Acquiring Robust, Force-Based Assembly Skills from Human Demonstration

Marjorie Skubic, *Member, IEEE*, and Richard A. Volz, *Fellow, IEEE*

I dedicate this paper to Professor Dick Volz. I have fond memories of our working together to edit this paper. I learned a great deal from that process. Thanks for being a good teacher and role model and for giving me the confidence I needed.
-Marge

Acquiring Robust, Force-Based Assembly Skills from Human Demonstration

Marjorie Skubic, *Member, IEEE*, and Richard A. Volz, *Fellow, IEEE*

***Abstract*—Robots have been used successfully in structured settings, where the environment is controlled; this research is inspired by the vision of robots moving beyond structured, controlled settings. The work focuses on the problem of teaching robots force-based assembly skills from human demonstration. To avoid position dependencies, force-based discrete states (contact formations) are used to describe qualitatively how contact is being made with the environment. Sensorimotor skills are modeled using a hybrid control model, which provides a mechanism for combining continuous low-level force control with higher-level discrete event control. A change in qualitative, discrete state constitutes an event and triggers a new control command to the robot, which moves the assembly toward a new contact formation. In this way, the skill execution is not dependent on absolute position but rather responds to changes in the force-based qualitative state. Experimental results are presented which validate the approach and show how skill acquisition can be accomplished even with an imperfect demonstration.**

***Index Terms*—Assembly, contact formation, discrete event systems, robot programming.**

I. Introduction

ROBOTS have been used successfully in manufacturing settings, where the environment is very structured and the tasks performed are repetitive and relatively simple. As long as the environment is controlled and the workpieces are confined to precise positions and orientations, the robot can continue to execute its task. Our research is inspired by the vision of robots moving beyond structured, controlled settings, and still performing successful assembly operations.

Although many challenging problems still exist, it is realistic to think of robots being used as intelligent tools and assistants which make the job easier for the human worker. Depending on the nature of the work, the robot may be asked to perform repetitive or nonrepetitive tasks. If it is a new task not performed previously, the robot may need some quick instruction or task refinement (i.e., programming), most likely done at the work site with instruction provided by the actual user rather than a specialized robot engineer.

As a means of providing a fast, interactive method of programming force-based assembly skills, we have been investigating robot programming by demonstration (PbD). To deal with uncertainties in position and orientation, as well as inconsistent training data, we have used an event-based approach, utilizing discrete event control as suggested in [1] and [2]. Unlike the approach used in [1], however, we do not use detailed geometric information. The research discussed here addresses the problem of learning force-based assembly skills by studying sensory patterns of demonstrated tasks, concentrating on the part of the assembly operation which involves making contact with the environment. Sensorimotor skills are modeled using force-based discrete states, which describe qualitatively how contact is being made with the environment.

Two characteristics distinguish our work from similar work (e.g., [2]). First, our hybrid control model combines reference velocity commands with force control. Second, our interactive PbD interface does not require a perfect task demonstration for skill acquisition to take place.

II. Related Work

As background material, we describe related work in PbD of assembly skills, in three specific areas: vision-based, simulation-based, and force-based systems.

A. Vision-Based Systems

Vision has been proposed as a passive modality to observe a human demonstrator performing tasks in a natural setting. Kuniyoshi *et al.* track the demonstrator's hand and extract a high-level task plan [3]. Ikeuchi and Suehiro use vision to recognize object configurations before and after the demonstrated assembly [4]. Face contact relations between the grasped object and its environment are analyzed. Contact transitions are represented as a directional graph.

Resolution limitations in the sensing make it difficult to extract the fine motion plan needed to perform assembly operations. Paul and Ikeuchi use thresholds to make assumptions about contact points, and then compute the motion path in C-space [5], [6]. However, the motion plan is based on position information, and, in order to succeed, requires accuracy in the robot motion and in obtaining object positions.

More recently, Miura and Ikeuchi have extended the work on face contact relations to support four degrees of freedom (three translations and one rotation) [7]. The contact transitions are categorized according to whether they need visual feedback or only passive compliance. It is assumed that the robot can perform passive compliant motions. A laser range finder is used to

Manuscript June 18, 1998; revised September 20, 1999. This paper was recommended for publication by Associate Editor K. Kosuge and Editor S. Salcudean upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant CDA-9115123, Grant CDA-9422123, and Grant EID-9017249 and by the State of Texas under TATP Grant 999903-267 and Grant 999903-095. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998.

M. Skubic is with the Computer Engineering and Computer Science Department, University of Missouri–Columbia, Columbia, MO 65211 USA (e-mail: skubic@cecs.missouri.edu).

R. A. Volz is with the Department of Computer Science, Texas A&M University, College Station, TX 77843-3112 USA.

Publisher Item Identifier S 1042-296X(00)10239-3.

achieve adequate resolution for the contact transitions requiring visual feedback.

B. Simulation-Based Systems

A virtual environment can also be used, allowing the demonstration to be performed in an interactive, simulated world. Onda et al use a simulated environment to learn assembly strategies by extracting a sequence of contact states (i.e., contact formations) [8]. The contact state is determined by the collisions of the geometric models that occur as the demonstrator moves the virtual workpieces. The approach creates a symbolic representation of the assembly process; the operator can also refine the plan by pruning out contact states deemed unnecessary. However, problems can still occur when the plan is executed in a real robot environment, unless the workpiece positions are known accurately. Preprogrammed skill primitives which move the robot from one contact state to another are offered as a solution to this problem.

Lloyd *et al.* propose a simulated demonstration environment which incorporates contact dynamics [9]. The demonstrated path is deformed to create a smoother path, repelled by some contact surfaces and attracted to others as desired. The approach assumes good registration between the physical environment and the virtual environment (acquired by vision) and also assumes an execution monitor that verifies contact states using force monitoring.

C. Force-Based Systems

Force information has been used in PbD as a means of generating more robust programs in contact situations. In [10], Asada and Izumi use teaching data to generate a program for placing an object at the corner of a rectangular box. The human directs the robot with a hand controller, demonstrating translational trajectories for hybrid position/force control. Constraint surfaces are restricted to being planes and must be perpendicular to each other.

Kosuge *et al.* use the same task and extract two levels of control strategies from the teaching data as follows: 1) a high-level sequence of discrete states (i.e., contact formations) and 2) the compliant motion strategies used to move between discrete states (modeled as damping control) [1]. The sequence of contact formations is known *a priori* and identified from the monitored force profile, using detailed geometric information [11].

Hovland *et al.* use discrete events to capture an assembly operation, where the events are changes in contact state [2]. They use a discrete event controller and assume the existence of a process monitor that identifies contact changes [12]. A hidden Markov model (HMM) is used to model assembly motions; the Markov states correspond directly to the discrete states (i.e., contact formations). The goal is to learn the velocity command which allows the robot to achieve the next desired discrete state. Thus, different task strategies can be demonstrated and learned within the same framework; if one strategy fails because of position or orientation uncertainties, another strategy can be determined on-line. However, the HMM requires the velocity commands to be discretized, which may be impractical if extended beyond the planar task used.

Delson and West teach a compliant motion program by using a virtual trajectory to model the human demonstrator's task strategy [13]. Wang and De Schutter incorporate an auto-pilot into the demonstration system to make it easier for the operator to demonstrate "perfect" skills [14]. However, in both cases, only one strategy is learned by the robot, and the task execution will fail if the orientation of the environment is significantly different from that at demonstration time. Delson and West address this problem in part by reorienting the environment and repeating the demonstration using the same task strategy [15].

Asada proposed a three-layer neural network (NN) to learn the nonlinear mapping of measured force to corrected motion (velocity commands) [16]. The approach was demonstrated on a planar assembly task, using simulated data to train the neural network.

Whalen implemented this NN approach for a planar edge-mating task [17] and showed that it was successful when using simulated training data, but failed when using actual human demonstrated data. One possible reason for its failure might be that human demonstrators can generate inconsistent motions unintentionally. As a result, the training sets may contain greatly varying output data for similar inputs, which are particularly difficult to use for NN training. Examples of such inconsistent motion are shown in Section IV.

Koeppe *et al.* proposed a modified NN for learning compliant motion strategies from human demonstrations [18]. Velocities and force signals are input to the network in the form of fuzzy variables (e.g., small, medium, large or negative, zero, positive). Koeppe showed that velocities should be included to solve the correspondence problem (i.e., delay) between the human's haptic perception and the motion correction. The results in a simulated environment have been promising; however, the execution environment must be identical to the demonstration environment.

Kaiser *et al.* also use NNs for sensorimotor skill acquisition from human demonstrations [19]. To overcome unintentional inconsistency demonstrated by the human teacher, the training data is first preprocessed. Inconsistent sample (input, output) pairs are removed, and a smoothing algorithm is applied. Also, reinforcement learning is used for skill refinement after the network has been trained. The method shows promise but has a lengthy training process. More recently, Friedrich *et al.*have proposed an interactive programming environment that captures user intention, such as object selection, by object related relations [20].

Although not usually reported in the PbD literature, the aspects dealing with force control bear some relation to the use of compliance mechanisms to complete assembly operations [21]. The force control mechanisms used in PbD typically have the goal of accommodating uncertainties in object positions. Passive compliance techniques [21] also have the goal of accommodating minor uncertainties in the relative position of objects. Programming by demonstration, however, may address motions over large portions of the workspace, whereas compliance mechanisms are normally focused on the fine motions during the final stages of an assembly operation. From the active compliance perspective [22], our use of force sensing in PbD

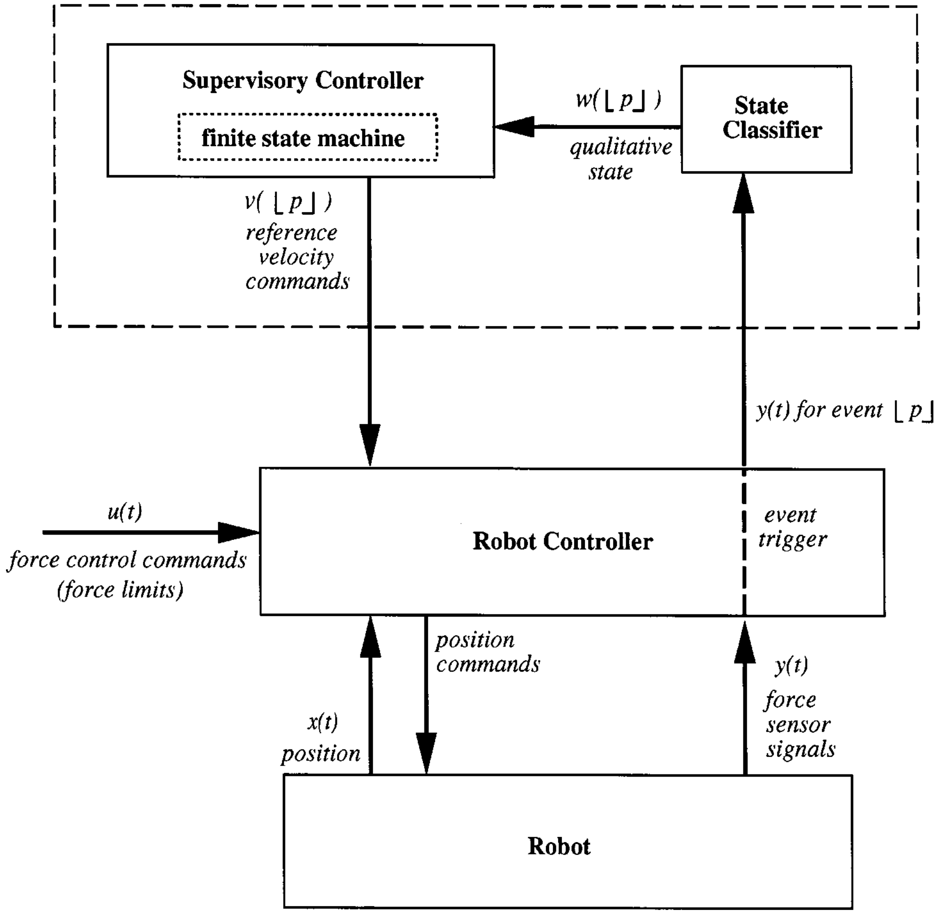

Fig. 1. Hybrid control model of an assembly skill.

can be viewed as imposing a higher-level strategy and control on traditional active compliance control.

III. Skill Model

The mathematical framework used for our robot skill model is the hybrid control model proposed by Brockett [23], in which symbol processing interacts with signal processing. The model, with special application toward modeling force-based assembly skills, is shown in Fig. 1. It consists of the following four parts: 1) the robot; 2) the robot controller; 3) the state classifier; and 4) the supervisory controller. Symbolic processing is done at the supervisory controller level, where both the inputs and outputs are in the form of symbols. The input to the supervisory controller is generated by the state classifier, which converts a time-varying signal into a symbolic representation of a qualitative state. The times at which this occurs are driven by an event trigger, which indicates that the system has changed to a new qualitative state.

Let x be the position of the robot end effector, y be the reading from the robot's force sensor, and u be the force control command. Then x, y, and u are vector-valued functions of time, which take on values in the subsets of Cartesian spaces X, Y, and U, respectively. Following Brockett's convention, we let p be a monotone increasing triggering signal, a function of time, and write $\lfloor p \rfloor$ to denote the largest integer less than or equal to p. Then $\lfloor p \rfloor$ may be viewed as an event index. Let $v(\lfloor p \rfloor)$ be the symbolic controller command (representing the reference velocity command), and $w(\lfloor p \rfloor)$ be the symbolic input to the supervisory controller (representing the qualitative state in the assembly process). The symbolic variables, v, and w take on values in the sets V and W, respectively, where V is the set of reference velocity commands and W is the set of force-based qualitative states. Let δ be the sampling time period for the robot controller and k be the integer sampling increment. The following equations describe the model:

$$\begin{aligned} x(k\delta+\delta) &= a(v(\lfloor p \rfloor),\, x(k\delta),\, y(k\delta),\, u(k\delta)) \\ y(k\delta) &= c_e(x(k\delta),\, u(k\delta),\, v(\lfloor p \rfloor)) \\ p(k\delta+\delta) &= r(y(k\delta)) \\ w(\lfloor p \rfloor) &= h(y(k\delta)). \end{aligned}$$

The position x is dependent on the symbolic controller command and the values for the position, force sensor reading, and force control command at the last sampling period. The force sensor reading y is dependent on the position, the force control command, and the symbolic controller command. The subscript e on c emphasizes that the force sensor readings will also depend on the robot's environment. The rate equation r which describes the increase in the triggering signal p acts as an event trigger and indicates when the system has changed to a new qualitative

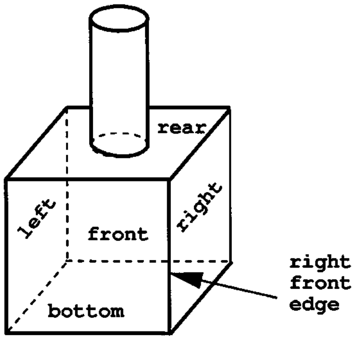

Fig. 2. Test workpiece showing descriptive SECF names.

state. The rate equation is constrained so that no more than one event may occur during the sampling period δ. The qualitative state w is dependent on the force reading.

The qualitative state drives the assembly skill; it takes the form of a single-ended contact formation (SECF). Contact formations [24] provide a qualitative description of how two or more objects contact each other (e.g., edge 1 of one object touches side B of another object). In contrast, single-ended contact formations provide a one-sided description of how a grasped object touches its environment (e.g., edge 1 of a grasped object touches any side in the environment). Descriptive names are given to each SECF to provide a meaningful link to the human demonstrator (e.g., *bottom* surface or *right front edge*, as shown in Fig. 2). Two sensor-based methods which can be used for SECF classification are described in previous work [25]. The mapping is performed by the state classifier every time the event trigger detects a change in SECF state.

As shown in Fig. 1, there are two levels of control. At the higher, symbolic level, SECF transitions are used to establish a nominal plan. A supervisory controller drives the assembly, using a finite state machine (FSM), represented as a directed graph. The states stored in the FSM correspond to the SECFs that describe the qualitative condition in the assembly process. The FSM stores the sequence of SECFs which are used by the human demonstrator in the assembly skill. The skill model also includes the controller output function, which generates (relative) reference velocity commands that describe how to transition to the next desired SECF (as discussed in Section IV).

At the lower level, force control is included to facilitate compliant motion. The robot controller regulates the robot using a position control loop; an outer force control loop is used to enforce force and moment limits. If a force exceeds the threshold value, then the robot controller attempts to control the force to a set value using an impedance controller based upon the underlying position control. This has the effect of providing local control as deflections from the nominal trajectory, to accommodate an unstructured environment. Compliant motion is not explicitly programmed but occurs naturally as a result of the force control accommodating environment uncertainties. Overlaid on the force control is a guarded move mechanism in which, if motion cannot continue without excessive forces, then a force guard is triggered. Encountering a force guard, in turn, triggers the SECF classifier and the higher-level supervisory controller. Transitions between SECFs essentially result from guarded move triggers.

The skill model is designed to support six-degree-of-freedom (DOF) motion but does assume that a reliable SECF classifier is available. SECFs may be vertex-to-surface, edge-to-surface, surface-to-surface, or combinations thereof. The operator demonstrates an assembly skill by showing a nominal SECF sequence. The motion to proceed from one SECF to the next desired SECF can be either translation (sliding) or rotation (pivoting) or a combination. However, we have observed that it is generally easier for the operator to demonstrate a motion that is either mostly translation or mostly rotation, as the operator tends to use the constraints of the environment to aid in the assembly operation.

Because we are not making any assumptions about the position and orientation of the robot's environment,[1] we are also not making guarantees that the desired goal condition will be reached. However, we can increase the likelihood for success, in spite of position uncertainties, by demonstrating more than one SECF sequence and incorporating each into the assembly plan graph. Each sequence results in the same goal SECF, and the sequences are merged at this goal state, as described in Section IV-C. A skill may, in fact, contain several SECF sequences that will result in a successful task completion. The actual sequence used during execution will be dependent on the position and orientation of the environment workpieces. The critical issue is to include in the assembly graph the possible SECFs that may be encountered, given position uncertainties; including more SECF sequences will provide robustness in spite of additional environment displacement. Ausuti and McCarragher have shown that the system will in fact converge to the goal state if all desired contact transitions are enabled and if the resulting graph has no loop [26]. This assessment, of course, requires analysis based on the task geometry.

IV. Skill Learning

As modeled here, the learning of an assembly skill involves the learning of the following three functions: 1) the mapping of force sensor signals to SECFs; 2) the sequence(s) of SECFs; and 3) the transition velocity commands which move the robot from the current SECF to the next desired SECF. The first function is acquired using supervised learning. The operator demonstrates each SECF while force data is collected, and the data is used to train a state classifier. The operator then demonstrates a skill, and the classifier is used to extract the sequence of SECFs and transition velocities which comprise the rest of the skill.

In previous work, we focused on learning the first function, the mapping of the force sensor signals to SECFs. Two methods of classification were presented; one method uses fuzzy set theory [27], and one uses a neural network [28]. In this paper, we focus on learning the remaining two functions, illustrating the process with an example of an actual skill demonstration.

One advantage of this two-step approach is that we have a partial skill model when the task demonstration is performed, and this information can be used during the demonstration. Feedback is provided to the operator by classifying and displaying the SECF sequence in real-time; the symbolic representation of

[1] Indeed, the assumption here is that we do *not* know the position and orientation of objects in the environment.

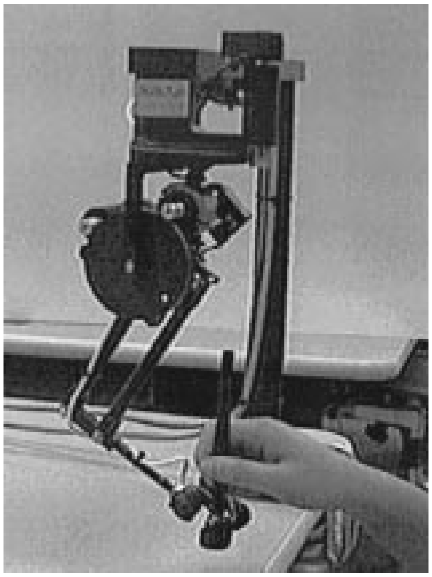

Fig. 3. The PHANToM, as used by the operator for skill demonstration.

the SECF is useful for providing meaningful and easily recognizable feedback. This additional feedback can be used to increase the efficiency of the skill acquisition, because the operator knows what sequence is being learned as the demonstration is performed. If the sequence is not the desired one, then corrections can be made immediately.

One underlying assumption is that the pose of the grasped object (w.r.t. the force sensor frame) is the same for both the SECF classifier training and the demonstration of the skill, which in turn must agree with the pose used in the execution of the skill. For example, if the orientation of the grasped object changes significantly after the SECF classifier has been trained, the classifier will no longer correctly identify the SECF state. This issue has been addressed in a previous paper [30], in which we describe a method of retraining the classifier to accommodate a new object pose. A small sample of force signals is collected for 2 SECFs in the set, a transformation is calculated, and the previous training data are transformed to achieve training for the new pose. Alternatively, each new force reading can be transformed and used with the original classifier. Another possible solution is to use a geometry-based approach for the SECF classification (e.g., [31]) and use position information to accommodate changes in object pose.

We describe the remainder of the skill acquisition process by describing our experimental procedure. The workpiece is a plastic, square block, as shown in Fig. 2. The classifier has been trained to identify 17 SECF classes. After training, the classifier performs at a 97% success rate with static data.

The demonstration of the skill was performed using the PHANToM haptic interface, shown in Fig. 3, as a force-reflecting hand controller to teleoperate an American Robot *merlin* manipulator in 6-DOF motion. A JR3 force sensor provided force signals. The operator was located in front of the robot, with full view of the workspace.

The environment is an open, rectangular wooden box, made of soft, bare wood. Although it has been sanded, it has a relatively high coefficient of friction. This was intended, as we specifically wanted to include the effects of friction in the experiments. The attached board can be clamped to the worktable in a variety of positions and orientations to generate different starting conditions.

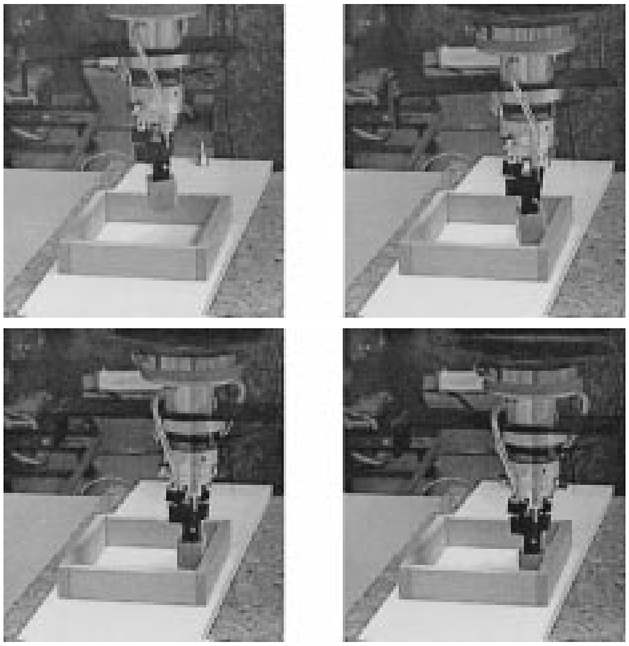

Fig. 4. The SECF steps for sequence A.

A. Learning a Sequence of Single-Ended Contact Formations

Snapshots of the sequence, which we will call *Sequence A*, are shown in Fig. 4. Starting from a *no contact* state, the intended sequence is *right front edge*, *right* side, and *right bottom*. The sequence was demonstrated three times, and the final demonstration was used to learn the SECF sequence and transition trajectories.

The force and moment profiles for the skill demonstration are shown in Fig. 5, with data logged at a frequency of 40 Hz. The sequence of SECF states was identified as described in [27], with results shown in Fig. 5. For graphing purposes, an index is assigned to each SECF. The *no contact* class is shown with an index of 0. The unclassified cases are shown with an index of -1.

A filtering algorithm is applied to the classification results, yielding new results as shown in Fig. 5. Transitory classes are filtered out, which has the effect of eliminating undesired human-generated actions. Human factors studies have shown that the bandwidth for human limb motion is at most 5 Hz for internally generated trajectories and around 10 Hz for reflex actions [32]. In this illustration, we have filtered out transitory states occurring faster than 10 Hz. The filtering algorithm also eliminates the unclassified cases. For the points of no classification, we assume that the SECF has not changed, and these points are graphed as the previously identified SECF.

The resulting sequence is an accurate representation of the demonstration; however, this is not the sequence that was intended. The *bottom* surface was encountered accidentally (shown as index#1 in Fig. 5), and there were several instances of oscillation between SECFs, such as the changes between

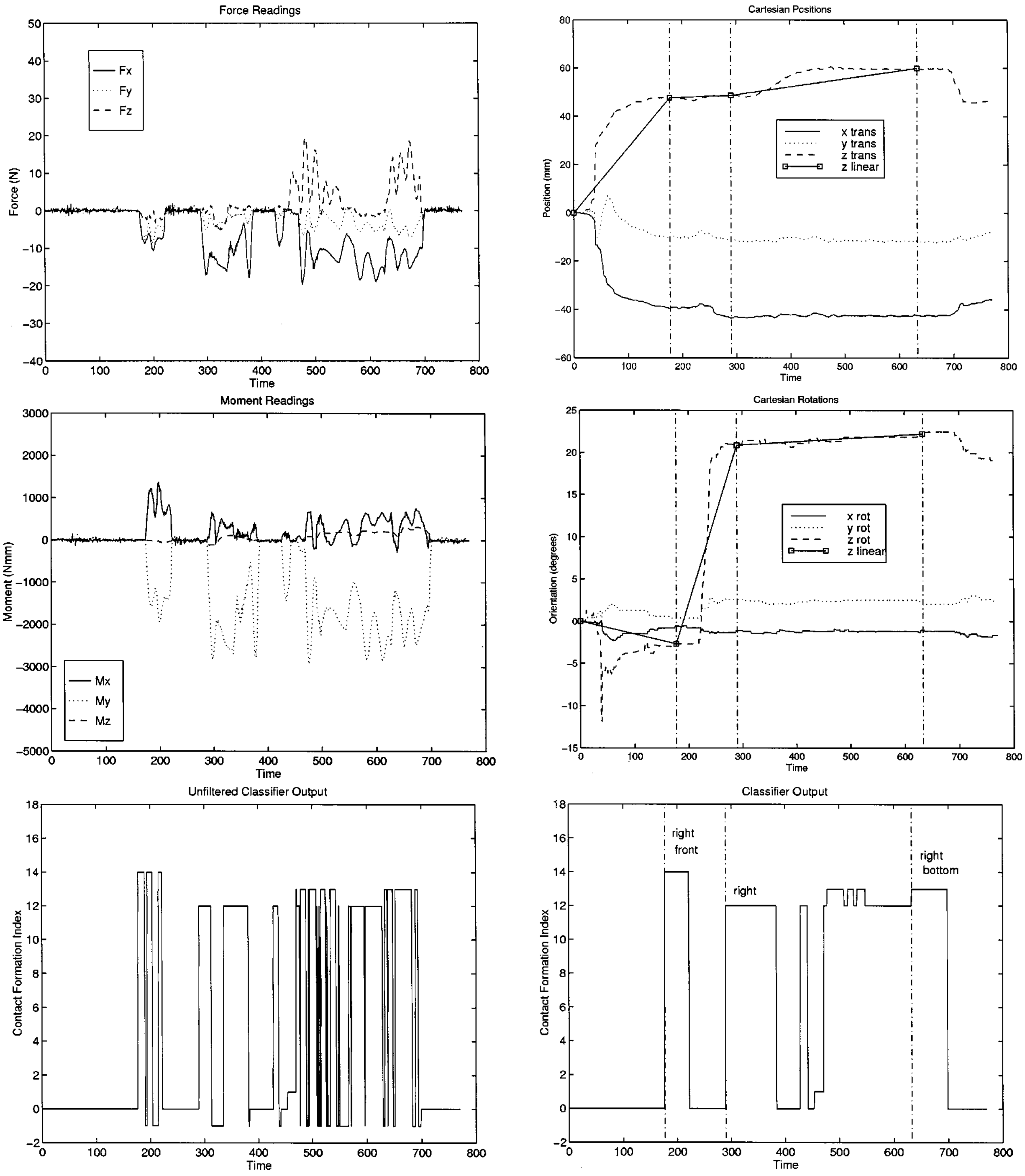

Fig. 5. Sequence A demonstration.

right bottom and the *right* side. Also, there were periods of *no contact* that we might prefer to avoid in order to achieve a smoother skill execution. As a result, this is not yet the sequence that we want the robot to duplicate.

This illustration provides an example of an imperfect skill demonstration. If we could assume that every demonstrated skill sequence was perfect, then the method presented would be adequate. However, perfect skill demonstrations can be difficult to achieve. Often, it is easier for the operator to look at the results of an imperfect demonstration and modify those results directly.

For this purpose, an interactive, programming environment is suggested, in which the operator can observe the learned SECF sequence and refine it if necessary. An operator can look at the sequence and determine which SECF steps are essential and which steps are unnecessary. This has been done, and Fig. 5 identifies the selected sequence by showing the selected force events with vertical hash lines.

B. Learning Transitions between SECFs

The final function to be learned, as part of the skill model, is the transition velocity information which will move the robot from the current SECF to the next desired SECF. The profiles in Fig. 5 show that the motion generated by the human demonstrator is not necessarily smooth. We do not want to duplicate inconsistent or unintended actions, but to extract an action which will drive the system to the next desired SECF state. For this purpose, we approximate the trajectory as a series of linear segments. The selected force events, which define the change in SECF, can be used as reference points in segmenting the demonstrated skill. The result is a piecewise linear trajectory which drives the system toward the next desired state. Because force events are used as reference points instead of position information, the trajectory generation is not dependent on absolute position.

The translational component of the reference velocity command is generated from the demonstrated position profile. Let t_p and t_{p+1} be the times at which force events p and $p+1$ occur in the demonstrated position profile. Then the distances, d_x, d_y, and d_z, traveled in the x, y, and z directions, are calculated as follows:

$$\begin{aligned} d_x &= x(t_{p+1}) - x(t_p) \\ d_y &= y(t_{p+1}) - y(t_p) \\ d_z &= z(t_{p+1}) - z(t_p). \end{aligned}$$

These distances are then normalized as shown: $d_{xn} = d_x/d$, $d_{yn} = d_y/d$, and $d_{zn} = d_z/d$ where $d = \sqrt{d_x^2 + d_y^2 + d_z^2}$.

The normalized distances define a relative direction. Assuming a constant speed, the normalized distances are used to generate the translational component of the reference velocity from SECF state $w(p)$ to state $w(p+1)$. The process is illustrated in Fig. 5, where the linear segments are shown for the z translation component.

The rotational component of the reference velocity command is generated in a similar manner. Let $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ be the rotational profiles for the x, y, and z orientations, respectively. Then the rotational distances, r_x, r_y, and r_z are calculated as follows:

$$\begin{aligned} r_x &= \alpha(t_{p+1}) - \alpha(t_p) \\ r_y &= \beta(t_{p+1}) - \beta(t_p) \\ r_z &= \gamma(t_{p+1}) - \gamma(t_p). \end{aligned}$$

Again assuming a constant rotational speed, the rotational distances can be used to generate the rotational component of the reference velocity. Fig. 5 provides an illustration for the z rotational component. While this strategy provides a relatively simple method for generating the reference velocity commands, no concept of speed is learned.

C. Merging Multiple Assembly Strategies

To provide robust execution of the assembly skill in a nonstructured environment, we must accommodate position and orientation variations. That is, we need a strategy for reaching the goal state, even if the execution environment is positioned differently from that used for demonstration. For example, in the sequence shown in Fig. 4, if the environment fixture was moved to the right, and the same initial trajectory was used, the grasped workpiece might touch the *bottom* surface first instead of the intended *right front edge*. To accommodate this possibility, we include additional SECFs in the FSM, along with their corresponding transition commands. In this way, the robot learns a more complete skill model and is able to reach the desired goal even without a structured environment. The SECF sequence paths are merged at the goal state, which is common for all sequences. Although one sequence is arbitrarily assigned as the primary path, if a node from another path is encountered during execution, then control shifts to this secondary path.

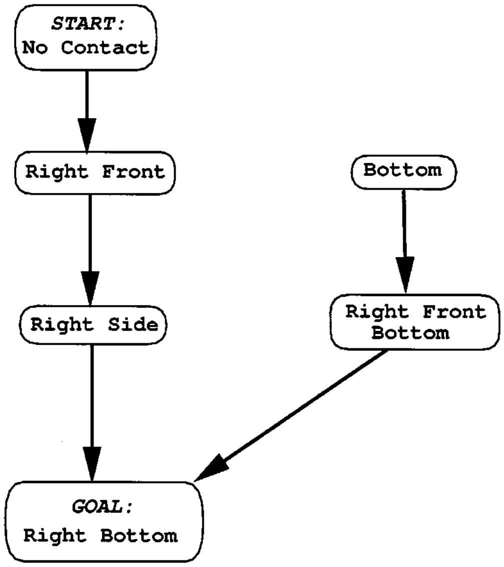

Fig. 6. Merged FSM with sequence A and sequence B.

For our experiments, a second sequence, *Sequence B*, is demonstrated as follows: *no contact*, *bottom*, *right front bottom*, and *right bottom*. *Sequence B* provides a successful skill strategy for the cases where the workpiece contacts the bottom surface first, instead of the right front edge. The sequence was again demonstrated three times and the third demonstration was used for skill acquisition. Following the procedure outlined in the previous sections, the SECF sequence and transition commands were extracted.

Sequence A and *Sequence B* are then merged into one FSM, as shown in Fig. 6. While *Sequence A* is arbitrarily considered to be the primary sequence, control can shift to *Sequence B* if the *bottom* SECF is encountered. Likewise, if any other SECF is encountered out of order, the FSM is still able to drive the control through a successful skill sequence.

To accommodate a greater variety of starting positions and orientations, additional sequence strategies can be demonstrated and merged into the FSM. We are assuming that another mechanism (e.g., vision) is used to position the grasped workpiece close to the area of interest so that contact will be made. The actual mechanism used will determine the effective position and orientation uncertainties, which will, in turn, determine what additional SECFs should be added.

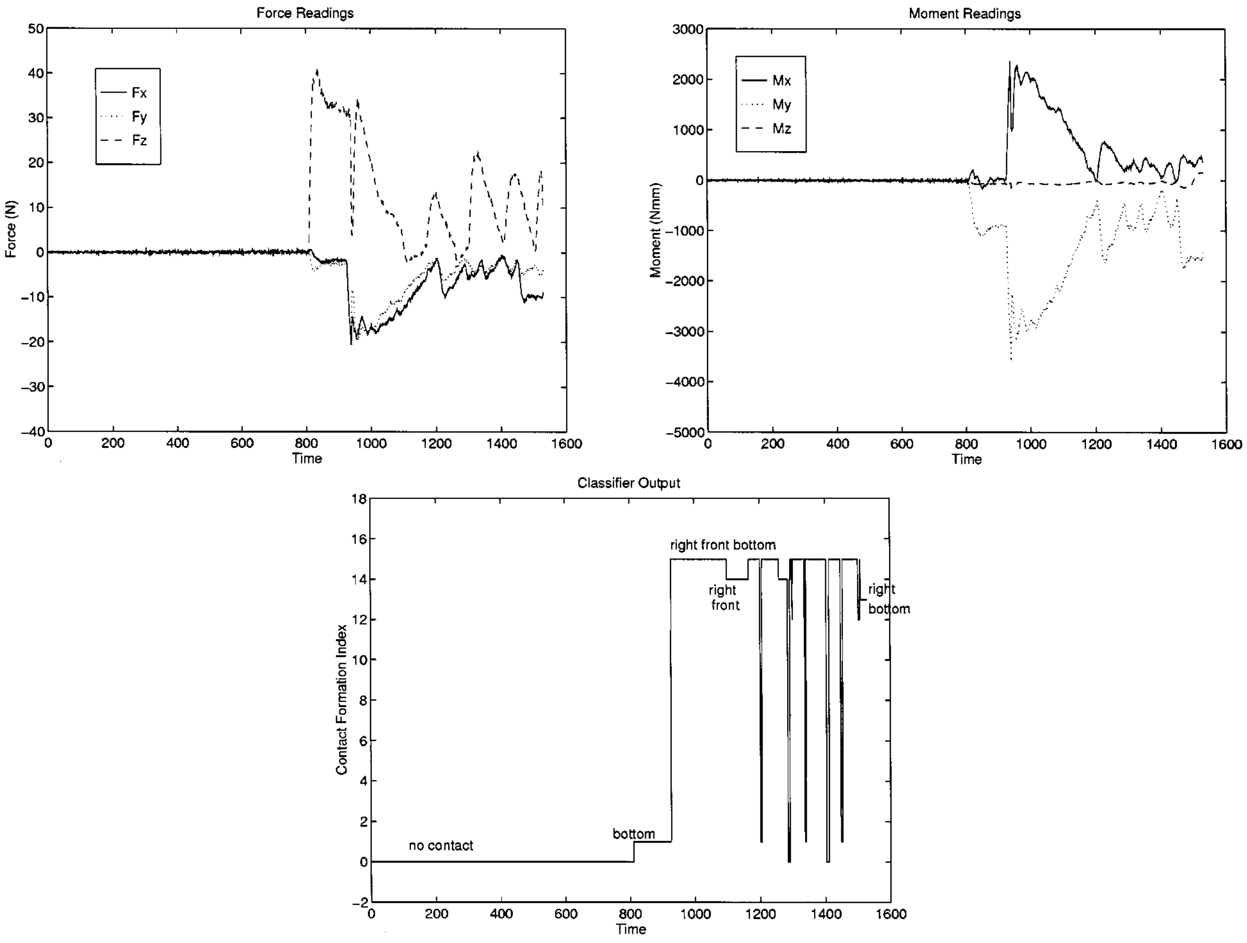

Fig. 7. Test run force profile.

V. Execution of the Learned Skill

In this section, we present results of test runs performed where the robot autonomously executed the learned skill. The same workpiece and environment fixture were used for both skill learning and for autonomous skill execution. The starting position of the robot, with the grasped workpiece, was the same in all tests; the workpiece was not contacting the environment. To test the execution of the skill in different starting conditions, the position and orientation of the environment fixture were varied between test runs.

A data logging function was added to the robot control program to monitor the performance. The logged data included the force sensor signals, the robot end effector position and orientation, and the classified SECF. All data was logged at a frequency of 50 Hz.

Variations in the starting position and orientation of the environment fixture were introduced in an attempt to produce additional sequences of SECF states. In the first four test runs, the task stopped short of the goal by about four degrees in z-axis rotation. This was attributed to an incorrect classification by the state classifier, because of high friction in the execution environment. To induce higher friction into the training data, the classifier was retrained using data collected by sliding the workpiece along a surface. Five additional test runs were performed. In all five cases, the skill was successfully completed, in spite of the position and orientation variations. Figs. 7 and 8 show results from a typical test run. Unexpected SECFs were encountered during the skill execution, but the skill model was able to accommodate the variation, generating the appropriate control command and, therefore, driving the task to successful completion.

VI. Discussion

Although the experiments were generally successful, limitations became apparent during testing. One potential problem results from the use of supervised learning for training the SECF classifier. The approach assumes that the operator knows which SECF classes to include in the set. While that may be trivial for simple skills, it may not be so obvious for complex skills.

In addition, our experiences show that the training of the state classifier is critical. Training data for the classifier should be collected in a condition as close as possible to that expected in the execution environment. This will ensure that the classifier will be trained to handle the same real-world, difficult-to-model effects that will be encountered during skill execution. One factor that complicated the issue was the presence of friction. If significant friction is present in the skill execution environment, then it must be included when collecting the training data for the classifier.

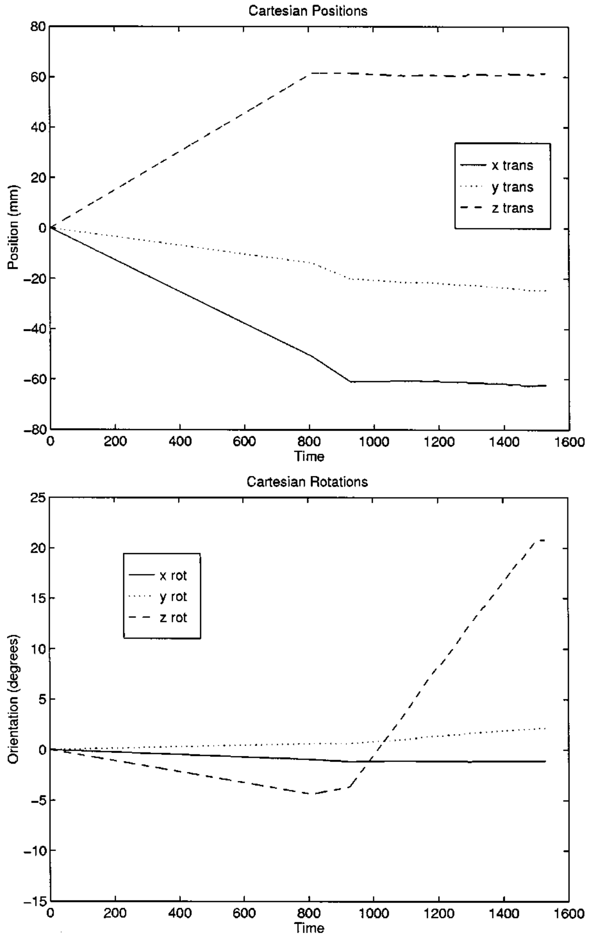

Fig. 8. Test run position profile.

While it is possible to collect separate training data (for a known SECF), it may be easier to collect training data while performing an entire skill sequence. We are currently experimenting with clustering algorithms which will be used to segment an entire profile of data into individual force-based clusters. The clusters of vectors will then be used to train the classifier. This approach will ensure that actual conditions, such as friction, are included in the training data. It will also eliminate the problem of determining which SECFs to include. The tradeoff is that the SECF classes will not be classified on-line during the actual demonstration, so this feedback will not be available to the operator.

Another limitation was the method of force control used. In this implementation, we were limited (by the robot controller) to a 100-Hz servo rate. This was adequate for the simple task demonstrated here, but complex tasks would most likely require a faster servo rate. In addition, there are limitations to using a force limit control algorithm with constant parameters throughout the skill sequence. For instance, this simple algorithm does not try to maintain a constant force if the workpiece breaks contact with the environment. A desirable extension would include changes in the force control parameters for each transition between SECF steps. This may be especially useful for more complicated skills, where it is necessary to maintain contact during the transition period.

In spite of the limitations and problems, the skill acquisition method worked remarkably well. We have succeeded in completing tasks for which others have reported failure [17]. Although the experiments involved a relatively simple skill, we feel the results validate the assembly skill model. The event-based approach proved to be a good idea and useful in accommodating variations in the environment. The interactive programming environment also proved to be a good idea, especially in letting the operator select the events that would be used. Our experience is that this made skill acquisition a great deal easier because the skill demonstration did not have to be done perfectly.

VII. Concluding Remarks

This research has been inspired by the vision of intelligent robot tools and assistants working successfully in unstructured environments. We have addressed this problem by investigating methods of transferring assembly skills to robots by observing a human-performed demonstration of the skill. Specifically, the focus here has been on learning force-based skills, rather than position-based skills, without requiring detailed geometric information. The ability to quickly transfer force-based skills to robots will dramatically increase the functional capabilities of our envisioned intelligent robot tools and assistants.

As a skill model, we have presented a theoretical framework for combining force control with reference velocity commands. This combined control, along with the use of force-based qualitative states, makes the approach suitable for unstructured environments. In this paper, we have concentrated on the contact portion of the task. We would envision that, eventually, the technique would be combined with either vision or shared control with a human operator.

The methodology presented supports 6-DOF motion and includes difficult-to-model effects such as friction. The experiments discussed show how a skill can be acquired in the presence of such effects and even if the demonstrated task is not perfect. The results also show how the learned skill can complete a task successfully in spite of significant position and orientation variations.

References

[1] K. Kosuge, T. Fukuda, and H. Asada, “Acquisition of human skills for robotic systems,” in *Proc. 1991 IEEE Int. Symp. Intelligent Control*, Arlington, VA, Aug. 1991, pp. 469–474.

[2] G. E. Hovland, P. Sikka, and B. J. McCarragher, “Skill acquisition from human demonstration using a hidden Markov model,” in *Proc. 1996 IEEE Int. Conf. Robotics and Automation*, vol. 3, Minneapolis, MN, Apr. 1996, pp. 2706–2711.

[3] Y. Kuniyoshi, M. Inaba, and H. Inoue, “Learning by watching: Extracting reusable task knowledge from visual observation of human performance,” *IEEE Trans. Robot. Automat.*, vol. 10, pp. 799–822, Dec. 1994.

[4] K. Ikeuchi and T. Suehiro, “Toward an assembly plan from observation,” *IEEE Trans. Robot. Automat.*, vol. 10, pp. 368–385, June 1994.

[5] G. V. Paul and K. Ikeuchi, “Modeling planar assembly paths from observation,” in *Proc. 1996 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, Osaka, Japan, Nov. 1996, pp. 520–525.

[6] ——, “A quasilinear method for computing and projecting onto c-surfaces: General case,” in *Proc. 1997 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, Grenoble, France, Sept. 1997, pp. 1183–1188.

[7] J. Miura and K. Ikeuchi, "Task-oriented generation of visual sensing strategies in assembly tasks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 126–138, Feb. 1998.

[8] H. Onda, H. Hirukawa, F. Tomita, T. Suehiro, and K. Takase, "Assembly motion teaching system using position/force simulator-generating control program," in *Proc. 1997 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, Grenoble, France, 1997, pp. 938–945.

[9] J. Lloyd, J. Beis, D. Pai, and D. Lowe, "Programming contact tasks using a reality-based virtual environment integrated with vision," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 423–434, June 1999.

[10] H. Asada and H. Izumi, "Automatic program generation from teaching data for the hybrid control of robots," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 166–173, Apr. 1989.

[11] S. Hirai and H. Asada, "Kinematics and statics of manipulation using the theory of polyhedral convex cones," *Int. J. Robot. Res.*, vol. 12, no. 5, pp. 434–447, Oct. 1993.

[12] G. E. Hovland and B. J. McCarragher, "Hidden Markov models as a process monitor in robotic assembly," *Int. J. Robot. Res.*, vol. 17, no. 2, pp. 153–168, Feb. 1998.

[13] N. Delson and H. West, "Robot programming by human demonstration: Subtask compliance controller identification," in *Proc. 1993 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Yokohama, Japan, July 1993, pp. 33–41.

[14] Q. Wang and J. De Schutter, "Toward real-time robot programming by human demonstration for 6d force controlled actions," in *Proc. 1998 IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, May 1998, pp. 2256–2261.

[15] N. Delson and H. West, "Robot programming by human demonstration: Adaptation and inconsistency in constrained motion," in *Proc. 1996 IEEE Int. Conf. Robotics and Automation*, vol. 1, Minneapolis, MN, Apr. 1996, pp. 30–36.

[16] H. Asada, "Representation and learning of nonlinear compliance using neural nets," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 863–867, Dec. 1993.

[17] P. V. Whalen, "Teaching accommodation task skills: From human demonstration to robot control via artificial neural networks," Ph.D. dissertation, Air Force Inst. Technol., Wright–Patterson Air Force Base, 1995.

[18] R. Koeppe, A. Breidenbach, and G. Hirzinger, "Skill representation and acquisition of compliant motions using a teach device," in *Proc. 1996 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, Osaka, Japan, Nov. 1996, pp. 897–904.

[19] M. Kaiser and R. Dillmann, "Building elementary robot skills from human demonstration," in *Proc. 1996 IEEE Int. Conf. Robotics and Automation*, vol. 3, Minneapolis, MN, Apr. 1996, pp. 2700–2705.

[20] H. Friedrich, J. Holle, and R. Dillmann, "Interactive generation of flexible robot programs," in *Proc. 1998 IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, May 1998, pp. 538–543.

[21] D. E. Whitney, "Quasistatic assembly of compliantly supported rigid parts," *ASME J. Dyn. Syst., Meas., Contr.*, vol. 104, pp. 65–77, May 1982.

[22] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, pp. 418–432, June 1981.

[23] R. W. Brockett, "Hybrid models for motion control systems," in *Essays on Control: Perspectives in the Theory and Its Applications*, H. L. Trentelman and J. C. Willems, Eds. Boston, MA: Birkhauser, 1993, ch. 2, pp. 29–53.

[24] R. S. Desai and R. A. Volz, "Identification and verification of termination conditions in fine motion in presence of sensor errors and geometric uncertainties," in *Proc. 1989 IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 800–807.

[25] M. Skubic and R. Volz, "Identifying single-ended contact formations from force sensor patterns," *IEEE Trans. Robot. Automat.*, vol. 16, pp. 597–603, Oct.. 2000.

[26] P. Astuti and B. J. McCarragher, "Sufficient conditions for the success of robotic assembly," in *Proc. 1994 IEEE Int. Conf. Robotics and Automation*, vol. 2, San Diego, CA, May 1994, pp. 1693–1699.

[27] M. Skubic and R. A. Volz, "Learning force sensory patterns and skills from human demonstration," in *Proc. 1997 IEEE Int. Conf. Robotics and Automation*, vol. 1, Albuquerque, NM, Apr. 1997, pp. 284–290.

[28] M. Skubic, S. P. Castrianni, and R. A. Volz, "Identifying contact formations from force signals: A comparison of fuzzy and neural network classifiers," in *Proc. 1997 Int. Conf. Neural Networks*, vol. 3, Houston, TX, June 1997, pp. 1623–1628.

[29] M. Skubic and R. A. Volz, "Learning force-based assembly skills from human demonstration for execution in unstructured environments," in *Proc. 1998 IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, May 1998.

[30] L. Everett, R. Ravuri, R. Volz, and M. Skubic, "Generalized recognition of single-ended contact formations," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 829–836, Oct. 1999.

[31] J. De Schutter *et al.*, "Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motion," *Int. J. Robot. Res.*, vol. 18, pp. 1161–1184, Dec. 1999.

[32] N. I. Durlach and A. S. Mavor, Eds., *Virtual Reality: Scientific and Technological Challenges*. Washington, DC: National Academy Press, 1995.

Marjorie Skubic (S'90–M'97) received the Ph.D. degree in computer science from Texas A&M University, College Station, in 1997.

Prior to graduate school, she worked in industry for 14 years as a Software Engineer, specializing in real-time systems. Industrial experience during that time included Texas Instruments, TRW, and Staefa Control System. Currently, she is an Assistant Professor with the Computer Engineering and Computer Science Department, University of Missouri, Columbia. Her research interests include sensory perception and human–robot interaction, especially robot programming by demonstration. She is also interested in robotics education and educational applications of robotics.

Dr. Skubic is a member of the ACM and ASEE. She is the Co-Chair of the Education Committee for the IEEE Robotics and Automation Society.

Richard A. Volz (M'60–SM'86–F'97) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Northwestern University, Evanston, IL, in 1960, 1961, and 1964, respectively.

He is the Royce E. Wisenbaker Professor of Engineering in the Computer Science Department, Texas A&M University, College Station. He served as Department Head from 1988 to 1997, and has recently embarked upon a new initiative on "Training Systems Science and Technology." Prior to joining Texas A&M, he was Director of the Robotics Research Laboratory and Professor of Electrical Engineering and Computer Science at the University of Michigan. He has served on five federal advisory boards (Air Force Scientific Advisory Board, Ada Board, Aerospace Safety Advisory Panel—a Congressional oversight committee on NASA, NASA Space Station Advisory Panel, and NASA Center of Excellence in Information Technology Advisory Panel). He is the author or co-author of over 100 research papers and has led over $15 000 000 in funded research projects.

Dr. Volz received the Decoration for Exceptional Civilian Service form the U.S. Air Force, two NASA Special Service Awards, and the IEEE Mellenium Medal. He served as the Editor-in-Chief of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 1994 to 1999, as well as in numerous other editorial and conference organization positions.