AUTOMATIC IDENTIFICATION OF ENVIRONMENT HAPTIC PROPERTIES[1]

Pierre E. Dupont
Aerospace and Mechanical Engineering
Boston University
Boston, MA 02215
pierre@bu.edu

Capt Timothy M. Schulteis
San Antonio Air Logistics Center
Kelly AFB, TX 78241

Paul A. Millman
Computer Motion Inc.
Goleta, CA 93117

Robert D. Howe
Division of Applied Science
Harvard University
Cambridge, MA 02138

Abstract

A system that can process sensory information collected during telemanipulation tasks in order to automatically identify properties of the remote environment has many potential applications. These include generating model-based simulations for training operators in critical procedures and improving real-time performance in unstructured environments or when time delays are large. The research issues involved in developing such an identification system are explored, focusing on properties that can be identified from remote manipulator motion and force data. As a case study, a simple block-stacking task, performed with a teleoperated two-fingered planar hand, is considered. An algorithm is presented which automatically segments the data collected during the task, given only a general description of the temporal sequence of task events. Using the segmented data, the algorithm then successfully estimates the weight, width, height, and coefficient of friction of the two blocks handled during the task. This data is used to calibrate a virtual model incorporating visual and haptic feedback. This case study highlights the broader research issues which must be addressed in automatic property identification.

[1] Accepted for publication in *Presence: Teleoperators and Virtual Environments* as of March 3, 1999.

1 Introduction

Traditional teleoperated manipulation relies entirely on the human operator for perception and control. The operator receives information from the remote environment visually and kinesthetically, and then generates the appropriate commands to accomplish tasks. However, in many teleoperation applications, machine perception of the remote environment can also play an important role. For example, in remediation of toxic waste dumps, quantitative measurements of the size and weight of the containers helps to infer their contents and to determine optimal handling strategies (Griebenow, 1994). Similar considerations apply to collecting rock samples on interplanetary missions and to transporting and defusing explosives.

Machine perception can also be used to develop models of the environment for use as real-time simulators and training systems. Simulators have already been used to overcome stability problems related to teleoperation with significant time delays (Noyes and Sheridan, 1984; Bejczy, Kim, and Venema, 1990; Funda and Paul, 1991). Training systems calibrated using actual feedback data would provide a realistic and safe practice environment for surgical applications (Satava and Jones, 1997; Docimo et al., 1997), toxic waste remediation (Griebenow, 1994), and munitions loading (Leahy and Hamill, 1995). In contrast, current methods to calibrate virtual models are typically based on idealized physical models whose parameters are often selected arbitrarily and then subjectively adjusted until the virtual model "feels" like the real object. One notable exception is MacLean's work (1996), in which force-displacement data from a toggle switch was used to recreate the haptic sensations for a virtual switch.

The problem of automatic identification of remote object properties has received little attention in telemanipulation research. By "automatic" we mean that the identification procedure is performed by the telemanipulation system with little or no input from the human operator.) Ideally, such a system would be able to identify all parameters of interest by observing normal telemanipulation procedures, making the system transparent to the operator. In actuality, input may be required from operators in the form of information, e.g., task context, or special exploratory motions. In an extreme example of this approach, Fyler (1981) demonstrated the ability to build up a picture of the shape of a remote object using a touch probe on the end of the robot manipulator arm.

Automatic property identification is a very complex problem. In this paper, our goals are to provide a definition of the problem, to outline the components of its solution, and to illustrate its solution using real teleoperator data. To motivate the problem definition appearing in the next section, consider the case of estimating the mass of a block during a pick-and-place operation. Assuming force sensing on the remote gripper, block mass can be estimated using inertial and/or gravity forces. The choice of estimation formula would depend on the relative size of the inertial and gravity forces as well as the precision of the force sensors. In either case, the mass must be estimated from that portion of the data stream during which the block is in contact with the gripper, but not in contact with any other objects. To accomplish this, two things are needed. First, a structure must be applied to the manipulation task which

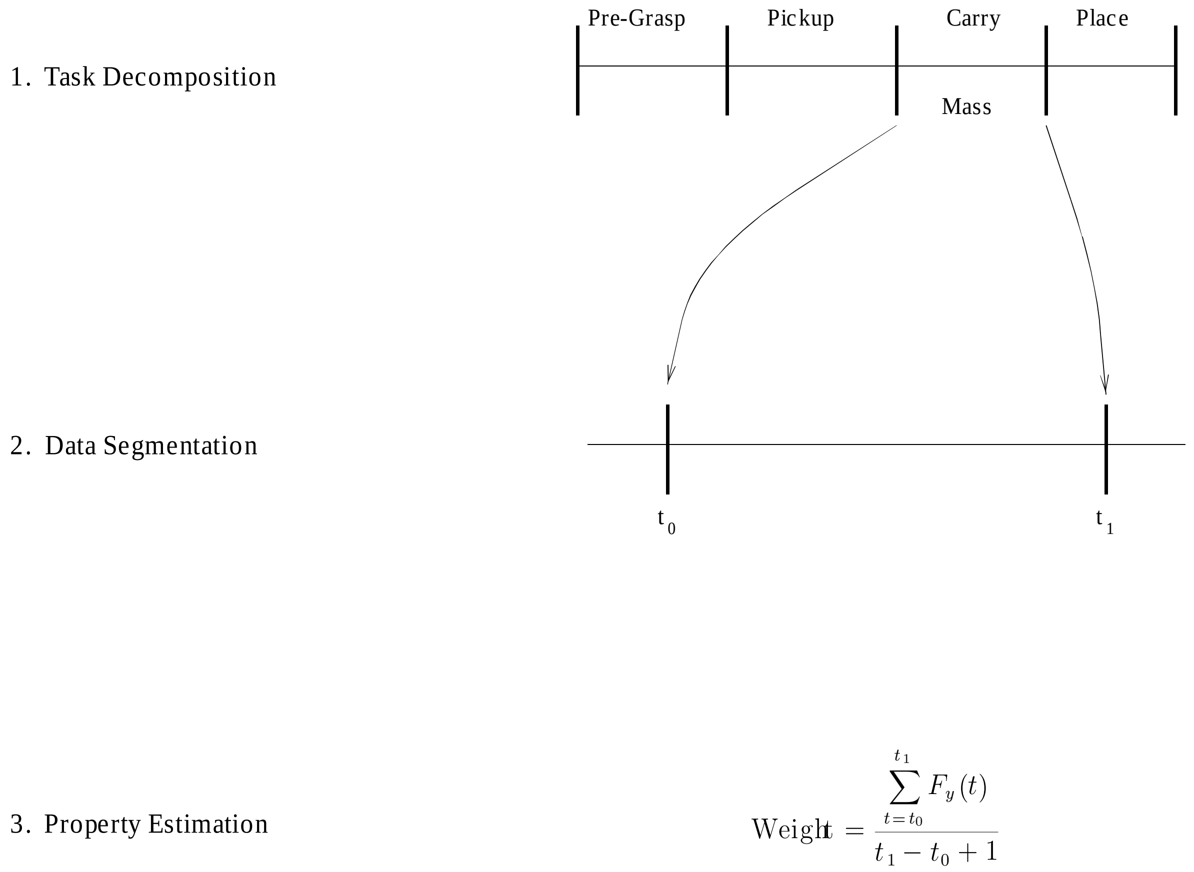


Figure 1: Automatic property identification applied to a pick-and-place task.

decomposes it into subtasks (e.g., “carry”) during which specific properties (e.g., mass) can be identified. Given this structure, a technique is then needed to extract sequences from the sensor data stream which correspond to property-estimating subtasks.

By means of this example, we can see that automatic property identification is composed of three distinct subproblems: task decomposition, data segmentation, and parameter estimation (Schulteis et al., 1996). Figure 1 depicts this process for the pick-and-place example. In the first step, the task is decomposed into *grasp*, when the manipulator makes contact with the object; *pickup*, when the object is lifted from the ground; *carry*, when the object is held above the ground; and *place* when the object is put back down on the ground. For the second step, the data segment associated with the *carry* subtask must be identified within the data stream corresponding to the entire task. Finally, an estimation technique is applied to this data segment to compute the object’s mass. In this case, a simple average of vertical gripper forces is shown.

In the next section, the environment identification problem is formally defined. The three subproblems of automatic property identification are presented in the context of the two proposed applications, namely real-time machine perception and virtual training systems. To the authors’ knowledge, this is the first work to consider the general problem of automatic property identification. Prior research has, however, considered task decomposition, data segmentation and parameter estimation individually. Such work is also summarized in this section.

Section 3 presents a case study of a block stacking task performed by a planar, two-fingered teleoperated hand. This study illustrates the issues involved in implementing a solution to each of the three subproblems. In section 4, the results of the case study are applied to the creation of a virtual training system. Identified object parameters are used to calibrate a dynamic simulation model. Using this model, operator motions applied to the teleoperator master generate reaction forces on the master and visual feedback on a computer monitor. Force trajectories produced during virtual manipulation are used as a basis of comparison with actual manipulation. The paper concludes by enumerating the unresolved research challenges of automatic property identification.

2 Automatic Environment Identification

The identification problem can be formally defined as follows.

Given a task description, $\mathcal{T}$, a sensor data stream, $d(t)$, and a set of properties to be determined, p, compute estimates of the states, $\hat{x}(t)$, and the properties, $\hat{p}(t)$, for $t \in [0, t_{final}]$.

A *task description*, $\mathcal{T}$, contains, at a minimum, a specification of the desired interactions between objects in the remote environment. Additional detail could indicate which objects the robot should handle and available grasp configurations. It could also include parameterized models of objects relating to, e.g., geometry and contact forces.

The *sensor data stream* can be represented by a vector $d(t) \in \Re^n$ where n is the number of sensor channels.

State, $x(t)$, is defined by the manipulated object and a description of the set of active constraints between it and all other objects in the environment, including the remote manipulator. The relative positions and velocities at the constraints may also be included. The manipulated object as well as the number of active constraints can change with time. At a particular time, the state $x(t)$ does not include a description of contacts between other objects in workspace. Knowledge of the state is typically necessary for property estimation.

A related definition is needed for describing the steps to solving the identification problem. A *subtask*, s, corresponds to a specific state ($s \approx x$), but also connotes the relative position of the state within the sequence making up the task. In the preceding pick-and-place example, "Pickup" and "Place" are both subtasks which correspond to "hold-on-ground", the state in which the robot is gripping the block and the constraint between the block and ground is active. For the application of real-time machine perception, state estimation provides a means of inferring the current subtask and thus providing subtask-specific feedback to the operator.

Properties estimated in the identification problem are those of the manipulated objects and of those objects with which they have contact. The latter class includes the remote manipulator. With regard to the remediation examples cited above, such properties include shape, size, weight, mass distribution, stiffness and friction.

A model for the manipulated object (and contacting objects) can, in some cases, be derived from the task description. Alternatively, an object model can be inferred from visual and force/displacement data and updated as new data becomes available. Given a model, the properties will often correspond to specific model parameters. They can then be written as a vector of real numbers, $p \in \Re^m$. Their estimates, $\hat{p}(t)$ can be considered time dependent when estimation is performed on line. Often, a particular property can be estimated during a single specific subtask. As will be demonstrated in the case study, however, others may require sensor data from several subtasks.

Whether or not the desired object properties are specified as inputs may depend on the application. For a training system, the salient properties may be inferred to be all those necessary for virtual simulation of the manipulated objects. In the case of real-time machine perception, it may be desirable to specify those properties which are particularly hard for the operator to ascertain. Here, we assume that they are inputs and time independent.

2.1 Solution Procedure

The outputs of the identification problem are estimates of states, $\hat{x}(t)$, and the properties $\hat{p}(t)$. Note that property estimates can only be made during particular states or sets of states. For example, to estimate friction between two bodies, they must be in sliding contact. Thus, to estimate both states and properties, we need to identify portions of data stream $d(t)$ associated with particular states.

In some simple tasks, it may be possible to infer the state directly from the sensor signals (without regard to task context). If the previously described pick-and-place operation was performed quasistatically, for example, the state "hold-above-ground", corresponding to the subtask "Hold" (see Figure 1), might be identified by a positive grasp force and constant downward vertical force corresponding to the block weight.

In many instances, however, state identification in the data stream without regard to task context is impossible. Task context can provide information on the manipulated object (i.e., what block are we picking up?) as well as contact information (i.e., what constraints are active?). For example, when stacking a collection of blocks, the "hold-on-ground" state occurs many times. During a particular instance of this state, contextual information can indicate the identity of the manipulated block as well as that of the block or surface on which it is being held.

In summary, we see that property estimation requires estimates of the states. And state estimation requires contextual information which must be derived from the task description. Thus, the following decomposition of the identification problem into three subproblems is suggested.

1. Task decomposition - the process of resolving a task $\mathcal{T}$ into a minimal sequence of subtasks, s_i, described by contact states and their associated sets of properties, $\mathcal{T} = \{s_1 = (x_1, p_1), s_2 = (x_2, p_2), \ldots, s_r = (x_r, p_r)\}$ where $p_i,\ i = 1, \ldots, r$ are subsets of

p. Note that an individual state and/or property may be associated with multiple subtasks.

2. Data segmentation - Given a task decomposition $\mathcal{T} = \{s_1, s_2, \ldots, s_r\}$ and the sensor data stream $d(t)$, find the time intervals, $\{(t_{1,i}, t_{1,f}), (t_{2,i}, t_{2,f}), \ldots (t_{r,i}, t_{r,f})\}$ corresponding to each subtask. To allow for sensor noise as well as unanticipated states, it is not required that $t_{j,f} = t_{j+1,i}$. Since there is uncertainty in determining the time intervals, the j^{th} interval provides only estimates of the subtask and state, $\hat{s}_j$ and $\hat{x}_j$, respectively. Data segmentation may be performed either on or off line. If performed on line, the estimates can be expressed as functions of time.

3. Property estimation - Given the time intervals $\{(t_{1,i}, t_{1,f}), (t_{2,i}, t_{2,f}), \ldots (t_{r,i}, t_{r,f})\}$ associated with each subtask, estimate the desired properties, $\hat{p}$. If this procedure is carried out on line, the properties can be written as $\hat{p}(t)$.

Figure 2 depicts the concept of the automatic environment identification system. As a normal teleoperated task is performed, the system collects such data as task descriptions and desired properties from the operator. The resulting forces and motions are received from sensors in the remote environment. Visual feedback, as well as interaction with the human operator and data segmentation module, may also be necessary to efficiently decompose the task into its constituent parts. Based on the deduced sequence of subtasks, the data segmentation module associates subtasks (and thus states) with time segments of the data stream. The desired properties are then estimated and used to build and calibrate a model of the remote environment. As shown, the model can then be used either to provide immediate assistance to the operator or to form the basis of a training system.

The following subsections provide an overview of prior work on the individual sub-problems of environment identification. Individually, task decomposition has received limited attention while data segmentation and parameter estimation have been studied extensively in other contexts. The application of these techniques to remote environment identification, however, has not previously been pursued. In this paper, our focus is on environment identification using only haptic feedback, i.e., forces and displacements. Thus, while significant benefits can be gained from vision, it will not be considered here.

2.2 Task Decomposition

While not extensively studied, task decomposition has been investigated from a variety of viewpoints according to the goal of the decomposition. The earliest work known to the authors is that of Kondoleon (1976) who analyzed ten common products and found they could all be assembled using twelve manufacturing subtasks. The goal in this case was to find the statistical occurrence of each task in assembly operations. Nevins and Whitney (1980) employed Kondolean's approach to task decomposition to estimate assembly cost. The estimate was computed by calculating the cost of performing each subtask and the frequency with which it occurred in an assembly operation.

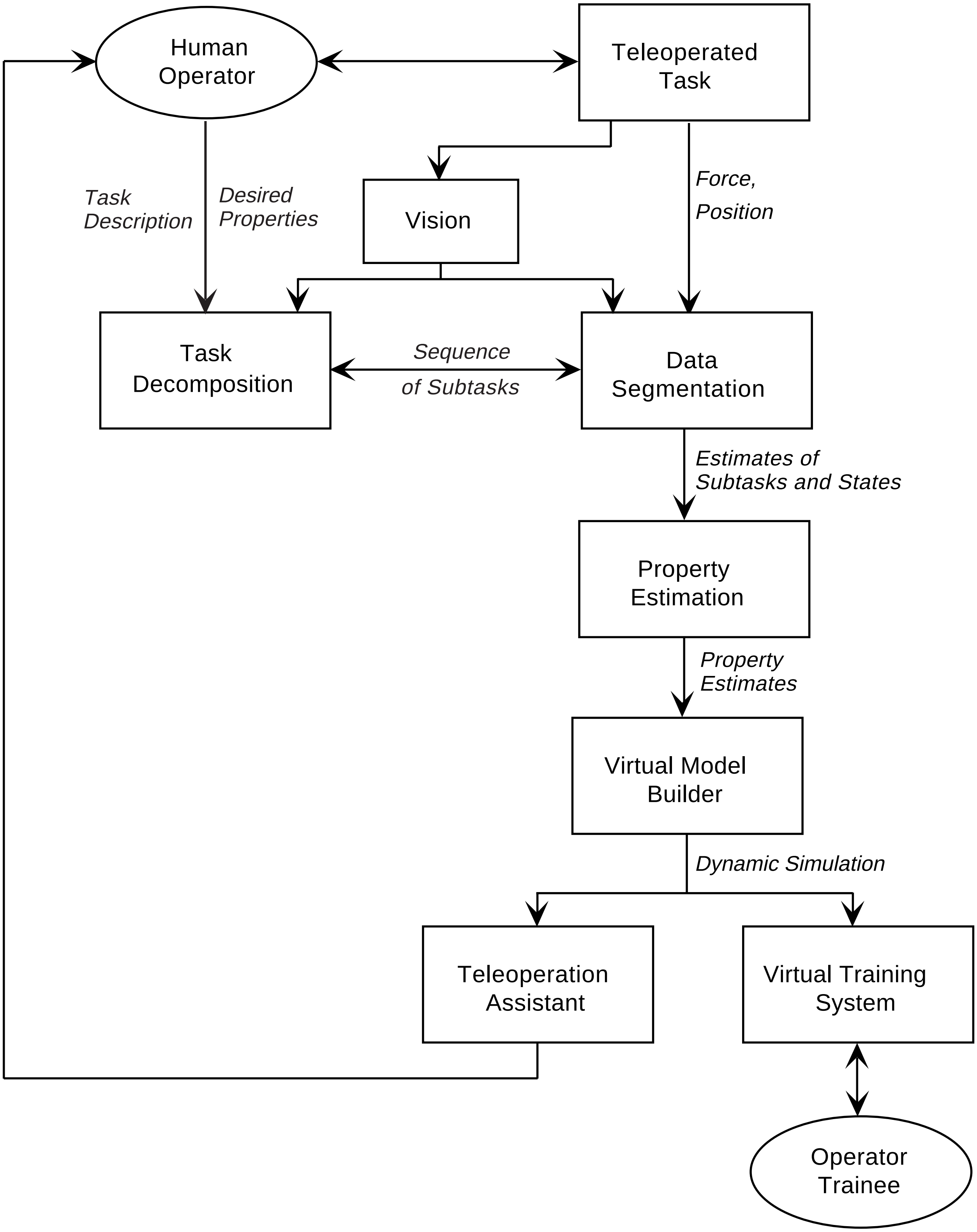


Figure 2: Flow diagram of the environment identification problem.

More recent work on task decomposition has been motivated by an interest in subtask-specific control strategies. McCarragher (1994a) divided tasks according to contact states in order to understand subtask control strategies employed by humans when successfully completing a task. To implement the on-line detection of switching times for subtask-specific control laws, Cutkosky and Hyde (1993) divided a task into phases separated by key transition events. McCarragher and Asada (1993) used contact states for recognition of state transitions of assembly processes.

2.3 Data Segmentation

Segmentation of manipulation data has received considerable attention for skill transference from humans to autonomous robots. For example, Pook and Ballard (1993) employed data segmentation in order to understand the qualitative control characteristics of an example task performed on a teleoperated system. Kang and Ikeuchi (1993) were interested in assembly task programming and used segmented data from a grasp task for the purpose of understanding the grasp motions. Yang et al. (1994) segmented data for similar purposes. Delson and West (1996) used human demonstration to program robots and in the process had to segment the data into subtasks that facilitated the generation of a robot program.

Segmentation methods described in the literature include hidden Markov models (Hannaford and Lee, 1991), qualitative reasoning with thresholding (McCarragher, 1994a), neural networks for off-line segmentation (Fiorini et al., 1992), Petri nets (McCarragher, 1994b) and statistical estimation theory (Eberman and Salisbury, 1994; Eberman, 1997).

Qualitative reasoning is employed in this paper to segment data from a two-fingered telemanipulator. In this approach, developed by McCarragher (1994a), position, velocity and force data is thresholded to minimize the effect of signal noise. The thresholded values are then combined in Boolean expressions to determine the current state from a variety of possible states. These states are then associated with subtasks according to the task decomposition. While not the most general approach to data segmentation, qualitative reasoning allows us to efficiently demonstrate the issues involved in data segmentation.

2.4 Property Estimation

Most work on property estimation assumes a parameterized model, e.g., a geometric or contact force model, and so is more properly termed parameter estimation. An extensive body of research on parameter estimation exists. See, for example, (Ljung, 1987).

Within this body, a significant portion is devoted to robot parameter identification. For example, the identification of link inertial parameters has been studied by Khosla and Kanade (1985) and An et al. (1985). Others have investigated the identification of kinematic parameters (Driels, 1993). In addition, a few authors have addressed identification of robot payload and environment properties. Methods for estimating payload inertia appear in the work of Atkeson et al. (1985) and Lin and Yae (1992). Lin and Yae also estimate certain parameters relating to constraints of the operating environment. They do assume, however, that the constraint surface is at least partially known and that it can be modeled using a set of known and unknown parameters. Still others have studied how to determine shapes of objects in the operating environment by using haptic exploration (Allen, 1990; Okamura, 1997).

In this literature, the kinematic constraints of the robot and payload are specified a priori and are used explicitly in the identification process. When modeling an unstructured environment, however, constraints may not be known in advance. In many cases, though, their identification is essential because they reveal functional relationships among objects in the environment. Identifying such constraints from manipulation data presents a new and difficult challenge. While this topic will not be discussed further here, additional information regarding constraint existence and modeling can be found in (Dupont et al., 1997), and regarding constraint parameter identification in (Bruyninckx, 1995).

3 Automatic Identification – a Case Study

Pick-and-place tasks are a convenient starting point for the study of automatic identification techniques, because the grasping and lifting actions that comprise these tasks are an essential part of many telemanipulation operations. These tasks are also amenable to automatic identification since the properties of interest and subtasks are readily defined. To include interactions between objects in the remote environment, a block stacking task is analyzed here. The identification problem is defined as follows.

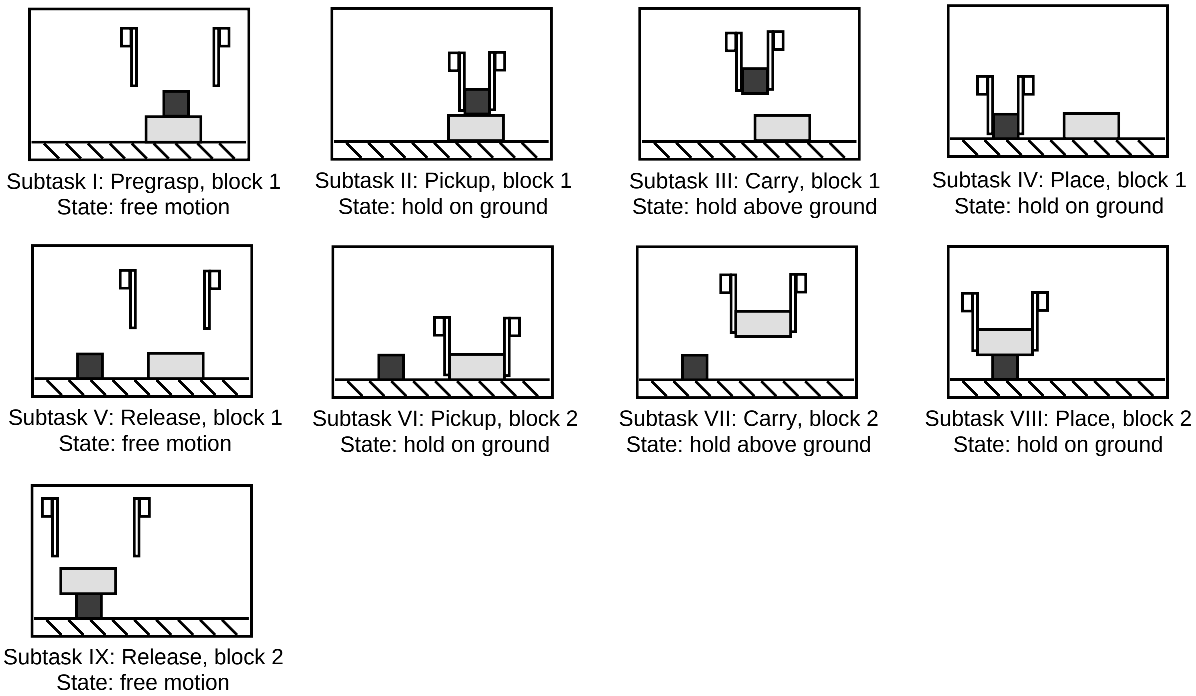
Task Description: Using a teleoperated hand system, reverse the positions of two rectangular aluminum blocks which are stacked one atop the other on a horizontal planar surface. The operator will move the top block off of the stack and onto the ground, and then place the other block on top of it. Figure 3 shows the anticipated progression of the task.

Properties: The remote environment properties to be estimated are the height, weight, width and friction coefficients of the blocks.

Sensor Data Stream: The remote manipulator's joint angles, velocities and fingertip forces are measured.

3.1 Teleoperated Hand System

A planar, two-fingered teleoperated hand system with finger tip force feedback was used to perform the block stacking task (Howe, 1992). This system trades a limitation on the number of joints for a clean and simple mechanical design. The system has high bandwidth and large dynamic range, which permits accurate control of contact forces and small motions. The system is designed to execute tasks that humans usually accomplish with a precision pinch grasp between the thumb and index finger. For most tasks, the operator's wrist rests



Subtask I: Pregrasp, block 1
State: free motion

Subtask II: Pickup, block 1
State: hold on ground

Subtask III: Carry, block 1
State: hold above ground

Subtask IV: Place, block 1
State: hold on ground

Subtask V: Release, block 1
State: free motion

Subtask VI: Pickup, block 2
State: hold on ground

Subtask VII: Carry, block 2
State: hold above ground

Subtask VIII: Place, block 2
State: hold on ground

Subtask IX: Release, block 2
State: free motion

Figure 3: Block stacking task.

on the table top and the operator makes contact with the master only at the tips of the fingers (Figure 4). The master and remote manipulators are kinematically identical, with two degrees of freedom in each finger, so finger tip position or force can be controlled within the vertical plane. The workspace is roughly circular and 75 mm in diameter. Parallelogram linkages maintain a constant vertical orientation of the finger tips, which precludes inappropriate torques on the operator's finger tips as the joints rotate. Two-axis strain gauge force sensors measure finger tip forces on both master and remote manipulator hands.

The controller uses a conventional bilateral force reflection control scheme. The joint angles of the master manipulator are the command inputs for position control of the joints of the remote manipulator, and the forces measured at the remote manipulator finger tips are the command inputs for force control of the master. Each finger is capable of applying a continuous tip force of at least 4 N. Flat, thin finger tips extending downward are mounted on the two remote manipulator fingers to facilitate manipulation of the rectangular blocks used in the experiments (Figure 4b). The manipulator finger tips were covered with a 2 mm layer of closed-cell foam rubber to increase compliance and friction.

The trials for this case study were performed by one of the authors after practicing sufficiently to become proficient at the task. Six sensor signals were recorded for each of the two remote manipulator fingers: two joint angles, two joint velocities, and horizontal and vertical components of finger tip force. The signals were collected at 50 ms intervals during the course of the task, for a total of 10 seconds. The forward kinematic relations permitted calculation

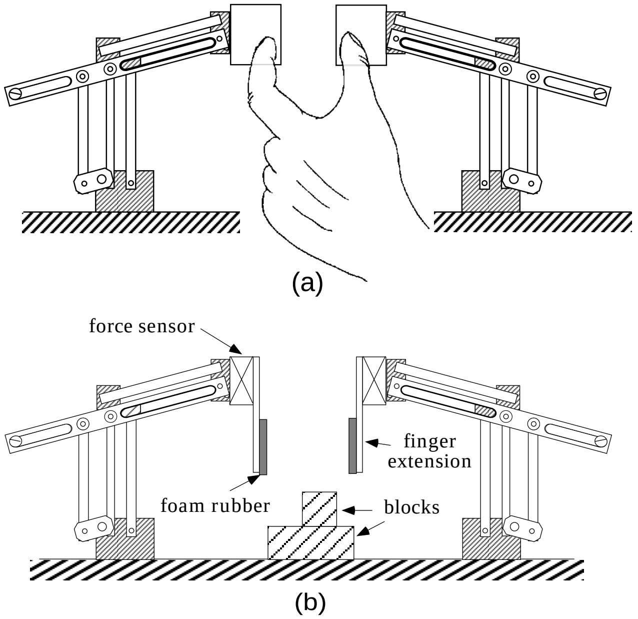


Figure 4: (a) Master manipulator with operator's hand. (b) Remote manipulator.

of endpoint position and velocity of each finger, as shown in the subsequent figures. In the following sections, particular solutions to the sub-problems of task decomposition, data segmentation, and property estimation are presented.

3.2 Task Decomposition

To relate the force and displacement data streams to the properties of interest, the task $\mathcal{T}$ must be divided into subtasks $s_i,\ i = 1 \ldots r$, consisting of contact states x_i and their property sets p_i to obtain $\mathcal{T} = \{s_1 = (x_1, p_1), s_2 = (x_2, p_2), \ldots, s_r = (x_r, p_r)\}$. To realize such a decomposition, the task description is first discretized based on each anticipated change in the active constraint set of the robot or of any object it is manipulating. (Recall Figure 3.) This provides a sequentially arranged set of potential states. A quasistatic or dynamic analysis of each contact state is then performed to solve for the associated property set (possibly null).

Table 1 depicts the results of this analysis for the first four task states. The column entries corresponding to specific environment properties indicate the requisite sensor signals. Note that the properties weight and width of block 1 can only be estimated in a single state, "hold-above-ground 1" while friction can be estimated in any of the last three states. In contrast, sensor data from both the second and fourth state is needed to estimate the height of block 2. Properties of the second block (and the height of the first block) can be estimated during the last five subtasks.

To obtain a minimal sequence of subtasks, it may be possible to eliminate certain states

Table 1: Relevant Signals and System States for Estimation of Object Properties During the First Four Subtasks

		Property			
Subtask	State	Weight Block 1	Width Block 1	Height Block 2	Friction of Fingers Against Block 1
Pregrasp, Block 1	free-motion	-	-	-	-
Pickup, Block 1	hold-on-ground 1	-	-	vertical positions	horiz. and vertical forces
Carry, Block 1	hold-above-ground 1	vertical forces	horizontal positions	-	horizontal and vertical forces
Place Block 1	hold-on-ground 1	-	-	vertical positions	horizontal and vertical forces

which do not contribute to the estimation of any properties. This would seem to be the case with the first state, "free-motion", which also repeats as the fifth state. As will become clear in the discussion of data segmentation, however, this state serves to delineate the temporal boundaries of states which do relate directly to property estimation.

The complete set of subtasks and states is shown in Figure 3. The task decomposition can be expressed as

$$\mathcal{T} = \{ \begin{array}{l} \text{Pregrasp } 1 = (\text{free-motion}, \{\emptyset\}), \\ \text{Pickup } 1 = (\text{hold-on-ground } 1, \{\text{height } 2, \text{friction } 1\}), \\ \text{Carry } 1 = (\text{hold-above-ground } 1, \{\text{weight } 1, \text{width } 1, \text{friction } 1\}), \\ \text{Place } 1 = (\text{hold-on-ground } 1, \{\text{height } 2, \text{friction } 1\}), \\ \ldots, \\ \text{Release } 2 = (\text{free-motion}, \{\emptyset\}) \ \} \end{array} \tag{1}$$

3.3 Data Segmentation

Given the task decomposition described by equation 1 and Figure 3, data segmentation involves assigning time intervals for each subtask to the data stream. The implemented algorithm proceeded in three stages, similar to the qualitative reasoning approach that has been applied to the analysis of sensor signals from human task performance (McCarragher, 1994a). In the first stage, the measured motions and forces at the finger tips were transformed into task coordinates and thresholded. The resulting threshold functions were tri-valued, that is, they were assigned values of "+", "0", or "-". In the second stage, the thresholded data were combined using Boolean operators to assign state labels at each sampling time. Once the states were identified, additional processing identified the subtasks. Here we describe the details of the state and subtask identification procedures.

Because these experiments used a multifingered hand, the finger tip motion and force data must be transformed to find the net motion and force of the grasped object in task space. This requires computation of averages and differences of the horizontal (x) and vertical (y) components of the two finger motions. The average motion of the two fingers is a measure of rigid body motion, and the difference in finger tip motions corresponds to gripping motion. Horizontal and vertical components of the finger tip forces were similarly transformed. The sum of the forces from the two fingers was the net force applied to the environment (including the block and the ground), and grip force was the minimum horizontal component of the two (opposing) finger tip forces. Because each finger of our system has only two degrees of freedom and flat finger tips, and because the objects were flat-sided blocks, the kinematic transformations of endpoint forces and motions to task space was trivial. In general, these transformations for multifingered hands require the use of robotic grasp analysis (Kerr and Roth, 1986).

Once the task motions and forces were obtained, the velocity and force data were passed through thresholding filters with equal positive and negative thresholds. The velocity thresholds were ± 5 cm/sec, and the force thresholds were ± 0.05 N. Position and force data for one trial of the stacking task are shown below in Figures 5 and 6. The progression of the task can be discerned in these figures. The side to side pattern of motion can be seen in Figure 5a, up and down motion in Figure 5b, and grasp and release in Figure 5c. The vertical offset of the two fingers (Figure 5d) is not of interest in this task, but has been included for completeness. The forces applied to the environment are shown in Figure 6. Horizontal forces (Figure 6a) are generally zero during the carry and free-motion subtasks, and non-zero during pickup and placement subtasks. Net vertical force (Figure 6b) is also zero during free-motion, but positive during carry subtasks, and negative or zero during pickup and placement. The holding subtasks are clearly visible in Figure 6c as large, positive grip forces.

These correlations between sensor data and task states were formalized into rules for automatically identifying the various states, as summarized in Table 2. Free-motion was defined as any time that both components of the force on both finger tips were zero. The hold-on-ground state was active whenever the grasp force was positive (i.e., greater than the positive threshold) and the total vertical force exerted by the fingers was not upward (i.e., negative or zero) and the average vertical velocity of the fingers was near zero. The sliding-on-ground state is the same as hold-on-ground, with the added condition that the average horizontal velocity was non-zero. Hold-above-ground was active when the grasp force was positive and the sum of the vertical forces was upward (positive). Note that this criterion for the hold-above-ground state can include a brief period just before and after the block lifts, when the vertical force is positive but the block is still in contact with the ground. This is unavoidable if the state identification algorithm uses only instantaneous data without reference to the prior state of the system. This approach proved adequate for this simple task.

Data for several trials of the block-stacking task were processed using the automatic state identification procedure. The segmentation algorithm divided the data into sections which

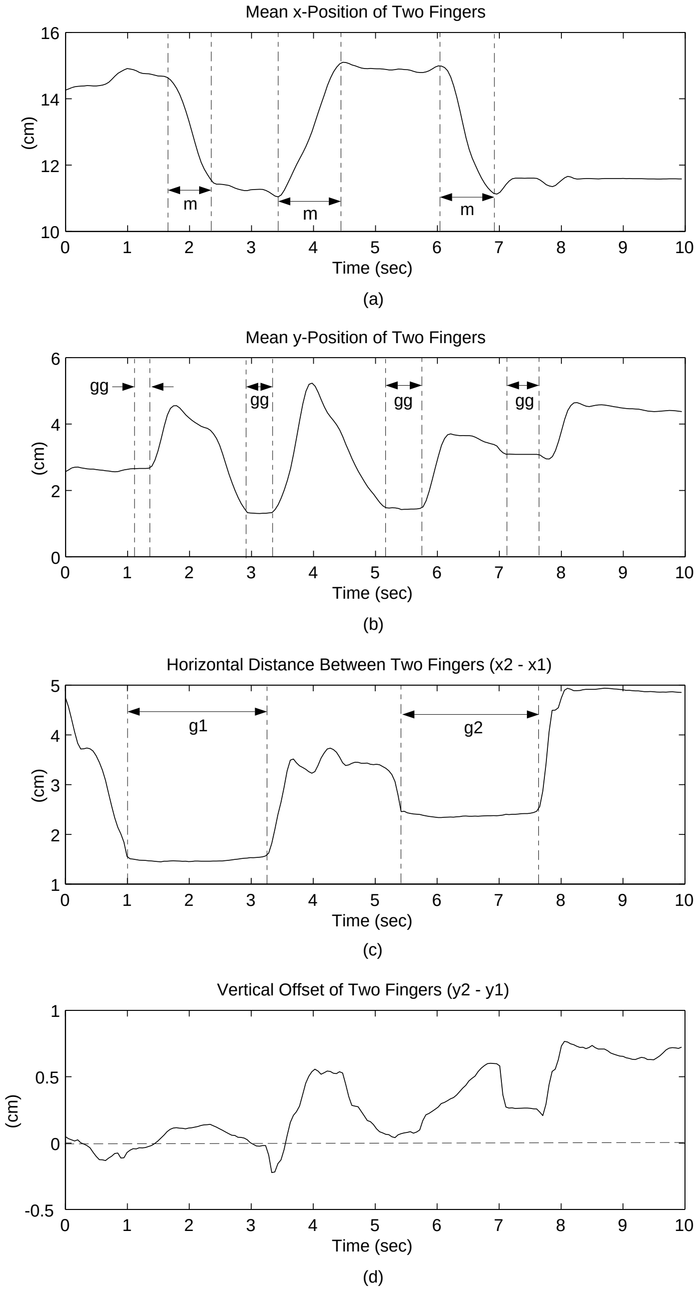
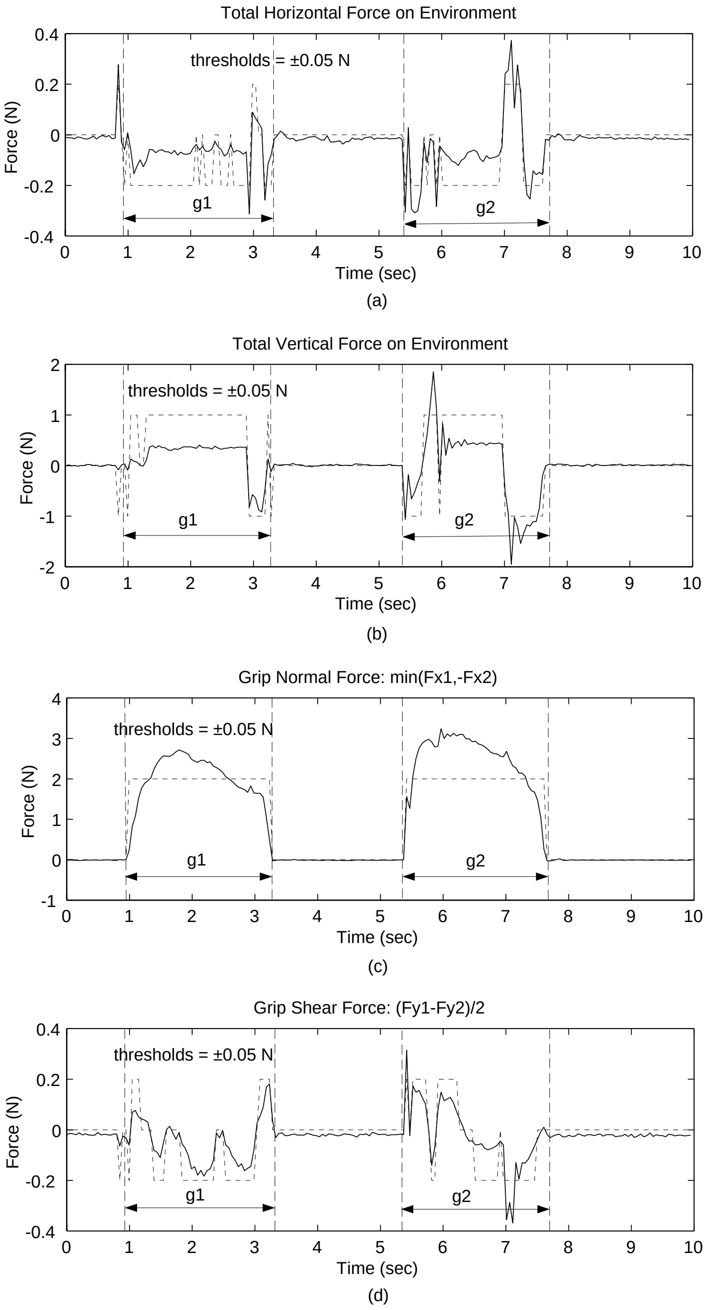


Figure 5: Horizontal and vertical positions of the two fingers. Legend: m = Movement of fingers to a new position; gg = Grasping an object on the ground; g1 = Grasping object 1; g2 = grasping object 2.



(a)

(b)

(c)

(d)

Figure 6: Forces: (a) Total horizontal force on environment; (b) Total vertical force on environment; (c) Grip force; (d) Vertical shear force. Legend: g1 = Grasping object 1; g2 = Grasping object 2.

Table 2: Qualitative Values of Sensed Parameters Which Define Task States

	Sensed Parameters					
State	Vx_ avg	Vy_ avg	Fx_ avg	Fy_ avg	Fx_ grip	Fy_ shear
free-motion			0	0	0	0
hold-on-ground		0		- or 0	+	
hold-on-ground, sliding	+ or -	0		- or 0	+	
hold-above-ground				+	+	

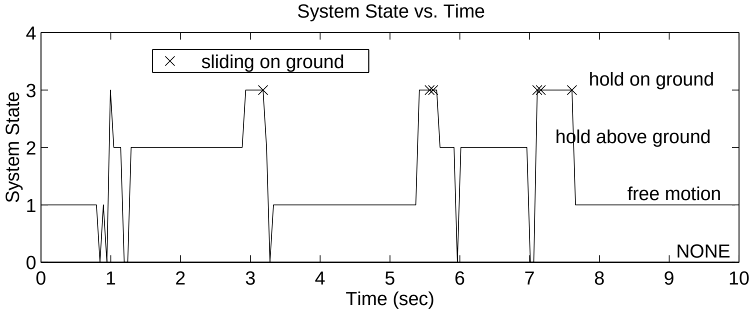


Figure 7: Result of automatic system state identification procedure.

corresponded closely to those selected by hand. For these data sets, the results of thresholding were not especially sensitive to the threshold values, although selecting appropriate thresholds can be difficult when noisy manipulators and a larger range of tasks and operators are involved (Hannaford and Lee, 1991).

Figure 7 shows the results of applying the procedure to the data in the preceding figures. The value of the plotted function denotes the identified state of the system at that time. A value of 1 corresponds to free-motion, a value of 2 to hold-above-ground, and a value of 3 to hold-on-ground. Times at which sliding on the ground was identified are labeled with an “x”. A value of zero indicates that none of the four possible states was identified. Accordingly, data from these times were not used in the identification of any object properties. These “dropouts” are caused mainly by transients in the signals just after lift-off or impact of the blocks. While these unidentified points are not desirable, they are not a problem as long as at least some data samples are successfully identified for each subtask. Using this convention, the expected (and observed) progression of states for this task was: 1, 3, 2, 3, 1, 3, 2, 3, 1. Subtasks I-IX were assigned to the data stream from the state information using the list of the anticipated order of the states.

A weakness of the simple segmentation algorithm used for this study is that it could potentially oscillate between the hold-on-ground state and the hold-above-ground state if the manipulated objects did not make and break contact cleanly. Although the data collected for this example did not exhibit this complication, future segmentation algorithms should

Table 3: Methods of Property Estimation

Property	*Subtask*	*State*	*Formula for Estimate*
Weight (Block 1)	Subtask III Carry Block 1	hold-above-ground 1	$W = \Sigma \frac{F_{y1}+F_{y2}}{N_{III}}$ (Average sum of vertical forces for finger1 and finger2)
Height (Block 2) Part a:	Subtask II Pick up Block 1	hold-on-ground 1	$y_{II} = \Sigma \frac{y_2}{N_{III}}$ (Average vertical position of right finger)
Part b:	Subtask IV Place Block 1	hold-on-ground 1	$y_{IV} = \Sigma \frac{y_2}{N_{IV}}$ (Average vertical position of right finger)
Final:			$h = \|y_{II} - y_{IV}\|$ (Absolute value of the difference in average height)
Width (Block 1)	Subtask III Carry Block 1	hold-above-ground 1	$w = \Sigma \frac{x_2-x_1}{N_{III}}$ (average difference in horizontal position)
μ (Block 1)	Subtasks II-IV Pick up Block 1, Carry Block 1, and Place Block 1	hold-on-ground 1 and hold-above-ground 1	$\mu \geq max\|\frac{F_{y1}}{F_{x1}}\|$ (Max. absolute value of vertical force over horizontal force)

be robust to this problem.

3.4 Property Estimation

Block weight, height, width, and μ, the coefficient of static friction between the block and the fingers of the robot hand, were estimated from the data. Table 3 lists each property along with the associated subtasks and estimation formulas. Because our goal is to outline the steps of the identification process and illustrate the key issues, no attempt has been made to find an optimal estimator. This table describes object property estimation during manipulation of the first block (Subtasks II-IV); for estimation of the analogous properties during manipulation of the second block (Subtasks IV-VIII), each instance of "block 1" in the table is simply replaced by "block 2" and vice versa.

Ideally, the weight of block 1 was the sum of the vertical forces measured at each finger tip during the "carry block 1" subtask. The estimate was the average sum over all samples in the subtask (i.e. over N_{III} , the number of samples in subtask III). The weight estimate

Table 4: Comparison of Estimated (mean and standard deviation) and Actual Environment Parameters

	Block 1		*Block 2*	
Property	*Estimated Value*	*Actual Value*	*Estimated Value*	*Actual Value*
Weight	0.352 ± 0.024 N	0.343 ± 0.0005 N	0.425 N ± 0.071 N	0.437 ± 0.0005 N
Height	1.75 ± 0.01 cm	1.60 ± 0.01 cm	1.29 ± 0.02 cm	1.27 ± 0.01 cm
Width	1.47 ± 0.01 cm	1.59 ± 0.01 cm	2.36 ± 0.01 cm	2.54 ± 0.01 cm
Lower Bound on μ	0.49	-	1.06	-

is sensitive to segmentation boundaries, due to noise during the transition period between subtasks. Therefore, the weight estimate arbitrarily excluded force data during the first and last 5% of the carry subtask.

The width of each block is simply the average horizontal distance between the finger tips over the entire carry subtask. In contrast, the height estimate uses interactions between objects in the remote environment to determine the desired parameters. This requires multiple estimation steps because it is formed from two discrete measurements. For example, the height of block 2 is found from the difference between the vertical position of the fingers when block 1 is lifted from atop block 2 and when it (block 1) is subsequently placed on the table. Similarly, moving block 2 gives the height of block 1.

Note that for these experiments, we assume static friction can be modeled as a single constant μ. Ideally, μ would be measured at the onset of slip, but since slippage of the blocks between the fingers was not detectable, only a lower bound of μ was obtained. If slip did occur, the maximum value of μ recorded should correspond to the coefficient of static friction, recorded at the onset of slip. In many telemanipulation tasks operators expressly avoid slips, so accurate determination of μ could require a special test procedure.

3.5 Results

The estimated parameters for one trial of the block-stacking task are listed in Table 4, along with actual values, as measured by a laboratory balance and calipers. Applying the algorithm to a different trial of the block stacking task gave similar estimates of the parameters. Certain trials exhibited significant vibrations following state transitions suggesting the need for data filtering prior to segmentation.

Figures 8 - 9 are plots of the measurements used to estimate object weight and height. The portions of the data that were used in the estimates are indicated by solid lines. Figure 8 shows the force measurements used to find the average weight during the hold subtask. Note that there are two intervals of measurements, indicating that two objects were held. The large oscillations that occur during the second interval are due to inertial forces and

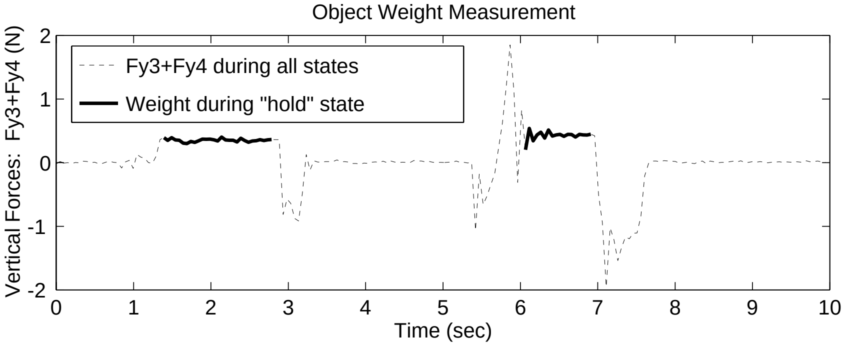


Figure 8: Object weights. The solid portions are the segments of data used for weight estimatio-

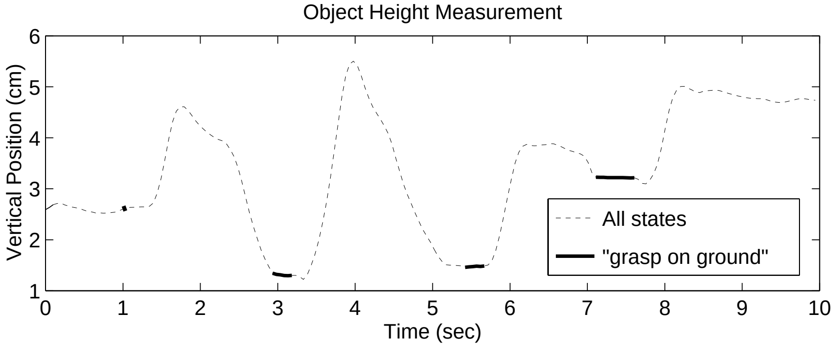


Figure 9: Object Heights. The solid portions are the segments of data used for height estimation. Object 2 height is the difference between the first and second solid lines, while object 1 height is the difference between the third and fourth solid lines.

the object impacting the surface. The average estimates of the weights are within 3% of the actual values. The standard deviation is relatively large for block 2 due to the oscillations in the force signals.

The first two solid lines in Figure 9 show the heights during the two hold-on-ground states, when block 1 is moved from the top of block 2 to the ground. The difference in these heights gives the height of the block originally beneath it. Likewise, the last set of two solid lines represents grasping and moving block 2 from the ground to the top of block 1. Notice that the intervals for these subtasks are fairly short, but even though there are only a few points in the interval, the estimated heights of blocks 1 and 2 are within 10% and 2% of their respective measured dimensions. The estimated height of block 1 shows greater error, perhaps due to slipping of the block within the fingers at the instant when the block makes contact with the ground. The errors in the estimates of the widths of the blocks were approximately 7%.

4 Calibration of Virtual Model

A virtual block was calibrated using the weight, height, and width parameters identified for one of the real blocks (block 2). Figure 10 depicts the teleoperated hand during block

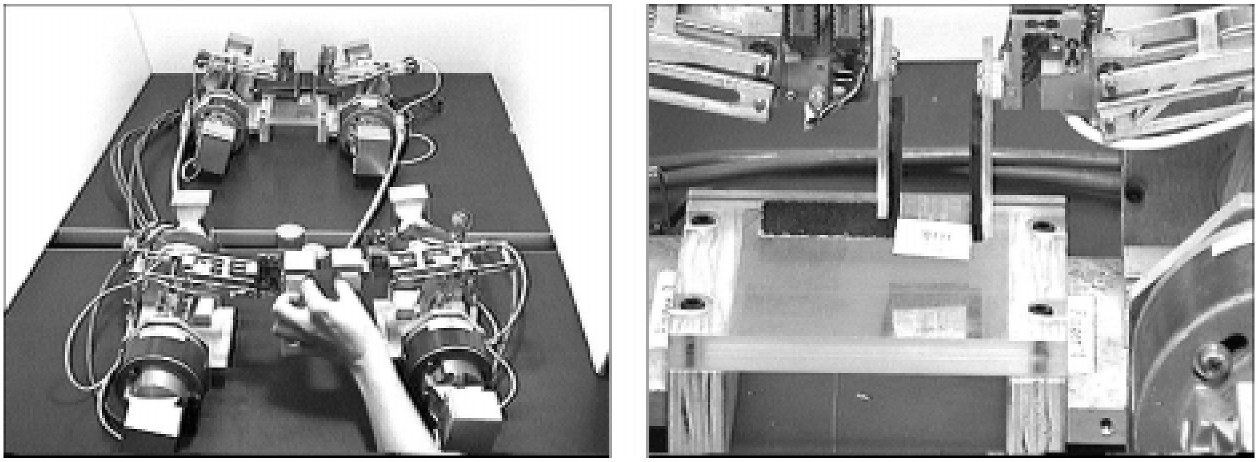


Figure 10: Master and Remote Manipulators (left) and Close-up of Remote Manipulator (right) During Block Calibration.

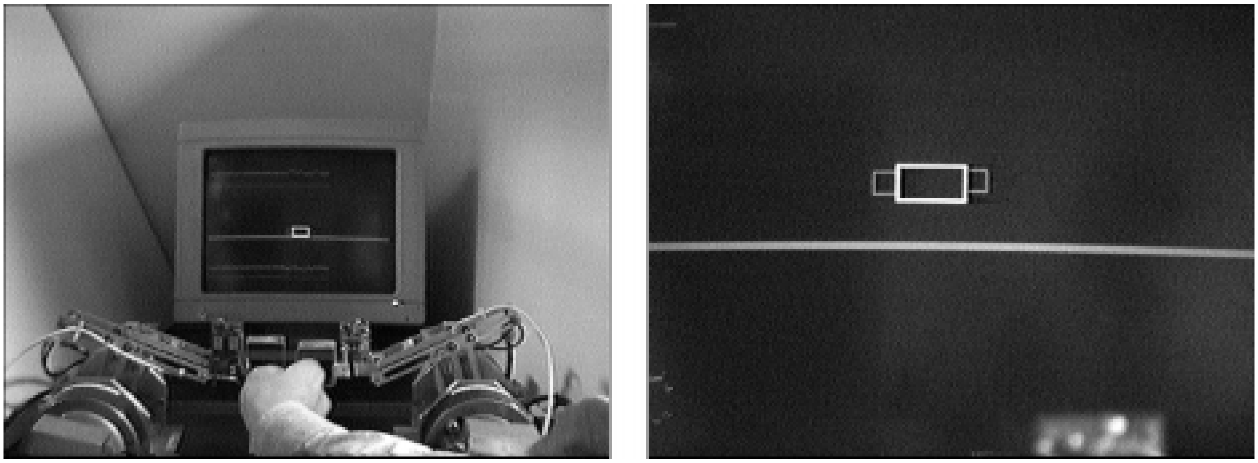


Figure 11: Virtual Simulation of Calibrated Block.

calibration. A dynamic simulation of the the calibrated block was then implemented using as input/output devices the master manipulator and a video monitor. (See Figure 11). As the operator moves the fingers on the master, the simulation displays this finger motion on the computer screen. The virtual block appears on the screen with the calibrated width and height of real block 2. As the virtual fingers contact the virtual block, reaction forces are computed and applied through the master manipulator to the operator's fingers. The operator was able to lift and maneuver the block around the screen in the same way the real block had been manipulated. Since the visual and force feedback were calibrated to the measured interactions between the actual fingers and block, the operator felt as though he were manipulating a block very similar in size, shape, and mass to the real block.

The contact and reaction forces were calculated using the relative velocities and positions of the fingers and block. Calibrated block mass was used to compute its position and velocity through Euler integration of the dynamic equation. The virtual forces computed during a typical simulation trial are depicted in Figure 12. In this trial, virtual block 2 is picked up and placed twice. These plots can be compared with the measured data of Figure 6 which depicts the sequential pick up and placement of blocks 1 and 2. The similarity of the force plots shows that the operator experienced comparable forces during interactions with the virtual block and the real blocks.

While the grip normal force is smaller for the virtual block, this is simply because the operator did not happen to squeeze as hard while manipulating the virtual block. Similarly, variation in the virtual grip shear force is less pronounced. In this case, the operator intentionally avoided applying shear moments to the virtual block as they tended to produce instability. This instability is the result of a stick-slip condition that occurs as one finger pushes up and the other pushes down. It is important to note that the virtual block was picked up twice during the time interval during which data was collected and it corresponds to the second block that was manipulated during interactions with the real blocks.

As a means of comparing estimated and simulated parameter values, data plots depicting the weight and width of the virtual block are shown in Figures 13 and 14, respectively. The virtual block weight of Figure 13 compares favorably with block 2 in Figure 8. In Figure 14, the actual width of block 2 in (a) agrees with both width measurements of the virtual block in (b).

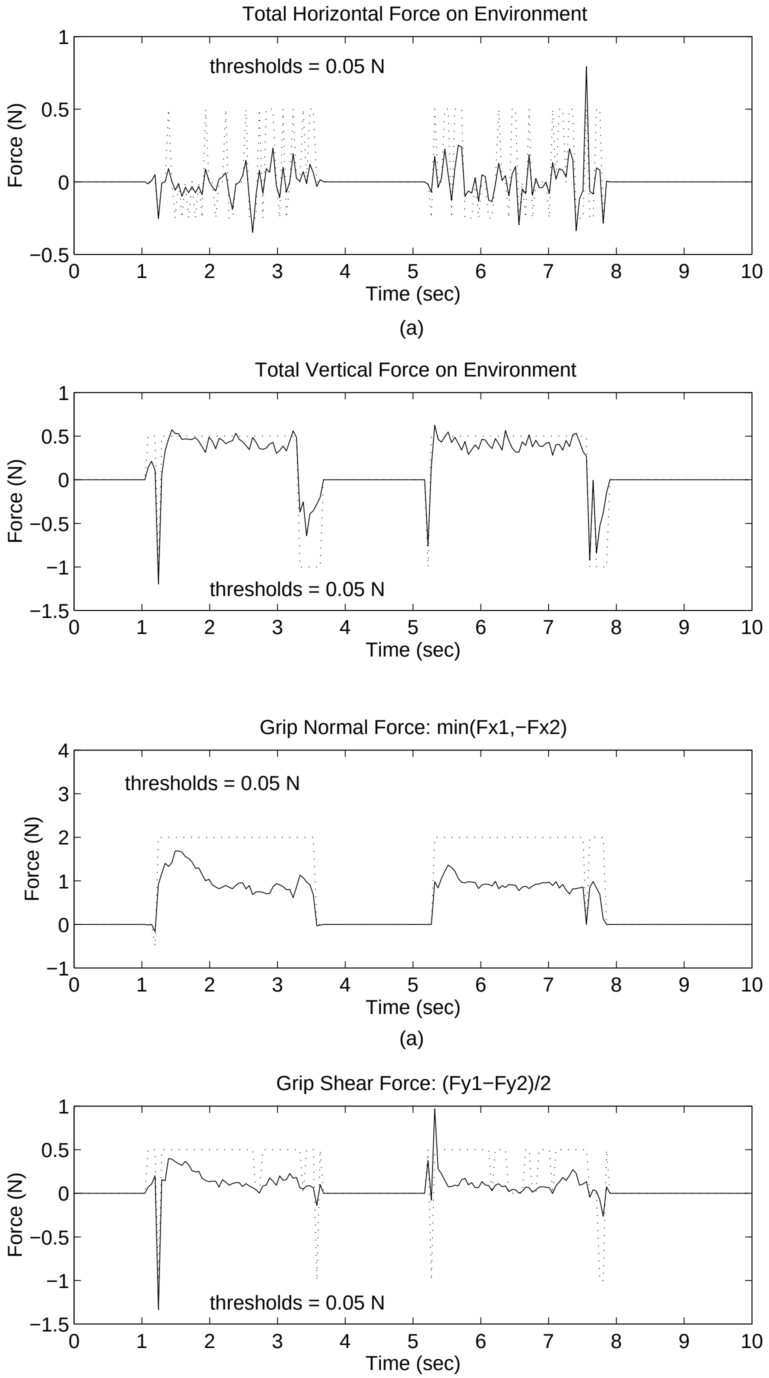
The virtual model simulation described here was limited to a single block due to limitations imposed by our particular computer software and hardware configuration. Note that these limitations relate to dynamic simulation of the virtual model rather than to environment identification. Nonetheless, this demonstration makes clear the utility and value of automatic environment identification. Currently, this work is being extended to several more sophisticated teleoperator systems with expanded dynamic simulation capability.

5 Conclusions

The automatic environment identification problem defined here has two promising applications. During complicated tasks in unstructured environments, real-time estimates of object properties could enable a human operator to perform these tasks with greater safety and efficiency. As demonstrated in the previous section, such a system would also help to automate the construction and calibration of environment models for operator training systems.

Note that while automatic property identification has been illustrated for a teleoperated task, the approach would apply equally well to systems in which there is no remote manipulator, i.e., the master manipulator is in contact with the environment. For example, properly instrumented surgical tools could constitute the master manipulator. As with a teleoperator, identified environment properties might be used either for operator assistance or training system construction.

Through careful problem definition as well as the analysis of a block stacking task, automatic property identification has been shown to consist of three principal subproblems: task de-



(a)

(a)

Figure 12: Virtual forces: (a) Total horizontal force on environment; (b) Total vertical force on environment; (c) Grip force; (d) Vertical shear force.

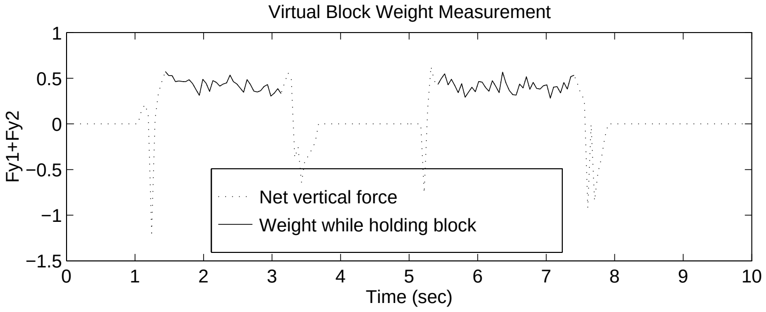


Figure 13: Virtual Block Weight Measurement. The solid portions are the intervals during which the virtual block is being held above ground.

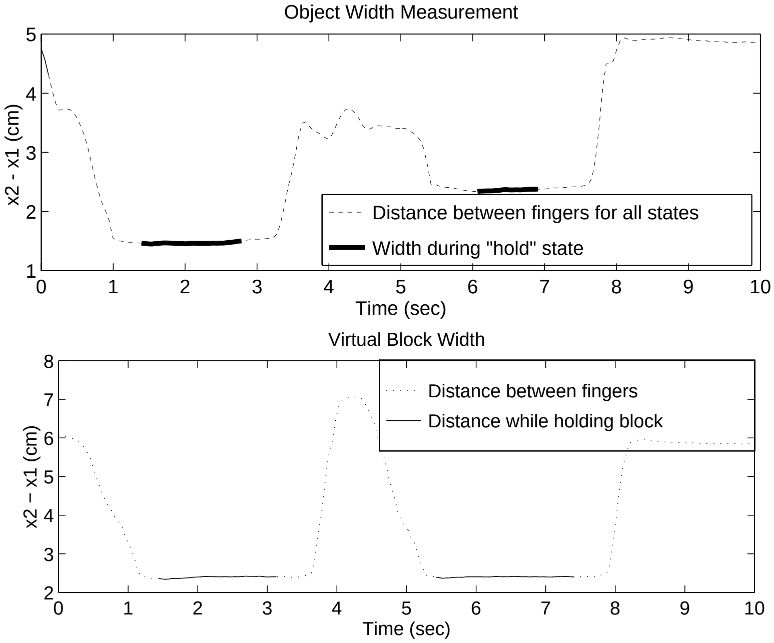


Figure 14: Block Width Measurement. (a) Actual blocks 1 and 2; (b) Virtual block 2. The solid portions correspond to “hold-above-ground” states.

composition, data segmentation, and property estimation. While specific solutions to these subproblems have been demonstrated for the example task, significant additional research is needed in each area to achieve an identification system of sufficient robustness and generality.

Specifically, an algorithmic approach to task decomposition should be developed that relates sensor output and environment properties to task-description-derived states. This approach should allow for environment objects which do not appear in the task description. Furthermore, practical data segmentation calls for a technique which is robust to variations in task performance and to sensor transients induced by contact state transitions. The application of results from decision and estimation theory holds promise here.

A methodology for property estimation is also needed, which relates desired properties to parameters based on iterative parametric modeling of environment objects. Such models should be physically-based whenever possible and selected in accord with data stream history as well as simulation efficiency.

A final research topic is to define the appropriate level of interaction between the user and each component of the identification system, and to study the behavior of the resulting system. For example, object modeling as part of property estimation may suggest modifications to the task decomposition. This in turn may lead to data segmentation ambiguities which call for operator resolution. These issues are currently under study.

6 References

Allen, P. K. (1990). Mapping Haptic Exploratory Procedures to Multiple Shape Representations. *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, OH, *3*, May, 1679–1684.

An, C., Atkeson, C., and Hollerbach, J. (1985). Estimation of Inertial Parameters of Rigid Body Links of Manipulators. *Proc. 24th Conference on Decision and Control*, Ft. Lauderdale, FL, December, 990–995.

Atkeson, C., An, C., and Hollerbach, J. (1985). Rigid Body Load Identification for Manipulators. *Proceedings of the 24th Conference on Decision and Control*, Ft. Lauderdale, FL, *2*, December, 996–1002.

Bejczy, A. K., Kim, W. S., and Venema, S. C. (1990). The Phantom Robot: Predictive Displays for Teleoperation with Time Delay. *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, *1*, May, 546–551.

Bruyninckx, H. (1995). Kinematic Models for Robot Compliant Motion with Identification of Uncertainties. Doctoral Dissertation, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium, April.

Cutkosky, M. R. and Hyde, J. M. (1993). Manipulation Control with Dynamic Tactile Sensing. *6th International Symposium on Robotics Research*, Hidden Valley, PA, October.

Delson, N. and West, H. (1996). Segmentation of Task Into Subtasks for Robot Programming by Human Demonstration. *ASME Japan/USA Symposium on Flexible Automation*, Boston, MA, *1*, July, 41–47.

Driels, M. R. (1993). Using Passive End-Point Motion Constraints to Calibrate Robot Manipulators. *Transactions of the ASME*, *115*, September, 560–566.

Docimo, S., Moore, R., and Kavoussi, L. (1997). Telerobotic Surgery Is Clinical Reality: Current Experience with Telementoring in Adults and Children. *Presence*, *6*(2), April, 173–178.

Dupont, P., Schulteis, T., and Howe, R. (1997). Experimental Identification of Kinematic Constraints. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April, 2677–2682.

Eberman, B., and Salisbury, J. K. (1994). Application of Change Detection to Dynamic Contact Sensing. *International Journal of Robotics Research*, *13*(5), October, 369–394.

Eberman, B. (1997). A Model-Based Approach to Cartesian Manipulation Contact Sensing. *International Journal of Robotics Research*, *16*(4), August, 508–528.

Fiorini, P., Losito, S., Giancaspro, A., and Pasquariello, G. (1992). Neural Networks for Off-line Segmentation of Teleoperation Tasks. *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, Glasgow, UK, *1*, August, 17–22.

Funda, J., and Paul, R. P. (1991). Efficient Control of a Robotic System for Time-Delayed Environments. *91 ICAR. Fifth International Conference on Advanced Robotics. Robots in Unstructured Environments.*, *1*, June, 219–224.

Fyler, D. (1981). Computer Graphic Representation of Remote Environments Using Position Tactile Sensors. SM thesis, MIT, August.

Griebenow, B.E. (1994). Buried-waste Integrated Demonstration Retrieval Projects. Annual Meeting of American Nuclear Society, New Orleans, LA, 19-23 June, *Trans. Am. Nucl. Soc.*, *70*, p. 402.

Hannaford, B., and Lee, P. (1991).Hidden Markov Modal Analysis of Force/Torque Information in Telemanipulation. *International Journal of Robotics Research*, *10*(5), October, 528–539.

Hannaford, B., Wood, L., Guggisberg, B., McAffee, D., and Zak, H. (1989). Performance Evaluation of a Six-Axis Generalized Force-Reflecting Teleoperator. JPL Publication 89-18, Pasadena, CA.

Hannaford, B., Wood, L., McAffee, D. A., and Zak, H. (1991). Performance Evaluation of a Six-Axis Generalized Force-Reflecting Teleoperator. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3), May/June, 620–633.

Howe, R. D. (1992). A Force-Reflecting Teleoperated Hand System for the Study of Tactile Sensing in Precision Manipulation. *Proceedings of the 1992 IEEE International Con-*

ference on Robotics and Automation, Nice, France, *2.*, May, 1321–1326.

Kang, S. B., and Ikeuchi, K. (1993). Toward Automatic Robot Instruction from Perception – Recognizing a Grasp From Observation. *IEEE Transactions on Robotics and Automation*, *9*(4), 432–443.

Khosla, P., and Kanade, T. (1985). Parameter Identification of Robot Dynamics. *Proceedings of the 24th Conference on Decision and Control*, Ft. Lauderdale, FL, December, 1754–1760.

Kondoleon, A. S. (1976). *Application of Technology-Economic Model of Assembly Techniques to Programmable Assembly Machine Configuration*, S.M. Thesis, MIT Mechanical Engineering Department.

Kerr, J., and Roth, B. (1986). Analysis of Multifingered Hands. *International Journal of Robotics Research*, *4*(4), 3–17.

Leahy, M. B., Jr. and Hamill, N. (1995). The Next Generation Munitions Handler Prototype Acquisition Campaign: Targets & Courses of Action. Research Paper, ACSC/DEA/205/05-05, Air Command and Staff College.

Lin, S. T., and Yae, K. H. (1992). Identification of Unknown Payload and Environmental Parameters for Robot Compliant Motion" *Proceedings of the 1992 American Control Conference*, Chicago, *4*, June, 2952–2956.

Ljung, L. (1987). *System Identification: Theory for the User*, Englewood Cliffs, NJ: Prentice-Hall.

McCarragher, B.J., and Asada, H. (1993). Qualitative Template Matching Using Dynamic Process Models for State Transition Recognition of Robotic Assembly. *Journal of Dynamic Systems, Measurement, and Control*, *115*, June, 261–269.

McCarragher, B. J., 1994(a), Force Sensing from Human Demonstration Using a Hybrid Dynamical Model and Qualitative Reasoning. *Proceedings of the 1994 IEEE Conference on Robotics and Automation*, *1*, May, 557–563.

McCarragher, B. J., 1994(b), Petri Net Modeling for Robotic Assembly and Trajectory Planning. *IEEE Transactions on Industrial Electronics*, *41*(6), December, 631–640.

MacLean, K. (1996). The 'Haptic Camera': A Technique for Characterizing and Playing Back Haptic Properties of Real Environments. *Proceedings of the ASME Dynamic Systems and Control Division*, Atlanta, GA, DSC-*58*, November, 459–467.

Nevins, J. L. and Whitney, D. E. (1980). Assembly Research. *Automatica*, *16*, 595–613.

Noyes, M. V. and Sheridan, T. B. 1984. A Novel Predictor for Telemanipulation through a Time Delay. *Proceedings of the Annual Conference on Manual Control*, Moffett Field, CA, NASA Ames Research Center.

Okamura, A. M., Turner, M. L., and Cutkosky, M. R. (1997). Haptic Exploration of Object with Rolling and Sliding. *Proceedings of the 1997 International Conference on Robotics and Automation*, Albuquerque, *3*, 2485–2490.

Pook, P. K., and Ballard, D. H. (1993). Recognizing Teleoperated Manipulations. *Proceedings of the IEEE International Conference on Robotics and Automation*, *2*, May, 578–585.

Rabiner, L.R. and Juang, B. H. (1986). An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, *3*(1), January, 4–16.

Rabiner, L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, *77*(2), February, 257–285.

Satava, R., and Jones, S. (1997). Virtual Environments for Medical Training and Education. *Presence*, *6*(2), April, 139–146.

Schulteis, T., Dupont, P., Millman P., and Howe, R. (1996). Automatic Identification of Remote Environments" *Proceedings of the ASME Dynamic Systems and Control Division*, Atlanta, DSC-Vol. *58*, November, 451–458.

Yang, J., Mu, Y., and Chen, C. S. (1994). Hidden Markov Model Approach to Skill Learning and Its Application to Telerobotics. *IEEE Transactions on Robotics and Automation*, *10*(5), October, 621–631.