2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Towards a Real-Time Bayesian Imitation System for a Humanoid Robot

Aaron P. Shon Joshua J. Storz Rajesh P. N. Rao
Department of Computer Science and Engineering, University of Washington, Seattle WA USA
{aaron,jstorz,rao}@cs.washington.edu

Abstract— **Imitation learning, or programming by demonstration (PbD), holds the promise of allowing robots to acquire skills from humans with domain-specific knowledge, who nonetheless are inexperienced at programming robots. We have prototyped a real-time, closed-loop system for teaching a humanoid robot to interact with objects in its environment. The system uses nonparametric Bayesian inference to determine an optimal action given a configuration of objects in the world and a desired future configuration. We describe our prototype implementation, show imitation of simple motor acts on a humanoid robot, and discuss extensions to the system.**

I. INTRODUCTION

Imitation learning has become increasingly popular as a mechanism for imparting complex skills to robotic systems. A majority of the imitation learning systems presented in the literature perform *off-line* learning: first, a knowledgeable instructor demonstrates a skill; second, the system processes the data to permit deferred imitation. In contrast, this paper presents a preliminary implementation of a system for imitation learning that learns and reproduces behaviors in real time using motion capture and a humanoid robot.

We believe that on-line interaction is critical for practical imitation learning systems. Human instructors' time is a precious resource, making immediate feedback on the ability of a robot to imitate demonstrated actions potentially important. The system we present runs in real time on a single laptop (with an additional PC for performing motion capture). We employ various approximations to ensure real-time performance. We begin by describing the system architecture, including software and hardware components. Next, we describe the inference and learning algorithms we employ. Finally, we show results from our prototype implementation and mention numerous improvements we intend to implement in the future.

II. ARCHITECTURE

Figure 1 diagrams the architecture of our system. The major components include a Vicon motion capture system[1], a HOAP-2 humanoid robot, and the databases which collect world state information gathered from the Vicon system. The system can operate in one of 3 modes: i) "mimicry," where the upper body of the robot exactly matches the joint angles of the human instructor using real-time inverse kinematics computed by the motion capture system and a linear mapping solution; [2] ii) "watching," where the system observes the human instructor and computes an approximate policy being followed by the human from joint angles and object states; and iii) "imitation," where the system attempts to use learned policies to reach a desired goal state, and continues to collect data from the motion capture system regarding the effects of the robot's actions in the world.

A. Action encoding

The system captures training data from the human instructor using a Vicon vision-based (reflective marker) motion capture system. We chose Vicon over other non-marker vision-based approaches for two reasons: i) real-time performance and ii) high accuracy of reconstructed poses. Using marker-based motion capture allows us to focus on learning and control algorithms rather than state estimation. This setup allows us to simultaneously store joint angles (strictly for action reproduction and recognition), and object state information (to recognize how instructor and robot actions impact the world state, as described below).

In watching mode, the system begins recording human motions and object motions when the human starts moving. When the human stops, models are trained to encode the human's motion and the effects that motion had on objects in the world. Later, in imitation mode, the system will use these models to compute which actions are needed to transition from a current world state to a desired next state. For the results shown here, we used manual action segmentation (i.e. the human presses a key to indicate the beginning and end of motion). Alternatively, the system allows automatic motion segmentation. Automatic segmentation computes the variance of each human DOF over a sliding window, and compares each DOF's variance to a threshold. When all DOF variances fall below threshold, the human is no longer considered moving. Automatic segmentation potentially improves the flow of recording actions, at the cost of some additional noise in the action encodings.

Presently we use only the arms and waist of the HOAP-2, giving a total of 9 degrees of freedom. Inverse kinematics software provided by the Vicon system and a simple linear mapping translates in real time from human movement at time t to an observation of joint angles fit to the robot DOFs, $S_t = \{\theta_1 \dots \theta_9\}$. We employ Hidden Markov Models (HMMs) to encode and recognize actions executed by the human instructor. To ensure real-time training, we disregarded model selection, and found empirically that left-to-right HMMs with 10 states sufficed to encode the actions of interest here. The observation

[1] http://www.vicon.com/

[2] Included in attached video

1-4244-0602-1/07/$20.00 ©2007 IEEE.

density of each HMM state i is modeled as a Gaussian: $P_i(S_t) \propto \frac{1}{\det(\mathbf{\Sigma}_i)} \exp\left((S_t - \mu_i)^{\mathrm{T}} \mathbf{\Sigma_i}^{-1} (S_t - \mu_i)\right)$, with μ_i a mean pose and $\mathbf{\Sigma}_i$ a covariance matrix. The Baum-Welch algorithm trains HMM parameters given sequences of training data [12]. When a sequence of length T, $S_1 \dots S_T$, is observed, the system evaluates its database of known gestures to determine the log likelihood that each HMM generated the observed sequence (computed using the well-known Viterbi algorithm). The observed sequence is assigned to the model with the highest log-likelihood, and the winning model is retrained using the sequences assigned to it thus far. This assignment of trajectories to HMMs effectively performs hard clustering in the space of human joint angle sequences.

If the log likelihood is below a threshold ϵ for all models, we create a new HMM, train it on the sequence, and add the model to the gesture database. This approach is an approximation to the true estimate of each HMM generating an observation (in particular, it does not account for different prior probabilities of each model), but is quick to execute and works well in practice for a variety of gestures we have tested; empirically we found that $\epsilon = 0.0$ suffices to determine whether or not an observed sequence of data represents a novel action.

B. Replaying actions

In imitation mode, the system selects one of the actions learned during watching according to the process detailed below. When an action is selected, the associated HMM is recalled from the gesture database. We play out the joint positions encoded by the mean of each HMM state in left-to-right order, linearly interpolating the joint positions between each to produce smooth movement over the entire action. To avoid damage to the robot, all target poses are first checked in a physics simulator[3] to ensure that no self-collisions occur. If a collision is detected, the robot simply remains in its previous state.

Figure 6 shows the bounding boxes used for collision detection. The collision detection step is performed in-line as a last check. The realtime system does not know whether a collision has occurred and can only observe effects on the world state; it assumes that any motor command sent succeeds in being replayed. Actions that have predicted collisions will not be replayed and will consequently be marked by the system as not succeeding.

C. State encoding

Let $O_i(t)$ denote the coordinate frame of a single rigid 3D object i at time t; the motion capture system returns an observation vector $O_i(t) = \{X, Y, Z, \theta_x, \theta_y, \theta_z, \sigma_x, \sigma_y, \sigma_z\}$. $\sigma_x, \sigma_y, \sigma_z$ represent linear scaling factors for each spatial dimension. This vector is relative to one of 3 origins, O_v, O_h, O_r, respectively denoting the Vicon system origin, human origin, and robot origin. All observations are translated, rotated, and scaled to match the current coordinate origin; in watch mode, O_h is used, and in imitation mode, O_r is used. While our previous work explored nonlinear regression models for matching whole body poses [16], we found that simple linear transformations sufficed for the upper body imitation shown here.

To perform computation in our model, we discretize the observation vector. We represent all state distributions nonparametrically using sparse grids. This representation is motivated by real-time concerns, since learning simply involves updating histogram counts. Histograms also facilitate integration and convolution between arbitrary distributions, which are important for implementing Bayes-optimal action selection as described below. The drawback of histograms is their lack of generalization, though this disadvantage diminishes in the limit of many training samples. We are currently investigating alternative representations for forward and prior model distributions, including Gaussian processes [19].

III. BAYESIAN INFERENCE AND LEARNING

Our previous work described algorithms for combining forward models of environmental dynamics with priors over actions to yield a distribution over actions to take given a current state of the environment, a subgoal, and a desired goal state [15], [13]. The approach represents a greedy approximation to the problem of inferring actions using imitation in a model-based Markov decision process framework [18], and has been used to train software agents in a first-person shooter game that imitate human play strategies [17].

A. Action selection

Let s_t, s_{t+1}, s_G respectively denote the state of the world at time t, the desired next state at time $t+1$, and a final goal state. Our framework attempts to compute the following distribution over a discrete set of actions: $P(a_t|s_t, s_{t+1}, s_G)$. That is, calculating the distribution over possible actions given the current state, a next state (in a potentially multi-step goal), and the end goal. For many purposes s_{t+1} and s_G can be collapsed into a single step goal. We identify this distribution with an *inverse model* [10], and note that Bayes' rule allows us to compute an MAP (maximum a posteriori) estimate of the distribution using two other distributions:

$$
\begin{aligned}
P(a_t|s_t, s_{t+1}, s_G) &= \frac{P(s_{t+1}|a_t, s_t, s_G)P(a_t|s_t, s_G)}{P(s_t, s_{t+1}, s_G)P(s_t, s_G)} && (1)\\
&\propto P(s_{t+1}|a_t, s_t, s_G)P(a_t|s_t, s_G) && (2)\\
&\propto P(s_{t+1}|a_t, s_t)P(a_t|s_t, s_G) && (3)
\end{aligned}
$$

$P(s_{t+1}|a_t, s_t)$ is a *forward model*, predicting how the environment will react to actions taken in the world; $P(a_t|s_t, s_G)$ is a *prior model* encoding a preference for certain actions given that the world is in state s_t and the goal is s_G. The MAP estimate in this case represents a greedy approximation to the general solution for planning in a Markov decision process, where s_G can be identified with a state whose value function estimate is high.

B. Learning from experience

As the robot acts on objects, it updates its estimate of actions' effects based on their sensory consequences (world

[3] Webots, from: `http://cyberbotics.com/`

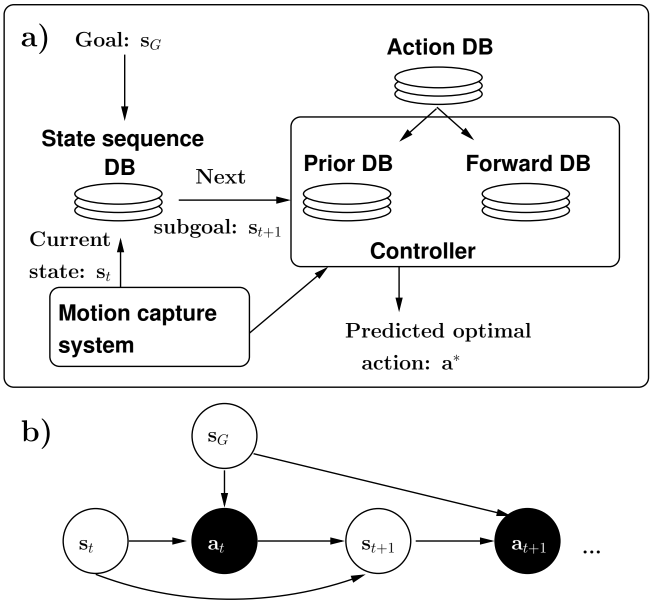


Fig. 1. **System architecture:** (a) During "watching" mode, the system continually updates conditional distributions $P_v(a_t|s_t, s_G)$, $P_v(s_{t+1}|a_t, s_t)$ for each object v (stored in the "Prior" and "Forward Model" databases, respectively), a distribution $P(a_t|O_1 \ldots O_t)$ for the action the human teacher is showing based on motion capture frames $O_1 \ldots O_t$ (the "Action" database), and a distribution $P(s_{t+1}|s_t)$ that encodes a sequence of subgoals over time (the "State sequence" database). (b) One time slice of the underlying graphical model for imitation. White nodes' values are directly observed from the environment; black nodes' values must be inferred, either from observation of the human teacher (while "watching") or from the Bayesian-optimal action selection described below (during "imitation"). Due to our system's real-time requirement, we compute a 1-step greedy approximation of node values in the model rather than full backward smoothing over all time steps.

state in this case). When learning starts, the robot employs the forward model learned by watching the human; as it accumulates more samples from the environment, the robot begins to use its own internal forward model. Denote the model derived from human actions $P_h(s_{t+1}|a_t, s_t)$; the model derived from robot experience is $P_r(s_{t+1}|a_t, s_t)$. During imitation mode, after executing an action, the robot updates $P_r(s_{t+1}|a_t, s_t)$ to reflect changes in environmental state. When computing the MAP estimate of each action's chance to succeed according to Equation 3, the system uses P_h if the number of samples collected for $P_h(\cdot|a_t, s_t) \geq P_r(\cdot|a_t, s_t)$, and P_r otherwise. This allows the robot to sample widely from P_h until the more robot-reproducible actions are encoded in P_r.

IV. RESULTS

Figure 2(b) and our video show examples of upper body mimicry using the Vicon system. Mimicry is useful for testing correspondences between the human's body and that of the robot, as well as for determining how variations in motion will be classified by the gestures database. Next we describe forward model learning and goal-directed imitation.

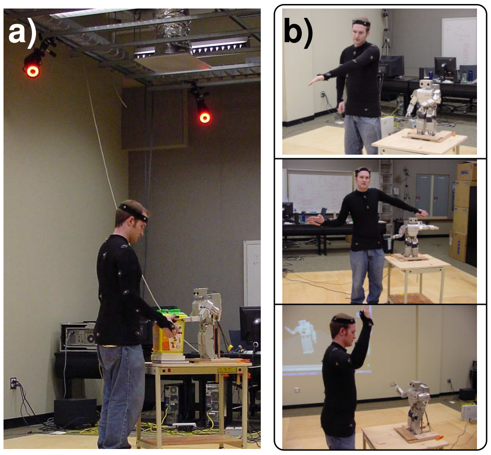


Fig. 2. **Setup and mimicry mode:** (a) Experimental setup. 12 motion capture cameras fit marker positions to a skeletal model in real time ($\sim$ 120 frames/sec). (b) Example snapshots of mimicry mode, in which the robot matches human body poses in real time. Mimicry is useful for testing out actions without committing them to the model databases.

A. Learning forward models

Figure 3(a) shows a top-down view of using the system to learn actions and the sensory consequences of those actions on an object. The human exhibits one of three actions (push box forward, push left, push right). The system automatically determines that 3 HMMs explain the observed paths, and successfully labels all 90 action examples. The figure shows normalized histograms that represent where the box ends up at the completion of each action, to a resolution of 1 cm^2. All state variables are integrated out except for the X-Y positions; that is, each histogram plots:

$$\sum_{Z'}\sum_{\theta'_x}\sum_{\theta'_y}\sum_{\theta'_z} P\left(\{Z', \theta'_x, \theta'_y, \theta'_z\}|\{Z, \theta_x, \theta_y, \theta_z\}, \hat{a}_t\}\right) \tag{4}$$

where $\{Z \ldots\}, \{Z' \ldots\}$ respectively denote state vector components at times t and $t+1$. As might be expected, errors on this simple example approximate normal distributions. For more complex objects, or more abstract actions and state spaces, we expect that nonparametric representations will be critical for predicting future world state.

Figure 3(b) shows an equivalent forward model overlay for box pickup as gathered from our main real time experiment. As our aim was consistency, the distribution has much less variance than the top-down example.

B. Imitation

To illustrate our algorithm for Bayesian action selection, we show results on a simple task: picking up a large box. The goal is to lift the box off a table surface and hold it at chest height (taking into account the linear scaling that compensates for size differences between human and robot). The human shows two broad types of action, lifting the box with one hand, and lifting with two hands. 25 example trajectories of the human lifting the box comprise the training data. At the end of training, the human also captures the desired goal state using the motion capture

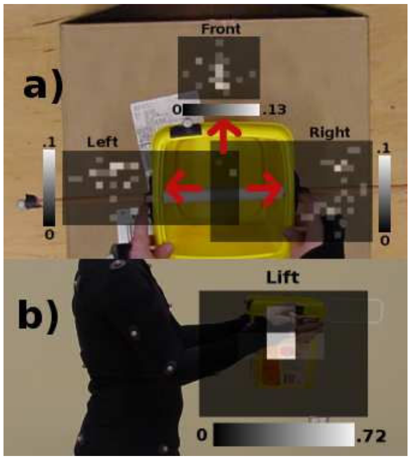


Fig. 3. **Learning forward models from motion capture:** (a) Forward models learned by the system after observing 3 different actions performed by the human (push box forward, push left, push right). 30 samples were collected for each action's consequences, all starting from the same initial state. Grayscale indicates probability of the human's action moving the box to a particular 1 cm^2 grid cell in the X-Y plane. The system used HMMs to automatically create a separate model for each gesture, and correctly classified each of the total of 90 actions to update the model shown here. (b) Forward model learned by the system for the box lift experiment (as described below).

system (by pressing a key). The robot's small size prevents it from lifting the box with one hand. Because this task involves just one step, we collapse s_G and s_{t+1} into a single state. Our action selection algorithm therefore finds the MAP estimate of $P(s_{t+1}|s_t, a_t)P(a_t|s_t)$ rather than using the multistep approach described in our previous work.

Figure 4 shows how the robot combines prior information from the human instructor with the effects of its own actions. Although 8 total actions were discovered by the gestures database (based on log likelihood estimates), we show the two with the highest prior probability (a one-hand lift and a two-hand lift). The remaining 6 actions are minor variations of the one-hand lift and two-hand lift that were classified differently by the action database. In 4(a), we show the total log likelihood of each action. The robot interacts with the box for a total of 40 iterations. Each time, it computes the Bayesian MAP estimate of the optimal action. [4] At iteration 14, the robot has acquired sufficient samples to use its own forward model estimate rather than the estimate acquired from the teacher. Because the robot is incapable of lifting the box with one hand, its forward model estimate causes the log likelihood of the one hand action to drop. The two hand action is much more reliable (shown as an increase in log likelihood), although sometimes this action fails as well (shown as slight dips in the plot).

In 4(b), we give snapshots of how the log likelihood of each action is determined at iterations 1 and 40. The bar graphs show the log prior $\ln(P(a_t|s_t))$ (black bar) and the log probability of the forward model $\ln(P(s_{t+1}|a_t, s_t))$ (gray), which together yield the total log likelihood of executing each action. A smaller total bar size means the action has a higher log likelihood according to our action selection algorithm. Photos show the robot attempting each action. Note that while the prior distribution does not change over time (expressing the human's preference for how to execute the action), the forward model does change over time (reflecting the robot's estimate of which action is likely to reach the correct goal state). The prior distribution encodes the human's preferences over which actions to take in a situation, and could be used, for example, to privilege certain actions as socially or contextually desirable.

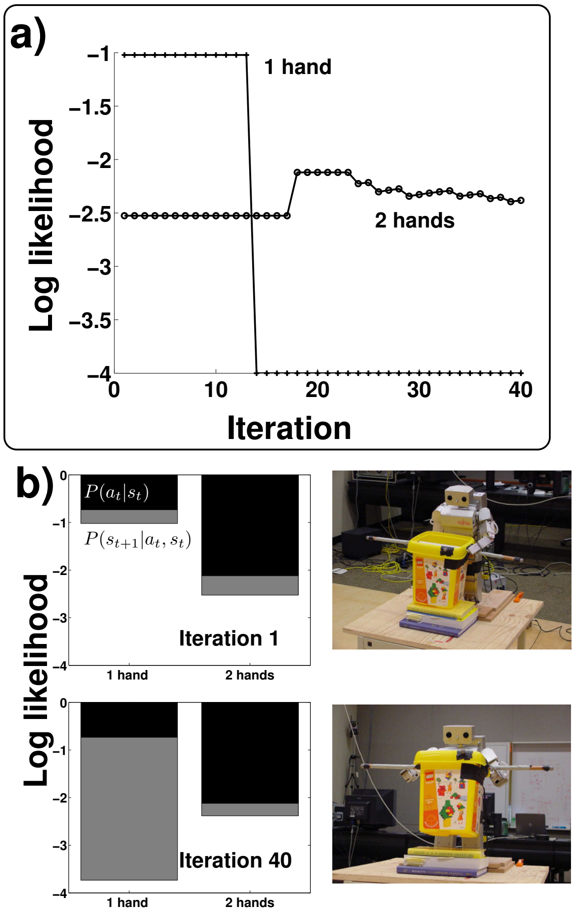


Fig. 4. **Using Bayesian inference to discover which actions to imitate:** (a) The robot learns over a series of 40 interactions with the box. Initially, the forward model learned from watching the human has more samples than the model gleaned from the robot's own experiences. Based on a prior bias toward one-handed lifting, the system selects a one-handed lift that is impossible for the robot to execute successfully. After iteration 14, the number of samples in $P_r(s_{t+1}|a_t, s_t) > P_h(s_{t+1}|a_t, s_t)$ for the one-handed lift action, and the system reevaluates the action's chances of success. At this point, the system instead begins to execute a two-handed lift. Once it has observed 3 samples of the two-handed lift, it begins to use $P_r(s_{t+1}|a_t, s_t)$ for the two-handed lift as well. Occasional mistakes in lifting cause small dips seen in the log likelihood, but overall the estimated a posteriori probability of success for the two-hand lift is higher than the one-hand lift. (b) Breakdown of log likelihoods into the prior distribution $P(a_t|s_t)$ (black bar) and the forward model distribution $P(s_{t+1}|a_t, s_t)$ (gray bar) at two points in the learning process. At iteration 1, the robot attempts to use a one-handed lift (photo at right). After 40 iterations, the robot's estimate of each action's log likelihood of success has changed: the two-handed lifting action is much more likely to yield the desired goal state.

[4] When the robot significantly disturbs the box (e.g. when it knocks the box off the table), the human replaces the box to its initial state. One goal of a more complete system would clearly be to consistently restore the world to its original state so as to avoid disturbing the human instructor.

Total time to train the system and for the robot to determine reliability of its actions in this simple task is on the order of 10 minutes. Due to the system's realtime nature, the human can intervene if necessary and show the robot a new action for executing the task at any time. There is no time or learning penalty for switching between system modes (mimicry, watching, imitation).

C. *Multiple instructors*

In 5, we see how different instructors can be used be the system to encode the same actions. As only joint-angle and world state information are used to encode actions, if one instructor closely mirrors another's demonstration, the system can correctly classify both as the same action. The same design that allows for intra-instructor variation with the same goal also helps with inter-instructor variation: with enough observations the system can use the best action (as learned through imitation and the robot's prior model P_r) while continuing to observe a different action. Only s_t, s_{t+1}, and s_G are relevant.

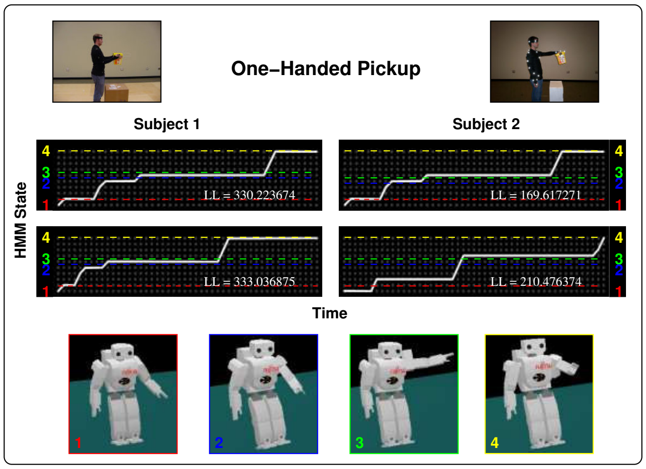
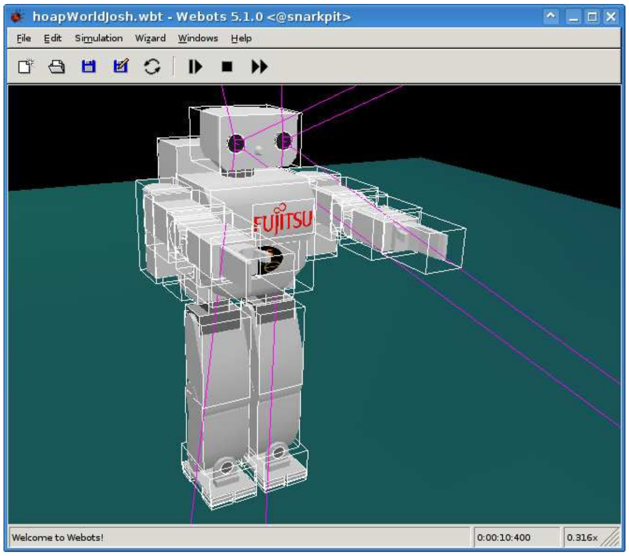


Fig. 5. **Matching actions from multiple instructors:** Viterbi paths for the one-hand pickup action, as observed from two different instructors at two different velocities. The similarities show the robustness of the system when presented with minor variations, and also illustrate how action models learned from one user can be recognized when observing another. There are 10 states in each chain and a variable number of time datapoints. Four key joint-angle states within the actions are also displayed on the robot.

V. CONCLUSION

We have described a prototype system for real-time, interactive imitation between a human teacher and a humanoid robot. The system learns to cluster human motion patterns into discrete actions. The system also learns predictive models of how motor acts affect objects in the environment, and uses its models to compute which action it should take given the state of the environment. By allowing the human to intervene at any point in the learning process, and by allowing the robot to use priors imposed by the human to guide during its own learning process, robotic skill learning can be achieved while ensuring efficient use of the human instructor's valuable time. Adding an element of self-directed learning also means the correspondence between human and robot actions need not be perfect; the robot itself can figure out, to some degree, which actions are most effective. Despite the presently limited repertoire of the system (see future extensions below), it serves as a proof of concept for combining self-directed learning, turn-taking interaction, and imitation using objects in a real-time framework.



Welcome to Webots!

Fig. 6. **Collision simulator:** A virtual view of the robot showing the bounding boxes used for collision detection. Each pair of links that can possibly collide is checked before any new motor command is sent to the robot.

A. *Related work*

Despite the overall dearth of efforts for real-time imitation learning of actions on objects, systems with similar goals to ours have recently appeared. Some systems have concentrated on full- or upper-body mimicry in real time using inverse kinematics [14], [4]. Examples of goal-directed efforts with a HOAP-2 humanoid are [3], [2], [1], where HMMs encode target positions for the robot's joints and hands. When imitating, the system attempts to minimize a cost function (using Lagrangian optimization) consisting of an accuracy term bringing the robot's hands close to a target point and a set of kinematic constraints. The system differs from ours in three important respects: i) teaching is performed through backdriving the robot's limbs, an option that may not be possible for certain robotics systems; ii) the system uses probabilistic models to encode actions, but does not learn probabilistic forward models for manipulated objects; and iii) Lagrangian optimization is employed rather than finding a Bayesian MAP estimate of action suitability. Additionally, the system's focus does not appear to be a closed-loop, teaching framework as we have described here. In another system [8], real-time imitation is performed using a humanoid robot. Actions are encoded as HMMs, and performed subject to one of several preprogrammed kinematic constraint profiles announced by the user at training time (indexed via a speech recognition system). Like ours, this system focuses on closed-loop interaction between a robot and a motion-captured instructor; however, it does not focus on learning models of objects in the environment. In [9], a humanoid reproduces gestures shown by a human instructor using an onboard vision system and a neural network based on findings from the mirror neuron system in primates. The self-directed learning aspect of our work is similar to [6], where a humanoid uses optic flow to autonomously learn how its manipulator changes the state of objects in the world. Finally, [5] uses Bayesian methods to learn a forward model for a robotic gripper, and to imitate a human waving gesture. It differs from our system in not focusing on object manipulation, and on learning the forward model from environmental experience alone rather than bootstrapping from models given by the human.

B. Extensions

Future work will concentrate on both low-level and high-level aspects of the model. Inclusion of an inverse kinematics solution will allow the system to modify paths from the HMM encoding of actions so that constraints such as relative distance from hands to objects is obeyed. Although we currently select the MAP action from the Bayesian estimate, in highly stochastic environments it might be advantageous to sample from the posterior distribution (a strategy known as "probability matching" in the biological literature [11], [7]. At the higher level, encoding longer-term subgoals using several hierarchical layers (rather than the 2 shown here) will enable more complex behaviors. We also anticipate implementing the algorithm for plan recognition described in [15], [13] to predict intent given observations of the human instructor's actions. In the long term, the system's model databases could facilitate acquisition of a library of goal-directed behaviors and predictive models of actions' effects on objects in the world.

VI. ACKNOWLEDGMENTS

We thank David Grimes, Keith Grochow, and Danny Rashid for their helpful advice and assistance with the motion capture and collision detection components. We gratefully acknowledge the support of the NSF AICS program. We thank the reviewers for their comments.

References

[1] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, 2006.

[2] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

[3] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 36, no. 5.

[4] G. Cheng, A. Nagakubo, and Y. Kuniyoshi, "Continuous humanoid interaction: An integrated perspective—gaining adaptivity, redundancy, flexibility—in one," in *Proc. First IEEE-RAS International Conference on Humanoid Robots*, 2000.

[5] A. Dearden and Y. Demiris, "Learning forward models for robotics," in *Proceedings of IJCAI*, 2005, pp. 1440–1445.

[6] P. Fitzpatrick and G. Metta, "Towards manipulation-driven vision," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[7] R. J. Herrnstein, "Relative and absolute strength of responses as a function of frequency of reinforcement," *Journal of Experimental Analysis of Behavior*, vol. 4, pp. 267–272, 1961.

[8] T. Inamura, N. Kojo, T. Sonoda, K. Sakamoto, K. Okada, and M. Inaba, "Intent imitation using wearable motion capturing system with on-line teaching of task attention," in *Proc. IEEE Humanoids*, 2005.

[9] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior*, vol. 12, no. 2, pp. 93–115, 2004.

[10] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol. 16, pp. 307–354, 1992.

[11] J. R. Krebs and A. Kacelnik, "Decision making," in *Behavioural Ecology (3rd edition)*, J. R. Krebs and N. B. Davies, Eds. Blackwell Scientific Publishers, 1991, pp. 105–137.

[12] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[13] R. P. N. Rao, A. P. Shon, and A. N. Meltzoff, "A Bayesian model of imitation in infants and robots," in *Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dimensions*, K. Dautenhahn and C. Nehaniv, Eds. Cambridge Univ. Press, (forthcoming).

[14] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling real-time full-body imitation: A natural way of transferring human movement to humanoids," in *IEEE Conference on Robotics and Automation*, 2003.

[15] A. P. Shon, D. B. Grimes, C. L. Baker, and R. P. N. Rao, "A probabilistic framework for model-based imitation learning," in *Proc. Ann. Mtg. Cognitive Science Society*, 2004.

[16] A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao, "Learning shared latent structure for image synthesis and robotic imitation," in *Advances in NIPS 18*, 2006.

[17] C. Thurau, T. Paczian, and C. Bauckhage, "Is Bayesian imitation learning the route to believable gamebots?" in *Proc. GAME-ON North America*, 2005, pp. 3–9.

[18] D. Verma and R. P. N. Rao, "Goal-based imitation as probabilistic inference over graphical models," in *Advances in NIPS 18*, 2006.

[19] C. K. I. Williams and C. Rasmussen, "Gaussian processes for regression," in *Advances in NIPS*, D. Touretzky and M. Hasselmo, Eds., vol. 8, 1996.