

Programming by Demonstration - Constructing Task Level Plans in a Hybrid Dynamic Framework

J. R. Chen

Department of Engineering, FEIT
The Australian National University
Canberra, ACT, 0200, Australia
E-mail: chen@faceng.anu.edu.au

B. J. McCarragher

Department of Engineering, FEIT
The Australian National University
Canberra, ACT, 0200, Australia
E-mail: brenan@faceng.anu.edu.au

Abstract

This paper presents a novel approach to constructing a task level plan for an assembly task from demonstration. A hybrid dynamic system is chosen as an attractive way to model assembly tasks. We propose a framework where mappings required within the hybrid dynamic system model are extracted from a demonstration. In this paper we extract the Event Path Planner mapping, which determines a task level plan. Demonstrated paths are broken down into their base elements, called transitions. A path plan is then constructed from transitions that were well demonstrated. We test the approach by having a robot execute an interesting assembly task using constructed task plans. The framework produces excellent results because, (i) the robot can perform selected paths better than when it simply copies the demonstrator, and (ii) it allows flexible path selection so the robot can be given a disposition in how it executes the task.

1 Introduction

One of the main problems to be solved if robots are to leave their current industrial domain is that of end user programming. An end user with little or no technical expertise needs a method of communicating to a robot what he wants done. A promising solution is Programming by Demonstration (PbD). Here the end user provides a demonstration of the task. A PbD interface then interprets what is to be done, and communicates it to the robot for execution. This is an easy and natural method for humans to program a robot.

In this paper we use PbD to derive a task level plan for assembly. A good task model for this purpose is a hybrid dynamic system [9]. Hybrid dynamic system modelling has found application in a wide variety of fields, including robotic assembly tasks [8]. This form of modelling naturally describes assembly tasks as consisting of a low level continuous system interacting with a higher level discrete system. The continuous system represents the continuous dynamics of task

objects in space. The discrete system represents the task as a set of discrete states, each state corresponding to a specific contact configuration between task objects. The two systems interact because the continuous dynamics will change as the task moves through different contact configurations. The discrete system representation provides an excellent task level description for assembly. We take advantage of its benefits in constructing a task level plan from demonstration.

Robot programming by demonstration is an active research area, however little has been presented on extracting a task level plan from demonstration. Extraction of such a plan is an important first step to PbD because it focusses on high level strategies used by the human to complete the task. Most work in the field has focussed on low level control issues, with no task level model used [1], [6] and [4]. Where task level models were used, demonstrated paths were simply repeated, no attempt was made to identify any good strategies mixed in with the inconsistent or suboptimal actions usual in a demonstration [5], [2] and [10]. Chen and McCarragher have previously looked at extracting a task level plan [3], where the suitability of paths were addressed on a complete path basis. The method was found to be very successful at identifying good demonstrator strategies. In this paper we take a different tack. Plans are selected on the basis of the suitability of the base elements of demonstrated paths, called transitions. This method allows good strategies from different demonstrations to be combined to form a path better than any of those demonstrated.

2 Programming by Demonstration

The task we use for PbD is shown in Figure 1. It involves inserting an axially compressible spindle between two supports. Insertion cannot take place unless the spindle is at least partially compressed. The task is modelled on the household chore of changing rolls on a paper roll holder. The task is also simplified to 2D as motion is only considered in the horizontal plane.



Figure 1: The spindle insertion task chosen for PbD

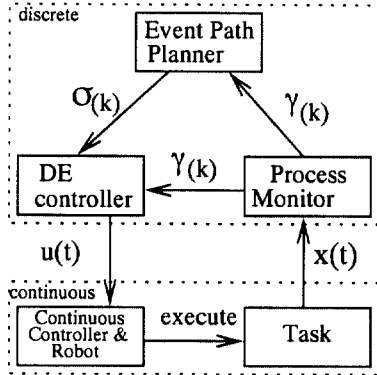


Figure 2: The Hybrid Dynamic System Model

We model the spindle insertion task as a hybrid dynamic system. Figure 2 shows the components of such a system as typically applied to robotic assembly. It consists of the robot and continuous controller, the task (spindle and supports), the Process Monitor (PM), the Event Path Planner (EPP), and the Discrete Event Controller (DEC). The Process Monitor provides the entry point into the discrete part of the system from the underlying continuous system, being characterised by the equations:

$$\begin{aligned}\tau(k) &= \psi(\mathbf{x}(t)) \\ \gamma(k) &= \alpha(\gamma(k-1), \tau(k))\end{aligned}$$

The k th discrete event $\tau(k)$ is identified by the mapping ψ from $\mathbf{x}(t)$, where $\mathbf{x}(t)$ is the position and orientation state vector of the spindle in the underlying continuous system. The k th discrete state $\gamma(k)$ is then determined by the mapping α from the previous discrete state $\gamma(k-1)$ and the event $\tau(k)$ that occurred. For the task presented here, the discrete state $\gamma(k)$ corresponds to one of the spindle-support contact configurations. The event $\tau(k)$ corresponds to a change from one contact configuration to another.

The Event Path Planner is characterised by:

$$\sigma(k) = (\tau_d(k+1), \tau_d(k+2), \dots, \tau_d(n)) = \xi(\gamma(k), \gamma_g)$$

The function ξ maps the current state $\gamma(k)$ and goal state γ_g to discrete event path $\sigma(k)$. $\sigma(k)$ consists of a sequence of desired events taking the assembly process from the current state $\gamma(k)$ to the goal state γ_g . The

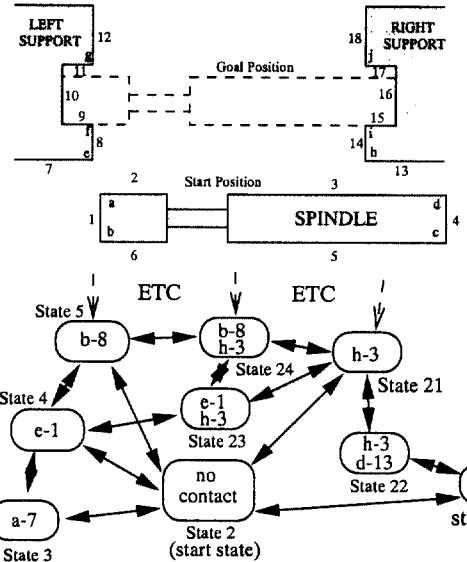


Figure 3: The discrete state automaton of task

components of the path include the next desired event $\tau_d(k+1)$ to the n th desired event $\tau_d(n)$ where n is the number of transitions required to reach γ_g . The d subscript on τ signifies that it is a *desired* event which is not necessarily guaranteed to occur.

The Discrete Event Controller provides an output from the discrete system back into the underlying continuous system. It is governed by the equation:

$$\mathbf{u}(t) = \phi(\gamma(k), \sigma(k))$$

The function ϕ maps the current state $\gamma(k)$ and an event path $\sigma(k)$ to the continuous command $\mathbf{u}(t)$ sent to the robot continuous controller. $\mathbf{u}(t)$ specifies a trajectory which takes the task from the current state $\gamma(k)$ to the next desired state by triggering the desired event $\tau_d(k+1)$ in $\sigma(k)$. The feedback loop is completed with the robot executing in the task space, modifying the state vector $\mathbf{x}(t)$.

The set of possible contact formations for a task form a finite state automaton. The nodes in the automaton represent contact formations. The arcs represent transitions from one contact formation to another. To construct the automaton for our task, support vertices were labelled with lower case letters, while edges were labelled with numbers (see Figure 3). Contact formations could then be described by vertex-edge pairs (eg, a-7 corresponding to the front left corner of spindle in contact with front edge of left support). A state was defined as a unique contact formation, ie. a unique vertex-edge pair or group of vertex-edge pairs. To simplify state referencing, each was given a unique state number (eg. a-7 was labelled γ_3). In total the automaton for the task had 74 states.

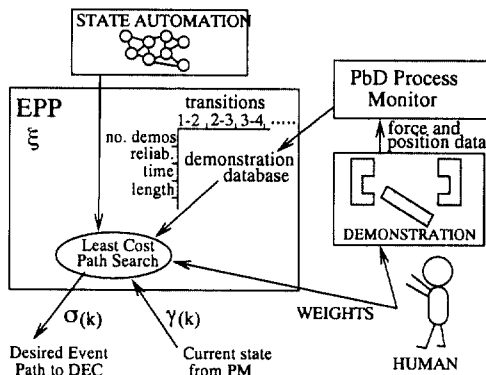


Figure 4: The Event Path Planning Framework

3 Event Path Planner Synthesis

We have seen that the hybrid dynamic system is defined by four mappings, ϕ , ξ , ψ and α . In order for the robot to execute a task, each mapping needs to be extracted from the demonstration information. Here we extract the EPP mapping ξ in order to obtain a task level plan for the spindle insertion task. Figure 4 shows the framework we use.

A demonstration is first provided by the human. This produces force and position data via a position sensor mounted on the spindle and two force sensors, one beneath each support. The state path which corresponds to this data is extracted by the PbD Process Monitor. This is a distinct element of the system to the Process Monitor presented in the previous section. The PbD process monitor operates only in the demonstration phase. Its role is to convert the continuous data from the demonstration (force/position) into a discrete level representation, i.e. a state path. The conversion is achieved manually here, but automatic conversion is possible [7] (note that only manual conversion for demonstrated states was required, not for all 74 states in the automaton). The demonstration database was then constructed from the demonstrated state paths and information in the state automaton.

The demonstration database records the desirability of each transition based on how it was demonstrated. It is a two dimensional array consisting of demonstrated transitions versus four key performance measures. The four performance measures chosen were: the number of times the transition was demonstrated (N), the length of the transition (L), the average time taken for the transition to be completed in the demonstration (T), and the reliability of the transition (R).

The database is used to assign a cost to the transitions in the discrete state automaton. If a transition is performed well, it can be assigned a low cost. Badly performed transitions can be assigned a high cost. The

EPP mapping ξ can then be achieved by performing a search for the least cost path $\sigma(k)$ through the automaton between the start state and goal state. To combine a transition's performance in each of the four areas, we define:

$$C_{i,j} = W_N C_{N_{i,j}} + W_T C_{T_{i,j}} + W_R C_{R_{i,j}} + W_L C_{L_{i,j}} \quad (1)$$

where $C_{i,j}$ is the overall cost of transition $\tau_{i,j}$ between states i and j . The costs $C_{N_{i,j}}$, $C_{T_{i,j}}$, $C_{R_{i,j}}$, $C_{L_{i,j}}$ are derived from how transition $\tau_{i,j}$ was demonstrated in the N , T , R and L performance areas respectively. The cost $C_{N_{i,j}}$ was defined as:

$$C_{N_{i,j}} = N_{max} - N_{i,j} + 1$$

where N_{max} was the number of times the most demonstrated transition was demonstrated, and $N_{i,j}$ was the number of times $\tau_{i,j}$ was demonstrated. This converts the number of demonstrations into a cost where the most demonstrated transition has a cost of one and other transitions a cost greater than one.

The cost $C_{R_{i,j}}$ was defined as:

$$C_{R_{i,j}} = A_i S_i + 1$$

where A_i is the number of degrees of freedom in state i and S_i is the number of transitions to other states from state i . High reliability implies successful execution of an attempted path. Unsuccessful execution results from incorrectly positioning the spindle relative to the supports. Spindle positioning is more certain when its motion is constrained by the supports. On this basis we use the spindle's degree of freedom in the state prior to the transition as the basis for our measure of its reliability. The greater the number of degrees of freedom, the less reliable the transition. A_i is multiplied by the number of possible next states S_i because incorrect positioning could cause any of the transitions out of that state to occur.

The time taken for a transition and the length of a transition convert naturally to costs and were defined simply as:

$$C_{T_{i,j}} = Tav_{i,j} \quad \text{and} \quad C_{L_{i,j}} = L_{i,j}$$

where Tav is the average time taken for transition $\tau_{i,j}$ in the demonstration, and $L_{i,j}$ is the length of the transition. The length of each transition is assigned a value of one, which has the effect of promoting a short event path. This will be the case since any prospective path will have a component of its cost equal to the number of transitions in the path.

The weights W_N, W_T, W_R, W_L in (1) allow the human to specify what aspect of performance is important. This gives the robot a disposition. If a short execution time is important, W_T can be increased relative to the other weights and a path with a short

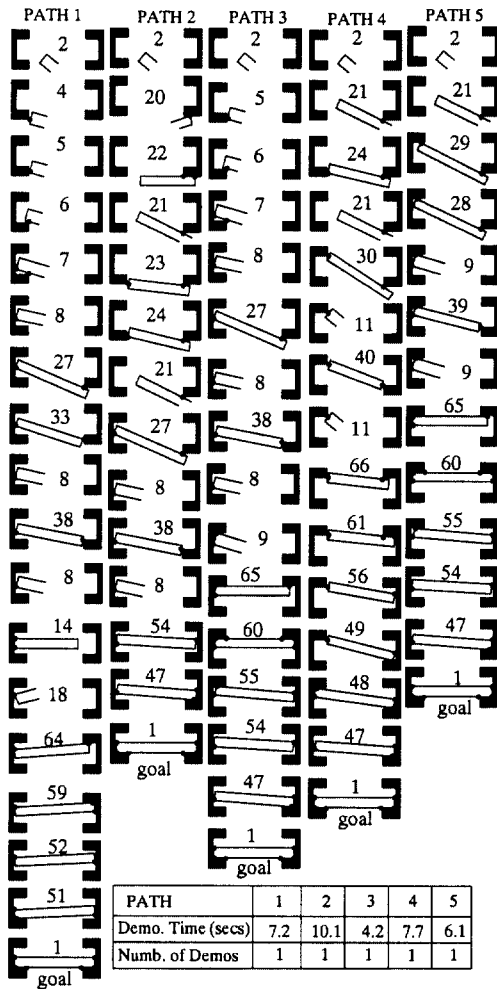


Figure 5: Demonstrated paths, with state numbers

execution time will be selected. The robot can also be made to focus on reliability, on a path with few transitions, or on what the demonstrator did most often.

4 Selecting an Event Path

The human was asked to provide five demonstrations (see Figure 5). The number five was chosen as a trade off between the number a human could be reasonably expected to provide, while still obtaining reasonable information content. The demonstration was captured using a Polehumus position sensor mounted on the spindle, and two force sensors, one beneath each support. We believe force and position sensing is the most appropriate form of sensing for capturing demonstrations of assembly tasks. Vision is another popular sensing medium [5], however in many circumstances it is impossible to see what contacts and fine motion is occurring because they are obscured by the task objects themselves. Given the human would use vision to some extent to complete the task, we blindfold him during the demonstration. Asada and Lui [6] identify

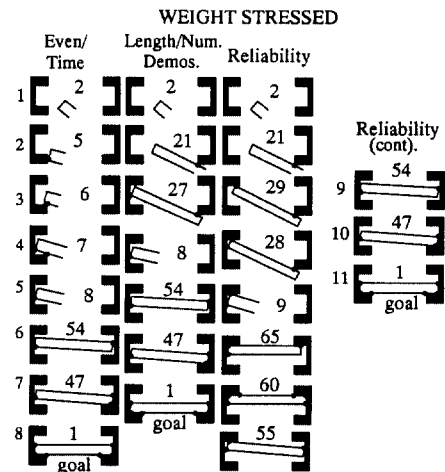


Figure 6: Event Path Planner selected paths

that all features used by the human to complete the task should be capturable by the sensor system used to record the demonstration. If this is not the case, perceptions used by the human to decide on a certain task action will not be available to the robot.

Once the demonstration database was built, the EPP could select paths by constructing them from the demonstrated transitions. A major feature of the EPP is its ability to select paths with a certain disposition according to the values given to the set of weights. To determine the sensitivity of the path selected to the weights' values, each was varied independently of the others to give the path selections shown in Figure 6.

When all weights were made the same the EPP selected path 2-5-6-7-8-54-47-1, which as you would expect is a middle of the road path. Its attractiveness lies mainly in its short length and low time for execution, although many of the path's components were also frequently demonstrated. For example the sequence 54-47-1 was a favoured final approach to the goal state.

When length or number of demonstrations were stressed the EPP selected path 2-21-27-8-54-47-1. This is a good selection with regard to path length because it is shorter than 2-5-6-7-8-54-47-1. In fact it is the shortest path possible between the start and goal states that can be constructed from demonstrated transitions. The path is a good selection from a number of demonstrations point of view also. In addition to the well used 54-47-1 path segment, it contains the 2-21 transition which was used frequently by the demonstrator when leaving the no contact state. The demonstrator used this transition often because the large side area of the spindle meant the desired contact between the vertex and support could be easily completed over a wide range of spindle positions.

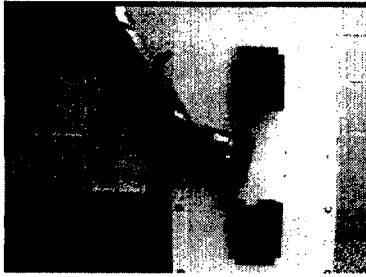


Figure 7: Eshed robot executing EPP selected paths

When reliability is stressed, path 2-21-29-28-9-65-60-55-54-47-1 is selected. Sequence 9-65-60-55-54 provides a reliable way to insert the non-compressible end of the spindle into the right support. It is reliable because a minimum of two contacts are maintained at all times which provides more constraint to help guide motion between spindle and support. In this regard it is better than the other sequences used to achieve this insertion (eg. 8-38-8-54 in Path 2 and 8-38-8-16-64 in path 1) because state 8 results in only one contact point.

When time is stressed, the EPP returns to selecting 2-5-6-7-8-54-47-1. This is due to the speed with which the human was able to demonstrate the 2-5-6-7-8 and 54-47-1 state sequences. The 2-5-6-7-8 sequence is fast because of the straight line motion it involves, with major changes in inertia not required when moving from state to state. This compares with the other main method for inserting the compressible end of the spindle of 2-21-27, where different directions of motion are required and the distance to be moved greater.

It should be noted that none of the paths selected by the EPP were demonstrated. Its paths were constructed from good components of different demonstrated paths according to what disposition was desired. This highlights a major strength of our EPP method. Good components of different paths can be combined to produce paths better than any of those demonstrated.

5 Implementing EPP Selected Paths

In order for the robot to execute selected paths, the Process Monitor (PM) and Discrete Event Controller (DEC) were implemented. The EPP is the first part of our research into PbD within a hybrid dynamic system framework, hence the details of the DEC and PM are still yet to be finalised. To allow experiments to take place the DEC and PM were achieved using simple schemes which worked well enough to highlight the benefits of the EPP. Process monitoring was achieved manually by a human using a graphical user interface. The human entered a new state when he saw that it had been reached. We have noted our objections

Path Name	Selected by Weight	Successful Executions	Execution Time (secs)
Demonstrated Paths			
Path 1	n/a	0/5	n/a
Path 2	n/a	2/5	72.6
Path 3	n/a	3/5	40.2
Path 4	n/a	2/5	45.4
Path 5	n/a	4/5	52.6
Event Path Planner Paths			
n/a	Time	3/5	23.3
n/a	Reliab	4/5	42.9
n/a	N, L *	3/5	33.6

* N = Number of demonstrations, L = Length

Table 1: Results of implementing selected paths on the robot

to vision as a sensing medium in assembly. Vision is possible for this particular task only because of its planar nature. A good view could therefore be obtained from directly above. A DEC was implemented by directly copying the demonstrator's paths in each state (x, y , and θ of the spindle). These paths were recorded relative to the demonstrator's initial spindle position at point of entry to the state. The robot then applied these paths from whatever point of entry position it experienced in that state at execution time. This approach coped well with small discrepancies in task position-orientation between the demonstration and execution phases.

Table 1 shows the results of our experiments. Both demonstrated and EPP selected paths were executed by the robot. This allowed a comparison to be made between our EPP framework and the usual scheme of directly copying demonstrated paths. Five attempts at each path were made.

Robot execution of demonstrated paths was broadly successful. The robot successfully executed paths 2,3,4 and 5 at least once. Path 1 proved more problematic because of the difficult transition 27-33. Robot attempts at this transition resulted in the sequence 27-8-33, ie. the spindle lost contact with the support before making contact again in state 33. The human executed this transition by maintaining a contact force between the spindle and support in state 27. This force naturally moved him into state 33 once the constraints changed. The velocity command output by the DEC resulted in little contact force in state 27, with the robot's inertia moving the task into state 8 once the constraints changed. This difficulty resulted from the lack of force control by the DEC, and also caused some failures in other paths. It affected the overall reliability of some paths, however in it did not impede us from showing experimentally the validity of our reliability definition.

The robot performed EPP selected paths very well. In general they were much better performed than the demonstrated paths. Our definition of a transition's desirability in each performance area proved to be an accurate reflection of how the robot performed it. The fast transition sequences 2-5-7-8 and 54-47-1 were able to be executed quickly by the robot and formed the basis for why the time path (2-5-6-7-8-54-47-1) was executed at least one third faster than any other. The reliability result was obscured a little by the lack of force control in the DEC, however the experiments were still able to show the validity of our reliability definition. The reliability path (2-21-29-28-9-65-60-55-54-47-1) was successfully executed four times out of five, as high or higher than any other. Sequence 9-65-60-55-54 formed the basis of its reliability because robot motion was better guided by the extra constraints that this sequence contains. The presence of 2 constraints in state 9 helped anchor the left side of the spindle as the right side was inserted. Alternative paths used state 8 which contained one contact only. For example, failures in Path 2 were caused by the left end of the spindle slipping from state 8 to state 9 or 7 (in sequence 27-8-38) due to this lack of constraint (see Figure 5). The validity of the path length and number of demonstrations definitions were also confirmed by the experiments. Path (2-21-27-8-54-47-1) was obviously the shortest path executed by the robot, the small number of states making it relatively fast and also easy to process monitor. This path also allowed the robot to use many of the demonstrators favourite transitions. The resulting path was better than any demonstrated, one the human had still not thought of.

6 Conclusion

Our methods for synthesising the Event Path Planner from demonstration have been successful. The EPP identified good strategies used by the human across a number of demonstrations, and grouped them together to construct a path better than any demonstrated. The value of such an approach was confirmed in experiments. The robot performed better using EPP paths than when copying demonstrated paths directly, the usual scheme in the literature to date. Further, the EPP allowed a very flexible path selection regime. It could be made to construct a path out of human strategies good in one particular performance area. In this way the robot's disposition could be tailored according to the circumstances. A reliable disposition is ideal where penalties for failure are high. If failure holds little consequence, then fast execution can be promoted. If the demonstrator was an expert, the robot can be made to copy him. The EPP could construct these paths even if the human never explic-

itly demonstrated a complete path with such a disposition. Experimental results confirmed the flexibility of the approach. Dispositions specified to the EPP at the path selection stage translated into actual robot dispositions at execution time. Validation of our EPP synthesis method was achieved in spite of a limited DEC regime. Some execution failures were caused by poor DEC performance, specifically its inability to specify fine motions where force control was necessary. These difficulties provide impetus for the next stage of our research of DEC synthesis from demonstration. The rectification of these DEC issues promises to augment the performance of our PbD system by complementing the already excellent EPP performance presented in this paper.

References

- [1] Christopher G. Atkeson and Stefan Schaal. Learning tasks from a single demonstration. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 1706–1712, May 1997.
- [2] Nathan Delson and Harry West. Robot programming by human demonstration: Adaptation and inconsistency in constrained motion. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996.
- [3] J.Chen and B.J.McCarragher. Robot programming by demonstration, selecting optimal event paths. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 0–0, April 1998.
- [4] M. Kaiser and R. Dillman. Building elementary skills from human demonstration. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 2700–2705, April 1996.
- [5] K.Ikeuchi, M.Kawade, and T.Suehiro. Toward assembly plan from observation, task recognition with planar, curved and mechanical contacts. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2294–2301, 1993.
- [6] Sheng Liu and Haruhiko Asada. Transfer of human skills to robots: Learning from human demonstrations for building an adaptive control system. In *Proceedings of the 1992 American Control Conference*, pages 2607–2612, 1992.
- [7] Brennan J. McCarragher and Haruhiko Asada. Qualitative template matching using dynamic process models for state transition recognition of robotic assembly. *Journal of Dynamic Systems, Measurements and Control*, 115(2):261–269, June 1993.
- [8] Brennan J. McCarragher and Haruhiko Asada. The discrete event modelling and trajectory planning of robotic assembly tasks. *Journal of Dynamic Systems, Measurements and Control*, 117(3):394–400, October 1995.
- [9] R.W.Brockett. Hybrid models for motion control systems. In H.L.Trentelman and J.C.Willems, editors, *Essays on Control: Perspectives in the Theory and Its Applications*, chapter 2, pages 29–5. Birkhauser, Boston, MA, 1993.
- [10] Marjorie Skubic and Richard A. Volt. Learning force based assembly skills from human demonstration for execution in unstructured environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1282–1288, May 1998.