

Learning Tasks From A Single Demonstration

Christopher G. Atkeson and Stefan Schaal

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, USA
ATR Human Information Processing, 2-2 Hikaridai, Seiko-cho, Soraku-gun, 619-02 Kyoto, Japan
{cga,sschaal}@cc.gatech.edu, <http://www.cc.gatech.edu/fac/{Chris.Atkeson,Stefan.Schaal}>

Abstract

Learning a complex dynamic robot maneuver from a single human demonstration is difficult. This paper explores an approach to learning from demonstration based on learning an optimization criterion from the demonstration and a task model from repeated attempts to perform the task, and using the learned criterion and model to compute an appropriate robot movement. A preliminary version of the approach has been implemented on an anthropomorphic robot arm using a pendulum swing up task as an example.

1 Introduction

One approach to programming robots is to have them learn a task by watching the task being performed by a human or by another robot. The robot can either mimic the motion of the demonstrator, or learn how the demonstrator acts in many situations (a policy). We are interested in exploring techniques for learning from demonstration in cases where the robot may not be doing exactly the same task as the demonstrator and where there are a small number of task demonstrations available. In these cases exactly imitating the motion of the demonstrator may not achieve the task, and there may be too little training data to learn an adequate policy. This paper explores an approach based on learning a task model and an optimization criterion for the task, and using the model and criterion to compute an appropriate policy. A preliminary version of the approach has been implemented on an anthropomorphic robot arm using a pendulum swing up task as an example. This paper describes that example implementation, the lessons learned, and the applicability of various approaches to learning from demonstration.

Learning from demonstration, also known as “programming by demonstration”, “imitation learning”, and “teaching by showing” is a topic that has received significant attention in automatic robot assembly over the last 20 years. Recent reviews include Bakker and

Kuniyoshi (1996); Dillmann et al. (1996); Hirzinger (1996) and Ikeuchi et al. (1996). The goal is to replace the time-consuming manual programming of a robot by an automatic programming process, driven by showing the robot the task performed by an expert. Approaches include direct teaching (teaching by showing) in which the robot imitates human motions or teleoperated motions, directly learning a demonstrated policy (Widrow and Smith, 1964; Pomerleau, 1991; Nechyba and Xu, 1995; Grudic and Lawrence, 1996), and the approach followed in this paper: learning the intent of the demonstrator and using that intention model to plan or generate a policy (Friedrich and Dillmann, 1995; Tung and Kak, 1995; Delson and West, 1996). Using the same robot as the one used in this work, learning from demonstration was investigated by Kawato et al. (1994) and Miyamoto et al. (1996).

2 The Human Demonstration

The goal of the swing up task is to move the hand so that a pendulum, initially hanging down, swings up to the inverted position and then is balanced in the inverted position (Spong, 1995). The hand holds the axis of the pendulum, and the pendulum rotates about this hinge in an angular movement (Figure 1). A general version of this task allows both horizontal and vertical hand motion perpendicular to the pendulum axis. However, to simplify the task for this first implementation, we restricted the hand motion to a horizontal line with the pendulum axis perpendicular to this line. Humans naturally use both horizontal and vertical hand motion to do the task, but can restrict their motion to mostly horizontal hand motions if asked. Adding additional degrees of freedom to the hand motion makes the task easier for the human demonstrator, but more difficult for the robot learner, as assessing the demonstrator’s intent, building a model, and generating a policy are all more complex. Figure 2 shows the hand motion and pendulum motion from several human demonstrations, and a

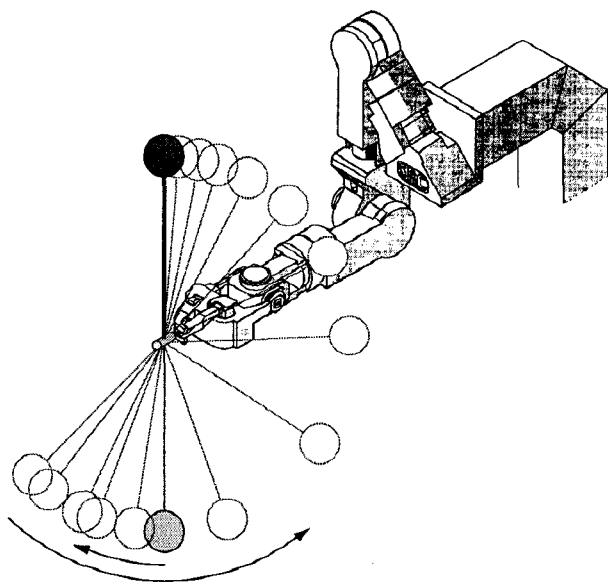


Figure 1: The SARCOS robot arm with a pendulum gripped in the hand. The pendulum axis is aligned with the fingers and with the forearm in this arm configuration.

“movie” of one of the human motions. Human motion was measured using a stereo vision system (QUICK-MAG) running at a 60Hz sampling frequency. The pendulum is marked by two colored balls that can be tracked in real-time. Note that the demonstration focuses on the variables important to the task, and the variables relevant only to the human arm such as joint angles are ignored. Our goal is task level learning from demonstration, rather than imitating or following patterns of arm movements.

3 Implementing The Task

We implemented learning from demonstration on a hydraulic seven degree of freedom anthropomorphic robot arm (SARCOS Dextrous Arm located at ATR, Figure 1). The robot observed its own performance with the same stereo vision system that was used to observe the human demonstrations. The combined vision and robot control system has about a 0.12s delay between an event and the response to the perception of that event. We implemented redundant inverse kinematics as well as real time inverse dynamics (An et al., 1988) to allow the robot to follow desired hand motions. The inverse kinematics implementation is based on a modified version of the extended Jacobian method (Baillieul, 1985). It uses a second order local optimization technique, incorporating an optimization criterion suggested for biological motor control in Cruse and Bruewer (1987).

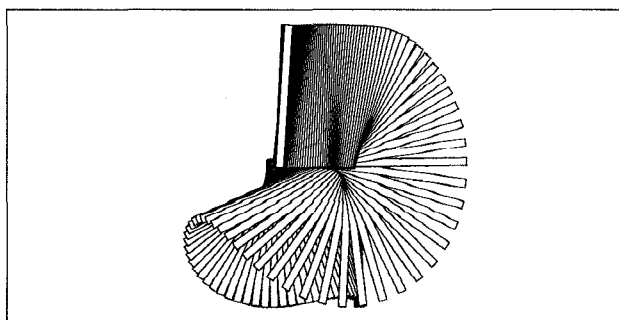
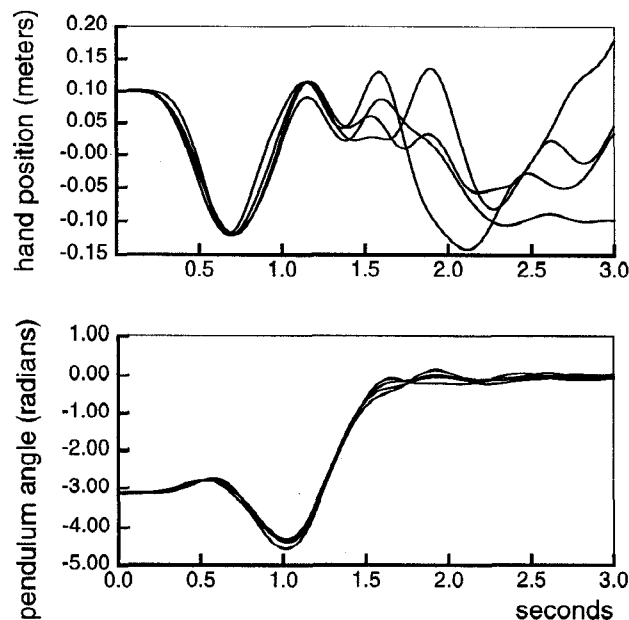


Figure 2: The human hand positions and pendulum angles for several demonstration swing ups, and pendulum configurations during one of the human demonstrations. The pendulum starts at $\theta = -\pi$ and a successful swing up moves the pendulum to $\theta = 0$. The hand motions are quite similar to each other during the swing up portion of the task, while the pendulum motions are quite similar to each other throughout the demonstration.

We decided to structure this task with two parts:

1. Learning to swing the pole up. The goal of this subtask is to move the hand so that the pole becomes upright with a small angular velocity. This motion is done without feedback (open loop).

2. Learning to balance the pole upright. This phase of the task is entered when the pole is nearly upright with a small angular velocity. Feedback is used to balance the unstable inverted pendulum. If the pendulum is within the capture region of the feedback controller the task has been completed successfully.

The most obvious approach to learning from demonstration is to have the robot imitate the human motion, by following the human trajectory $\mathbf{x}(t)$ (\mathbf{x} is the state of the task and t is time) and applying the correct commands $\mathbf{u}(t)$ (\mathbf{u} is the control input to the task). The dashed line in Figure 3 shows the robot hand motion as it attempts to follow the human demonstration of the swing up task, and the corresponding pendulum angles. Figure 3 also shows a movie of the corresponding motion. Clearly, direct imitation failed to swing the pendulum up, as the pendulum did not get even halfway up to the vertical position, and then oscillated about the hanging down position.

Another approach is to mimic the human response to each situation (a policy $\mathbf{u}(\mathbf{x})$, where \mathbf{u} is the action and \mathbf{x} is the task state). This typically requires large amounts of demonstration data, which is not available in this case and is usually expensive to collect. Also, expert human demonstrations tend to have limited variability (Figure 2) so the training data is too limited to cover enough states to learn an adequate policy. The discussion section explores this issue further.

The approach we will use is to apply optimal control to finding a swing up trajectory that works for the robot, based on learning both a model and an optimization criterion and using the human demonstration to initialize the optimization process.

- **Learning a model.** The robot learns a model of the task ($\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$) from its attempts to perform the task.

- **Learning an optimization criterion.** The robot learns an optimization criterion $C = \sum_k r(\mathbf{x}_k, \mathbf{u}_k, k)$ from the demonstration that ranks performance similar to human performance as more optimal.

- **Using the human performance to seed and limit the optimization process.** Most optimization processes only find local optima, and require an initial point or trajectory to seed the search. We use the human demonstration as the initial seed. The existence of a demonstration focuses the optimization process on

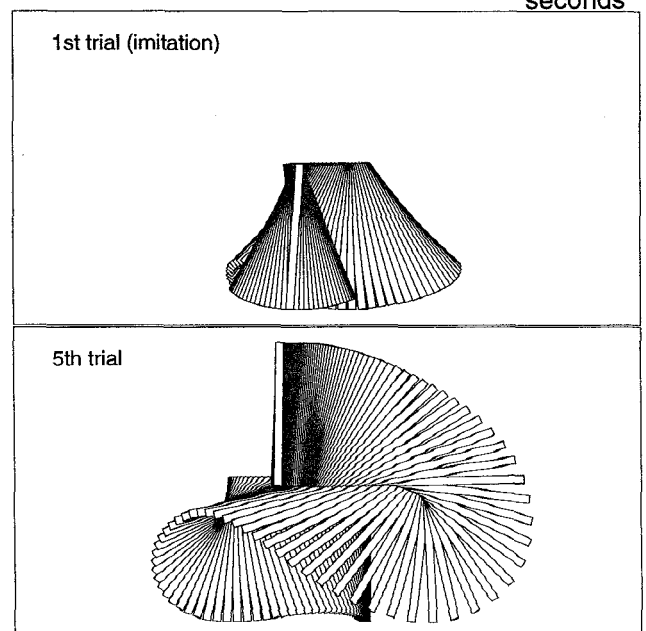
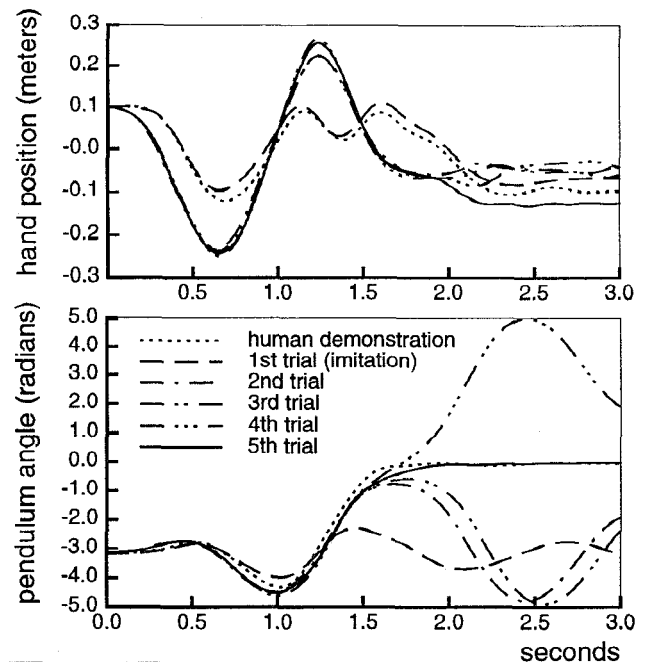


Figure 3: The hand positions and pendulum angles during robot learning from demonstration, pendulum configurations during robot imitation of a human demonstration (1st trial), and pendulum configurations after learning (5th trial).

a small volume of state space, and greatly reduces the need for exploration.

4 Learning a Model

This paper describes a knowledge-based parametric approach to learning a model. A knowledge-based parametric model is based on a priori knowledge of the physics of the task. A dynamic model of an idealized pendulum (all mass concentrated at the tip) attached on a horizontally moving hand is:

$$\dot{\theta}_{k+1} = (1 - \alpha_1) \dot{\theta}_k + \alpha_2 (\sin(\theta_k) + \ddot{x}_k \cos(\theta_k)/g) \quad (1)$$

where θ is the pendulum angle, $\dot{\theta}$ is the pendulum angular velocity, \ddot{x} is the horizontal hand acceleration, α_1 is the viscous damping, α_2 is $\Delta g/l$, Δ is the time step $0.0167s$, g is the gravitational acceleration $9.81m/s^2$, and l is the length of a pendulum $0.35m$. Idealized values for the α based on these parameters are $\alpha_1 = 0.0$ and $\alpha_2 = 0.47$. Identified values for the α based on large numbers of robot movements with the pendulum are $\alpha_1 = 0.0094$ and $\alpha_2 = 0.56$. Parametric models were constructed using linear regression in MATLAB. Note that the real pendulum did not have all of its mass concentrated at the tip, did not have ideal viscous friction, and the identified pendulum model included the robot dynamics between commanded hand accelerations and actual hand accelerations.

5 Learning to Balance

The robot learned to balance the inverted pendulum using methods described in Schaal (1997). 30 seconds of human balancing data allowed the robot to identify a model of the pendulum dynamics. The RFWR non-parametric modeling approach was particularly useful in producing a local linear model of the inverted pendulum dynamics (Schaal and Atkeson, 1996; Atkeson et al., 1996)

$$\dot{\theta}_{k+1} = 0.0051x_k + 0.0058\dot{x}_k + 0.47\theta_k + 0.997\dot{\theta}_k + 0.052\ddot{x}_k \quad (2)$$

Based on previous work on learning pole balancing from demonstration (Schaal, 1997) the following one step cost criterion was minimized:

$$r(\mathbf{x}, \mathbf{u}, k) = 125x^2 + 50\dot{x}^2 + 1200\theta^2 + 25\dot{\theta}^2 + 1.5\ddot{x}^2 \quad (3)$$

This optimization criterion produces the following balance controller, calculated using the discrete linear quadratic regulator design routine `dlqr` in MATLAB:

$$\ddot{x}_{commanded} = -6.9x - 9.3\dot{x} + 53\theta + 9.6\dot{\theta} \quad (4)$$

whose gains are similar to human gains identified from the balancing data. The $0.12s$ visual sensing delay can

be handled in two ways. Either the state of the system can be augmented by delayed commands, or a predictive controller can be used to predict the state of the system $0.12s$ in the future and apply the feedback controller to those states. The former method increases the complexity of a policy significantly, while the latter method requires a model to predict the state of the system. We used a predictive controller to compensate for the delay. The balance controller provided stringent constraints on the performance of the swing up controller because the robot workspace limited the capture region for successful balancing.

6 Learning to Swing Up

Given the failure of direct imitation to swing up the pendulum, some other approach must be considered. We used optimal control to automatically design a swing up trajectory for the pendulum. The robot learned an optimization criterion $C = \sum_k r(\mathbf{x}_k, \mathbf{u}_k, k)$ that penalizes deviations from the demonstration trajectory:

$$r(\mathbf{x}_k, \mathbf{u}_k, k) = (\mathbf{x}_k - \mathbf{x}_k^d)^T \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_k^d) + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \quad (5)$$

where the state is $\mathbf{x} = (\theta, \dot{\theta}, x, \dot{x})$, \mathbf{x}^d is the demonstrated motion, and the control is $\mathbf{u} = (\ddot{x})$. The penalty matrices were adjusted to make the robot trajectory similar to the human demonstration by increasing the penalty on the deviation in hand position, and were:

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R} = (1) \quad (6)$$

Note that the hand acceleration used in the demonstration is not included in the learned criterion. We found that including it allowed the optimized accelerations to exceed the capabilities of the robot, so setting the nominal control to zero helped reduce the size of the planned accelerations.

The first trial (direct imitation of the human motion) provided data that was used to build a model of the robot ($\alpha_1 = 0.0073$, $\alpha_2 = 0.55$). Standard second order trajectory optimization techniques (Dyer and McReynolds, 1970) were used to compute a locally optimal trajectory, using the human trajectory as the initial trajectory to refine. This optimization generated a new hand trajectory for the robot to follow. The executed hand motion and the corresponding pendulum motion are indicated as the 2nd trial in Figure 3. The hand motion had a larger excursion, and the pendulum motion came closer to the goal, but did not reach it.

The model of the robot was re-identified using only data from the 2nd trial ($\alpha_1 = 0.010, \alpha_2 = 0.57$). This trajectory is closer to the data distribution of a successful trajectory, so it may more accurately predict the hand motions necessary to swing the pendulum up. Again, optimization techniques were used to plan a new trajectory, using the previously planned trajectory to seed the optimization process. The executed hand motion and pendulum motion are indicated as the 3rd trial in Figure 3. The pendulum motion reached closer to the goal, but again failed to reach the capture region of the balance controller.

Re-identifying the model using only data from the 3rd trajectory did not result in significantly different model parameters. Because the model structure was not capable of fully representing the complexity of the task dynamics, the planner must compensate for residual modeling error. There are many possible ways for the planner to do this compensation. In the data presented in this paper the planner used binary search to adjust the viscous drag coefficient used in planning so that the planned hand trajectory would add more energy to the pendulum, since the energy of the pendulum was too little for it to reach the inverted position. The first viscous drag adjustment was to set ($\hat{\alpha}_1 = 0.015$). A new optimized trajectory was planned and executed, resulting in the 4th trial of Figure 3. In this trial the pendulum reached vertical with an angular velocity of 4rad/sec , which was too high for the feedback controller to capture, so the pendulum continued around and then oscillated around the down position with an angle of π . The second viscous drag adjustment was to set ($\hat{\alpha}_1 = 0.0125$), which resulted in the 5th trial of Figure 3. In this trial the pendulum reached vertical with a low enough angular velocity that the feedback controller was able to capture it. Figure 3 shows a movie of this trial. Subsequent trials with the same planned hand trajectory were also successful. Another modeling error compensation approach adjusts the desired velocity at the end of the swing up movement. This gives a similar performance to the approach presented here.

7 Discussion

The major lesson from this work is that it is difficult to learn a model well enough or fast enough to successfully perform the swing up task after only a few trials. Therefore, the planner must compensate for significant modeling error. The planner adjusted the task specification to increase the energy pumped into the pendulum motion. This discussion section addresses some of the questions raised by this implementation.

7.1 Why did direct imitation of human motion fail?

In our experiments direct imitation of human motion failed to swing the pendulum up. There are several possible reasons for this.

- **The robot controller is imperfect.** The robot does not exactly reproduce the human motion. We use a real time inverse dynamics model of the robot arm to provide feedforward control. This rigid body dynamics model is identified from actual robot motion (An et al., 1988). There are many effects it does not account for, including the dynamics of the hydraulic actuation system. Also, we do not take into account the effect of the pendulum on the robot motion, as in direct imitation it is assumed that a model of the pendulum dynamics is not available. Because we use low feedback gains comparable to human feedback gains, modeling errors have a substantial effect.

- **The task is different.** It is not clear that the response of the pendulum will be the same even if the robot hand motion matches the demonstrated hand motion. The robot grip of the pendulum axis is not exactly the same as the human's grip and the orientation of the pendulum axis relative to the motion of the hand is slightly different in the two cases. The human demonstration included slight deviations from a straight line and constant hand orientation, which the robot did not replicate in its movements.

- **Unstable tasks often require feedback control.** Open loop imitation of demonstrated motion will often not suffice to control an unstable system such as an inverted pendulum, bicycle, or unicycle. We note that there are vertical hand motions that will stabilize an inverted pendulum without feedback control (Blackburn et al., 1992) but in our work the only robot hand motion is horizontal. A feedback controller must therefore be learned, either directly from the demonstration data or indirectly as suggested in this paper.

7.2 Why not learn a policy directly?

As discussed previously, another approach is to learn a policy directly from demonstration by mimicking the human response to each situation. There are several reasons why this is not appropriate for this form of learning from demonstration.

- **The robot controller may be imperfect.** The robot may not be capable of exactly replicating human motion, and the task level learning approach of this paper may help compensate for the robot's lower level execution errors.

- **The task may be different.** In addition to the small task differences discussed previously, there may be large task differences. We have also used hu-

man demonstrations with another pendulum with a substantially different mass, mass distribution, length, moment of inertia, and friction. These demonstrations can be used to train optimization criteria and to provide initial trajectories for optimization, but not to train models or directly train policies or motions.

• **Expert performance generates limited training data.** Good human performance does not usually include errors, so it is difficult to learn responses to errors from expert demonstrations (Friedrich and Dillmann, 1995). Not enough states are visited to build an adequate policy. The robot's attempts to learn a task model also suffer from the limited training data, but data from the robot's attempts can be used to improve the model, while robot data cannot be used to improve a directly learned human policy. Expert performance must be degraded or perturbed to adequately train a robot policy directly. Also, because of the limited demonstration data collected, learning a policy from this data would have lead to motions similar to the demonstrated human hand motion, which is unlikely to have worked for the reasons discussed in Section 7.1.

7.3 Why not use "learning control"?

Learning control, which learns to follow a desired trajectory through repeated attempts to execute the trajectory, can be applied in two ways to the swing up task (see An et al. (1988) for further references to learning control). The robot can learn to follow the human hand trajectory more closely by adjusting the torques at its joints. This strategy is unlikely to swing the pendulum up for the reasons discussed in Section 7.1. The robot can learn to mimic the demonstrated pendulum motion by adjusting its hand trajectory. Learning control to match pendulum motion is difficult because the relationship between pendulum motion and hand acceleration is ill-conditioned when the pendulum is near horizontal (horizontal hand acceleration has little or no effect on the pendulum angle in this configuration). Errors in the executed pendulum trajectory while the pendulum is near horizontal can only be compensated for by large changes in hand acceleration, exceeding the capabilities of the robot.

Introducing an optimal control framework to learning from demonstration allows the robot not only to replicate the human demonstration, but perhaps to find a better or easier way to do the task.

7.4 What model is learned?

The balance controller learned a relationship between the measured hand acceleration and the pendulum motion, while the swing up controller learned a relationship between the commanded hand acceleration and

the pendulum motion. There are arguments for each approach. The relationship of the measured hand acceleration to the pendulum motion is closer to the idealized physics, and should be invariant over who is moving the pendulum base, a human demonstrator or a robot. The relationship between commanded and actual robot accelerations may be complex, may depend on the entire state of the robot, and may introduce additional state such as actuator dynamics that might not be included in the identified model. The commanded acceleration is known perfectly, while the actual acceleration may be difficult to measure, often requiring filtering to reduce noise, which introduces a delay in the measurement. Discrepancies between the actual and commanded hand acceleration must be eliminated if accurate trajectory following is required. Models based on actual hand acceleration produced stable feedback controller designs more often than models based on commanded acceleration for the balance controller. The swing up controller was open loop, so the issues were quite different and compensating for the relationship between commanded and actual hand accelerations was important.

7.5 Why isn't the learned robot hand motion more similar to the demonstration?

• **Humans used more capabilities than were available to the robot.** Even when asked to make only horizontal hand movements, the human demonstrations included small vertical movements and small changes in the orientation of the pendulum axis, which may have had a large effect on the pendulum motion. The robot ignored these vertical and rotational hand movements in the demonstration. Also, humans have higher acceleration and jerk (rate of change of acceleration) capabilities, so it was difficult to get the robot to exactly execute the human trajectory.

• **The task dynamics were different.** The human demonstrator and the robot used a different grip where the pendulum axis might have been oriented differently, so the pendulum dynamics may have been different. For the robot, the task dynamics included the relationship between the commanded robot hand acceleration and the actual robot hand acceleration. These dynamics were not present in the human demonstration.

• **The optimization criterion changed the robot motion.** The human hand acceleration was not included in the optimization criterion, so exact replication of the human motion was penalized relative to similar motions with smaller hand accelerations. With the optimization criterion that was used the robot was able to find a way to do the task that was easier for it

to execute due to the smaller hand accelerations, but still managed to swing the pendulum up with a pendulum trajectory similar to the pendulum trajectory of the human demonstration. If a more exact imitation of the demonstrated human hand motion was desired the penalty on deviations between the robot and human hand motion could be increased in the optimization criterion.

7.6 Why wasn't feedback used during the swing up?

The second order trajectory optimization process produces a time varying feedback controller for use during trajectory execution as well as a locally optimal trajectory (Dyer and McReynolds, 1970). However, we had difficulties applying feedback gains during pendulum swing up, due to the need to compensate for the sensing delay and since the feedback commands often exceeded the capabilities of the robot.

8 Conclusion

Human demonstration of a pendulum swing up task accelerated robot learning of the task in an initial implementation of programming by demonstration. The robot used optimal control to learn the task, based on only a single demonstration of the task. The human demonstration provided the optimization criterion and an initial trajectory to seed the optimization process.

Acknowledgments

The robot and support for both investigators was provided by the ATR Human Information Processing Research Laboratories. Additional support for C. Atkeson was provided under Air Force Office of Scientific Research grant F49-6209410362, and by a National Science Foundation Presidential Young Investigator Award. Additional support for S. Schaal was provided by the German Research Association, the German Scholarship Foundation, and the Alexander von Humboldt Foundation.

References

- An, C. H., Atkeson, C. G., and Hollerbach, J. M. (1988). *Model-Based Control of a Robot Manipulator*. MIT Press, Cambridge, MA.
- Atkeson, C. G., Moore, A. W., and Schaal, S. (1996). Locally weighted learning for control. *Artificial Intelligence Review*. in press.
- Bakker, P. and Kuniyoshi, Y. (1996). Robot see, robot do: An overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*, pages 3–11.
- Blackburn, J. A., Smith, H. J. T., and Gronbeck-Jensen, N. (1992). Stability and Hopf bifurcations in an inverted pendulum. *American Journal of Physics*, 60(10):903–908.
- Delson, N. and West, H. (1996). Robot programming by human demonstration: Adaptation and inconsistency in constrained motion. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 30–36.
- Dillmann, R., Friedrich, H., Kaiser, M., and Ude, A. (1996). Integration of symbolic and subsymbolic learning to support robot programming by human demonstration. In Giralt, G. and Hirzinger, G., editors, *Robotics Research: The Seventh International Symposium*, pages 296–307. Springer, NY.
- Dyer, P. and McReynolds, S. (1970). *The Computational Theory of Optimal Control*. Academic, NY.
- Friedrich, H. and Dillmann, R. (1995). Robot programming using user intentions and a single demonstration. In Kaiser, M., editor, *Proceedings of the 3rd European Workshop on Learning Robots (EWLR-3)*, Heraklion, Crete, Greece.
- Grudic, G. Z. and Lawrence, P. D. (1996). Human-to-robot skill transfer using the SPORE approximation. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 2962–2967.
- Hirzinger, G. (1996). Learning and skill acquisition. In Giralt, G. and Hirzinger, G., editors, *Robotics Research: The Seventh International Symposium*, pages 277–278. Springer, NY.
- Ikeuchi, K., Miura, J., Suehiro, T., and Conanto, S. (1996). Designing skills with visual feedback for APO. In Giralt, G. and Hirzinger, G., editors, *Robotics Research: The Seventh International Symposium*, pages 308–320. Springer, NY.
- Kawato, M., Gandolfo, F., Gomi, H., and Wada, Y. (1994). Teaching by showing in kendama based on optimization principle. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'94)*, volume 1, pages 601–606.
- Miyamoto, H., Schaal, S., Gandolfo, F., Gomi, H., Koike, Y., Osu, R., Nakano, E., Wada, Y., and Kawato, M. (1996). A kendama learning robot based on bi-directional theory. *Neural Networks*, 9(8):1281–1302.
- Nechyba, M. C. and Xu, Y. (1995). Human skill transfer: Neural networks as learners and teachers. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 314–319. IEEE Computer Society Press, Los Alamitos, CA.
- Pomerleau, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3:88–97.
- Schaal, S. (1997). Learning from demonstration. In Mozer, M. C., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA. NIPS96 proceedings, in press.
- Schaal, S. and Atkeson, C. G. (1996). From isolation to co-operation: An alternative view of a system of experts. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA.
- Spong, M. W. (1995). The swing up control problem for the acrobot. *IEEE Control Systems Magazine*, 15(1):49–55.
- Tung, C. P. and Kak, A. C. (1995). Automatic learning of assembly tasks using a dataglove system. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 1–8. IEEE Computer Society Press, Los Alamitos, CA.
- Widrow, B. and Smith, F. W. (1964). Pattern recognizing control systems. In *1963 Computer and Information Sciences (COINS) Symposium Proceedings*, pages 288–317. Spartan, Washington DC.