Robot Learning From 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

The goal of robot learning from demonstration is to have a robot learn from watching a demonstration of the task to be performed. In our approach to learning from demonstration the robot learns a reward function from the demonstration and a task model from repeated attempts to perform the task. A policy is computed based on the learned reward function and task model. Lessons learned from an implementation on an anthropomorphic robot arm using a pendulum swing up task include 1) simply mimicking demonstrated motions is not adequate to perform this task, 2) a task planner can use a learned model and reward function to compute an appropriate policy, 3) this model-based planning process supports rapid learning, 4) both parametric and nonparametric models can be learned and used, and 5) incorporating a task level direct learning component, which is non-model-based, in addition to the model-based planner, is useful in compensating for structural modeling errors and slow model learning.

1 LEARNING FROM DEMONSTRATION

One form of robot learning is to have robots 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, reacts, and handles errors 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 only 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 directly from the demonstration. This paper explores an approach to learning from demonstration based on learning a task model and a reward function, and using the model and reward function to compute an appropriate policy. This approach has been implemented on an anthropomorphic robot arm using a pendulum swing up task as an example. This paper describes the approach, its implementation, the lessons learned, and the applicability of various approaches to learning from demonstration. The paper also explores the roles of a priori knowledge and models during learning from demonstration, and compares the use of parametric and nonparametric models.

Learning from demonstration, also known as "programming by demonstration", "imitation learning", and "teaching by showing" is a topic that has received significant attention in the field of automatic robot programming over the last 20 years. Recent reviews include Bakker and Kuniyoshi (1996); Dillmann et al. (1996); Hirzinger (1996) and Ikeuchi et al. (1996). Approaches include direct teaching 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). Atkeson and Schaal (1997) report on a preliminary implementation of learning from demonstration for the pendulum swing up task.

2 THE PENDULUM SWING UP TASK

The pendulum swing up task is a more complex version of the pole or broom balancing task (Spong, 1995). The hand holds the axis of the pendulum, and the pendulum rotates about this hinge in an angular movement (Figure 1). Instead of starting with the pendulum vertical and above its rotational joint, the pendulum is hanging down from the hand, and the goal of the swing up task is to move the hand so that the pendulum swings up and is then balanced in the inverted position. The swing up task was chosen for study because it is a difficult dynamic maneuver and requires practice for humans to learn, but it is easy to tell if the task is successfully executed (at the end of the task the pendulum is balanced upright and does not fall down).

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 implementation, we restricted the hand motion to a horizontal line with the pendulum axis perpendicular to this line. Figure 2 shows the pendulum and hand motion from several human demonstrations, and Figure 3 shows a "movie" of one of the human motions. Figure 4 shows several human demonstrations in which more "pumping" motions were used. Human motion was measured using a stereo vision system running at a 60Hz sampling frequency. Note that the demonstration focuses on the variables important to the task (pendulum angle and angular velocity, and hand position and velocity), and ignores variables relevant only to the human arm such as elbow or shoulder position, or joint angles. Our goal is task level learning from demonstration, rather than matching patterns of human arm movements (Aboaf et al., 1989).

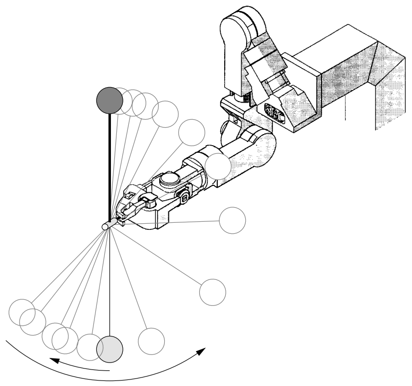

Figure 1: The SARCOS robot arm with a pendulum gripped in the hand. The pendulum axis is aligned with the fingers and with the forearm in this arm configuration.

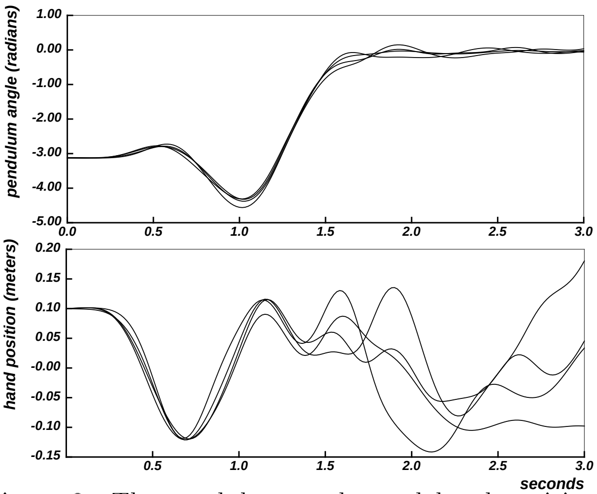

Figure 2: The pendulum angles and hand positions for several demonstration swing ups by a human. The pendulum starts at $\theta = -\pi$ and a successful swing up moves the pendulum to $\theta = 0$.

3 THE ROBOT IMPLEMENTATION OF 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. A Kalman filter was used to predict future sensory measurements to compensate for this delay. An idealized model of the pendulum dynamics was incorporated in the Kalman Filter and not changed during learning. Future research will address the learning of perceptual models. We implemented redundant inverse kinematics as well as real time inverse dynamics

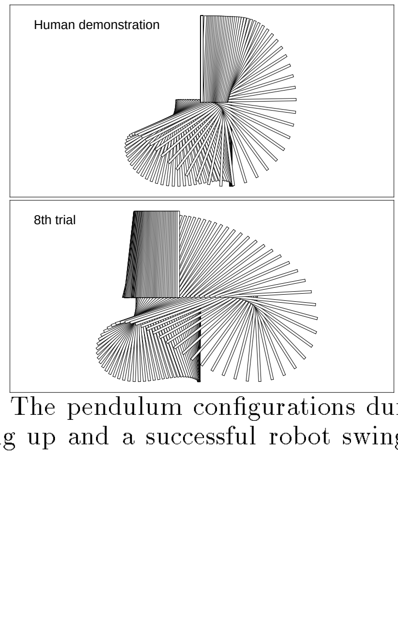

Figure 3: The pendulum configurations during a human swing up and a successful robot swing up after learning.

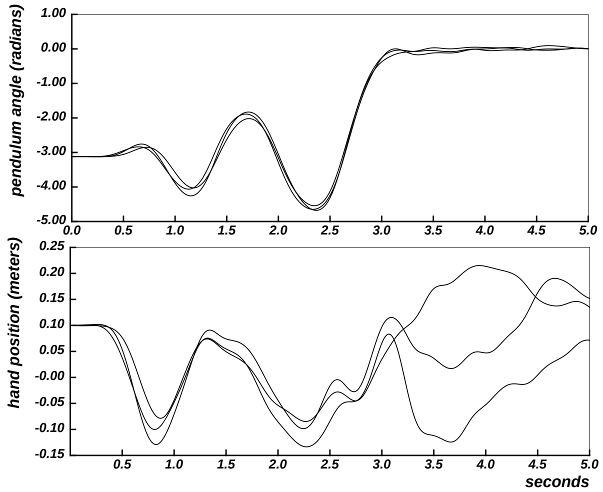

Figure 4: The human pendulum angles and hand positions for several demonstration swing ups in which two pumping motions were used.

(An et al., 1988) to allow the robot to follow desired hand motions.

In general we are interested in building complex motions out of simple components (motion primitives), as we feel this approach accelerates learning of new tasks. In this work we structured the swing up 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 hand trajectory. The dashed lines in Figures 5-10 show the robot hand motion as it attempts to follow the human demonstration of the swing up task, and the corresponding pendulum angles. 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. The discussion section explores possible explanations why direct imitation might fail.

The approach we will use is to apply a planner to finding a swing up trajectory that works for the robot, based on learning both a model and and a reward function and using the human demonstration to initialize the planning 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 a reward function.** The robot learns a reward function $C = \sum_k r(\mathbf{x}_k, \mathbf{u}_k, k)$ from the demonstration that ranks performance similar to the demonstration as better.
- **Using the human performance to seed the planning process.** Many planners only find locally optimal plans, and require an initial plan to seed the search process. We use the human demonstration as the initial plan. The existence of a demonstration focuses the search on a small volume of state space, and greatly reduces the need for exploration.

4 LEARNING THE TASK MODEL

We have used several techniques to learn task models, including knowledge-based parametric approaches and nonparametric approaches. Knowledge-based parametric models, which are based on a priori knowledge of the physics of the task, typically support more powerful generalization. Nonparametric models typically require less apriori knowledge. We will briefly outline both approaches.

Our knowledge-based parametric model is based on a discrete time dynamic model of an idealized pendulum (all mass concentrated at the tip) attached on a horizontally moving hand:

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

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 $0.29m$. Idealized values for the α based on these parameters are $\alpha_1 = 0.0$ and $\alpha_2 = 0.56$. 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 error dynamics of the deviations between commanded and actual hand accelerations.

Nonparametric models were constructed using locally weighted learning as described in (Atkeson et al., 1996a). These models did not use knowledge of the model structure (Equation 1), but instead assumed a general relationship:

$$\dot{\theta}_{k+1} = f(\theta_k, \dot{\theta}_k, x_k, \dot{x}_k, \ddot{x}_k) \quad (2)$$

The nonparametric models did use knowledge that the variable to predict was $\dot{\theta}_{k+1}$ and that variables that

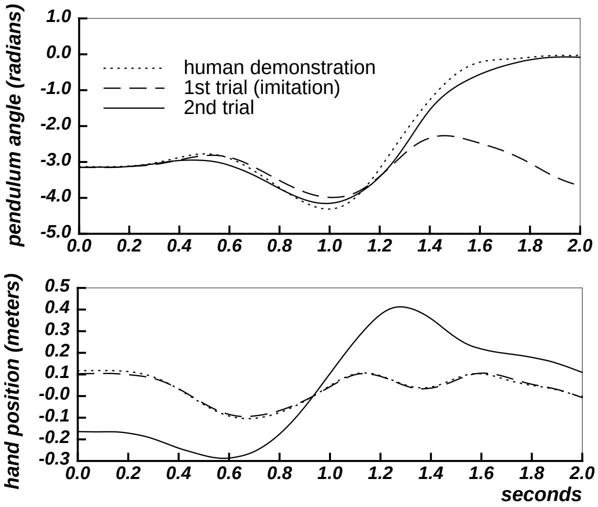

Figure 5: The hand and pendulum motion during robot learning from demonstration using a parametric model.

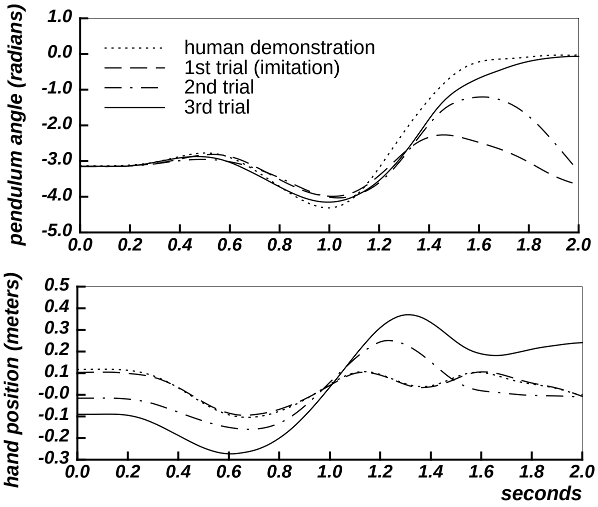

Figure 6: The hand and pendulum motion during robot learning from demonstration using a nonparametric model.

might be relevant to that prediction included θ_k, $\dot{\theta}_k$, x_k, $\dot{x}_k$, and $\ddot{x}_k$. Training data from the demonstrations was stored in a database, and a local model was constructed to answer each query. Meta-parameters such as distance metrics were tuned using cross validation on the training set. For example, cross validation was able to quickly establish that hand position and velocity (x and $\dot{x}$) played an insignificant role in predicting future pendulum velocities ($\dot{\theta}_{k+1}$).

5 LEARNING TO BALANCE

In this paper we will focus on learning the swing up movement primitive, and only briefly describe how the robot learned to balance the inverted pendulum. Schaal (1997) presents learning to balance in more detail. 30 seconds of human balancing data allowed the robot to identify a model of the pendulum dynamics. The RFWR nonparametric modeling approach was particularly useful in producing a local linear model of the inverted pendulum dynamics (Schaal and Atkeson, 1996; Atkeson et al., 1996b), for example:

$$\dot{\theta}_{k+1} = 0.47\theta_k + 0.997\dot{\theta}_k + 0.0051x_k + 0.0058\dot{x}_k + 0.052\ddot{x}_k \quad (3)$$

Based on previous work on learning pole balancing from demonstration (Schaal, 1997) the following reward function was minimized:

$$r(\mathbf{x}, \mathbf{u}) = 1200\theta^2 + 25\dot{\theta}^2 + 125x^2 + 50\dot{x}^2 + 1.5\ddot{x}^2 \quad (4)$$

Based on the learned model and this reward function, a balancing policy can be computed for these types of regulation tasks using standard techniques (in this case the discrete linear quadratic regulator design routine **dlqr** in MATLAB):

$$\ddot{x}_{commanded} = 55\theta + 10\dot{\theta} - 6.9x - 9.3\dot{x} \quad (5)$$

The feedback gains of this policy are similar to human gains identified from the balancing data. The balance policy provided stringent constraints on the performance of the swing up controller because the robot workspace limited the capture region for successful balancing. The transition from the open loop swing up motion to the balance controller occurred when the pendulum came within 0.5 radians of vertical with an upward angular velocity of less than $3 rad/sec$. Any non-zero pendulum angular velocity or robot hand velocity must be dissipated by the balance controller as the pendulum gets closer to vertical.

6 LEARNING THE SWING UP TASK

6.1 LEARNING WORKS WELL ON AN EASY TASK

We used a planner based on optimal control techniques to plan a swing up trajectory for the robot. The planner used a reward function $C = \sum_k r(\mathbf{x}_k, \mathbf{u}_k, k)$ that penalizes deviations from the demonstration trajectory sampled at $60 Hz$:

$$r(\mathbf{x}_k, \mathbf{u}_k, k) = (\mathbf{x}_k - \mathbf{x}_k^{\mathrm{d}})^{\mathrm{T}}(\mathbf{x}_k - \mathbf{x}_k^{\mathrm{d}}) + \mathbf{u}_k^{\mathrm{T}}\mathbf{u}_k \quad (6)$$

where the state is $\mathbf{x} = (\theta, \dot{\theta}, x, \dot{x})$, $\mathbf{x}^{\mathrm{d}}$ is the demonstrated motion, k is the sample index, and the control is $\mathbf{u} = (\ddot{x})$.

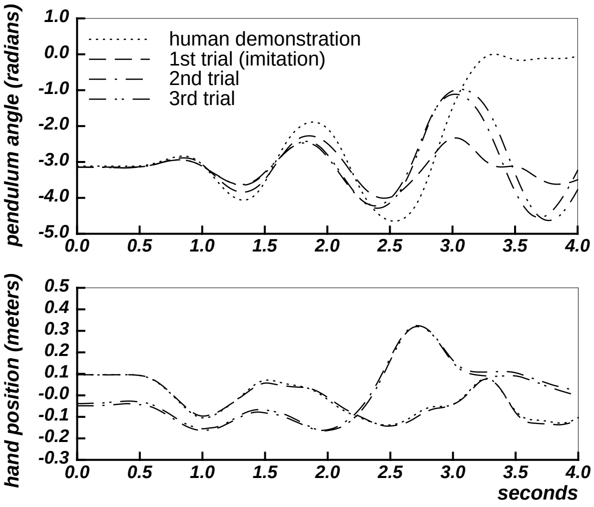

Figure 7: The hand and pendulum motion during robot learning from demonstration of a "two pump" trajectory using a parametric model. No improvement in performance is seen after the 3rd trial.

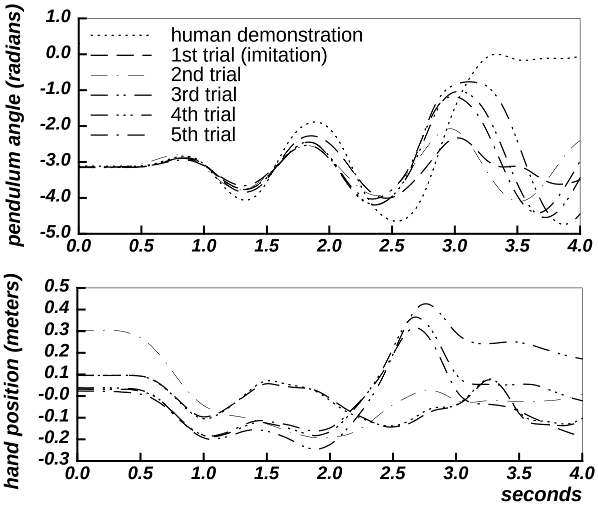

Figure 8: The hand and pendulum motion during robot learning from demonstration of a "two pump" trajectory using a nonparametric model. No improvement in performance is seen after the 5th trial.

In the first trial (dashed line in Figure 5 and Figure 6) the robot imitated the human hand motion. Although this motion failed to swing up the pendulum, it did provide data that was used to build a model of the robot.

Learning, using a parametric model, used this data to estimate model parameters for the knowledge base parametric model (Equation 1): $\alpha_1 = 0.0056$ and $\alpha_2 = 0.5506$. Trajectory optimization techniques (Cohen, 1992) were used to compute a locally optimal trajectory based on this model, using the human trajectory as the initial trajectory to refine. The state control trajectories were parameterized with splines, and the knot points of the splines were adjusted using gradient descent on the optimization criterion. 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 5. This motion succeeded in swingup up the pendulum.

Learning using the nonparametric model was not as fast (Figure 6). The same initial training data was used (trial 1). The next attempt got the pendulum up only half way. Adding this new data to the database and replanning resulted in a movement that succeeded (trial 3 in Figure 6).

6.2 LEARNING FAILS ON A HARD TASK

To demonstrate that the human demonstration had a large impact on the form of the learned trajectory and to explore learning a more difficult trajectory, the human demonstrator executed a different swing up trajectory. The first demonstration trajectory had one back and forth motion of the pendulum (Figure 2). A more difficult swing up task is to "pump" the pendulum twice before swinging it up (Figure 4). This maneuver stores energy in the pendulum that must be recovered in the final swingup. The model-based learning approach becomes slow or even gets stuck using either parametric (Figure 7) or nonparametric (Figure 8) models, as these models, with the given input variables, don't capture the full complexity of the task dynamics. In the parametric case the estimated parameters stop changing (compare α_1 and α_2 in trials 3 and 4 in Table 1). In the nonparametric case the new data does not result in a changed planned trajectory for the robot.

Trial	α_1	α_2	$\Delta\theta$
1	-	-	-
2	0.0063	0.5546	0
3	0.0082	0.5631	0
4	0.0085	0.5651	0.5
5	0.0083	0.5680	0.7

Table 1: Model parameters and shifts of final position during learning.

6.3 DIRECT TASK LEVEL LEARNING

Acknowledging that models are never perfect, we believe the failure of purely model-based learning to be due to a mismatch between the model structure (the

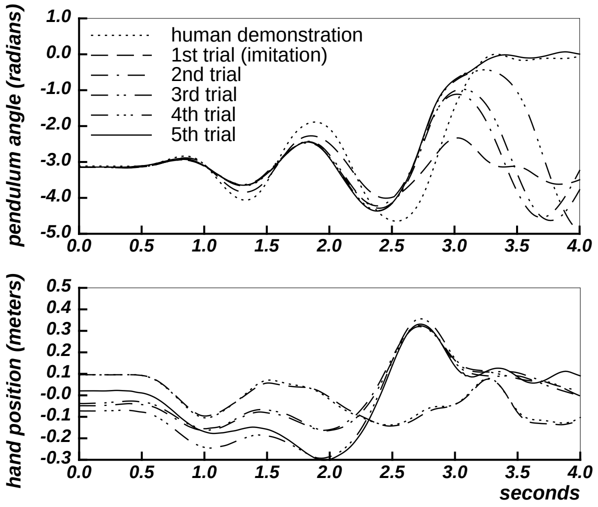

Figure 9: The hand and pendulum motion during robot learning from demonstration of a "two pump" trajectory using a parametric model.

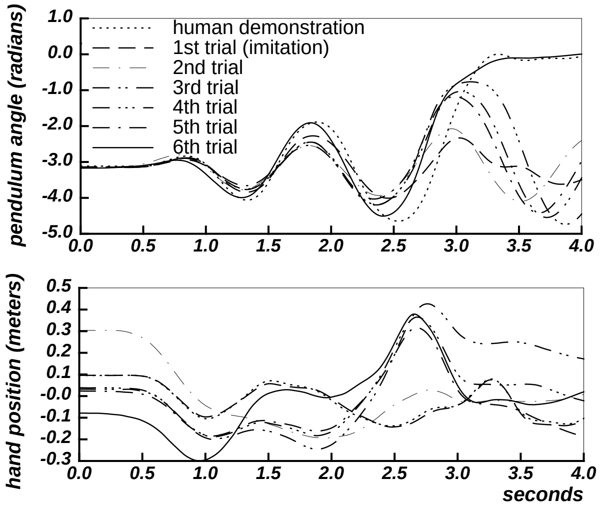

Figure 10: The hand and pendulum motion during robot learning from demonstration of a "two pump" trajectory using a nonparametric model.

form of the model) and the true system. In the case of the nonparametric model the inputs may not include all relevant factors (such as the state of the robot arm, which affects the robot trajectory following errors). We introduce direct task-level learning to overcome the occasional failure of purely model-based learning (Aboaf et al., 1989). For fast learning, the planner can compensate for deficiencies in the modeling approach that block or slow down further performance improvements by directly adjusting task parameters. For the swing up task we many natural ways to do this. The task is segmented into two phases, the swing up and the balance phase. The task planner can use a different desired pendulum target angle at the planned transition between the swing up and the balance phase to change the energy added to the pendulum during swing up. This parameter is suggested by the failure of the trajectory to match the planned trajectory for this variable. We implemented this approach for parametric model-based learning, and the adjusted position target is listed in the last column of Table 1. The adjustments were selected using an initial value of $0.5 radians$ and changed using binary search with an initial step size of $0.2 radians$. The additional trajectories in Figure 9 show successful learning using this type of direct task level learning on top of the model based learning. We have also used schemes which adjust the desired terminal velocity with equal success.

For nonparametric model based learning the adjustment of a final position target or a final velocity target was often not sufficient to allow the planner to find a trajectory that worked. The range of validity of the model was sufficiently limited that an increased terminal position or velocity caused the planned trajectory to leave the region of validity of the model, and to be quite inaccurate. An approach that was more robust was to adjust the demonstration not only at the end, but throughout its extent. For example, the 6th trial in Figure 10 shows the effect of adjusting the demonstration by multiplying the demonstrated pendulum velocity at each time t by $1 + c * t/T$, where $c = 0.3$ and T is the length of the swing up trajectory segment. We have implemented a straightforward binary search algorithm to search for appropriate coefficients c.

7 DISCUSSION

Conclusions from the implementations include:

- Simply mimicking the demonstrated human hand motion is not adequate to perform the pendulum swing up task.
- Learning a model and a reward function and using planning to compute a policy can get good performance rapidly (get the learner "into the ball park"), but convergence to final performance can be blocked by deficiencies in the modeling approach.
- Incorporating a task level direct learning component, which is non-model-based, is useful in compensating for structural modeling errors and slow model learning.

Why did direct imitation of the demonstrated human motion fail?: Directly imitating demonstrated motions, or directly learning a demonstrated policy, may not work because of imperfect execution or slight differences in the demonstrated and actual task. The robot controller is not perfect, and does not exactly reproduce the human motion. To assess how good the robot controller is, compare the dotted line (human demonstration) and the dashed line (robot imitation) in the hand position plots of Figures 5-10. It is not clear that the response of the pendulum will be the same even if the robot hand motion exactly matches the demonstrated hand motion. The robot grip of the pendulum axis is not the same as the human's grip due to differences in hand structure 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, and the robot did not replicate those deviations in its movements.

In addition to small task differences, there may be major differences between the demonstrated and actual task. We have also used human demonstrations with another pendulum with a substantially different mass, mass distribution, length, moment of inertia, and friction to train the robot to swing up the pendulum used in this study. These demonstrations can be used to train reward functions and to provide initial trajectories for optimization, but not to train models or directly train policies or motions, since the dynamics of the demonstration task are quite different from the actual task.

Why isn't the learned robot hand motion more similar to the demonstration?: The robot hand motion after learning was often quite different from the original demonstration. There are several possible explanations for this. As described previously, the different grips (and hand structure) of the robot and the human may make the task dynamics different. The human could move faster and in more ways than the robot. But an interesting consequence of the use of a model–based planner is that the robot could learn something new from demonstration, and find a new and better way (for the robot) to execute the task according to the reward function.

Why can't nonparametric modeling techniques eliminate the need for the task level direct learning component?: We applied locally weighted regression (Atkeson et al., 1996a) in an attempt to avoid the structural modeling errors of the idealized parametric models during swing up, and also to see if a priori knowledge of the structure of the task dynamics was necessary to learn the task. Although these models enabled the balance controller to be learned very quickly and allowed learning with less a priori knowledge, the planner still needed to compensate for residual modeling errors in order to successfully perform the swing up. Two reasons that nonparametric models failed to eliminate the need for task level direct learning are:

- **The planner needed oversmoothed models.** The planner used derivatives of the model, and took advantage of any modeling error to reduce the cost of the planned trajectory, so the planning process amplified modeling errors. It was necessary to bias the nonparametric modeling tools to oversmooth the data to allow the planning process to find reasonable trajectories. Therefore, the benefit of nonparametric modeling was reduced.

- **There wasn't enough data.** In order to learn a model of the complexity necessary to accurately model the full robot dynamics between the commanded and actual hand accelerations a large amount of data is required, independent of modeling technique. Because there are few robot trials during learning, there is not enough data to make such a model even just in the vicinity of a successful trajectory. If it was required that enough data is collected during learning to make an accurate model, robot learning from demonstration would be greatly slowed down.

Why does task level direct learning work?: Task level direct learning works well in this implementation because a single parameter is available that directly relates to the task structure. Initially the robot adds energy to the pendulum, and then the pendulum coasts to the goal position, ideally having just the right amount of energy to stop at the top. The desired transition pendulum angle or angular velocity provide fairly direct control over the amount of energy added to the pendulum. This form of task level direct learning reduces a highly complex and multidimensional learning problem to a simple low dimensional search (in this case a one dimensional search), greatly simplifying the learning problem. Highly appropriate task parameters are available because of the context provided by the model–based planning process. If there was no planner, the desired transition position or velocity parameter would be meaningless. If a random task parameter was arbitrarily chosen for adjustment, it is unlikely to be effective, although there are several task parameters that could also have been used. The task parameters to adjust can be found automatically by comparing the planned and actual trajectories. If a table of adjustable parameters was used to represent the robot trajectory or a more general policy, it would be much more difficult to find a comparable dimensionality reduction and a comparable reduction in learning complexity.

Implications for reinforcement learning: In reinforcement learning, there has been a debate between model-based (indirect) and non-model-based (direct) approaches. In the model-based case, a task model and/or reward function is learned from the environment, and an appropriate policy is computed. In the non-model-based case, such as Q learning, a policy is represented and adjusted directly, and the consequences of the policy adjustment are measured to guide future policy adjustments. Just as in learning from demonstration it is useful to combine model-based and non-model-based approaches in a hybrid approach, we expect that hybrid approaches will be useful for reinforcement learning.

The role of a priori knowledge: A delicate issue is the role of a priori knowledge in this approach to learning from demonstration. Clearly, a priori knowledge on the part of the human programmer is used to structure the control system and select inputs, outputs, and states. Less obviously, the problem of figuring out what perceptual values are relevant to the task has been addressed using a priori knowledge on the part of the human programmer. The vision system used in this study only measured relevant variables from the demonstration. The planner implicitly weights the relevance of each measurement in the demonstrated trajectory during the planning process. The model structure for the parametric model was based on a priori knowledge, but a nonparametric model has also been used with somewhat slower learning. The reward function for the balancing learner was completely set by the human programmer (based on his learning), and while the form of the reward function was set by the human programmer for swing up, the values of the reward function parameters were set based on the demonstration data. Human knowledge was used to break the task into a swing up and a balance component, and to use the desired angular velocity at the transition to adjust the model-based plan. It is not clear that this human knowledge can be easily automated.

What did the robot learn?: The human demonstration provided a trajectory that defined the task goal and seeded the learning process. The robot learned a model and the task level adjustments necessary to compensate for the modeling error. Based on this learned knowledge, the robot computed the detailed trajectory that performed the task.

This research leads to further research questions:

- What is the effect of increased dimensionality? In future implementations the robot will be allowed to make vertical motions, and perhaps orientation changes with its hand during swing up, which should allow the robot to more closely imitate the human demonstration. Will the increased dimensionality make learning easier or harder?
- Can a more robust planning process be developed which does not require oversmoothing the data, to make more effective use of nonparametric model learners?
- How can control and perceptual models be developed simultaneously, without changes in the perceptual model causing apparent changes in the robot or task dynamics?
- Can the task structure be found automatically, rather than having to be set by a human programmer?

8 CONCLUSION

Human demonstration of a pendulum swing up task enabled robot learning of the task. This paper explored an approach to learning from demonstration based on learning a task model and a reward function from the demonstration, and using the model and reward function to compute an appropriate policy. This approach was implemented on an anthropomorphic robot arm. Important lessons learned from the implementation included 1) simply mimicking motions is not adequate to perform some tasks, 2) a task planning module can use a learned model and reward function to compute an appropriate policy, 3) this model-based planning process supports rapid learning, 4) both parametric and nonparametric models can be learned and used, and 5) incorporating a task level direct learning component, which is non-model-based, in addition to the model-based planner, is useful in compensating for structural modeling errors and slow model learning.

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

Aboaf, E. W., Drucker, S. M., and Atkeson, C. G. (1989). Task-level robot learning: Juggling a tennis ball more accurately. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation.*

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. (1996a). Locally weighted learning. *Artificial Intelligence Review.* in press.

Atkeson, C. G., Moore, A. W., and Schaal, S. (1996b). Locally weighted learning for control. *Artificial Intelligence Review.* in press.

Atkeson, C. G. and Schaal, S. (1997). Learning tasks from a single demonstration. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation.*

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.

Cohen, M. F. (1992). Interactive spacetime control for animation. *Computer Graphics*, 26(2):293–302.

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.

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 cooperation: 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.