Proceedings of the 2001 IEEE/RSJ
International Conference on Intelligent Robots and Systems
Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001

Robust learning of arm trajectories through human demonstration

Aude Billard & Stefan Schaal
Computer Science and Neuroscience, HNB-103, Univ. of Southern California, Los Angeles, CA 90089-2520
Kawato Dynamic Brain Project (ERATO/JST), 2-2 Hikaridai, Seika-cho, Soraku-gun, 619-02 Kyoto, Japan
{billard,sschaal}@usc.edu

Abstract

We present a model, composed of hierarchy of artificial neural networks, for robot learning by demonstration. The model is implemented in a dynamic simulation of a 41 degrees of freedom humanoid for reproducing 3D human motion of the arm. Results show that the model requires few information about the desired trajectory and learns on-line the relevant features of movement. It can generalize across a small set of data to produce a qualitatively good reproduction of the demonstrated trajectory. Finally, it is shown that reproduction of the trajectory after learning is robust against perturbations.

1 Introduction

Robot teaching by demonstration makes an increasing body of robotics research. A large part of these efforts focus on assembly task-learning. Movements of a human performing object moving/stacking tasks are recorded, either using video images [1, 2] or using a manipulandum, for measuring directly the joint torques [3, 4]. Data are then analyzed to determine the torques required for an industrial robot arm to reproduce the motions with similar precision. A major issue in robot teaching by demonstration is to define an algorithm which can remove inconsistencies across the demonstrations, while keeping enough information to allow reproduction of fine features of the movement [5]. As the level of precision of data segmentation is highly task-dependent, it requires the development of adaptive and learning algorithms [6].

Human ability to imitate oustands that of all other animals [7]. Human imitation ranges from gross reproduction of general body postures (pantomine) to precise reproduction of joint motions (as in dance, surgery, etc). Recent trends in robotics takes inspiration in studies of human imitation to develop architectures for visuo-motor control and learning in robots which would show some of the flexibility of natural systems [8, 9, 10, 2, 11, 12]. In [9, 13], we developed a biologically plausible model of human imitation. In this paper, we discuss the potential of this model for controlling a 3 degrees of freedom robot arm.

2 The model

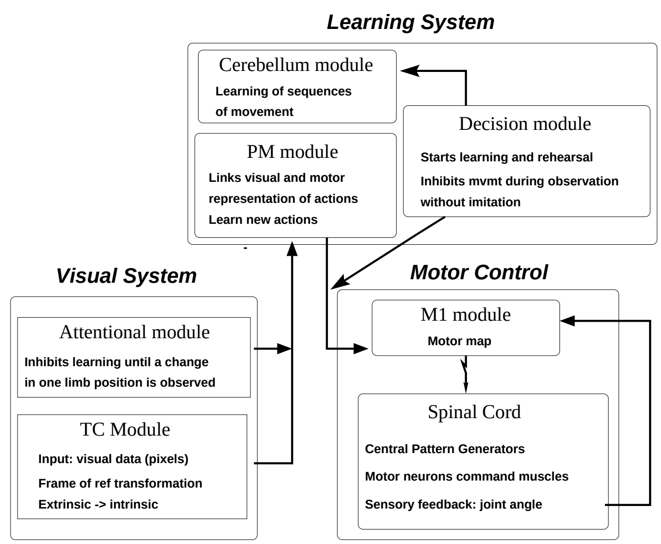

Figure 1: The architecture consists of seven modules which give an abstract and high-level representation of corresponding brain areas involved in visuo-motor processing. The seven modules are: the attentional and temporal cortex modules, the primary motor cortex and spinal cord modules, the premotor cortex and cerebellum module, and the decision module.

This work builds upon our general model of learning by imitation ([9, 13]), see Figure 1. The model is biologically inspired in its function, as its composite modules have functionalities similar to that of specific brain regions, and in its structure, as the modules are composed of artificial neural architectures. It is loosely based on neurological findings in primates and incorporates abstract models of some brain areas involved in visuo-motor control, namely the temporal cortex (TC), the spinal cord, the primary motor cortex (M1), the premotor area (PM) and the cerebellum.

In this paper, we briefly describe the parts of the model important for the specific implementation. The reader can refer to [9] for a detailed description.

2.1 *Data process*

The visual module performs four levels of processing of the data: 1) transformation of frame of reference, from extrinsic to intrinsic, 2) filtering of small and noisy motions, 3) segmentation of the motion, based on changes in velocity and movement direction, and 4) parameterization of the movements in terms of speed and direction. In the experiments reported in this paper, see Section 3, data are recorded in joint angles, and, thus, step 1 is not performed.

Filtering of small motions
Input data to this module are all recorded joint angular trajectories of arm movement. We apply a threshold function (see Equation 1) to the position trajectory of each degree of freedom (DOF) to eliminate small movements, due to the interaction torques across the body. These small movements are noise to us, as we wish to recognize and model only the key features of voluntary movements.
Let $D_i(t)$ be the angular displacement of joint i at time t. D_i^M, D_i^m and $\tilde{D}_i$ are the maximal, minimal and mean values of D_i over the whole trial. The high-pass filter eliminates from further processing into segmentation (see below) all joint trajectories D_j which are such that

$$D_j^M - D_j^m < \frac{pi}{16} \qquad (1)$$

Segmentation
Data are segmented to detect the starts and stops of voluntary motion and changes in direction of movement, following Equations 2, 3 and 4. The segmentation process depends on a set of 3 parameters per DOF. These are the minimum displacement θ_j^m (in joint angle) for detecting a motion, the minimum time window T_0 during which no displacement greater than θ_j^M has been observed. Since different degrees of freedom have different dynamic properties, due to their different lengths and muscular composition, we applied different segmentation parameters to each. These properties are calculated as follows.

$$\theta_j^m = 8 \cdot \frac{pi}{D_j^M - D_j^m} \qquad (2)$$

$$\theta_j^M = 2 \cdot \theta_j^m \qquad (3)$$

$$T_0 = 2 \cdot T_j \text{ with T such that } D_i(T) = \tilde{D}_i \qquad (4)$$

Parameterization
After segmentation, the speed and direction of movement of each joint is coded in the output of the TC neurons to the PM neurons There are two neurons per degree of freedom (DOF) per joint, coding for positive and negative direction of movement, respectively. Let $y_i^{TC}(t)$ be the output of TC neuron i at time t, and $t_1, \ldots, t_n$ the series of time steps at which the segmentation has occurred, then

$$y_i^{TC}(t_j) = D_i(t_k) - D_i(t_{k-1}) \qquad (5)$$

and k is such that $t_j = t_k$ and $j > 1$.
Figure 2 shows segmentation of the trajectory of the shoulder joints (flexor/extensor, abduction/adduction) of the left arm during drawing of a figure eight. Figure 2 shows the speed input coded by the neurons after segmentation.

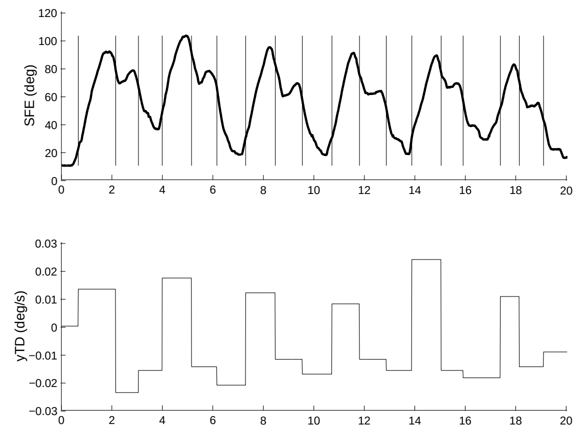

Figure 2: Top: Trajectory of shoulder joint (SFE=flexion/extension) during drawing of a figure eight. Vertical bars show the time steps of segmentation. (Bottom) Output of the TC neurons following segmentation. The output is proportional to the value of the speed of the motion at the time of the segmentation.

2.2 *Motor control*

In our model, motor control is hierarchical and is composed of (from lowest level to highest level) *muscle model*, *spinal cord*, motor cortex (M1), premotor cortex (PM) and cerebellum modules.

Spinal cord module
The module receives excitatory input from the M1 neurons (defining the amplitude and speed of the movement for each DOF) and outputs to the muscle model which then computes joint torques sent to the dynamic simulation. It is composed of fixed neural circuits, one for each DOF for each joint, which produce an oscillation of that joint, following [14].
The neurons of the spinal cord module are modeled as leaky-integrators, which compute the average firing frequency [15]. According to this model, the mean membrane potential m_i of a neuron N_i receiving input from M1 nodes x_j is governed by the equation

$$\tau_i \cdot dm_i/dt = -m_i + \sum w_{i,j} x_j \qquad (6)$$

where $x_j = (1 + e^{(m_j+b_j)})^{-1}$ represents the neuron's short-term average firing frequency, b_j is the neuron's bias, τ_i is a time constant associated with the passive properties of the neuron's membrane, and $w_{i,j}$ is the synaptic weight of a connection from neuron N_j to neuron N_i. In the model, a neuron in the spinal cord receives input from all other neurons and from neurons in M1 module.

Muscle torques Each DOF of each joint is controlled by a pair of muscles which receive input from corresponding pair of motor neurons in the spinal cord module. A muscle is simulated as a combination of a spring and a damper [16]. The torque exerted on each joint is determined by a pair of opposed flexor and extensor muscles. These muscles can be contracted by input signals from motor neurons, which increase their spring constant, and therefore reduce their resting length. The torque acting at a particular joint is therefore determined by the motoneuron activities (M_f and M_e) of the opposed flexor and extensor muscles:

$$T = \alpha(M_f - M_e) + \beta(M_f + M_e + \gamma)\Delta\varphi + \delta\Delta\dot{\varphi} \quad (7)$$

where $\Delta\varphi$ is the difference between the actual angle of the joint and the angle at rest (zero in our experiments). The coefficients α, β, γ, and δ determine, respectively, the gain, the stiffness gain, the tonic stiffness, and the damping coefficient of the muscles.

M1 and PM modules

M1 module contains a *motor map* of the body. It is composed of 3-neuron networks for each DOF of each joint for independently regulating the amplitude (two nodes) and the frequency (one node) of the oscillation of the corresponding joint DOF, similar to [14].

The PM module creates a direct mapping between the parameterization of the observed movement in TC, following visual segmentation, and that used for motor control in M1. PM nodes receive sensory feedback, in terms of joint angle position, from the spinal cord module. $y_j^S(t)$ is the angle of degree of freedom j of limb l. The term $\delta(y_{l_j}^S, y_{l_k}^{TC} \cdot (\pi/8))$ in equation 2.2 compares the actual deviation of joint j in the avatar and the deviation of the human arm, given by $y_{l_k}^{TC}$ for corresponding direction of motion k of joint j [1]. This term modulates the amplitude or speed of the movement, by increasing or decreasing (for smaller or larger speed) the output of the PM and consecutively of the M1 nodes (and finally the torque sent to the motor neurons).

The network TC-PM-M1 consists of a Dynamical Recurrent Associative Memory Architecture (DRAMA) [17]. Similarly to time delay networks [18], each connection is associated with two parameters, a weight w_{ij} and a time parameter τ_{ij}. Weights correspond to the synaptic strength, while the time parameter specifies a synaptic delay, that is a delay on the time required to propagate the activity from one neuron to the other. It is a fully connected network (all nodes in PM are connected to all nodes in TC and to all nodes in M1, nodes in TC and M1 are not interconnected) with asymmetrical connections ($w_{ij} = w_{ji}$) and self-connections on each unit.

If $(\tilde{w}, \tilde{\tau})$ and (w, tau) are the weights of the connections between M1 and PM, and between TC and PM modules respectively, if y_k^{TC} is the output of TC neuron for direction of movement k of limb l, then y_j^{M1}, the output of M1 neuron of degree of freedom j of limb l is:

$$dy_{l_j}^{M1}/dt = -\tilde{\tau}_{l_j l_j}^{M1} + \sum_i \tilde{w}_{il_j} \cdot y_i^{PM} \cdot \delta(\tilde{\tau}_{il_j}, y_i^{PM}) \cdot \delta(y_{l_j}^S, y_{l_k}^{TC} \cdot \frac{\pi}{8}) \quad (8)$$

and y_j^{PM}, the output of neuron i in module PM

$$dy_{l_j}^{PM}/dt = -\tau_{l_j l_j}^{PM} + \sum_i w_{il_k} \cdot x_i^{TC} \cdot \delta(\tau il_k, y_i^{TC}) \cdot \delta(y_{l_j}^S, y_{l_k}^{TC} \cdot \frac{\pi}{8}) \quad (9)$$

The function $\delta(x, H)$ is a threshold function that outputs 1 when $x \geq H$ and 0 otherwise.

The sets of weights $\tilde{w}$ and w define the mapping between the visual representation and the torques sent to the muscles. The relationship $\tilde{w} = f(w)$ is a first order approximation of the inverse dynamics calculation. In previous experiments, all connection parameters (w) and temporal (τ) were modulated by learning, see [9]. In the experiments reported here, this mapping is fixed and follows a first order relationship, such that for activation of degree of freedom j of limb l, represented by activity in neuron j in M1, neuron i in PM and neuron k in TC, we have

$$\tilde{w}_{ji}(t) = I_{l_j}(t) \cdot w_{il_k} \quad (10)$$

where $I_{l_j}(t)$ is the moment of inertia of the limb l using DOF j. The moment of inertia varies depending on the position of the limb attached to this limb. Here we consider only the shoulder/elbow complex. In a first approximation, we estimate

$$I_{l_k}(t) = I_{l_j}(t_0),\ l = 0,1 \quad (11)$$

$$I_{l_k}(t) = I_{l_j}(t_0) \cdot (D(t) - D(t_0)), \quad l = 2 \quad (12)$$

t_0 is the time at which the arm is in the relaxed position (aligned along the body). $l = 0,1,2$ are the 3 DOFs (SFE, SAA, SHR) of the shoulder and l is the DOF of the elbow.

[1] The term $y_{l_k}^{TC}$ is a measure of the speed at the beginning of the movement. We map it into a measure of position by multiplication with the factor $(\pi/8)$.

Cerebellum and Decision modules

Learning of the complete movement is done in the cerebellum module. This module is a DRAMA architecture. It receives input from the PM module and learns the time series of activity of neurons in PM, which represent the sequence of motions after segmentation. Learning consists of updating the parameters of the connection between cerebellum and PM modules (w and τ) following Hebbian rules, given by Equations 13 and 14.

$$\delta w_{ji}(t) = a \cdot y_i(t) \cdot y_j(t) \tag{13}$$

$$\tau_{ji}(t) = \left(\frac{\tau_{ji}(t-1) \cdot \frac{w_{ji}}{a} + \frac{y_j(t)}{y_i(t)}}{\frac{w_{ji}}{a} + 1}\right) \cdot y_i(t) \cdot y_j(t) \tag{14}$$

where a is a constant factor by which the weights are incremented.

Finally, the decision module controls the transition between observing and reproducing the motor sequences, i.e., it inhibits PM neural activity due to TC (visual) input to flow downwards to M1 (for motor activation). It is implemented as a set of if-then rules and has no direct biological inspiration.

2.3 3-D biomechanical simulation of a humanoid

The model is implemented in a dynamic simulation of a 41 degrees of freedom (DOF) avatar [19] (see Figure 3 right). Shoulders, hips, wrists, ankles and head have 3 DOFs. Elbows and knee have one. The trunk is made of three segments with 2 DOFs each. All limbs are attached by hinge joints. The external force applied to each joint is gravity. Balance is handled by supporting the hips; ground contact is not modeled. There is no collision avoidance module. The dynamics model is derived from the Newton-Euler formulation of Rigid Body Dynamics.

3 Experiments

Data for these experiments are recordings of human arm motion, gathered using a Sarcos SenSuit (see Figure 3 left). The complete SenSuit is worn like an exoskeleton which, for most movements, does not restrict the motion while an array of lightweight Hall-effect sensors reliably records the positions of 35 degrees of freedom (DOF) of the human body. In the experiments, we used only recordings of the 3 degrees of freedom (DOF) of the shoulder joint (flexion, abduction and humeral rotation) and of the elbow joint. Data are captured at a frame rate of 100Hz.

Because of space constraints, we present here the model's implementation on one set of data. These are recordings of left arm movement for five repetitions of drawing a figure eight. Data were segmented following the algorithm of Section 2.1, as shown in Figure 2. The model was presented with the whole trajectory once

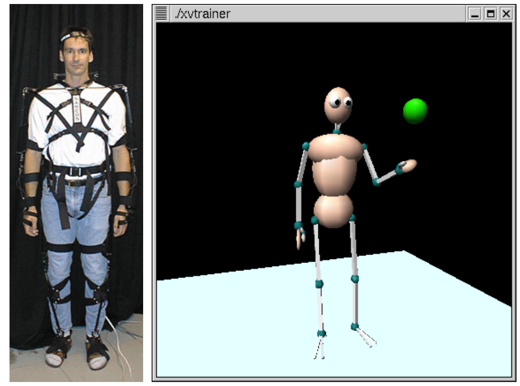

Figure 3: (Left) Subject wearing the Sarcos sensuit recording system. (Right) Simulation of a 41 degrees of freedom humanoid.

(composed of 6 repetitions of the complete movement). The amplitude and speed of the movement varies over the repetitions. The model was let reproduce the trajectory both during training and after learning. Figure 5 shows superimposed human subject and avatar trajectories of shoulder (SFE=flexor/extensor, SAA=abduction/adduction) and elbow (EB) joints during training. Figure 4 shows the values of the torques sent to the same 3 DOFs of the avatar during the training phase. In this example, only these 3 DOFs received active control for reproducing a desired trajectory (after segmentation). The remaining joints of the arm and of the rest of the body were kept immobile, receiving torques to cancel the internal perturbations due to the motions of other joints and (i.e. $y^{TC} = 0$ for these joints).

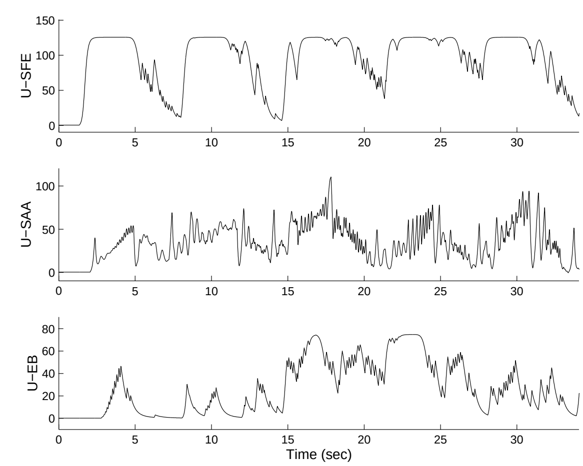

Figure 4: Torque sent to the shoulder joints, (SFE=flexor/extensor), (SAA= abduction/adduction) and elbow joint (EB) during the reproduction of figure eight with the left arm.

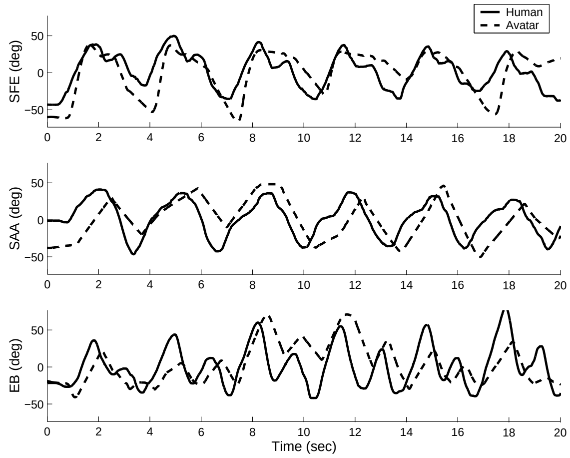

Figure 5: Trajectory of the shoulder joints, (SFE=flexor/extensor), (SAA= abduction/adduction) and elbow joint (EB) during the reproduction of figure eight with the left arm.

Results show that the system follows closely the demonstrated trajectory during training, mapping the changes in amplitude and speed. The model manages to extract on-line the main features of the movement, while dismissing most of the (from our point of view) unimportant fine structure of the movement and noise (from the recording system). Figure 6 shows the increase of neural firing during training. After two presentations of the movement, the network has generalized. That is, it has learned to recognize the four main features of the movement and to represent those with a specific neural firing pattern: node1-node2-node3-node2. Note that the nodes code for the combined movement of the 3 DOFs (SFE/SAA/EB) and not for each joint separately. For clarity of the figure, we show only the SFE joint superimposed to the neural firing rate.

The same pattern of node firing is activated during the 4 remaining presentations of the movement, showing that the network recognizes correctly the movement. During training on the 4 last movements, the network adjusts the parameters (the τ_{ij} from equation 14) of these nodes' interconnections to better represent the precedence and time lag between each sub-movement. After training, the model produce a movement which presents the main sub-features of the training movement (see Figures 7 and 8). Amplitude and speed of this reproduced movement represents the weighted combination of that of the six demonstrations.

Robustness in the face of perturbation is an important criteria for a robotics system. We tested the model's robustness after training, by applying an external force during rehearsal of the trajectory. Figures 7 and 8 show the trajectory of SFE and EB DOFs when applying a vertical force on the wrist of 200 and 1200 Newton respectively. Results show that the system recovers quickly to both constraints effect. This is due to the PID control term ($\delta(y^S_{l_j}, y^{TC}_{l_k} \cdot (\pi/8)))$ of equation 2.2. The desired trajectory ($y^{TC}_{l_k}$) is obtained by rehearsal of the activity of neurons in the TC module (feeding back the activity from PM neurons).

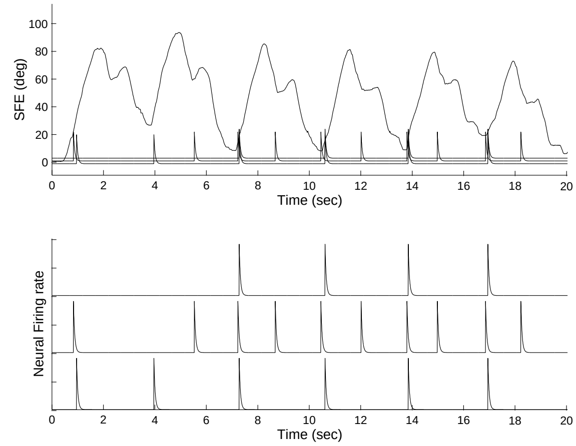

Figure 6: Firing rate of the neurons in the cerebellum module during training.

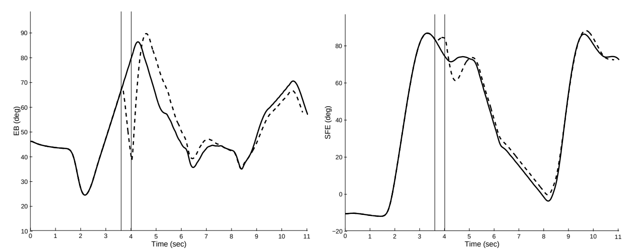

Figure 7: Trajectory of the shoulder joints (SFE=flexor/extensor) and elbow joint (EB) during rehearsal after training. Solid line: desired trajectory, dashed line: trajectory after perturbation (vertical force of 20Kg) lasting 0.2 seconds.

4 Conclusion

This paper presented a biologically inspired model of human imitation. The model was implemented in a dynamic simulation of a humanoid avatar with 41 degrees of freedom for reproducing 3D human arm motion. It was shown that the model could learn the principal features of a 3 DOFs arm trajectory, by generalizing across the different demonstrations. Learning is fast and done on-line (here on a pentium III, 700 MHz station). In [17], we showed that the DRAMA architecture allowed on-line learning and reproduction of movements in robots with very limited computational power (124M

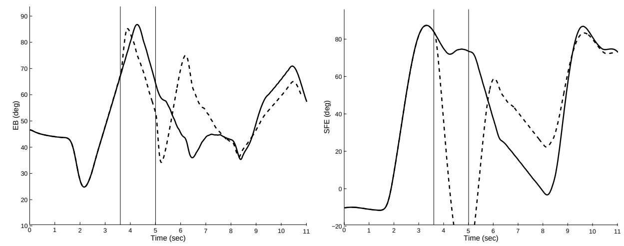

Figure 8 Trajectory of the shoulder joints, (SFE=flexor/extensor) and elbow joint (EB) during rehearsal after training. Solid line: desired trajectory, dashed line: trajectory after perturbation (vertical force of 120Kg) lasting 0.7 seconds.

RAM, 30MHz). Further, it was shown that reconstruction of the movement was robust against perturbation. Generalization, on-line learning and robustness against perturbation are desirable for a robot controller. However, a number of questions remain to be answered in order to determine the usefulness of our model to control a robot:

1) Scaling up to full body motion: Although we have shown that the model could well reproduce motion of a 65 degrees of freedom humanoid when given simulated (not noisy) data [9], it remains to show whether the performance of the model would degrade when controlling actively (as opposed to "passive" response to internal torques) more joints than the 3 used here and for a more general set of data.

2) Range of validity of the model: The experiments showed that a qualitatively good reproduction of the features of the movement can be obtained by giving only the following information to the system: 1) the starts and stops of the movement (and it is sufficient to determine those within a precision of 10% of the duration of the movement), 2) the initial speed at each start point of the movement and 3) a first order approximation of the inverse dynamics. It remains now to determine the range of motions within which the above set of minimal constraints holds. The present experiments showed only that this is sufficient for 3D slow motions of the arm, when the torso is in upright position and when only 3 joints are active, the other joints being kept immobile.

Acknowledgments

This work was made possible by Awards #97 10312 and # 0082995 of the U S National Science Foundation, the ERATO Kawato Dynamic Brain Project funded by Japanese Science and Technology Cooperation, and the ATR Human Information Processing Research Laboratories. Aude Billard was also supported in part by a fellowship from the Swiss National Science Foundation.

References

[1] K. Sing Bing Kang Ikeuchi, " A robot system that observes and replicates grasping tasks, " in *Proceedings., Fifth International Conference on Computer Vision*, 1995.

[2] M.I. Kuniyoshi and Inoue, " Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation, vol.10, no.6*, pp799– 822, 1994.

[3] H. Asada, H.; Izumi, " Automatic program generation from teaching data for the hybrid control of robots," in *IEEE Transactions on Robotics and Automation*, 1989, vol.5 2, pp. 166 –173.

[4] N. Delson and H. West, " Robot programming by human demonstration: adaptation and inconsistency in constrained motion," in *IEEE International Conference on Robotics and Automation*, 1996, vol. 1, pp. 30 –36.

[5] M. Skubic and R.A. Volz, "Acquiring robust, force-based assembly skills from human demonstration," in *IEEE Transactions on Robotics and Automation* 2000, vol. 16:6, pp.772 –781.

[6] I. Kamon, T. Flash, and S. Edelman, " Learning visually guided grasping: a test case in sensorimotor learning," in *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 1998, vol. 28:3.

[7] M Tomasello, Cultural transmission in the tool use and communicatory signaling of chimpanzees, " *Language and intelligence in monkeys ans apes: comparative developmental perspectives*, pp.274– 311, 1990.

[8] L. Berthouze, P. Bakker, and Y Kuniyoshi, " Learning of oculomotor control: a prelude to robotic imitation," in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96*, 1996, pp. 376 –381.

[9] A. Billard, " Learning motor skills by imitation: a biologically inspired robotic model," *Cybernetics & Systems Journal, special issue on Imitation in animals and artifacts*, vol. 32:1-2, pp. 155–193, 2001.

[10] J. Demiris, *Movement imitation mechanisms in robots and humans*, Ph. D. thesis, Dept. of Artificial Intelligence, University of Edinburgh, May 1999.

[11] S. Schaal, " Is imitation learning the route to humanoid robots? " *Trends in Cognitive Sciences*, vol. 3(6), pp.233–242, 1999.

[12] S. Schaal, " Learning from demonstration," *Advances in Neural Information Processing Systems*, vol. 9, pp. 1040–1046, 1997.

[13] M. Arbib, A. Billard, M. Iacoboni, and E. Oztop, " Mirror neurons, imitation and (synthetic) brain imaging," in *Neural Networks (to appear)*, 2000.

[14] A.J. Ijspeert, J Hallam, and D Willshaw, " Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology," *Adaptive Behavior*, vol.7, no2, pp. 151–172, 1999.

[15] J.J Hopfield, " Neurons with graded response properties have collective computational properties like those of two-state neurons," in *Proceedings of the National Academy of Sciences* vol. 81, pp. 3088–3092. Washington : The Academy, 1984.

[16] F Lacquaniti and J F Soechting, " Simulation studies on the control of posture and movement in a multi-jointed limb," *Biological Cybernetics*, vol.54, pp. 367–378, 1986.

[17] A. Billard and G Hayes, "Drama, a connectionist architecture for control and learning in autonomous robots," *Adaptive Behavior, Vol. 7:1*, pp. 35–64, 1999.

[18] D. T. Lin, P A. Ligomenides, and J E. Dayhoff, " Learning spatio-temporal topology using an adaptive time-delay neural network, " in *Proceedings of World congress on neural networks, Portland, OR*, 1993, vol. 1, pp.291–294

[19] S. Schaal, "The sl simulation and real-time control software package, " Tech. Rep. Computer Science Tech Report, http://www-slab.usc.edu/publications/schaal-TRSL , University of Southern California, 2001.