

# MODEL-ADAPTIVE HYBRID DYNAMIC CONTROL FOR CONSTRAINED MOTION SYSTEMS\*

Brenan J. McCarragher  
brenan@faceng.anu.edu.au

and

David J. Austin  
david@faceng.anu.edu.au

Department of Engineering, Faculties,  
Australian National University,  
Canberra, ACT 0200,  
Australia.

October 19, 2001

**Keywords:** Hybrid dynamic system; adaptive control; constrained motion; Lyapunov

## Abstract

*A new task-level adaptive controller is presented for the hybrid dynamic control of constrained motion systems. Using a hybrid dynamic model of the process, velocity constraints are derived from which satisfactory velocity commands are obtained. Due to modelling errors and parametric uncertainties, the velocity commands may be erroneous and may result in sub-optimal performance. A task-level adaptive control scheme, based on the occurrence of discrete events, is used to change the model parameters from which the velocity commands are determined. Automated control of an assembly task is given as an example and simulations and experiments for this task are presented. These results demonstrate the applicability of the method and also indicate properties for rapid convergence.*

---

\*This research was funded in part by a grant from the Australian Research Council

## 1 Introduction

This paper will consider the adaptive control of constrained motion systems. Examples of such systems are numerous. In particular, constrained motion systems are common in the field of robotics. For example, walking machines, robot manipulators performing assembly or grinding tasks, and robotic hands grasping objects are all constrained motion systems. A great deal of research has been conducted in the field of constrained motion systems. Hogan [3] developed impedance control for the manipulation of objects constrained by the environment. Mason [6] and Raibert and Craig [10] developed controllers which use both position and force control for the manipulation of constrained objects. The concept of using position and force control simultaneously has been extended by a number of researchers (e.g. [11]) in search of a better control scheme for constrained motion systems. Also, Tarn et al [12] have investigated closely the problems encountered during contact transition. Despite these efforts, the application of constrained motion systems has not had the burgeoning success of comparable technologies. This poor level of success is due to a continuing focus on the low-level details. We would argue that work for constrained motion systems should proceed at a more abstract level. Hybrid dynamic systems provide a good framework for modelling and analysis of abstract concepts linked to continuous systems. Brockett [2] provides a good discussion of the advantages of hybrid dynamic modelling.

Constrained motion systems frequently require some form of adaptive control due to poor modelling information or time-varying constraint functions. For example, assembly tasks are constrained motion systems which frequently exhibit both poor modelling and time-varying constraint functions. Assembly tasks are traditionally one of the most error prone of robotic applications. Conventional approaches to assembly fail to address two problem areas. First, adaptive methods are necessary for most applications due to the difficulties discussed above, and second, an abstraction of the assembly process is required to escape from the drudgery of low-level details.

Adaptive control of constrained motion systems has been the subject of considerable research. However, most prior work in the area focusses on adaptive control in a continuous-time framework (e.g. [5]). The

hybrid dynamic systems framework provides an ideal tool for adaptation and allows the separation of the continuous and discrete elements. In prior works, McCarragher [7, 8] has presented adaptation schemes using the hybrid dynamic system framework. These adaptation schemes have the demonstrated ability to recover from position and orientation errors and converge to an optimal assembly sequence. Little other work has been done in the area of adaptive control of hybrid dynamic systems.

In this paper we present a model-adaptive control scheme for constrained motion systems. The following section presents an overview of hybrid dynamic modelling for constrained motion systems, then we discuss techniques for synthesising the control commands. Section 4 describes the adaptation law necessary due to modelling errors and Section 5 presents simulated and experimental results for a peg-and-hole insertion task. The experimental results demonstrate the real-world applicability of the model-adaptive control scheme.

## 2 Hybrid Dynamic Modelling of Constrained Motion Systems

We consider a specific type of hybrid dynamic system, consisting of a discrete event controller interfaced to a constrained motion system involving the motion of two polyhedral parts. The two parts are a rigid, polyhedral *workpiece* with possible constraints introduced by contact between the workpiece and a fixed, rigid and polyhedral *environment*. Assembly processes are typical systems of this type. Hybrid dynamic modelling is particularly appropriate for systems of this type as there are a small number of possible combinations of edge-edge and surface-vertex contacts, and hence, we have dramatically reduced the complexity of the model by abstracting to a higher level. The structure of the adopted hybrid dynamic model is as shown in Figure 1. The system consists of three parts: a discrete event controller, a continuous-time system, and an interface between the two.

For this paper, the continuous-time system consists of a velocity-controlled workpiece interacting with

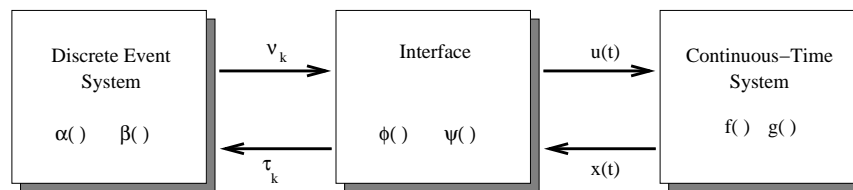


Figure 1: *Block Diagram Representation of Hybrid Dynamic System*

the environment (as discussed above). The continuous-time system is described by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

where  $\mathbf{x}(t)$  is the continuous-time state vector and  $\mathbf{u}(t)$  is the velocity input to the continuous-time system or the *control command*. As the workpiece interacts with the environment, the free-space dynamics of (1) become constrained. The geometric constraints may be expressed as

$$\mathbf{g}_j(\mathbf{x}) = 0 \quad (2)$$

where  $\mathbf{g}_j$  is the constraint function for the  $j^{\text{th}}$  edge-edge or surface-vertex contact. Distance functions are ideal candidates for  $\mathbf{g}_j$  (e.g. the shortest distance between a surface and a vertex).

The interface consists of two maps between the continuous-time system and the discrete event system. The first map  $\phi$  is the part of the interface which converts the discrete event controller commands ( $\nu_k$ ) into continuous control commands

$$\mathbf{u}(t) = \phi(\nu_k) \quad t_k \leq t < t_{k+1} \quad (3)$$

where  $t_k$  and  $t_{k+1}$  are the times of the  $k^{\text{th}}$  and  $k+1^{\text{th}}$  events, respectively. The second part of the interface is the process monitor, which uses the continuous signal  $\mathbf{x}(t)$  to detect contact changes in the system and determine corresponding discrete events. The process monitor is defined as the map  $\psi$ :

$$\gamma_k = \psi(\mathbf{x}(t)) \quad (4)$$

where  $\gamma_k$  is the  $k^{\text{th}}$  discrete state of the system.

The discrete event controller is the part of the system which determines the appropriate discrete command to issue, based upon the occurrence of discrete events. The state transition function  $\alpha$  determines

the next discrete state of the system from the current discrete state of the system and the discrete event and the function  $\beta$  determines which discrete command to issue.

$$\gamma_{k+1} = \alpha(\gamma_k, \tau_k) \quad (5)$$

$$\nu_k = \beta(\gamma_k) \quad (6)$$

For our purposes, the discrete states of the system correspond to the possible combinations of edge-edge and surface-vertex contacts between the workpiece and the environment.

### 3 Control Command Synthesis

Control commands are determined by first establishing a *desired event* for each state. The desired event is chosen such that the system moves towards the target state. The desired events may be determined manually or automatically, depending upon the application. For any given state, we use the desired event and geometric considerations of the workpiece and environment to establish conditions on the command to be executed. There are three conditions upon which the control law (3) for motion control is selected. First, the *maintaining condition* ensures that the currently active constraints (2) remain satisfied, if desired. Second, the *enabling condition* is a necessary condition that ensures that the next desired discrete event  $\tau_{k+1}$  is allowed to occur. Third, the *disabling condition* is a sufficient condition that ensures an undesired discrete event is not allowed to occur.

The motion of the system described by (1) is constrained by (2). The first possible task of the controller is to ensure that the control commands satisfy this geometric constraint. To derive admissible velocities that satisfy the geometric constraint, we can differentiate (2) to give

$$\frac{\partial}{\partial \mathbf{x}} [g_j(\mathbf{x})] \frac{d}{dt} \mathbf{x}(t) = \mathbf{0} \quad t_k \leq t < t_{k+1} \quad (7)$$

where  $g_j$  is the constraint function for this contact. This can be rewritten as

$$\mathbf{a}_j^T \dot{\mathbf{x}}(t) = \mathbf{0} \quad t_k \leq t < t_{k+1} \quad (8)$$

where  $\mathbf{a}_j = \frac{\partial}{\partial \mathbf{x}} g_j(\mathbf{x})$  is a column vector with length equal to the number of degrees of freedom. Equation (8) is our maintaining condition in that it must be satisfied to maintain the contact or geometric constraint. When  $g_j$  is a distance measure, equation (8) becomes a requirement that the distance between the points of contact remains zero (i.e. the points remain in contact).

In addition to maintaining a constraint, it is desired to determine the motion such that the workpiece encounters the next discrete state  $\gamma_{k+1}$ . There are two types of events that we must consider: gain of constraint and loss of constraint. If we assume that  $g_l$  is a distance measure then the enabling condition for a gain of constraint must reduce the distance. Hence

$$\mathbf{a}_j^T \dot{\mathbf{x}}(t) < \mathbf{0} \quad t_k \leq t < t_{k+1} \quad (9)$$

Similarly, to enable a loss of constraint, the distance must increase

$$\mathbf{a}_j^T \dot{\mathbf{x}}(t) > \mathbf{0} \quad t_k \leq t < t_{k+1} \quad (10)$$

Equations (9) and (10) are necessary conditions for discrete event  $\tau_{k+1}$  to occur.

The third condition, the disabling condition, is used to prevent unwanted gains of constraint and is derived directly from the enabling condition. Since (9) is a necessary condition for a discrete event to occur, a sufficient condition for a discrete event not to occur is obtained by changing the direction of the inequality.

$$\mathbf{a}_j^T \dot{\mathbf{x}}(t) \geq \mathbf{0} \quad t_k \leq t < t_{k+1} \quad (11)$$

where  $j$  indicates the discrete states (constraint equations) that are not desired to occur. When  $g_j$  is a

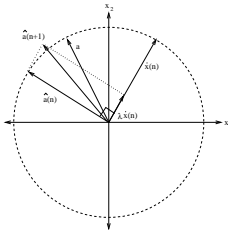


Figure 2: *Justification of Adaptation Law*

distance measure, equation (11) becomes a requirement that the distance between the points of contact does not decrease (i.e. the points stay apart).

For each possible edge-edge or surface-vertex contact, the desired event determines which one of the above conditions should be applied, forming a set of conditions. The control command is now determined by satisfying this set of conditions. Any method for satisfying the set of constraints will yield an acceptable discrete event velocity command. One method [1], which uses a search technique to maximise the minimum distance to each constraint for maximum robustness, is suggested.

#### 4 Adaptation

Despite determining a velocity command which satisfies the above constraints, errors can still occur due to model inaccuracies, tracking control errors, or other unknowns. In these situations, it is desired to have the system adjust to the new information and adapt the desired velocity commands. The ability to adapt is particularly important in an industrial setting where new products are frequently introduced and the assembly line needs to be “tuned” to the new tasks. The extra time, work and uncertainty involved a sub-optimal operation motivates the need for task-level adaptation of the discrete event controller.

An adaptation law needs to be selected to reduce the modelling error. An adaptation law is proposed

based on intuitive reasoning about the vector spaces used for discrete event control as shown in Figure 2. Consider the adaptation of a maintaining condition. Here, the estimate of the constraint vector  $\hat{\mathbf{a}}$  and the velocity vector  $\dot{\mathbf{x}}$  are orthogonal. Yet, the velocity vector is not orthogonal to the actual constraint vector  $\mathbf{a}$ , indicating the need for adaptation. By adding a portion of the velocity vector to the estimated constraint vector, the difference between the estimated and the actual constraint vectors decreases. Hence, the following adaptation law is proposed

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n - \Delta \lambda \dot{\mathbf{x}}_n \quad (12)$$

where  $\lambda > 0$  is the adaptation rate, and  $\dot{\mathbf{x}}_n$  is the velocity command vector for the  $n^{th}$  trial.  $\Delta_i = -sgn(\mathbf{a}_i^T \dot{\mathbf{x}})$  is a switching function determined by the control condition to be adapted. In addition to equation (12),  $\mathbf{a}$  is renormalised following the adaptation as the magnitude of  $\mathbf{a}$  is unimportant. Given this adaptation law, two issues arise. The first is demonstrating the convergence of the estimated model constraint to the actual parameters. The second is the selection of  $\lambda$  such that the adaptation remains stable. Both of these issues can be answered using Lyapunov theory.

The complete proof of Lyapunov stability is given in [9]. The result of that proof is that stability, and hence convergence to zero modelling error, is guaranteed if the following condition on the adaptation rate is met.

$$\lambda < \frac{2\Delta \tilde{\mathbf{a}}_n^T \dot{\mathbf{x}}_n}{\|\dot{\mathbf{x}}_n\|^2} \quad (13)$$

where  $\tilde{\mathbf{a}}(n) = \hat{\mathbf{a}}(n) - \mathbf{a}$  is the modelling error. We will now examine how to satisfy equation (13) for each of the discrete event conditions. For simplicity and to highlight the adaptation equations, we will assume that the velocity vector has been normalised to  $\|\dot{\mathbf{x}}_n\| = 1$ . This assumption has little effect as only the direction of the velocity vector is important.

Due to modelling errors the maintaining condition may not be satisfied. A violation of the maintaining condition implies that the desired velocity command does not maintain the desired contact. The desired

velocity command was derived according to the current understanding of the system, given by

$$\hat{\mathbf{a}}_n^T \dot{\mathbf{x}}_n = 0 \quad (14)$$

The execution of the system resulted in

$$\mathbf{a}^T \dot{\mathbf{x}}_n = C \quad (15)$$

where  $C$  is a non-zero constant indicating that the maintaining condition has not been met. For a loss of contact, the distance between the points of contact has increased. Hence,  $C$  is positive if a loss of contact occurs. Alternatively, an over-force condition (a large increase in force) will be recorded if the velocity command attempts to “go through” the environment. In this case, the distance measure is attempting to go negative, and hence,  $C$  is negative if a large increase in force is experienced.

Applying equations (14) and (15) to equation (13) yields the condition on the adaptation rate which guarantees that the model constraint estimates will converge to the true model parameters, for a violation of the maintaining condition.

$$\lambda < -2\Delta C \quad (16)$$

Satisfying (16), however, is difficult because the quantity  $C = \mathbf{a}^T \dot{\mathbf{x}}_n$  is not exactly known. Only the sign is known from the detection of discrete events. Furthermore, as the modelling error decreases,  $\tilde{\mathbf{a}}_n$  decreases and so the adaptation rate must also decrease in order to satisfy (13). Specifics for the actual adaptation rate will depend on implementation, such as the rate of force increase or how quickly contact is lost, indicating the size of modelling error that exists. Nonetheless, equation (13) and the guideline of a small and decreasing adaptation rate will suffice for now.

A similar derivation may be conducted for the enabling and disabling conditions, resulting in

$$\lambda < -2\Delta\hat{C} \quad (17)$$

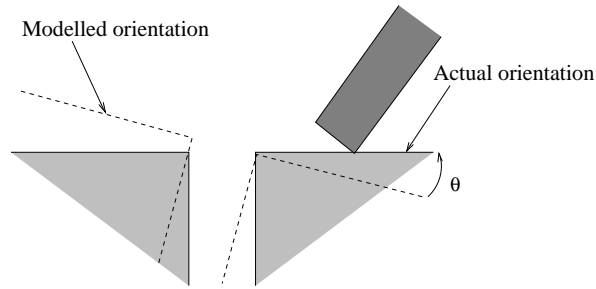


Figure 3: *Two-Dimensional Assembly Task with Orientation Error*

The important result is that only the maintaining condition (equation (16)) depends upon the unknown  $C$ . Equation (17) depend only upon known quantities and is easily satisfied. Thus, equation (12) gives an adaptation law for which the estimated model parameters converge to the actual model parameters provided equation (13) is satisfied. Equation (13) is easily satisfied for the enabling and disabling conditions. However, it is not possible to ensure that (13) is satisfied for the maintaining condition.

## 5 Experiments

To demonstrate the effectiveness of the adaptation process and some convergence characteristics, we will consider the motion control of an automated planar assembly task as depicted in Figure 3. The goal is to maintain contact between the corner of the workpiece and the horizontal surface, as shown in the figure. The

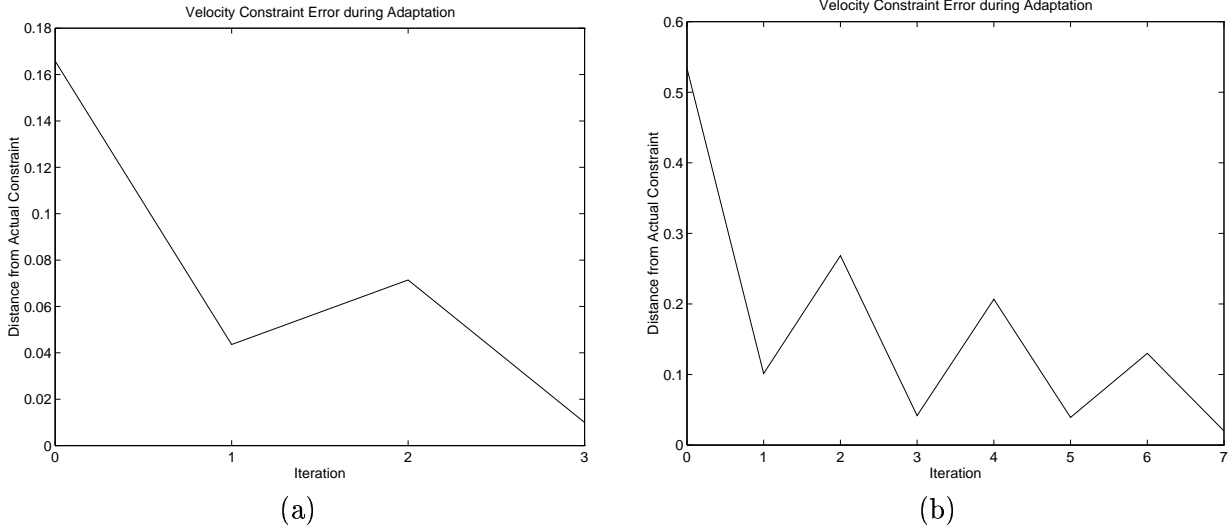


Figure 4: *Simulated Results for Adaptation Law: (a)  $\lambda(0) = 0.2$  and  $\beta = 0.8$  (b)  $\lambda(0) = 0.7$  and  $\beta = 0.8$*

estimated model parameters suggest that the surface is  $10^\circ$  off the horizontal as in Figure 3. This condition easily arises during actual operation due to alignment errors on the incoming fixtures (environments). The difficulty is due to the transport and support mechanisms for the fixtures, which are easily misaligned and often change during the course of a production run. As such, it is a good test case for adaptation.

For adaptation, we need to determine values of  $\lambda$  to satisfy the convergence conditions. It has been shown that, in some cases, the bounds on  $\lambda$  are unknown. The experimental task demonstrated here is such a case. For these cases, an empirical expression for  $\lambda$  has been developed. Using the basic assumption that the error in the estimate of the constraint should decrease exponentially, from (13) we see that  $\lambda$  must also decrease exponentially. The following expression for  $\lambda$  is proposed

$$\lambda(n) = \beta p(\epsilon) \lambda(n - 1) \quad (18)$$

where  $\beta$  is a scaling factor and  $p(\epsilon)$  is the probability of the most commonly occurring error (determined experimentally). The parameters of this expression for  $\lambda$  are the initial value  $\lambda(0)$  and the scaling factor,  $\beta$ . Simulation and experiments support this choice of expression for  $\lambda$  and demonstrate that a wide range of these parameters provide stable convergence.



Figure 5: *Puma 560 Robot*

Figure 4(a) presents the simulated results for the surface following task with  $\lambda(0) = 0.2$  and  $\beta = 0.8$ . These parameter values were selected by trial and error for fastest convergence for a number of different cases. Figure 4(b) demonstrates the graceful degradation of performance as the parameters are varied from the best experimentally determined value. In this case,  $\lambda(0)$  is substantially increased to  $\lambda(0) = 0.7$ . Intuitively, the results of Figure 4(b) demonstrate over-adaptation with the estimated constraint vector overshooting the actual constraint. In this example, the oscillations decrease and the estimated constraint converges to the actual constraint.

The adaptation law has been implemented for a Puma 560 robot (see Figure 5) with considerable tracking errors (it was deliberately not calibrated). The experimental apparatus was constructed from angle iron for the “hole” and machined aluminium for the “peg”. Again, the robot is attempting to move the peg horizontally whilst maintaining contact with the surface, as shown in Figure 3. Figure 6(a) shows the performance of the adaptation law with the experimentally determined decay rate of  $\beta = 0.8$  and with  $\lambda(0) = 0.2$ . Figure 6(a) highlights the potential speed of convergence of the adaptation law. Figure 6(b) presents the experimental results for the surface following task with  $\lambda(0) = 0.2$  and  $\beta = 0.3$ . These

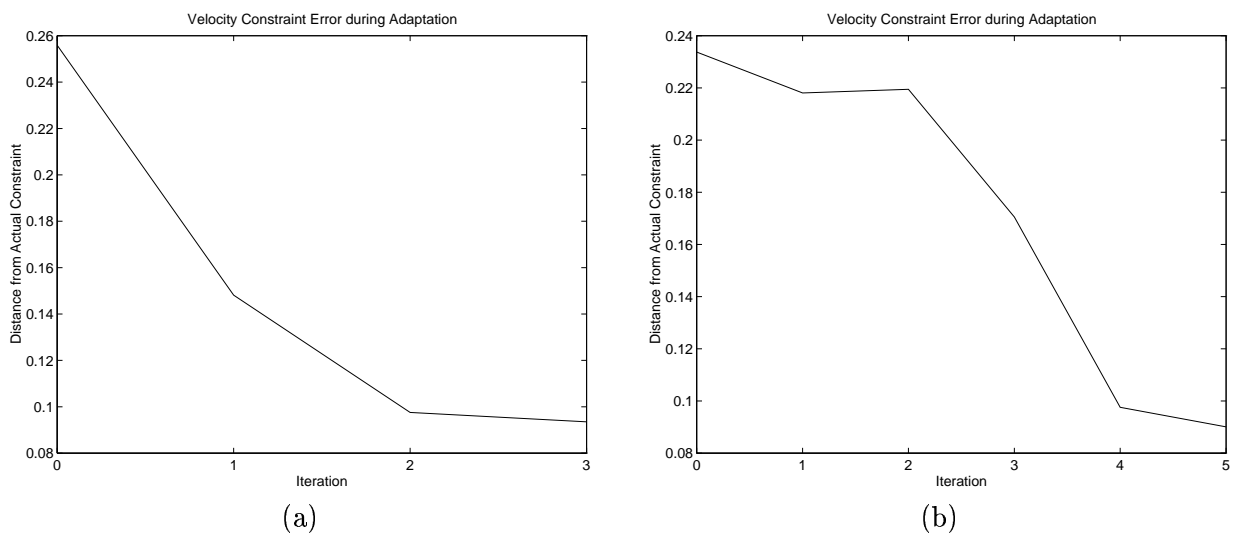


Figure 6: *Experimental Results for Adaptation Law: (a)  $\lambda(0) = 0.2$  and  $\beta = 0.8$  (b)  $\lambda(0) = 0.2$  and  $\beta = 0.3$*

parameter values lead to slower convergence as  $\lambda$  is conservatively small. Also, Figure 6(b) demonstrates a state detection error occurring at iteration 2, a feature of any practical work. This practical example demonstrates that the adaptation law is able to recover from errors elsewhere in the system. Comparing Figures 6(a) and 6(b) we see that the choice of  $\beta$  has a significant impact on the speed of convergence of the adaptation law.

## 6 Conclusions

A adaptation scheme using hybrid dynamic control was applied to constrained motion systems. The algorithm adapted the model-based geometric constraints according to the discrete event execution of the system. A simple adaptation law is shown to cause the error system to be stable in the sense of Lyapunov, thus guaranteeing that the estimate converges to the actual model parameters. Simulations and experiments for a realistic example of automated assembly are given which show that the adaptation law possesses good convergence characteristics. Observations for improving the convergence and error recovery characteristics are given. As the simulations and experiments demonstrate, a highly successful, hybrid dynamic adaptation system has been presented. Moreover, this adaptation ability along with the state monitoring of [4], and the error detection and recovery of [8] form a highly effective, intelligent motion

control system.

## References

- [1] M. J. Best and K. Ritter. *Linear Programming: Active Set Analysis and Computer Programs*. Prentice-Hall, Inc., 1985.
- [2] R. Brockett. Hybrid models for motion control systems. In *Essays on Control: Perspectives in the Theory and its Applications*, pages 29–53. Birkhauser Boston, 1993.
- [3] N. Hogan. Impedance control: An approach to manipulation, parts i, ii and iii. *Journal of Dynamic Systems, Measurement and Control*, 107:1–24, March 1985.
- [4] G. E. Hovland and B. J. McCarragher. Frequency-domain force measurements for discrete event contact recognition. In *IEEE Intl. Conf. on Robotics and Automation*, pages 1166–1171, April 1996.
- [5] D. Jeon and M. Tomizuka. Learning hybrid force and position control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 9:423–431, 1996.
- [6] M. T. Mason. Compliance and force control for computer-controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, June 1981.
- [7] B. J. McCarragher. Task-level adaptation using a discrete event controller for robotic assembly. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, pages 2281–2285, July 1993.
- [8] B. J. McCarragher. The error detection and recovery of robotic assembly tasks. In *Proc. IFAC Symposium on Robot Control, Capri, Italy*, pages 891–896, September 1994.
- [9] B. J. McCarragher. Model adaptive hybrid dynamic control for constrained motion systems. In *European Control Conf.*, volume 1, pages 469–475, September 1995.
- [10] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103:126–133, 1981.
- [11] B. Siciliano and L. Villani. A force/position regulator for robot manipulators without velocity measurements. In *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*, pages 2567–2572, 1996.

- [12] T.-J. Tarn, Y. Wu, N. Xi, and A. Isidori. Force regulation and contact transition control. *IEEE Control Systems*, pages 32–39, February 1996.