Presented at the 6th ISRR, Oct. 2-5 1993, Hidden Valley, Pennsylvania

Manipulation Control with Dynamic Tactile Sensing

Mark R. Cutkosky and James M. Hyde

Center for Design Research
Stanford University
560 Panama St.
Stanford, California 94305-2232

Abstract
In this paper we describe ongoing work toward event-driven dextrous manipulation. In this context, the events are primarily determined through tactile and force/torque sensing. We begin with a review of recent work in tactile event detection and its role in the control of manipulation. We then consider control issues, focusing on the problem of accomplishing smooth transitions as the constraints, dynamic equations and control objectives change from one phase of a manipulation task to the next. Smooth transitions are essential in dextrous manipulation because of the typically low inertias of the grasped object and the fingers — the object accelerates quickly and the fingertip sensors produce large signals in response to disturbances at the contacts. Finally, we describe a language that we are developing to facilitate programming of dextrous manipulation tasks with multiple control modes and tactile sensors.

1. Introduction

Dextrous manipulation is a process characterized by tactile events and control discontinuities. A simple task such as grasping a glass of water, lifting it and replacing it contains several such events and discontinuities. Initially the fingertips approach the glass, perhaps under velocity control. The sensation of contact at the fingertips constitutes an event, and signals the need to switch to force control so that a desired grasp force is attained. As the hand starts to lift, separation of the glass from the table top is sensed as another event and requires another change in the control.

Experiments with human subjects reveal that during such tasks people rely on a combination of fast- and slow-acting tactile sensors to detect such events as contact, the onset of motion and the onset of slipping [Johansson and Westling 1990; Westling and Johansson 1984, 1987; Srinivasan et. al. 1990]. Preliminary experiments with a simplified robotic hand [Howe et. al. 1990] have suggested that a combination of force sensors and dynamic tactile sensors can provide robots with a similar ability.

However, before the goal of smooth, robust, event-driven dextrous manipulation can be achieved, many improvements in event detection and low- and high-level control are necessary. Dextrous manipulation involves a hierarchy of sensing and control problems, ranging from task-level programming at the highest level ("pick up object") to detailed motion and force trajectories at the lowest. In between is a level that we shall call phase-based control, and this level will be the focus of this paper. Phase-based manipulation must accommodate both detailed force and motion trajectories specified in terms of time and discrete events that serve as milestones in task planning.

Phase-based control promises to simplify the programming and control of dextrous hands. Dextrous hands are systems with many degrees of freedom, complex (and changing) kinematic and dynamic models, and numerous sensors and actuators. Segmentation of tasks into phases helps to make the very difficult problem of coordinated manipulation manageable by providing a limited context and scope for each phase. Thus during an "approach" phase it is natural to pose the control problem in terms of the trajectories of the fingertips, while during object manipulation it is natural to pose the problem in terms of the grasped object and the internal and external forces exerted on it. Phases also simplify sensor interpretation; for example there is obviously no need to sense the status of a grasped object during an approach phase when the fingers have not made contact.

In the following sections we first examine manipulation events more closely and consider the kinds of sensors needed to detect them. Next, we consider the ramifications of such events for control of dextrous manipulation, focusing on the problem of accomplishing smooth transitions from one control phase or regime to the next. The continual need to detect and respond to such events in even the simplest of manipulation tasks motivates the development of an Phase/Event/Transition language, whose structure we propose in the final section.

2. Dextrous manipulation events

The kinematics and dynamics of dextrous manipulation with robot hands have been investigated extensively, leading to a variety of methods for choosing grasp configurations and forces and for controlling the hand. It has been observed that the equations of motion of the combined hand/object system are quite sensitive to assumptions about the contact conditions between the fingers and the object [Cutkosky and Wright 1986]. Thus, changes in the state of the contacts between the fingertips and the grasped object, and between the grasped object and the external environment, constitute one important type of manipulation event requiring detection and an appropriate response. For example, kinematic and friction constraints change as fingers make and break contact and as they roll or slide from the smooth surface of an object onto an edge or corner. If incorrect assumptions about the contact type are made, completely unrealistic estimates of the grasp stiffness and stability can easily result [Cutkosky and Kao 1989].

Looking beyond the need to detect and respond to such changes simply to maintain acceptable performance of the hand/object system, we observe that manipulation events are mileposts that indicate progress in a manipulation task. Returning to the previous example of grasping, lifting and replacing a glass of water we observe that the task is naturally decomposed into several phases, demarcated by events. In the following table, we list the phases involved in the example and the events that would signal the end of each phase:

Phase	***Event***
• fingers approach glass	finger/object contact
• close fingers upon glass	stable grasp achieved
• ramp up grasp and load force until glass lifts	glass/table contact loss
• raise and replace glass	glass/table contact
• release grip	finger/object contact loss

As noted earlier, such tasks have been addressed in the physiology literature. Westling and Johansson [1984, 1987] have recorded the nerve signals from various tactile sensors as subjects performed grasp-lift-replace tasks using objects instrumented with position, acceleration and force sensors. The results indicate that human manipulation is event-driven, with information from tactile sensors signaling the progress of the task. In general, the shallow, fast-adapting (FAI) sensors are most sensitive to the initiation of contact at the fingertip and to the onset of slip. The deep, fast-adapting (FAII) sensors are most reliably associated with changes in the state of the grasped object such as the occurrence of contact between the object and an external surface.

Throughout a manipulation task, humans also respond to events not directly associated with phase completion. For example, as an object is held, the occurrence of small slips at one or more fingertips indicates that the object is about to start sliding. The localized "microslips" are detected primarily by the FAI sensors and elicit automatic adjustments in the ratio of the normal and tangential forces at the fingertips.

Partly inspired by the human example, recent work in robotics has explored approaches for detecting and responding to tactile events. Howe and Cutkosky [1989; 1993] discuss dynamic tactile sensors for soft robotic fingertips. One such sensor measures the rate of change of stresses in the fingertip skin. From this sensor, information about fine surface features such as scratches and grooves can be reconstructed as a fingertip moves over the surface of an object. Another sensor responds to localized accelerations or vibrations in the skin. This device produces large signals when fingertips make or break contact with an object and when a grasped object makes or breaks contact with another surface. Both the skin acceleration and stress rate sensors are capable of detecting the localized microslips that presage sliding.

It can be argued that because events such as making contact or starting to slip are associated with rapid changes in state variables, dynamic tactile sensors which measure derivative quantities are inherently most sensitive. Additional information is, of course, available from other sources. For example, force/torque sensors located in the fingertips can also indicate when a fingertip makes contact with an object. Measurements of the fingertip velocities (from joint angle sensors) and of the finger actuator or tendon forces can also indicate when a fingertip makes contact, albeit with less rapid response and lower sensitivity than sensors at the fingertips. In the previous example of grasping a glass of water, even the coolness of the glass provides confirmation of contact.

However, while a combination of sensors is typically available for responding to manipulation events, the ability to reliably detect such events during manipulation remains difficult to achieve. Part of the problem is that tactile sensors, and especially dynamic tactile sensors, respond to all kinds of disturbances.

Detection of the onset of slip provides a good illustration of the difficulties. As noted earlier, the onset of slip is accompanied by microslips at the fingertip/object contact areas, typically initiating near the periphery of the contact where the pressure is minimal. Although both stress rate and skin acceleration sensors can detect the skin vibrations that accompany such microslips, the problem is that many other phenomena including contact of the grasped object with surfaces in the environment, vibrations emanating from the manipulator drivetrain or servo control, and changes in the contact forces applied by any fingers can produce vibrations in the skin as well. In some cases, these other events are genuinely accompanied by microslips; for example, increasing or decreasing the grasp force typically changes the contact area and skin deformation, which results in some localized movement at the skin/object interface.

Consequently, grasp force adjustment based solely on the input from dynamic tactile sensors is unlikely to be feasible except when the hand is quiescent. Even then, it is necessary to make some provisions for rejecting spurious vibrations emanating from the hand servomechanism or from random events in the local environment.

Tremblay and Cutkosky [1993] have found that robustness is enhanced by comparing the signals from dynamic tactile sensors located both on and off the immediate finger/object contact area. A schematic of an instrumented fingertip is shown in Figure 1. Spurious

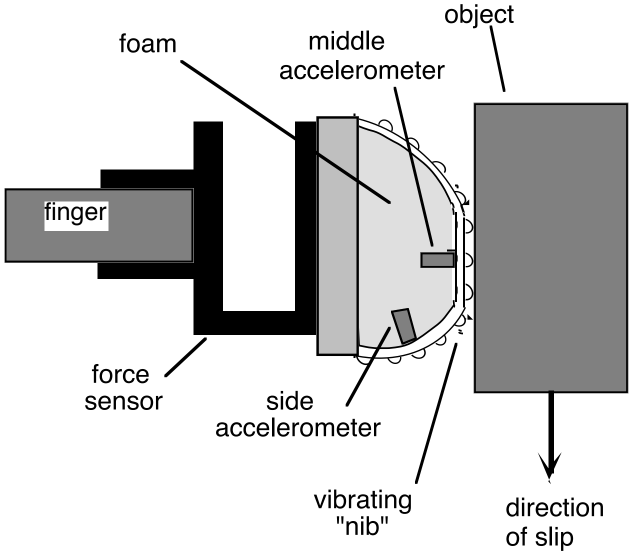

Figure 1. Cross-section of fingertip with dynamic tactile sensors on and off the contact area.

vibrations and accelerations transmitted through the fingers tend to affect both sensors equally. However, the sensor(s) located outside the contact area are more sensitive to the small vibrations associated with incipient slips. A simple combination of filtering and thresholding was found adequate for responding to incipient slips while rejecting spurious vibrations. The approach was found to work for objects with a variety of rough and smooth surfaces.

Figure 2 shows a typical test run. At time A, the grasp force is gradually reduced until an incipient slip is detected at B. The grasp force is subsequently increased by a safety factor (at C) to prevent object motion. At this time, the estimate, $\hat{\mu}$ of the coefficient of friction can also be updated using the equations:

$$\mu = f_t / f_n \ , \quad \hat{\mu} = \alpha \hat{\mu}_{n-1} + \beta \mu$$

where f_n and f_t are the contact normal and tangential forces, respectively (measured at time B), and α and β are factors that determine how much to weight the latest computation of μ in comparison to the previous estimate.

The estimate of the coefficient of friction is used in grasp force regulation. For example, at time D the external load on the object has suddenly been doubled. The robot adjusts the grasp force automatically according to the formula, $f_n = k_s \left(f_t / \hat{\mu} \right)$ where k_s is a safety factor. Object motion remains minimal (< 1 mm) throughout the experiment.

It can also been seen from Figure 2 that disturbances such as adjustments in the load and grasp forces do strongly excite the dynamic tactile sensors. However, by comparing the signals from the on- and off-contact sensors, incipient slips can be distinguished from such

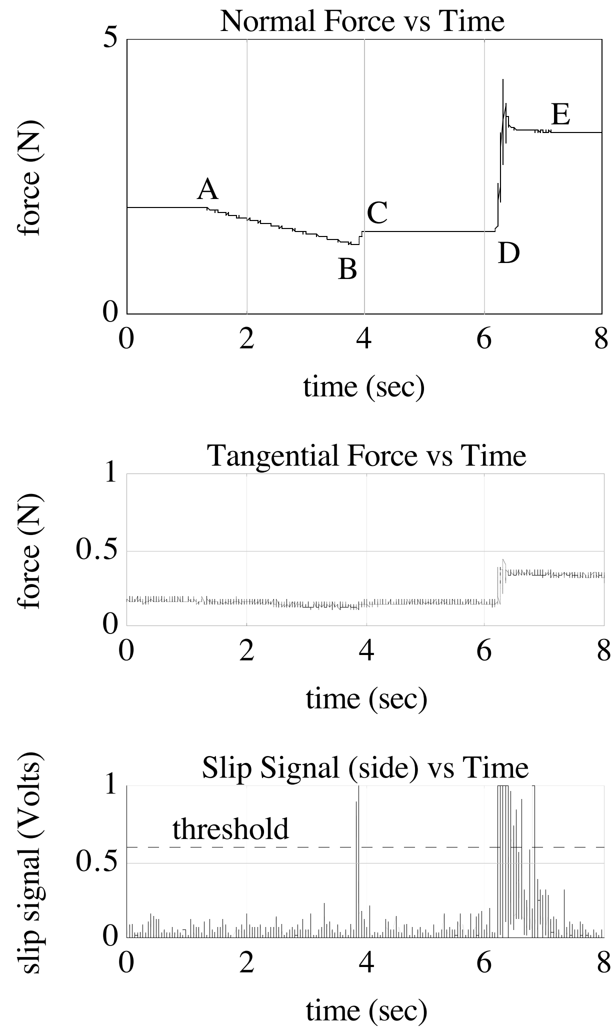

Figure 2. Plots of normal (grasp) and tangential (load) force for a fingertip holding an object, along with the signal from a side-mounted skin acceleration sensor during an incipient slip event and a disturbance in the load force.

disturbances. Figure 3 shows detailed plots from the middle and side accelerometers during an incipient slip and subsequent grasp force adjustment. Only the side accelerometer produces a significant signal during the incipient slip, but both accelerometers respond to the subsequent adjustment in grasp force.

In other work, Eberman and Salisbury [1993] have examined the signatures obtained from fingertip force/torque sensors during manipulation. Using a combination of signal processing methods and sequential hypothesis testing based on statistical analysis of the properties of the signals, they demonstrate reliable detection of such events as contact (impact) and changes in surface texture for a sliding fingertip. Eberman and

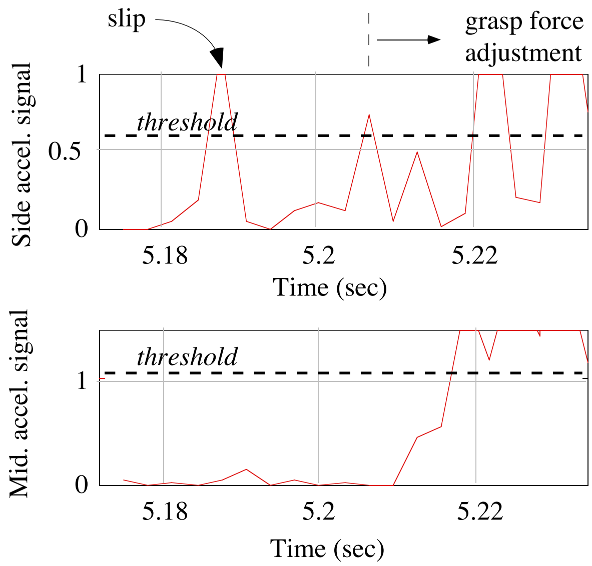

Figure 3. Comparisons of the signals from side and middle accelerometers (off and on the contact area, respectively) during an incipient slip and subsequent grasp force adjustment.

Salisbury point out that context (i.e., knowledge of prior events) is an important factor in choosing the most likely hypothesis. They also note that the inclusion of information from other sensors is straightforward and should increase robustness of event detection.

Other examples of using tactile sensors to detect events such as the onset of slip, or changes in texture can be found in the work of Bicchi et. al. [1989] and Dario and De Rossi [1985]. Recent surveys of this and other applications of tactile sensing are provided in [Howe and Cutkosky 1992] and [Howe 1994]. Table 1 summarizes some of the most common events and the sensors that we believe are most effective at detecting them. The list reflects our experience and is not intended to be exhaustive.

As Table 1 indicates, there are generally multiple sensors that can be used to detect important manipulation events. In most cases the signals are too noisy to be used directly in feedback control (especially for signals that are inherently noisy, as in the case of skin acceleration sensors), but by using signal processing techniques such as Bayesian methods, neural nets, or fuzzy systems, we may elicit reliable event-detection information from the noisy signals. Tactile event detection can also draw upon related developments in such applications as machine diagnostics.

A discussion of sensor interpretation methods is beyond the scope of this paper, but we can make a couple of observations that have ramifications for the control and language concepts described in the following sections. First, signal processing techniques will generate events that have some associated, and evolving, probability. The probability of each event depends on current and past

Event	**Sensors**
finger makes/breaks contact	<u>skin acceleration</u>, <u>stress rate</u>, <u>force</u>, velocity
finger incipient slip	<u>stress rate</u>, <u>skin acceleration</u>
finger starts/stops sliding	<u>stress rate</u>, skin acceleration, velocity
finger starts/stops rolling	<u>array</u>, <u>force</u>, velocity
change in contact type (e.g., flat surface → edge)	<u>array</u>, <u>force</u>
change in friction	<u>stress rate</u>, <u>force</u>
change in texture	<u>skin acceleration</u>, <u>stress rate</u>, array
object makes/breaks contact with external surface	<u>skin acceleration</u>, <u>force</u>, stress rate
object starts/stops sliding on external surface	<u>skin acceleration</u>, force, velocity
change in object properties (e.g., stiffness, inertia) or loading	array, force, stress rate

<u>skin acceleration</u> and <u>stress rate</u> are dynamic tactile sensors (e.g., as described in [Howe and Cutkosky 1989; 1993]).

<u>force</u> sensors are “intrinsic” force/torque sensors at the fingertips (e.g., as described in [Bicchi et. al. 1989].

<u>velocity</u> refers to fingertip velocities, typically obtained via joint angle sensing and grasp kinematics.

<u>array</u> refers to tactile arrays of pressure or surface displacement sensing elements (e.g., as described in [Fearing 1990]).

Table 1: Common manipulation events and sensors that can detect them. The sensors underlined are those believed to be most effective for detecting the corresponding events.

signals from a collection of sensors and on prior knowledge from the nominal task plan. As Eberman and Salisbury [1993] demonstrate, certain events such as impact can be difficult to isolate without such extra information.

To better understand event probability, consider the case of a fingertip approaching and making contact with an object. The first indication of contact will probably be a noisy signal from a skin acceleration sensor. The likelihood that this signal actually indicates the beginning of contact is increased by our knowledge that we are nearing the end of an “approach” trajectory. Over the next few time samples the noisy accelerometer signal is joined by an increase in the normal force, and subsequently by growing signals from a tactile pressure sensing array. We also observe that the fingertip velocity decreases. At some point during this sequence, the probability of contact exceeds a threshold and we determine that a contact event has actually occurred.

It is important to note that several "candidate" events might have to be supported until the probability distribution marks one event as dominant. Recognizing an event is essentially a symbolic commitment to that event to the exclusion of other candidates, a decision which will trigger a switch into a different control phase.

Our second observation relates to the framework needed to detect events. The possibility of multiple anticipated and unanticipated events occurring during each phase leads us to prefer a decentralized approach with independent, possibly asynchronous, event-detectors, each having access to any combination of sensors and/or lower-level signal processing units. One promising approach is presented by Brock [1993] who describes a framework involving a hierarchy of "perceptual units."

We have thus far decomposed manipulation tasks into discrete phases of control, separated by tactile events. We can detect several types of events using existing sensor technology. Ongoing work in sensor development and signal processing should allow us to distinguish critical events from a background field of disturbances. The events will act as triggers to tell us when to switch between phases, but an important segment of this work is the actual routines that will effect the phase transitions. These routines are the subject of the next section.

3. Transitions

As noted earlier, events typically signal abrupt changes in some combination of constraints, equations of motion and command inputs. In some cases, as when the fingertips make contact with a grasped object, the changes are obvious and dramatic. In other cases the changes may be more subtle, for instance when changing from a lightly loaded contact to a heavily loaded contact (resulting in a significant increase in the contact area so that a point contact kinematic model can no longer be assumed). Even if the control law itself is not changed after an event, as when using impedance control during both the approach and contact phases of a grasp, it is often desirable to change control parameters such as gains or setpoints. We will label such event-prompted controller changes "transitions." Transitions serve as bridges between the discrete control phases and to a large degree govern the apparent fluidity of a manipulation task. Successful execution of these transitions is essential for dextrous manipulation.

Most previous work on control with transitions has focused on the admittedly critical issue of stability. However, for dextrous manipulation, smoothness is at least as important. Disturbances such as force spikes during contact initiation will excite tactile sensors (especially dynamic tactile sensors, as seen in Figure 2) and can compromise the reliable detection of future events. Equally crucial are the constraints that must be obeyed during a transition. For example, contact forces must remain positive or above some minimum dictated by friction considerations to avoid dropping an object. At the same time, the contact forces must stay below some maximum that depends on available actuator forces and on the fragility of the fingertips and the object.

One of the most thoroughly studied transition control problems is that of approaching an object, making contact, and exerting some steady force. This common situation requires stable control of a manipulator in contact with the environment, with the added difficulty of dynamic discontinuities occurring at the instant of impact. Stable, smooth response during this contact transition is notoriously difficult to achieve. A variety of solutions to this problem have been proposed in previous work. In [Hyde and Cutkosky 1993] we conducted a study of several methods from the controls literature and also examined a feedforward command preshaping technique.

To use the preshaping technique, we first identified the dominant impact vibration frequencies occurring when a fingertip struck an object. This frequency information was used to modify a baseline command input (a step function rising from zero force prior to contact to the desired grasp force after contact) to suppress vibrations resulting from the change in force. Essentially, the command input was convolved with a train of impulses whose magnitudes and spacing were dictated by the impact frequency data. The preshaping technique does rely on identified parameters, but is relatively insensitive to parameter variation; over- or under-estimating the frequencies by as much as 20% results in only a 10% increase in the amplitude of post-impact oscillation.

In conducting our experiments, we used a basic form of the event-transition concept mentioned above. We constantly monitored the fingertip force level during the approach phase, and when the force exceeded a prescribed threshold, we declared a contact "event." This event prompted the "transition" to the contact force control law, using one of the published methods or the preshaping technique. Our goal was to minimize oscillations of the fingertip force signal during the transition to the force control phase.

The results of contact experiments demonstrated that any of the methods from the literature can improve response if gains are judiciously tuned. Figure 4 compares the results obtained with active impact damping [Khatib and Burdick 1986] and command input shaping. Each method clearly eliminated the large oscillations exhibited by the baseline controller. For the feedback-based methods to controlling transitions, a tradeoff among rise time, peak impact force and duration of oscillation results. The methods that rely on velocity or force feedback signals are,

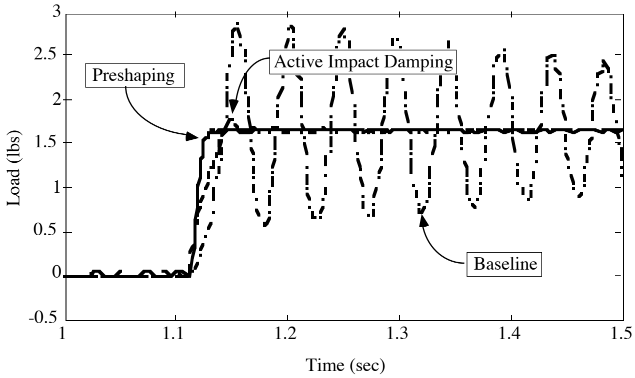

Figure 4. Force trajectory for fingertip contacting a hard surface: The effects of active impact damping [Khatib and Burdick 1986] and command input shaping [Hyde and Cutkosky 1993] are illustrated. Both methods greatly reduce the oscillations as compared to a simple baseline force control law, with the input shaping showing a slight advantage in rise time and overshoot.

not surprisingly, sensitive to noise in those signals. In contrast with these feedback based methods, input command shaping suppresses vibration by modifying feedforward information, and is therefore less susceptible to noise.

We concluded that the benefits of input command preshaping are comparable in magnitude to those of feedback-based methods, and the preshaping technique is perhaps superior where noise in velocity signals is a problem (perhaps because the velocity must be obtained by estimation from digital position information).

We have also investigated another common transition, from stationary to sliding contact. The dynamics involved in this transition depend largely on contact parameters such as contact area and friction coefficients, and also on manipulator and object impedances. Imprecise knowledge of these parameters, combined with non-linear friction phenomena, can make it difficult to move through the transition without causing large disturbances or even limit cycles at the contact. We have conducted some preliminary experiments in this area, once again using the event-transition approach, but as yet have no conclusive findings on the merits of the many possible techniques that could be applied to the transition. Certainly the preferred approach may vary depending on the contact conditions, speed of sliding after the transition, and physical composition of the fingertip and environment.

These experiments demonstrated the role of events and transitions in some simple manipulation tasks. In both the contact initiation example and the sliding onset work we monitored fingertip force information to detect contact events, and then implemented transition routines to bridge the gap between pre- and post-event control laws. It remains to be seen whether a unified framework can be developed to encompass all such transitions or whether each case (i.e., each transition type from one phase to another) must be treated separately.

To explore such issues further we are led to consider a language and programming environment for manipulation with multiple control phases, events and transitions. Our proposal for such a language is described in the following section.

4. Toward an event-based language for dextrous manipulation

The following proposal for an event-driven language has been inspired by our work on detecting tactile events and transition control. It also draws inspiration from a number of works on manipulation languages and event-based robot programming. In particular, the phase description is based on Brockett's Motion Description Language (MDL) [Brockett 1988; Eng 1988].

In MDL, motion and/or force trajectories are specified as sequences of triples, $((\mathbf{U}_1,\mathbf{K}_1,T_1)\ (\mathbf{U}_2,\mathbf{K}_2,T_2)\ \ldots\ (\mathbf{U}_n,\mathbf{K}_n,T_n))$, in which each $\mathbf{U}_n$ is a vector of command inputs, $\mathbf{K}_n$ is a matrix of gains and T_n is the time duration or epoch over which $\mathbf{U}_n$ and $\mathbf{K}_n$ apply. T_n can be indefinite ($T_n \rightarrow \infty$) for guarded-move instructions. During each time interval, the system is assumed to be governed by the differential equation:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\big(\mathbf{U}(t) + \mathbf{K}\mathbf{y}(t)\big)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the vectors of state and measured variables, respectively, $\mathbf{U}$ is the vector of setpoint commands, and $\mathbf{K}\mathbf{y}$ is the feedback law. As the gain matrix, $\mathbf{K}$, can be varied with each epoch, and the measured variables, $\mathbf{y}$, can include any combination of forces, velocities and positions, a wide range of behaviors including conventional PID motion control, force control and impedance control can be accommodated.

However, while MDL provides a versatile motion control language, it is not sufficient for event-driven dextrous manipulation for a couple of reasons:

- Although the $\mathbf{U}$, $\mathbf{K}\mathbf{y}$ combination can handle a range of control laws, it becomes awkward to try to accommodate tasks for which the preferred control formulation changes (as when shifting from fingertip-centered motion control to object-centered manipulation) using a single set of $\mathbf{U}$, $\mathbf{K}$, $\mathbf{x}$, and $\mathbf{y}$ matrices and vectors.
- The detection of events is important for higher-level task planning and execution. At this level, one is interested less in the time duration associated with a trajectory than with the occurrence of a symbolic event and the decision about how to respond to it.

Therefore, we propose a language that essentially augments MDL with certain aspects of discrete event systems, and contains explicit provisions for executing smooth transitions from one phase to the next. In our framework, a task consists of phases interrupted by events. Phases are task segments characterized by a single controller and set of constraints. Phases can easily be accomplished with a single string of (U,K,T) triples. Each phase is launched by a Starting-event and begins with a Starting-transition, a routine which depends on the details of the phase and the Starting-event. Phases also include one or more Expected-end-events and corresponding Expected-end-transitions. Expected-end-events are phase goals, and Expected-end-transitions are routines executed in anticipation of the end-event, the intention being to smooth the switch to the next phase. A simple example of a phase with a single expected ending event is a move command that moves a fingertip from location A to location B in space:

Phase: Move from A to B

- *Starting-event:* Fingertip is at (within some tolerance of) location A.
- *Starting-transition:* Accelerate to desired velocity.
- *Control:* Follow specified trajectory consisting of (U,K,T) triples.
- *Expected-end-event:* Attain location B (within some tolerance).
- *Expected-end-transition:* Decelerate

If a task proceeds according to plan, each phase is terminated by an expected ending event. However, unanticipated events are always possible. Thus, the motion phase in the preceding example could be terminated by a Contact event, leading to a new phase, perhaps consisting of compliant contour-following. For robustness, we should anticipate a number of possible end events (some desirable and some not) for each phase. This issue will be addressed in the next section. However, for the remainder of this section we will confine our discussion to a single end event for simplicity.

Figure 5 is a schematic illustration of how several phases can be concatenated to accomplish a simple task such as probing the surface of an object with a single fingertip. In Phases 1 and 2 the fingertip undergoes two motion trajectories with a waypoint at location A. The Expected-end-event during Phase 1 is the attainment of location A, and the Expected-end-transition is a slight deceleration to ensure that location A is achieved with sufficient accuracy. When the fingertip has arrived within some tolerance of position A, Phase 2 begins with a starting transition in which the fingertip accelerates back to its previous velocity.

The Expected-end-event for Phase 2 is contact with the object. At time t_2, believing that contact is imminent, the system begins the Expected-end-transition and decelerates to a low approach velocity. The fingertip servo stiffness is also reduced in anticipation of a switch to force control. However, the contact event does not occur as soon as expected and the fingertip crawls toward the object until contact is sensed at Event 2.

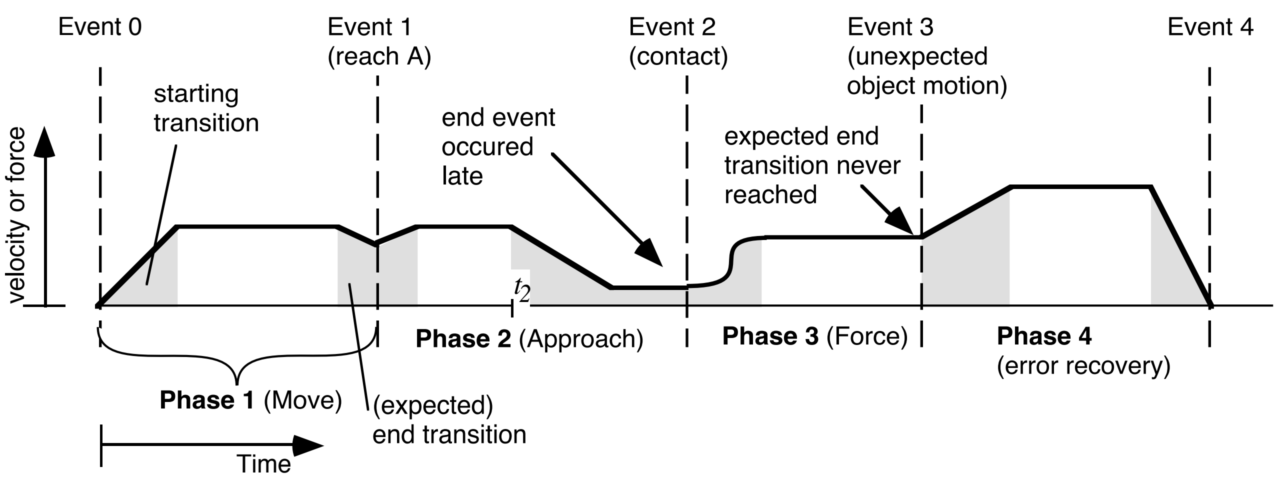

Figure 5. The Phase/Transition/Event sequence for part of a task is shown schematically above. Each phase begins and ends with an event such as fingers making contact or arriving at a specified location. In this example a fingertip approaches and touches an object, which slips unexpectedly.

Upon contact, there is a switch to force control for Phase 3. A smooth transition is needed to ramp up the contact force (perhaps using command input shaping) to some desired level. The expected ending event for this phase might be a timeout, signifying that the desired force was maintained for some prescribed time. However, this event is never reached; the object slips and an object-motion event is registered as Event 3. The final phase is therefore a default error recovery phase in which the fingertip retreats. This phase requires a smooth transition from the force control of the previous phase to a desired retreat velocity.

This example illustrates how several phases can be chained together to perform a useful task, but it also raises a few issues. First, we observe that Phase 1 is optional. The entire trajectory from the starting location, through location A, to the expected contact location could be specified as a single string of (U,K,T) triples. The reduced speed at location A and the decrease in the fingertip impedance could be achieved by varying the values of U and K respectively, rather than with explicit transitions. The decision to create two distinct phases depends on the importance of location A as a milestone for monitoring the task. Thus, while significant changes in the control law are always associated with events, it is not true that all events require changes in the control law. Indeed, if attainment of location A is important to register as an event, but it is not desired to slow down in the vicinity of A, then the ending transition of Phase 1 and the starting transition of Phase 2 can be Null.

Clearly, a task can be decomposed in multiple ways. Regarding the glass of water experiment mentioned in the introduction, Westling and Johansson would have divided the raise-and-replace phase into three separate phases for lifting, holding and lowering the glass, because it suited them to identify events with the accelerations detected by FAII sensors as movement stopped and started. The phase structure can be adjusted to reflect a researcher's goals or experimental focus.

A second issue, mentioned earlier, is the use of multiple ending events and transitions to improve the system response during deviations from the nominal phase sequence. If we had explicitly accounted for the possibility of object motion at Event 3 in the previous example, we could have executed a transition and subsequent phase to stop the unwanted motion and perhaps return to the force control of Phase 3.

Yet another issue concerns the timing of the Expected-end-transitions. How do we know when to activate these transitions? This question is inexorably tied to the problem of event uncertainty. We want to make use of prior information, to estimate when an event might occur so that we can smooth the switch into the next phase. We examine the problems of unexpected events and event uncertainty in the next section.

5. Extending the language toward multiple and uncertain events

In describing the phase/event/transition framework in Section 4 we assumed that some higher-level agent (or a human programmer) chooses the appropriate phase, including specifications for the controller and starting transition, when a particular event occurs. At this level, the hand can be viewed as a discrete-event system. As McCarragher and Asada [1993] point out, there is apparently no unified theory for control of discrete event control systems. However, a number of representations and approaches have been proposed for manipulators viewed as discrete event dynamic systems, including state-transition tables [Schneider 1989], task/context hierarchies [Brock 1993], systems based on finite state automata [Sobh and Bajcsy 1992] and Petri nets [Cao and Sanderson 1993], [McCarragher and Asada 1993]. Brockett [1993] also considered discrete systems by developing a theoretical framework for extending his Motion Description Language (MDL) to model hybrid signal/symbol systems.

Figure 6 is an effort to cast the phase/event/transition framework into the notation of a discrete event dynamic system. The states (or “places” in Petri net notation) are the nodes of the graph and correspond to the phases in our framework; the events are labeled arcs that take the system from one state, or set of “places,” to the next. (Note that the term “transition” in the Petri net literature corresponds to what we would call an “event,” and is therefore different from our notion of transitions as the beginning or ending sections of phases.)

In the example of Figure 6, the nominal plan involves first moving to a Pre-Grasp configuration in which the fingers are centered about an object. The expected event is attainment of the desired configuration. The next phase consists of closing the fingers until contact is sensed. The third phase consists of ramping up the load and grasp forces in parallel, while watching for incipient slippage between the fingertips and the object. In this example, incipient slip events prompt adjustments to the normal/tangential force ratio, but do not trigger a new phase. When the object separates from the table top the next phase begins.

In this very simple example, any event other than the expected event at each phase takes the system to a simple error recovery phase (retreating slowly to a safe configuration). It is a simple extension to alter the “unexpected events” into “secondary expected events” and have these secondary events launch other phases besides the “Goto Safe Place” phase. Using this framework, we can construct phase loops to work around problems, thus adding robustness to the task execution.

We note that while the example shown in Figure 6 is trivial in comparison to most discrete event systems, it already shows greater sophistication than most dextrous hands are able to manage today.

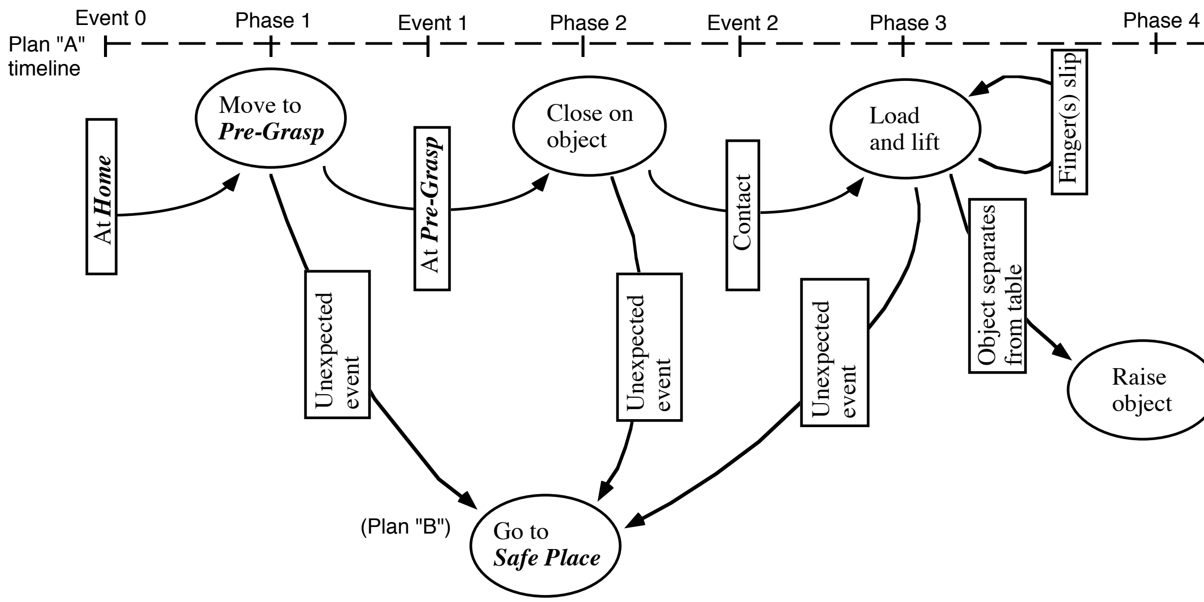

Figure 6. A fragment of a very simple plan for grasping an object. In the nominal plan, each phase (nodes in ovals) ends with an expected event (arcs with box labels). In this simple example, any event other than the expected (goal) event for each phase results in failure and retreat (Plan "B")

The representation in Figure 6 is useful for capturing the flow of phases and events but we still have not addressed the issues of event uncertainty and transition timing. As discussed at the end of Section 2, a realistic approach to event detection will result in events with probabilities that develop over time. (Because manipulation requires high servo rates, even the events themselves will typically occupy several time samples — this is in contrast to most discrete event applications in which it is assumed that an event will occur between one tick of the state clock and the next).

The probability of an event will depend on prior knowledge, in addition to current sensor measurements. For example, as we approach an object, we typically have some knowledge about when we expect to make contact — the location and dimensions of the object may be approximately known as a result of visual sensing or obtained from a geometric model. We can take advantage of this knowledge by increasing the contact event probability as we approach the object. As the probability grows to some significant level, we can adjust the approach velocity and perhaps the controller gains. This adjustment is our proposed Expected-end-transition from section 4. Note that although there is now a continuum spanning the onset and occurrence of an event, there is still a qualitative difference between the adjustment made when approaching an expected event and the transition made following an event. In the former case we seek to adjust the parameters of an ongoing phase so as to facilitate the transition to a new phase, should the expected event occur. In the latter case, a known event has occurred and we seek a smooth beginning for a new phase with, perhaps, a completely new control structure.

We can account for multiple, simultaneous events of varying uncertainty by establishing some thresholds and taking advantage of the state/event representation provided by Petri nets, state tables and similar task representations. Extensions to such representations make it possible to compute and propagate the probabilities associated with uncertain events. Examples include Petri nets augmented with fuzzy local and global variables [Cao and Sanderson 1993] and finite state automata with events that have probabilities [Sobh and Bajcsy 1992]. In a related approach, Brock [1993] discusses a hierarchical representation of tasks whose execution depends on whether the current system Knowledge satisfies the required context for each task. The system Knowledge corresponds to the current state estimate (with explicit uncertainty) and is a function of both the state estimate obtained from sensing and previous Knowledge.

Any of these methods can allow a discrete event system to recognize multiple uncertain events, and take advantage of prior knowledge. Under the Petri net representation, for instance, each phase will have a list of events to be monitored. During the execution of the phase, sensor interpretation and prior knowledge will attribute varying certainty levels to these events. When any one of the events exceeds a "plausibility threshold" we can look ahead in our phase structure and execute the Expected-end-transition that would be appropriate if that event were about to occur. While the transition routine is running, we can continue to monitor events. If a second event becomes plausible, we can modify our ending transition as appropriate. Eventually, one event will exceed a "commitment threshold" and we can declare that the event has actually occurred. At this time we consult the phase/event network to determine the next phase and execute the starting transition needed to begin that phase.

6. Conclusions

In this paper we have proposed a framework for event-based dextrous manipulation. In the context of dextrous manipulation, most of the events that we are concerned with involve changes in contact conditions either between the hand and a grasped object or between the object and the external environment. Such events are an inherent aspect of dextrous manipulation, and part of what makes it challenging -- the dynamic equations, the constraints and the desired control structure often change as fingers make or break contact or as the object starts to slide.

Because the events are typically associated with contact conditions, tactile sensors (interpreted broadly to include force/torque and position sensors in the fingers as well as conventional tactile arrays) provide the best means for detection. Dynamic tactile sensors, which measure derivative quantities such as skin stress rate and vibration, are especially sensitive to the abrupt changes in contact conditions that accompany manipulation events. As an example, we have briefly described our work on detecting incipient slip events with dynamic tactile sensors.

In summary, it appears that a combination of tactile sensors now exists that will permit a robot to detect most kinds of manipulation events. However, the reliable detection of events remains challenging, in large measure because contact events are inherently noisy and tactile sensors respond strongly to the force and velocity disturbances that accompany them.

Once contact events have been detected, it is necessary to respond to them. Typically, this requires a change in the control formulation (e.g., from fingertip motion control to object impedance control). It is important to accomplish the transition to the next control regime or phase smoothly, to minimize disturbances that can excite sensors and destabilize the control. This is particularly true when switching to force control. A general framework for accomplishing such transitions has yet to be developed but a number of promising solutions are available for common cases. As examples, we have briefly described our work on achieving smooth transitions from motion control to force control and from stationary to sliding contact.

To further explore the detection of and response to contact events, we are lead to consider a phase/event/transition language. The language sits at an intermediate level between detailed trajectory specification and control and task-level control with discrete events. In our approach, a simple manipulation task such as grasping and lifting an object is decomposed into several phases, demarcated by contact events. The description of each phase includes a specification of the control law and constraints as well as the force or motion trajectory to be followed. Phases also have explicit starting-transitions and expected-ending-transitions to ensure smooth performance across changes in the control law. A sequence of phases can easily be represented in terms of standard discrete event systems notation, as a basis for task-level planning and execution monitoring. We conclude with a discussion of possible extensions to address phases with multiple and uncertain ending events. We believe that such extensions will be essential for robust manipulation.

Acknowledgments

We are indebted to Prof. Robert Howe of Harvard University for his collaboration on this work and his many suggestions during the genesis and refinement of the ideas presented in this paper. The work described has been supported by the Office of Naval Research under contract N00014-92-J-1887.

References

[Bicchi et. al. 1989]
A. Bicchi, J.K. Salisbury and P. Dario. "Augmentation of grasp robustness using intrinsic tactile sensing". Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Vol. 1, pp. 302-307, May 14-19, 1989.

[Brock 1993]
D. L. Brock, "A Sensor Based Strategy for Automatic Robotic Grasping," Ph.D. Thesis, MIT, January 1993.

[Brockett 1988]
R. W. Brockett, "On the Computer Control of Movement," Proceedings of the 1988 IEEE International Conference on Robotics and Automation, pp. 534-540, Philadelphia, April, 1988.

[Brockett 1993]
R. W. Brockett, "Hybrid Models for Motion Control Systems," in Perspectives in Control, H. L. Trentelman and J. C. Willems, Eds., Birkhauser, 1993.

[Cao and Sanderson 1993]
T. Cao and A. C. Sanderson, "A Fuzzy Petri Net Approach to Reasoning about Uncertainty in Robotic Systems," Proceedings of the 1993 IEEE International Conference on Robotics and Automation, pp. 317-322, May, 1993.

[Cutkosky and Kao 1989]
M.R. Cutkosky and I. Kao, "Computing and Controlling the Compliance of a Robotic Grasp," IEEE Transactions on Robotics and Automation, Vol. 5, No. 2, April, 1989, pp. 151-165.

[Cutkosky and Wright 1986]
M.R. Cutkosky and P.K. Wright, "Friction, Stability and the Design of Robotic Fingers," International Journal of Robotics Research, 1986, Vol. 5, No. 4, 1986, pp. 20-37.

[Dario and De Rossi 1985]
P. Dario and D. De Rossi. “Tactile sensors and the gripping challenge,” IEEE Spectrum, Vol, 5, No. 22, pages 46-52, August 1985.

[Eberman and Salisbury 1993]
B. Eberman and J. K. Salisbury, Jr., “Application of Change Detection to Dynamic Contact Sensing,” A.I. Memo No. 1421, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1993.

[Eng 1988]
V. Eng, “Modal Segments for a Peg Insertion Task,” Harvard Robotics Laboratory Technical Report, May 25, 1988.

[Fearing 1990]
R. S. Fearing, “Tactile Sensing Mechanisms,” International Journal of Robotics Research, Vol. 9, No. 3, June 1990, pp. 3-23.

[Howe 1994]
R. D. Howe, “Tactile Sensing and Control for Robotic Manipulation,” Journal of Advanced Robotics, 1994 (in press).

[Howe et. al. 1990]
R.D. Howe, N. Popp, P. Akella, I. Kao and M.R. Cutkosky. “Grasping, manipulation and control with tactile sensing,” Proceedings of the 1990 IEEE International Conference on Robotics and Automation, pp. 1258-1263, Cincinnati, Ohio, May 1990.

[Howe and Cutkosky 1989]
R.D. Howe and M.R. Cutkosky. “Sensing skin acceleration for texture and slip perception.” In the Proceedings of the 1989 IEEE International Conference on Robotics and Automation, pages 145-150, Scottsdale, Arizona, May 1989.

[Howe and Cutkosky 1992]
R. D. Howe and M. R. Cutkosky, “Touch Sensing for Robotic Manipulation and Recognition,” The Robotics Review 2, O. Khatib, J. Craig and T. Lozano-Perez, eds., M.I.T. Press, Cambridge, MA, 1992, pp. 55-112.

[Howe and Cutkosky 1993]
R. Howe and M. R. Cutkosky, ``Dynamic Tactile Sensing: Perception of Fine Surface Features with Stress Rate Sensing," IEEE Transactions on Robotics and Automation, Vol. 9, No. 2, April, 1993, pp. 140-151.

[Hyde and Cutkosky 1993]
J. M. Hyde and M. R. Cutkosky, “Contact Transition Control: An Experimental Study,” Proceedings of the 1993 IEEE International Conference on Robotics and Automation, Atlanta, Georgia, May 1993.

[Johansson and Westling 1990]
R.S. Johansson and G. Westling. “Tactile afferent signals in the control of precision grip.” Attention and Performance, volume 8, pages 677-713, 1990.

[Khatib and Burdick 1986]
O. Khatib and J. Burdick, “Motion and force control of robot manipulators,” In the Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, California, April 1986.

[McCarragher and Asada 1993]
B. J. McCarragher and H. Asada, “A Discrete Event Approach to the Control of Robotic Assembly Tasks,” Proceedings of the 1993 IEEE International Conference on Robotics and Automation, Vol. 1, pp. 331-336, May, 1993.

[Schneider 1989]
S. A. Schneider, “Experiments in the Dynamic and Strategic Control of Cooperating Manipulators,” Stanford Ph.D. Thesis, September 1989.

[Sobh and Bajcsy 1992]
T. M. Sobh and R. Bajcsy, “Autonomous Observation Under Uncertainty,” Proceedings of the 1992 IEEE International Conference on Robotics and Automation, pp. 1792-1798, May, 1992.

[Srinivasan et. al. 1990]
M.A. Srinivasan, J.M. Whitehouse and R.H. Lamott. “Tactile detection of slip: Surface micro geometry and peripheral neural codes.” Journal of Neurophysiology, 1990.

[Tremblay and Cutkosky 1993]
M. Tremblay and M. Cutkosky, “Estimating friction using incipient slip sensing during a manipulation task,” Proceedings of the 1993 IEEE International Conference on Robotics and Automation, Atlanta, Georgia, May 1993.

[Westling and Johansson 1984]
G. Westling and R.S. Johansson, “Factors Influencing the Force Control During Precision Grip,” Experimental Brain Research, Vol. 53, pp. 277-284, 1984.

[Westling and Johansson 1987]
G. Westling and R.S. Johansson, “Responses in Glabrous Skin Mechanoreceptors During Precision Grip in Humans,” Experimental Brain Research, Vol. 66, pp. 128-140, 1987.