Learning Force-Based Assembly Skills from Human Demonstration for Execution in Unstructured Environments *

Marjorie Skubic
Computer Engineering & Computer Science
University of Missouri-Columbia
Columbia, MO 65211

Richard A. Volz
Dept. of Computer Science
Texas A&M University
College Station, TX 77843-3112

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 the structured, controlled settings. The work focuses on the problem of learning low-level force-based assembly skills from human demonstration. To avoid position dependencies, force-based discrete states 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. 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.

1 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. This research is inspired by the vision of robots moving beyond the 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, extend his capability, or allow him to avoid a hazardous environment. Depending on the nature of the work, the robot may be asked to perform repetitive or non-repetitive 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). Several researchers have incorporated force signals into PbD systems. Asada first proposed the use of neural networks [1], and others have used this approach as well (e.g., [2, 3, 4]). However, there have been problems with training neural networks directly from demonstrations because of the unintentional, inconsistent motion commands generated by human demonstrators.

To deal with both inconsistent training data and uncertainties in position and orientation, we have used an event-based approach, utilizing discrete event control, as suggested in [5, 6]. Unlike the approach used in [5], we do not use any geometric information. The research discussed here addresses the problem of learning low-level force-based assembly skills by studying sensory patterns of demonstrated tasks, concentrating on the part of the program 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., [6]). First, our hybrid control model combines reference velocity commands with force control. Second, our interactive PbD interface allows skill acquisition to take place even if the demonstrated task is flawed.

In the remaining sections, we describe the skill model, skill acquisition, and experiments. In Section

*This work has been supported by NSF under grants CDA-9115123, CDA-9422123, and EID-9017249 and by the State of Texas under TATP grants 999903-267 and 999903-095.

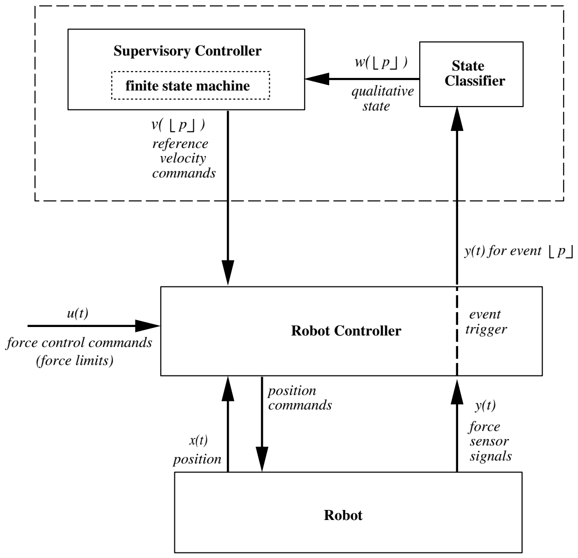


Figure 1: The Hybrid Control Model of an Assembly Skill

2, we describe the theoretical framework of the skill model. In Section 3, skill learning is illustrated with an example of a demonstrated task. Section 4 presents the results of autonomous execution of the learned skill, and is followed by discussion in Section 5. In Section 6, we conclude.

2 Skill Model

The mathematical framework for the robot skill model is the hybrid control model proposed by Brockett [7], in which symbol processing interacts with signal processing. The model, with special application towards modeling force-based assembly skills, is shown in Figure 1; it consists of four parts: (1) the robot, (2) the robot controller, (3) the state classifier, and (4) the supervisory controller. The interaction between the robot controller and the robot is based on functions of time. Symbolic processing is done at the supervisory controller level; both its 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 is 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 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 which drives the assembly skill will take the form of a single-ended contact formation (SECF), which describes how a grasped object is touching its environment. Two sensor-based methods which can be used for SECF classification are described in previous work [8, 9]. The mapping is performed by the state classifier every time the event trigger detects a change in SECF state.

The robot controller regulates the robot using two control loops. At the lowest level, there is a position control loop. An outer force control loop is used to implement hybrid position/force control, using force signals for feedback. Ideally, the force control commands would change whenever the qualitative state changes. However, as a first step, the force control commands are modeled as pre-established force and moment limits for each axis of the force sensor. If a force along any axis exceeds the threshold value, then the robot controller attempts to control the force to a set value.

The force and moment limits remain constant for an entire assembly task.

The supervisory controller uses a finite state machine (FSM), represented as a directed graph, to drive the assembly skill. The states stored in the FSM are the SECF's which describe the qualitative condition in the assembly process. The FSM stores the sequence of SECF's which are used by the human demonstrator in the assembly skill. The skill model also includes the controller output function, which generates reference velocity commands that describe how to transition to the next desired SECF.

Because we are not making any assumptions about the position and orientation of the robot's environment,[1] we cannot guarantee that the desired goal condition will be reached. This is similar to other sensor-based strategies used for robotic control in unstructured environments (e.g., [10]). However, we can increase the probability for success, in spite of position uncertainties, by incorporating more than one SECF sequence into the FSM. A skill may, in fact, contain several SECF sequences that will result in a successful task completion. The combination of sequences can be used to achieve a sequence that was not included in the demonstration process but, in fact, still provides successful completion.

3 Skill Learning

As modeled here, the learning of an assembly skill involves the learning of three functions: (1) the mapping of force sensor signals to SECF's (2) the sequence (or sequences) of SECF's 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 human teacher demonstrates each SECF while force data is collected, and the data is used to train a state classifier. (Two methods of classification are discussed in previous work [8, 9], using fuzzy set theory and a neural network.) The human teacher then demonstrates a skill, and the classifier is used to extract the sequence of SECF's and transition velocities which comprise the rest of the skill.

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 human teacher by classifying and displaying the SECF sequence in real-time. The symbolic representation of the SECF is useful for providing meaningful and easily recognizable feedback to the human teacher. This additional feedback can be used to increase the efficiency of the teaching process, because the teacher 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.

3.1 Learning a Sequence of Single-Ended Contact Formations

The learning of a sequence of SECF's is illustrated with an example of an actual skill demonstration. The workpiece used is a plastic, square block. For this illustration, the fuzzy classifier [8] has been trained to identify 17 SECF classes (plus the *no contact* class). After training, the classifier performs at a 97% success rate with static data.

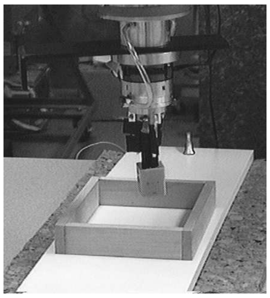
The demonstration of the skill was performed using the PHANToM haptic interface as a force-reflecting hand controller to teleoperate an American Robot *merlin* manipulator in 6 dof motion. A JR3 force sensor provided force signals. The human teacher was located in front of the robot, with full view of the operation. In the mind of the teacher, there was a specific intended SECF sequence that was part of the teacher's skill model, and this is the sequence that we would like the robot to learn.

The environment is an open, rectangular, wooden box, made of soft, bare wood. Although it has been sanded, it still 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 so that different starting conditions can be generated.

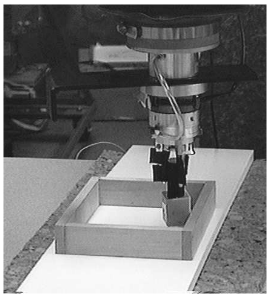
Snapshots of the sequence, which we will call *Sequence A*, are shown in Figure 2. Starting from a *no contact* state, the intended sequence is *right front*, *right* side, and *right bottom*. The sequence was demonstrated 3 times, and the final demonstration was used to learn the SECF sequence and transition trajectories.

The force and moment profiles for the skill demonstration are shown in Figure 3, with data logged at a frequency of 40 Hz. The sequence of SECF states was classified and filtered as described in [11]. The resulting sequence, shown in Figure 4, is an accurate identification of the demonstrated sequence; however, this is not the sequence that was intended. The *bottom* SECF was encountered accidently, and there were several instances of oscillating between SECF's, such as the changes between *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.

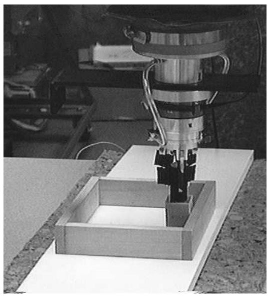
[1] Indeed, the assumption here is that we do *not* know the position and orientation of objects in the environment.



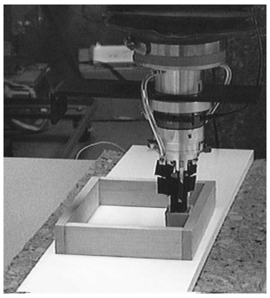
1: Start Position



2: Right Front



3: Right Side



4: Goal Position is Right Bottom

Figure 2: The SECF Steps for Sequence A

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, it is not always easy for the operator to provide perfect demonstrations. Often, it may be easier for the operator to look at the results of an imperfect demonstration and modify those results directly.

For this purpose, an interactive, iterative teaching environment is suggested, in which the operator can observe the learned SECF sequence and refine it if necessary. For example, an operator can easily look at the sequence and determine which SECF steps are essential and which steps are unnecessary. This has been done, and Figure 4 identifies the selected sequence by showing the selected force events with vertical hash lines.

3.2 Learning Transitions between SECF's

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. From the graphed profiles in Figure 5, we can see 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

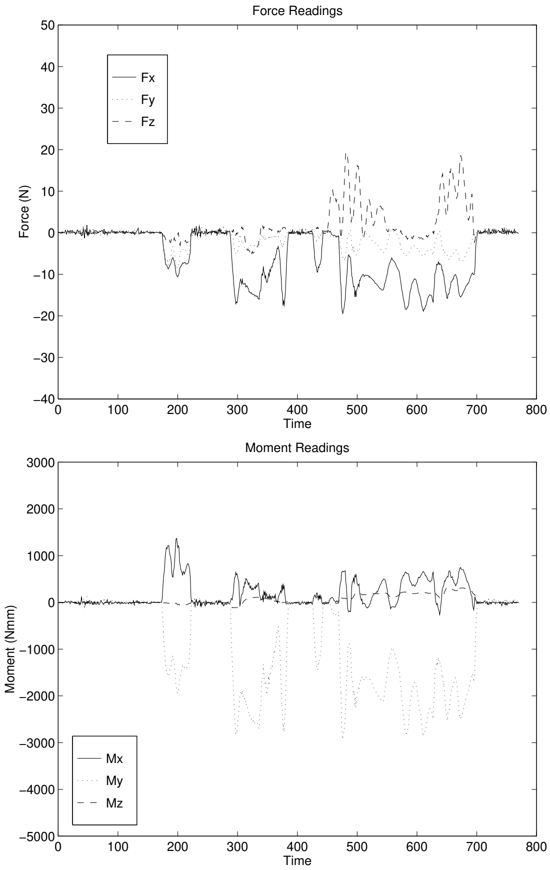


Figure 3: Sequence A Demonstration: Force Profile

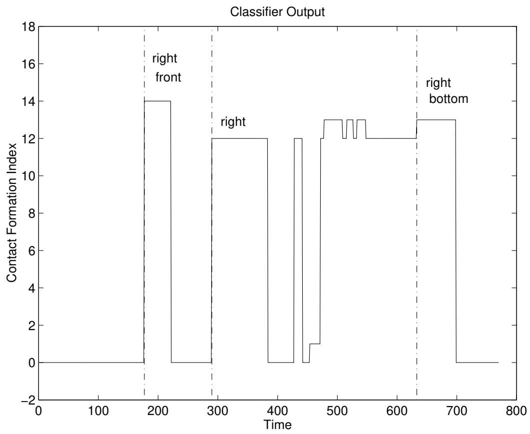


Figure 4: Sequence A Demonstration: SECF Classification

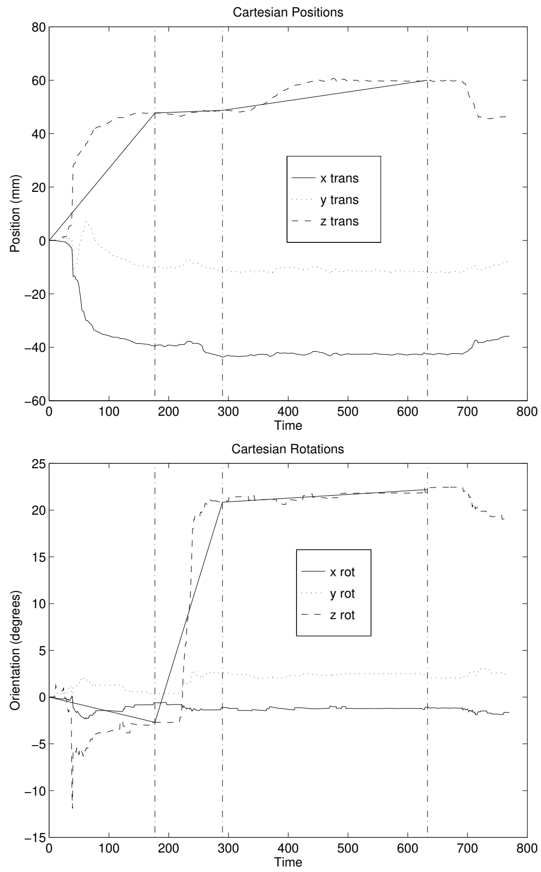


Figure 5: Sequence A Demonstration: Position Profile

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 towards the next desired state. Because force events are used as reference points instead of position information, the trajectory generation is not dependent on the precise 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 profile. Then the distances, d_x, d_y, and d_z, traveled in the x, y, and z directions, are calculated as follows:

$$d_x = x(t_{p+1}) - x(t_p)$$

$$d_y = y(t_{p+1}) - y(t_p)$$

$$d_z = z(t_{p+1}) - z(t_p)$$

These distances are then normalized as shown: $d_{xn} = \frac{d_x}{d}$, $d_{yn} = \frac{d_y}{d}$, and $d_{zn} = \frac{d_z}{d}$ 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 Figure 5, where the linear segments are shown for the z translation component.

The generation of the rotational component of the reference velocity command is done 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:

$$r_x = \alpha(t_{p+1}) - \alpha(t_p)$$
$$r_y = \beta(t_{p+1}) - \beta(t_p)$$
$$r_z = \gamma(t_{p+1}) - \gamma(t_p)$$

Again assuming a constant rotational speed, the rotational distances can be used to generate the rotational component of the reference velocity. Figure 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. As presented, the strategy assumes that a constant speed is used.

3.3 Merging Multiple Assembly Strategies

To provide robust execution of the assembly skill in a non-structured environment, we need a way to accomodate position and orientation variations. That is, we need a strategy for reaching the goal state, even if the execution environment is positioned differently than that used for the skill demonstration. For example, in the sequence shown in Figure 2, if the environment fixture was translated to the right, and the same skill sequence was used, the grasped workpiece can reach the *bottom* SECF first instead of the intended *right front*. To accomodate this possibility, we include additional SECF's 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.

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

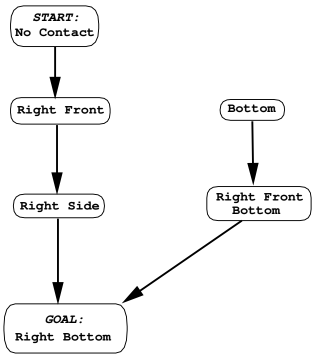


Figure 6: The Merged FSM with Sequence A and Sequence B

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 Figure 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 environment 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 SECF's should be added to the FSM.

4 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 4 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 re-trained using data collected by sliding the workpiece along a surface. Five additional test runs were performed. In all 5 cases, the skill was successfully completed, in spite of the position and orientation variations. Figures 7 and 8 show results from a typical test run. Unexpected SECF's were encountered during the skill execution, but the skill model was able to accommodate the variation and still produce a successful skill completion.

5 Discussion

Although the experiments were generally successful, there are limitations that became apparent in the course of the testing. One potential problem results from the use of supervised learning. The approach taken here, where the SECF classifier is trained before the skill demonstration, 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. The significant point is that the 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.

While it is possible to collect separate training data (for a known SECF), it may be easier to collect training data while actually 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 SECF's to include in the class set. 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. Certainly, a desirable extension to this work 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. 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 teaching environment also proved to be a good idea, especially in letting the teacher select the events that would be used in skill acquisition. Our experience is that this made the teaching process a great deal easier because the skill demonstration did not have to be done perfectly.

6 Conclusion

This research has been inspired by the vision of intelligent robot tools and assistants working successfully in unstructured environments and not necessarily just in repetitive tasks. 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 low-level, force-based assembly skills, rather than position-based skills, without using 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 which provides a mechanism 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 supervisory control

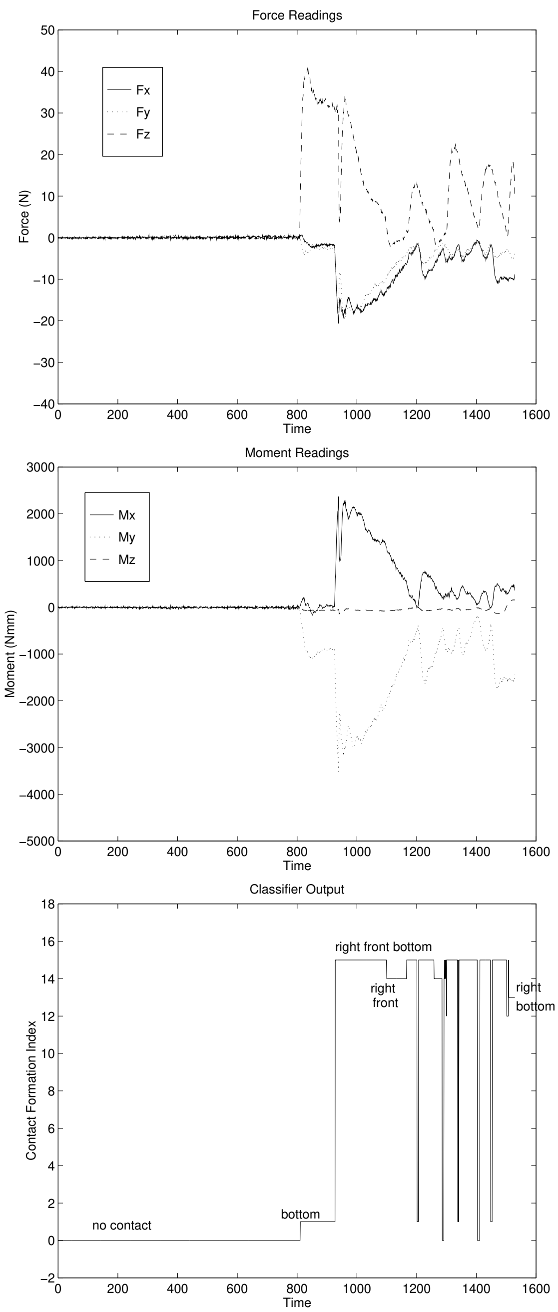


Figure 7: Test Run Force Profile

by a human operator.

In the experiments presented, we used 6 dof motion and included real-world, 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.

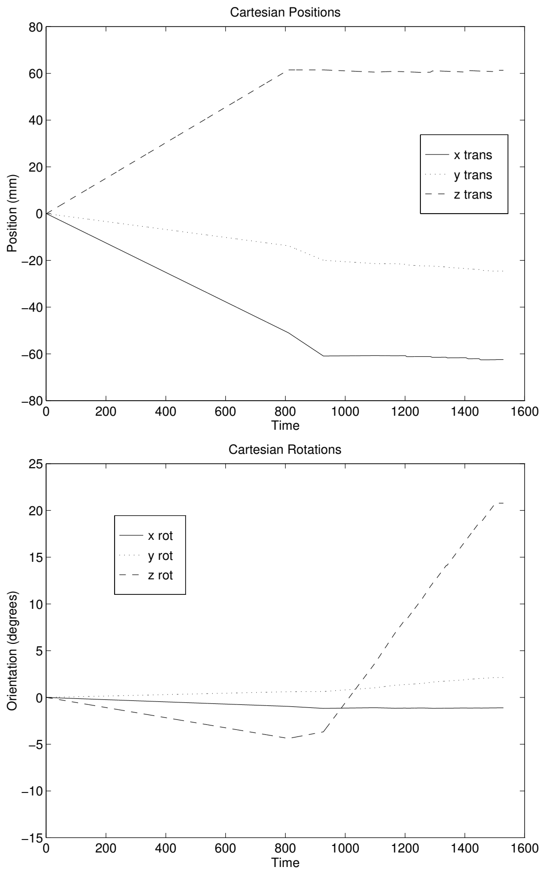


Figure 8: Test Run Position Profile

References

[1] H. Asada, "Teaching and learning of compliance using neural nets: Representation and generalization of nonlinear compliance," in *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cinncinnati, OH, May 1990, pp. 1237–1244.

[2] M. Kaiser and R. Dillmann, "Building elementary robot skills from human demonstration," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Apr. 1996, vol. 3, pp. 2700–2705.

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

[4] P.V. Whalen, *Teaching accomodation task skills: from human demonstration to robot control via artificial neural networks*, Ph.D. thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, 1995.

[5] K. Kosuge, T. Fukuda, and H. Asada, "Acquisition of human skills for robotic systems," in *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, Arlington, VA, Aug. 1991, pp. 469–474.

[6] G.E. Hovland, P. Sikka, and B.J. McCarragher, "Skill acquisition from human demonstration using a hidden markov model," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Apr. 1996, vol. 3, pp. 2706–2711.

[7] 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., chapter 2, pp. 29–53. Birkhauser, Boston, MA, 1993.

[8] M. Skubic and R.A. Volz, "Identifying contact formations from sensory patterns and its applicability to robot programming by demonstration," in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, Nov. 1996, vol. 2, pp. 458–464.

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

[10] R.A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14 – 23, Mar. 1986.

[11] M. Skubic and R.A. Volz, "Learning force sensory patterns and skills from human demonstration," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, Apr. 1997, vol. 1, pp. 284–290.