Further, the y-direction (motion direction) output has the characteristics of a 2nd order system. As motion in the y-direction begins, the sensor output rises to a peak value, and then oscillates about a steady state value. Contact with the wall is indicated with a distinct rise in the sensor output. Figure 10(a) compares multiple moves. The data spread for the steady state value of the output is much smaller than similar moves without perturbation.

With data as shown in figure 10(a), the use of a constant threshold value for determination of contact can be revised. If a move that is known to terminate in contact is long enough to create a model, contact with the wall can be determined by a data point that falls outside  $n$  standard deviations computed from data collected after the rise time of the move. A simple collision detection algorithm is shown in figure 11. The “X” in figure 9(b) indicates where a collision would have been detected using this algorithm, with  $n = 3.5$ . Figure 10(b) shows a closeup of the data from figure 10(a) where the wall is encountered. The “X” marks indicate where the wall would have been detected.

Two refinements to this algorithm can be made. The first is to retain an absolute maximum constant threshold, so that if the data readings are very noisy, or if an obstacle is encountered before an adequate model can be built, the robot can still stop on a given force. This would eliminate the possibility of damage to the robot and the environment. Second, the algorithm as it stands uses all of the data points after the rise time to compute the mean and standard deviation. Computing the mean and standard deviation from only the previous  $N$  data points would provide a local standard deviation that would be more useful for non-homogeneous friction.

Motion perturbation, while improving the performance of the system by reducing the effects of static friction, still does not overcome the “noise” associated with non-homogeneous friction. It does, however, produce sensor output that conforms well to a

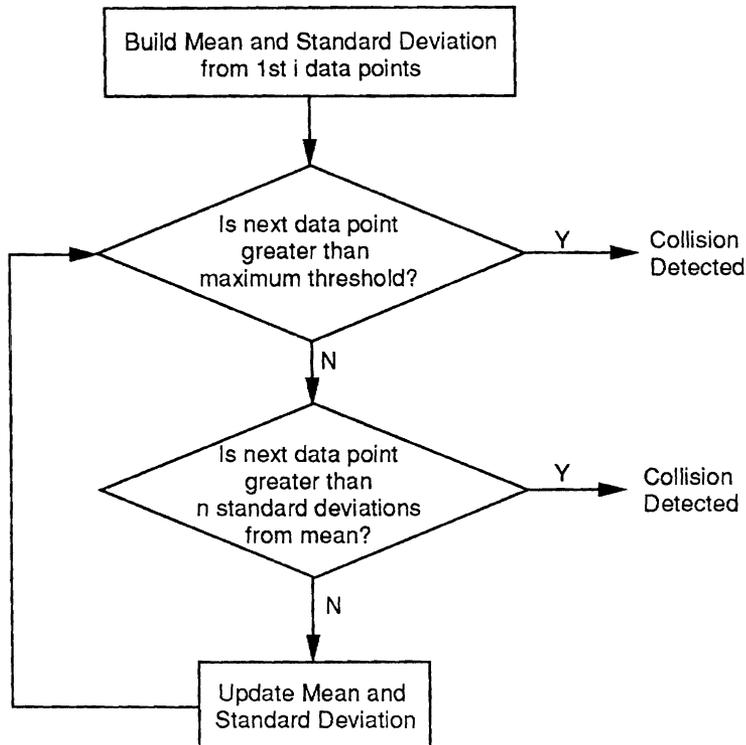


Figure 11: Collision Detection Algorithm

system model. The collision detection algorithm is also beneficial when the environment contains other surfaces with different coefficients of friction.

### 5.3 Exploratory Procedures

In some instances, surface conditions may impede command execution to the point that contact operations are impossible under the current model of the environment. An example of this would be trying to detect collision with a foam rubber wall while sliding across a very rough surface. In cases like this, it may become necessary for the remote site to autonomously explore surface conditions while the human operator waits.

A more refined model of the environment obviously leads to more accurate analysis of sensor data. The operator works in a model world dealing with kinematics only. As the slave manipulator operates in the real world, data about the environment can be gathered, analyzed, and used to refine new incoming data. Many surface attributes can be recovered through normal operation of the manipulator, including penetrability, hardness, compliance, compressibility, deformability, and surface roughness [14]. These criteria may be enough to refine the environment model to the point where contact operations can again be accomplished using the same types of commands from the master site as before. However, there may be surfaces where the current paradigm of contact operations cannot be used. At this point, the human operator must adapt the motion strategies to reflect the surface attributes. Instead of sliding along a surface to find a wall, for example, the operator may have to move above the surface, poking the surface occasionally to make sure that “contact” has not been lost, until the wall is encountered.

## 6 Conclusions

Although the criterion for contact operations, including collision and error detection, appear to be simple, it is shown that using real world sensors and control, a much more robust set of rules must be used. By utilizing robust criterion for error detection, limited execution model commands can be successfully carried out, and actual error states can be differentiated from spurious data.

## References

- [1] R. L. Andersson. Computer architectures for robot control: a comparison and a new processor delivering 20 real mflops. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1162–1167, 1989.
- [2] Forrest T. Buzan. *Control of Telemanipulators with Time Delay: A Predictive Operator Aid with Force Feedback*. PhD thesis, Massachusetts Institute of Technology, 1989.
- [3] C.B.Phillips and N.I.Badler. Jack: a toolkit for manipulating articulated figures. In *Proc. of ACM/SIGGRAPH Symposium on User Interface Software*, Banff, Alberta, Canada, 1988.
- [4] Peter I. Corke. *A New Approach to Laboratory Motor Control: The Modular Motor Control System*. Tech. Report, Univ. of Pennsylvania, Philadelphia, PA, 1989.
- [5] Janez Funda. *Teleprogramming: Towards Delay-Invariant Remote Manipulation*. PhD thesis, Univ. of Pennsylvania, 1991.
- [6] B. Hannaford and W.S. Kim. Force reflection, shared control, and time delay in telemanipulation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 133–137, 1989.
- [7] Vincent Hayward. *RCCL User's Manual*. Tech. Report TR-EE-83-46, Purdue Univ., 1983.
- [8] B. Hirzinger, J. Heindl, and K. Landzettel. Predictive and knowledge-based telerobotic control concepts. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1768–1777, 1989.
- [9] T. Lindsay and R.P. Paul. *Design of a Tool-Surrounding Compliant Instrumented Wrist*. Tech. Report MS-CIS-91-30, GRASP LAB 258, Univ. of Pennsylvania, 1991.
- [10] R.P. Paul, J. Funda, T. Simeon, and T. Lindsay. Teleprogramming for autonomous underwater manipulation systems. In *Intervention '90*, pages 91–95, The Marine Technology Society, June 1990.
- [11] R.K. Roberts, R.P. Paul, and B.M. Hillberry. The effect of wrist force sensor stiffness on the control of robot manipulators. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 269–274, April 1985.
- [12] P. S. Schenker and A. K. Bejczy. Workspace visualization and time delay in telerobotic operations. In *13th Annual AAS Guidance and Control Conf., Aerospace Human Factors Session*, February 1990. Keystone, CO.
- [13] Thomas B. Sheridan. Telerobotics and human supervisory control. To be published as a book.
- [14] P.R. Sinha, Y. Xu, R. Bajcsy, and R.P. Paul. *Robotic Exploration of Surfaces With a Compliant Wrist Sensor*. Tech. Report MS-CIS-90-92, GRASP LAB 244, Univ. of Pennsylvania, Philadelphia, PA, 1990.
- [15] R. Volpe and P. Khosla. Experimental verification of a strategy for impact control. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1854–1860, 1991.
- [16] Daniel E. Whitney. Quasi-static assembly for compliantly supported rigid parts. In M. Brady, J.M.Hollerback, T.L.Johnson, T.Lozano-Perez, and M.T.Mason, editors, *Robot Motion: Planning and Control*, pages 429–462, MIT Press, 1982.
- [17] Yangsheng Xu. *Compliant wrist design and hybrid position/force control of robot manipulators*. PhD thesis, Univ. of Pennsylvania, 1989.