Imitation of Human Demonstration Using A Biologically Inspired Modular Optimal Control Scheme

GAVIN SIMMONS and YIANNIS DEMIRIS
Department of Electrical and Electronic Engineering
Imperial College London
Exhibition Road, London, SW7 2AZ
{gavin.simmons, y.demiris}@imperial.ac.uk

Progress in the field of humanoid robotics and the need to find simpler ways to program such robots has prompted research into computational models for robotic learning from human demonstration. To further investigate biologically inspired human-like robotic movement and imitation, we have constructed a framework based on three key features of human movement and planning: optimality, modularity and learning. In this paper we describe a computational motor system, based on the minimum variance model of human movement, that uses optimality principles to produce human-like movement in a robot arm. Within this motor system different movements are represented in a modular structure. When the system observes a demonstrated movement, the motor system uses these modules to produce motor commands which are used to update an internal state representation. This is used so that the system can recognise known movements and move the robot arm accordingly, or extract key features from the demonstrated movement and use them to learn a new module. The active involvement of the motor system in the recognition and learning of observed movements has its theoretical basis in the direct matching hypothesis and the use of a model for human-like movement allows the system to learn from human demonstration.

1. Introduction

Our goal is to create robots that are both flexible and adaptable, and which can interact with humans in an environment that does not require any special considerations for the robot. Humans working with robots in a natural way will require a more natural means to programme them than through a computer terminal.[1] Robots should be able to learn from human demonstration and from each other,[2] widening their adaptability and reducing complex programming. Since we want robots to perform actions more like humans, our starting point is to look at how humans move, interact and learn from each other. This biologically inspired approach to robotics allows us to draw upon studies from neuroscience and neurophysiology to help design the function of the robotic system.[3]

Recent work on mirror neurons (neurons that fire both when an action is observed and performed) in humans[4,5] and growing evidence for a simulation theory of mind[6,7] point to a direct link between visual perception and the motor system in humans. This link has been used as the basic idea behind several successful robotic imitation systems.[2,8] To allow greater correspondence between demonstrated and imitated movements, we have looked at producing a motor system for a robot arm that acts along the principles of the human motor system. To this end, we have constructed a framework based on three key features of human movement and planning: optimality, modularity and learning. In this framework a well-established optimal control scheme is adapted to implement a biologically-plausible model of human movement. The movement planning part of the control scheme lends itself well to a modular implementation. When a movement is observed, each module produces motor commands to update the internal state representation, allowing the system to simulate the demonstration. If the goal of the movement is achieved, the system can then use the selected motor commands to move the robot arm, or extract features of the demonstrated movement to use in creating a new module.

We begin by exploring what is meant by "human-like" movement, and outline relevant computational theories of human movement production and planning. We then briefly discuss the suggested link between perception and action, and introduce a computational model for imitation based on this idea that incorporates the model for human-like movement. We then describe the implementation of the optimal control scheme and the imitation system, and present results both in simulation and for a robot arm.

2. Models of Human Movement

2.1. *Features of Human Movement*

A wide range of studies have shown that human upper limb movements have certain invariant features.[9,10,11,12] Some of these features are a result of the physical motor system; the spring-like nature of muscles or the interplay between agonistic and antagonistic muscles. Of more relevance to a robotic system that does not have these physical attributes are other features that emerge as a product of the underlying motor representation and the neural commands sent to the muscles. Any computational model of the motor system should aim to capture and explain these features in a biologically realistic manner.

One of the main features is that of smoothness of movement. Despite the almost infinite number of possible ways in which the hand can be moved to a target, in fast visually guided point-to-point reaching movements the path of the hand is roughly straight, while the motions of the joints are complex. The velocity of the hand follows a bell-shaped curve with a single maximum, resulting in smooth acceleration. Although there is variation in the exact hand path, the common feature of smooth movement with the hand moving in a straight line between points holds both between individuals and between trials.

Another key feature is the trade-off between speed and accuracy known as Fitts' Law - the faster a movement occurs, the less accurate the final hand position, and vice versa.[9]

2.2. *Optimal Control*

For even the simplest tasks, such as point-to-point reaching movements, there are many possible paths that the hand could move along and many possible velocity profiles for it to follow. Even if both the path and the velocity profile have been determined, the inverse kinematics problem means that there are many combinations of joints angles that can achieve the required trajectory. However, not all trajectories or sets of joint angles are equal. Motor planning can be thought of as the process of selecting one solution from these many possibilities that is consistent with the goal of the task.

This can be formulated as an optimal control problem, where the trajectory is produced by attaching a cost to some characteristic of the movement, such as the hand velocity or energy expended in the muscles, and then selecting control signals to move the arm in such a way as to minimise some function of this cost. To which aspects of the movement we assign a cost determines which trajectory is followed. Our goal then becomes to select the aspect of the movement on which to apply the cost function so that we capture the required features of human movement.

Cost functions, or optimisation criteria, for modelling limb movements and hand trajectories fall into one of two categories - kinematic or dynamic solutions. In kinematic solutions the cost function is based upon the geometric or time-based properties of the motion, and the state of the limb could be represented, for example, in terms of joint angles or the Cartesian position of the hand. In dynamic solutions the cost function is based on the dynamics of the arm and the state could be represented, for example, in terms of joint torques or forces acting on the hand. Examples of cost functions from each category are given in the next section.

2.3. *Minimum Jerk and Minimum Torque*

An example of a kinematic solution to the optimal control problem for human movement is the minimum jerk model. It has been suggested that smoothness of movement could be an explicit goal of the system.[11] The measure of smoothness chosen by Flash and Hogan was the "jerk" of a movement, defined as the square of the derivative of the Cartesian hand acceleration. Their results show a good match between the predicted and actual trajectories. This suggests that there is a kinematic representation of movement at higher levels of the motor system. Some features of the measured movements however, such as a slight asymmetry in the velocity profile, were not captured by the model.

In contrast to the purely kinematic minimum jerk model, a model was proposed that used a dynamic objective function.[12] Uno et al. argued that while the minimum jerk model successfully captured the behaviour of reaching movements, it is unlikely that movements are determined independently of dynamic quantities of the arm such as length, load, torque or external force. The authors defined an objective function that minimised the change in the torque at the joints. The hand paths predicted by the model were in accordance with those predicted by the minimum jerk model in an area close to and in front of the body. Significant differences were predicted however in areas further out and to the side of the body, where the dynamics of the arm differ from those in front of the body. Experimental results were obtained for human arm trajectories of the movements where the two models differed in their predictions. These results showed clearly that the shape of the hand path was dependent on the area of the workspace where the movement was performed, as predicted by the minimum torque-change model.

2.4. *Minimum Variance - Signal-dependent noise*

Despite producing human-like movements, both the minimum jerk and minimum torque-change models have a number of problems. The criteria used in those models do not include an accuracy constraint for the movement. In common with many other optimisation criteria used to explain human movements, they also lack a principled explanation as to why the motor system evolved to minimise these particular aspects of movement, other than that they both predict the general shape of human reaching movements.

Harris and Wolpert[13] looked at the constraints on movement that hindered the achievement of a target goal. They suggested that the constraint with the biggest impact would be the effect of neural noise on the accuracy of the movement. Biological noise is signal-dependent, meaning that the standard deviation of the noise is linearly proportional to the absolute value of the signal on which it occurs. It is present in all neural systems and would cause deviations in the path of the arm from the desired trajectory for a point-to-point reaching movement. The deviations will build-up over the course of the movement, leading to variability in the final position of the hand.

In the presence of signal-dependent noise, moving rapidly requires large control signals, which means that the amplitude of the noise is also large, decreasing the accuracy of the movement. Since inaccuracy requires further corrective movements to accomplish a task, moving as rapidly as possible is not the optimal solution for the motor control system. Lowering the amplitude of control signals decreases the amplitude of the noise, and hence increases accuracy, but this results in slower movements. The assumption of signal-dependent noise on the control signal therefore neatly explains the trade-off between movement speed and accuracy, as observed in actual human reaching movements.[9] On this basis, it was proposed that neural commands are selected to minimise the variance in the final hand position for a specified movement duration to achieve some task goal.[13]

As well as explaining the observed speed-accuracy trade-off, the minimum variance model also successfully predicts the smooth movement characteristics of human reaching movements. Smoothness of movement is not an explicit goal of the minimum variance objective function as it was with the models described above, but despite this the model predicts the characteristic straight hand paths and bell-shaped velocity profiles of human movement. This indicates that smooth movement is not an explicit goal of the human motor system, but is instead an *emergent* property of the movement when the motor system is subject to signal-dependent noise. Observing the variance of the hand is also much easier than measuring the jerk or the torque change.[14] Its success at explaining so many features of human movement makes the minimum variance model the most biologically realistic of the optimisation criteria that have been suggested as being represented in the motor system.

3. Imitation

The term *imitation* is used generally throughout the literature to refer to similar but related processes which have subtle distinctions.[15] When *learning by imitation*, a system is given the ability to engage in imitative behaviour which then serves as a mechanism for reinforcing further learning and understanding. *Learning to imitate* requires that the system learns how to solve the correspondence problem (see below) through experience. The focus of this work however is on *learning by demonstration*, which would seem at first to be identical to learning by imitation. However learning from a demonstration does not have to involve imitative behaviour. Where it does not, the system performs *task-level imitation*, and learns how to perform a task or achieve some target goal without imitating the exact behaviours of the demonstrator.

One of the major problems for researchers in imitation is the correspondence problem. How do humans map from the perceived state of the demonstrator to their own body state? How do we know which parts of our bodies to move to correctly imitate the demonstrator? This problem is even more acute for robotic imitation, where the morphology of the robot is likely to be different from that of a human. The robot is also likely to have less flexibility and fewer degrees of freedom than the human demonstrator. This leads on to the problem of determining a measure of the similarity between the demonstrator and the imitator.[16,1] For task-level imitation, this measure requires the specification of some goal, such as moving the hand to a particular target at a particular time.

3.1. *Simulation Theory and the Direct Matching Hypothesis*

One theory that seeks to explain the correspondence problem is *simulation theory*, which effectively suggests that humans put themselves in the place of the demonstrator and "simulate" the motor commands they would use to perform the observed movements by deploying the same cognitive mechanisms. This simulation is usually thought of as requiring the ability to give pretend inputs to our motor system as well as the ability to take the system offline, so that the movements are not actually performed while they are being imagined.[7] This is in contrast to *theory theory*, which suggests that our behaviour can be explained by the use of inferential and deductive processes that do not involve simulation,[17] the principle idea being that humans acquire a set of explanatory theories that relate external stimuli to the unobservable internal states of the demonstrator. They can then use

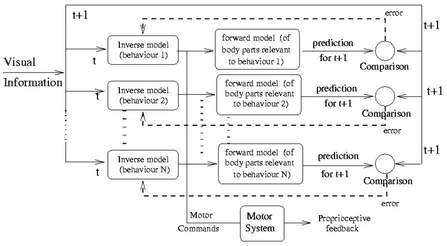
t+1
error
t+1
Visual
Information
t
Inverse model (behaviour 1)
forward model (of body parts relevant to behaviour 1)
prediction
for t+1
Comparison
t
Inverse model (behaviour 2)
forward model (of body parts relevant to behaviour 2)
prediction
for t+1
Comparison
t+1
error
t+1
t
Inverse model (behaviour N)
forward model (of body parts relevant to behaviour N)
prediction
for t+1
Comparison
error
Motor
Commands
Motor System
Proprioceptive feedback

Figure 1. Architecture for an active imitation system

these theories to infer these internal states and try to match their own state to that of the demonstrator.

The ideas behind simulation theory are also expressed in the *direct matching hypothesis*, which proposes that action understanding results from a mechanism that maps an observed action onto a motor representation of that action.[18] This is closely related to the *motor theory of perception*, which suggests that the motor system is actively used in the perception of motion and the recognition of actions. The mirror neuron system found in monkeys and posited to exist in humans as well[4,19] is such a mechanism. An experiment[18] to monitor a subject's gaze both when performing and observing a task showed that gaze is predictive in both situations, implying that during action observation humans carry out motor programs equivalent to those used in the action. From a computational viewpoint, the required mapping between perception and appropriate action is a very difficult process, as visual perception takes place in a different coordinate frame from motor control.

3.1.1. *Mirror Neurons*

The mirror neuron system is a neural circuit in the F5 area of monkey premotor cortex that is active both when the monkey observes another monkey or a human grasping or manipulating objects, and when the monkey performs the same manipulation. Evidence is accumulating that a similar system exists in humans.[5] This system has been put forward as the link between visual and motor representations that is required in order for imitation and learning from observation to take place.

3.2. *A Computational Model for Imitation*

A model of imitation based on the link perception and action has been described by Demiris and Hayes.[2] In their model, a distinction is made between passive and active imitation. In the passive system, information from the vision system is given to a posture estimation module which estimates the current postural state of the demonstrator. The postures that define the movement sequence being imitated are stored and fed into a movement matching module which produces the motor commands needed to match these postures with equivalent postures by the imitator.

This system has also been implemented for a robot head.[20] The passive system follows a one-way perceive-recognise-act sequence[21] and requires very little information. However, it makes no distinction between known and unknown movements, and the motor system is clearly separate from the perception system. In the active system, shown in Fig. 1, a forward model is paired with a behaviour (equivalent to an inverse model or motor primitive) and used to generate a prediction of that behaviour that is compared with the next state of the demonstrator. This structure uses prediction feedback and can therefore match a visually perceived demonstration behaviour with an equivalent one from the imitator's set of known behaviours. The prediction is compared to the demonstrator's actual state and used to produce an error signal that determines how closely a known behaviour matches the perceived behaviour. In this way, known and unknown behaviours can be distinguished and the motor system is directly involved in the perception of the demonstrated action.

In the next section, we outline how this imitation model is combined with an implementation of the minimum variance model and adapted to allow recognition and learning of human demonstrated trajectories by a robot arm.

4. Implementation

4.1. *Minimum Variance Model*

There have been several computational models that have been used to investigate the minimum variance constraint, each with a different goal. In their original paper Harris and Wolpert generated trajectories for comparison with experimental data, formulating the signal-dependent noise solution as a quadratic programming problem.[13] Todorov and Jordan explored optimisation principles for motor control, using a Kalman filter framework to produce an optimal feedback control law.[22] Miyamoto et al. used a number of processes to train a recurrent neural network to produce the desired accelerations.[14] These were then used to drive a dynamics model of the human arm. Feng et al. carried out a detailed mathematical treatment of the minimum variance model.[23]

While each approach is perfectly valid and produces excellent simulation results, they were either computationally intensive, required long training periods to determine relevant parameters, or were in some other way unsuitable for the direct control of a robot arm. Instead, the ideas presented by Harris and Wolpert, Todorov and Jordan, and Miyamoto et al. were used to produce an optimal control method that was computationally straightforward and suitable for implementation on a simple servomotor robot arm.

Since the human arm has seven degrees-of-freedom (DOF) and is highly kinematically redundant, a complete bio-mechanical model of the arm would require a very complicated controller. Instead a simpler model was used - a two-link, two-joint arm restricted to move in a horizontal plane, as shown in Fig. 2. Both joints were revolute, with the first joint representing the shoulder placed at the origin

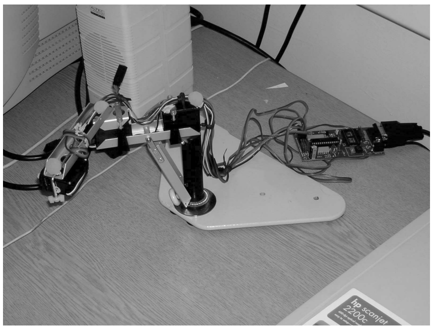

Figure 2. The two-link planar robot arm used in these experiments

of the coordinate system. The second joint, representing the elbow, connected the two links. The forward and inverse kinematics for such a robot arm were calculated using a standard method.[24] Many studies have used a manipulandum to restrict a person to move their arm in a plane, providing data for comparison between the movement of the model and actual human movement.[12]

4.1.1. *State Representation and System Dynamics*

The *discrete-time linear quadratic regulator* (DLQR) optimal control system was adapted to perform signal-dependent noise based control. Initially, the state feedback was assumed to be noiseless. The robot arm was modelled as a discrete time system with noise added to the control term:

$$s_{t+1} = As_t + B\left(u_t + n_t\right) \tag{1}$$

where s_t was the state of the system and u_t was the neural control signal. A and B were fixed matrices describing the dynamics of the system. The term n_t represented Gaussian white noise with zero mean and variance $k\, u_t^2$, which increased with the magnitude of the control signal.

Since the theory is expressed in terms of moving the hand to a desired position, in a desired time and with a desired accuracy, the state representation was chosen to be the position of the end effector of the robot arm and its velocity, both measured at time t.

$$s_t = \left[x_t, \dot{x}_t, var_t, 1\right]' \tag{2}$$

where x_t was the position of the end effector on a single axis, $\dot{x}_t$ was the velocity along that axis and var_t was the positional variance.[13] The fourth term, "1", was used in the optimal control function to encode the positional error. The discrete-time dynamics of the system were

$$x_{t+1} = x_t + \dot{x}_t \Delta t \tag{3}$$

$$\dot{x}_{t+1} = \dot{x}_t + u_t \Delta t \tag{4}$$

These dynamics were put into the matrix form of Eq. (1), setting A and B as

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ \Delta t \\ 0 \\ 0 \end{bmatrix}$$

The positional variance was taken from the first element of the diagonal of the covariance of the state distribution, calculated separately. The state representation and the discrete-time dynamics were adapted from the supplementary notes to Todorov and Jordan, which included a second-order model of muscle forces acting on the hand. These were removed since the robot arm was constructed using servomotors driven by a positional signal rather than torque, and controlling the arm to generate the required trajectories could be achieved without taking the extra forces into account. By extending the state representation in a suitable way, the algorithm would be equally applicable to an arm controlled by torque motors or artificial muscles.

4.1.2. *Cost Function*

The DLQR defines a cost function J, in matrix form, which has been adapted for the minimum variance model:

$$J = \sum_{t=0}^{T-1} (s_t' Q s_t + u_t' R u_t + s_T' Q_T s_T + s_N' Q_N s_N) \tag{5}$$

where Q is the state cost measuring state deviation, Q_T is the final state cost at the end of the movement, Q_N is the state cost during a post-movement period of length N, and R is the input cost measuring input size. The DLQR problem is to find the set of control signals $\left\{u_0^{lqr}, \ldots, u_{T-1}^{lqr}\right\}$ that minimises this cost function. The values of the four matrices Q, R Q_T and Q_N were chosen in accordance with the minimum variance model. Since the aim of the model is to minimise the variance in the hand position during the post-movement period, no constraints were placed on the state from the start of movement to the start of the post-movement period. This meant that the state cost Q was zero for the duration of the movement. The post-movement cost Q_N is defined from the end of the movement at time $t = T{+}1$ to the end of the post-movement period at time $t = T + N$, and penalises deviation from the target position as well as ensuring that the velocity tends to zero at the end of the movement. The cost function J therefore becomes:

$$J_{mv} = \sum_{t=0}^{T+N-1} \left(R.u_t^2\right) + \left((x^* - x_T)^2 + (w.\dot{x}_T)^2 + (var_T)^2\right) + \sum_{t=T+1}^{T+N-1} \left((w.\dot{x}_t)^2 + (var_t)^2\right) \tag{6}$$

where N is the length of the post-movement period, x^* is the desired end position of the movement and w is a scaling factor for the velocity. The structure of the final state cost is also adapted from Todorov and Jordan, again without the terms relating to the second-order muscle model. The first term in Eq. (6) penalises the size of the control signals, which needed to be kept small to increase the accuracy of the movement in the presence of signal-dependent noise. The second term ensures that the state moves towards the desired end position at the correct time and begins to penalise the velocity and variance at this time. The third term continues to penalise the velocity and variance during the post-movement time, forcing the velocity to zero and minimising the positional variance.

The state representation and cost functions chosen up to this point were suitable for a single straight point-to-point reaching movement. However, humans can also move through other points when reaching towards a target, for instance when we need to reach around an obstacle. The single target point structure described so far is easily extensible to allow a number of other points to be included in the trajectory. The cost function was amended to include the positions of the necessary via-points. The same idea could also be used to specify a target velocity for the via-points or for the target position if the goal of the movement was not a stationary point.

In contrast to the target position, which by definition occurs at the end of the movement, each via-point consisted of both the desired position and the time at which the trajectory should pass through that position.

$$J_{via} = J_{mv} + \sum_{i=1}^{m} \left((x_i^* - x_{t_i})^2\right) \tag{7}$$

where m was the number of via-points, x_i^* was the desired position for via-point i, and x_{t_i} was the state value at the time specified by via-point i. To put this into a convenient matrix form, an extra "1" was added to the state representation for each via-point, in the same way as for the target position. Instead of the state cost Q being zero for the whole of the movement period, Q at each time t_i specified by via-point i included the desired position x_i^*. Q at all other times remained as zero.

4.1.3. *The Control Algorithm*

The DLQR algorithm consists of a pre-processing step where the "cost-to-go", P, at each time step was calculated, starting with the cost at the desired end state and working back to the initial state. The "cost-to-go" from the current state is the minimum cost incurred to reach the current state added to the minimum "cost-to-go" from the next state. P at time t = T + N (i.e. the end of the post-movement period) is simply the final state cost . Then, for every time step back to zero, the value of P is calculated according to Eq. (8).

$$P_{T+N} = Q_f$$

$$P_{t-1} = Q_t + (A'P_tA) - \left(A'P_tB\left(R + B'P_tB\right)^{-1}B'P_tA\right) \tag{8}$$

This process is called Riccati recursion and takes place before the start of the movement. When the movement begins the value of the optimising state feedback gain K_t is calculated online for each time step from $t = \mathit{0}$ to $t = T + N$ - *1*, according to

$$K_t = -\left(R + B'P_{t+1}B\right)^{-1}B'P_{t+1}A \tag{9}$$

This is then used in turn to calculate the DLQR control signal

$$u_t^{lqr} = K_t s_t \tag{10}$$

From this the amplitude of the signal dependent noise is calculated as a value drawn from a normal distribution with zero mean and variance $k\,(u_t^{lqr})^2$, where k is a proportionality constant. The calculated noise signal is then added to the control signal and the noisy signal is used in the state update Eq. (1).

The state representation given above, and all other equations in this section, are shown for movement in the x-axis only. Since this model is approximating a linear system of equations, extension to the y-axis simply involves adding the required terms to the state representation, dynamics and cost function matrices, and extending the dimensionality of all other matrices appropriately. The state representation for two degrees of freedom is given by Eq. (11). As before, the "1"'s in the state representation encode for positional accuracy.

$$s_t = \left[x_t,\, y_t,\, \dot{x}_t,\, \dot{y}_t,\, 1,\, 1\right]' \tag{11}$$

After the state had been updated, an inverse kinematics algorithm was used to calculate the joint angles required to position the robot arm according to the current state. These angles were used directly to plot the simulation results. The joint angles were then mapped on to servomotor position commands that were sent to a servo control board to move the robot arm.

4.1.4. *Task Optimization in the Presence of Signal-dependent noise (TOPS)*

A powerful feature of using an optimal control scheme to implement the minimum variance model is the ability to specify movements in terms other than reaching to an exact set of coordinates in the workspace.[14] Both the minimum jerk and the minimum torque models require this kind of specification, but the addition of noise as an integral part of the systems means that repeated movements to the same target coordinates will not end at precisely the same spot. Instead, tasks are specified in terms of moving to a certain point with a certain level of variance. This gives a target area, rather than a precise point, within which the goal can be said to be achieved. This has been formalised as Task Optimization in the Presence of Signal-dependent noise (TOPS).[14] Here we return to Fitts' Law, where the required accuracy of the goal (set by the required variance) affects the amount of time it takes to complete, or vice versa. This is extremely relevant for imitation, as will be discussed in the next section.

4.2. *Imitation System*

As stated above, the minimum variance model has already been implemented in several different forms. The approach outlined above draws from these previous implementations, but is constructed specifically to be both computationally straightforward and suitable for expanding into the modular imitation model described earlier. In the original implementation of this model,[2] the inverse models, or behaviours, were proportional-integral-derivative controllers whose gain parameters could be adjusted within certain limits to change the behaviour. The forward models used the motor commands produced by the inverse models to predict the next state. Each prediction altered the corresponding behaviour's confidence level, depending on how closely it matched the demonstrated trajectory. As the demonstration continued, either one behaviour would exhibit confidence levels far above the others, in which case it could imitated, or no behaviour would match the demonstration, in which case learning would take place through the passive imitation system.

In the adapted model, the PID controllers have been replaced by the optimal controller described in the previous section. Instead of storing sets of gain parameters and postures for each behaviour, the "cost-to-go" for a particular movement is stored. The "cost-to-go" is more of a prediction than an exact plan. It is determined by the target of the movement and is independent of the starting position. Thus, once a "cost-to-go" for a particular target has been calculated, it can be reached from any starting position as long as sufficient time is allowed for the movement within the velocity constraints of the system. At each time step during a demonstration, the system "simulates" the motor commands it would use to produce the demonstrated movement by running each module on a forward kinematics model. Each "cost-to-go" is used to produce a value of K_t which is combined with the internal state information to produce a motor command. Each motor command is used to update the internal state, and the resulting predicted states are compared with the state of the demonstrator. The closest match is chosen to be the next state.

As stated above, using an implementation of the minimum variance model allows tasks to be specified in terms of a target area or accuracy constraint. Using this approach, the system performs task-level imitation. At the end of the demonstration, the predicted end position is compared with the end position of the demonstrator. If the required goal is achieved, the imitation system can perform the action using the robot arm. If the goal is not achieved, the system can extract the demonstrators end position and the location of any via-points, and use these to construct a new cost function, which in turn can be used to produce a new "cost-to-go" module that can be added to the others. The new movement becomes part of the system's repertoire.

The optimal control system lends itself extremely well to task-level imitation of this nature, since the trajectory it produces is not specified beforehand, but is a result of the controller trying to minimise the variance of the final position. It is possible that the system may already have a "cost-to-go" module for a target coordinate. However, the timing of the movement may not produce movements with the required level of accuracy for a task. In this case, a new "cost-to-go" module to the same target coordinates would have to be constructed, but with more time steps to increase the accuracy and achieve the movement goal.

However, not all tasks can be specified simply in terms of an end goal or target area. Allowing via-points to be incorporated into task goals increases the number and complexity of tasks that can be imitated by this system, the best example being in the case of obstacle avoidance. This also means the system has a convenient way of imitating the exact trajectory of a demonstrated movement if this is required.

5. Results

The inverse kinematics for the two-link, two-joint configuration of the robot arm were well defined, so it was possible to extract reliable values for the joint angles from the hand position coordinates. For an arm with more degrees-of-freedom not restricted to a plane, a more complicated and less well-defined inverse kinematics algorithm would have to be used. In the point-to-point reaching experiments shown here control signals were calculated for an initial movement phase of duration T = 100 time steps and a post-movement phase of duration $N = 50$ time steps. The shoulder joint was defined as the origin for the movement.

The robot arm trajectories were captured using a simple webcam and algorithms adapted from those used in the open source AR Tool Kit.[25,26] As can be seen in Fig. 3, the trajectories for the human demonstrations were captured in the same way.

Fig. 4 shows plots generated by the computational model described in the section above for a typical point-to-point reaching movement, from coordinates (-0.6, 0) to (0, 0.6). The smooth step changes in position and bell-shaped velocity profiles can clearly be seen for movement in both the x- and the y-axis. The smooth changes in the joint angles can also be seen. The configuration of the arm as it moves the hand through the required trajectory is shown in Fig. 5. The straight hand path is evident in the figure.

The results shown in Fig. 6 are for movement of the robot arm, set to follow

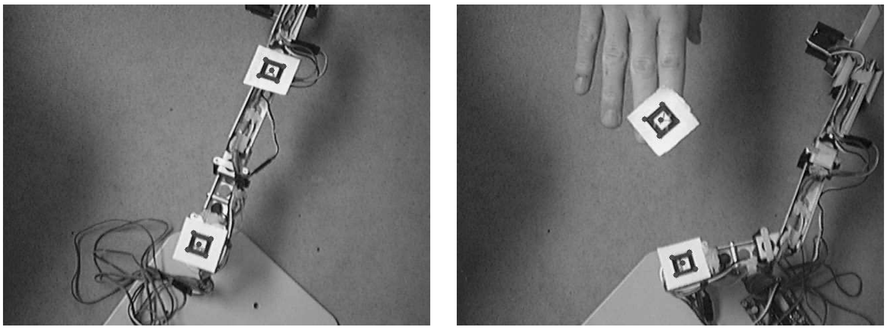

Figure 3. Captured images from the computer vision system, showing the makers used for tracking the trajectories

the path as in the simulation results from coordinates (-0.6, 0) to (0, 0.6). These results are typical of the trajectories produced by the robot arm. In each of plots in Fig. 6, the trajectory of the robot arm is shown against the trajectory from the computational model used to drive the servo motors. The path of the robot arm was captured using a computer vision system that detected and tracked coloured markers placed on the joints and at the end of the second link. As can be seen in Fig. 6, despite small sampling errors in the vision system the path produced by the robot arm is very close to that of the computational model.

The next set of results, shown in Fig. 7a, are plots for a similar point-to-point reaching movement as in Figs. 4 and 5, but repeated twenty-five times. These plots clearly show the variations in the repeated trajectory due to signal-dependent noise. These variations between trials are typical of human-like movements, as are the slightly curved hand paths. Fig. 7b also shows this variation between repeated

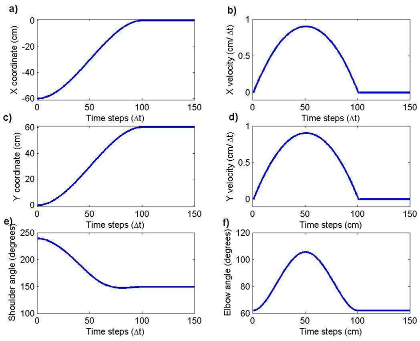

Figure 4. Plot showing a) change in x-axis position; b) change in velocity along the x-axis; c) change in y-axis position; d) change in velocity along the y-axis; e) change in shoulder angle; f) change in elbow angle

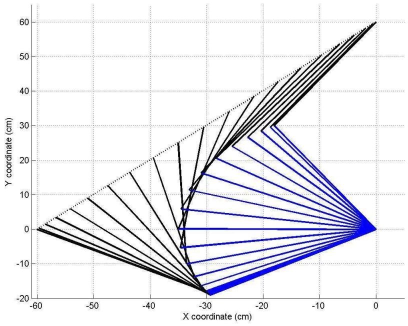

Figure 5. Plot showing simulated arm postures captured at equal time intervals during a point-to-point reaching movement

movements, as well as clearly showing the tradeoff between the speed of a movement and its accuracy, measured by the end point standard deviation.

Fig. 8 shows a trajectory that includes a via-point. Since the trajectory is longer, the end point variance will be larger for a given movement time compared to the straight line movement. This would need to be taken into account when deciding whether the goal has been achieved at the end of a simulated movement.

The next set of results show the application of the movement system to an imitation task. The imitation system was provided with a number of "cost-to-go" modules for random chosen target positions. A demonstration trajectory was then produced to a target not known by the imitation system. Figs. 9 and 10 show both the first attempt of the system to replicate the demonstrated trajectory, which fails, and the subsequent successful attempt after learning has taken place. In Fig. 9, the demonstrated trajectory is a simple point-to-point reaching movement to a

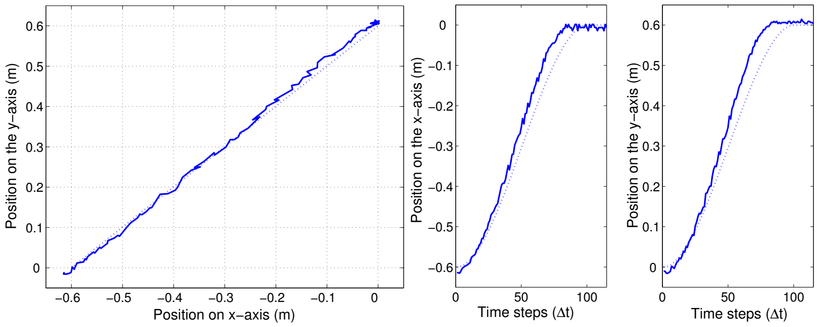

Figure 6. Plots showing typical trajectories produced by the robot arm (solid line) and the simulated trajectories (dotted line), a) The hand path of the robot arm in the x-y plane; b) Change in position on the x-axis; c) Change in position on the y-axis

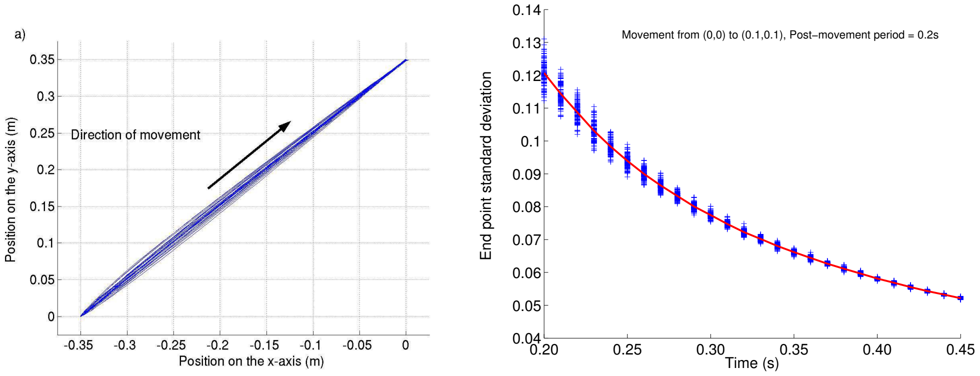

Figure 7. a) Plot for a point-to-point reaching movement repeated twenty-five times showing the variation between repeated trials
b) Speed-accuracy trade-off of the minimum variance system. For each movement time, 100 movements were repeated over a distance of 14cm and the end point standard deviations were recorded.

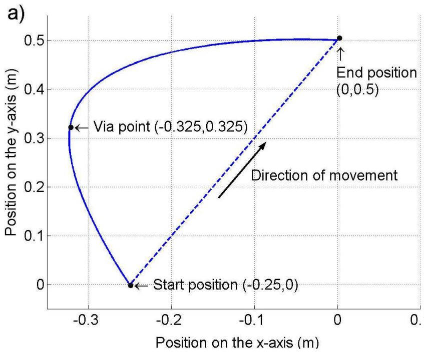

Figure 8. Plot showing a more complex trajectory including a via-point

target unknown to the system. In Fig. 10, the system has a module that allows it to reach the target, but does not pass close enough to the required via-point. A new module is therefore learnt, including the location of the via-point.

It should be emphasised that although the imitation system can learn a module for any given target point in the workspace, this does not mean that it could be provided with modules allowing it to reach every point and then be able to replicate every trajectory it observed. Even if this were the case, the addition of via-points means that an infinite range of more complex trajectories would need to be provided as well. This would also be time consuming and expensive in terms of memory storage of the modules. By providing the system with only a few basic targets, and then demonstrating task related movements, the system learns only those movements it needs to achieve a task. The nature of the modular system is such that it will then to able to generalise within this set of known movements. This is currently being extensively tested.

A further point to note is that since the goals are specified in terms of an accuracy constraint, even if a module exists to a demonstrated target position, it may not produce a trajectory with the required accuracy. In this case, another module would have to be learnt, with a greater number of time steps to decrease the end point variance. The location of via-points in time as well as space is also important, since changes in the temporal position of the same spatial via-point results in very different trajectories, which may be critical to the task. Even though a certain via-point location might be part of a module at a particular time step, a new module would have to be learnt if the same via-point were passed through at

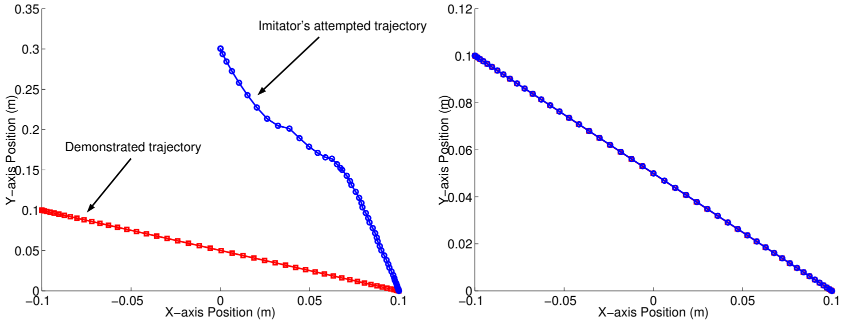

Figure 9. From its known set of "cost-to-go" modules, the system attempts to imitate a simple point-to-point reaching movement. The plot on the left shows this attempt, which clearly fails. The plot on the right shows that, after learning a new "cost-to-go" module, the system is able to replicate the demonstration.

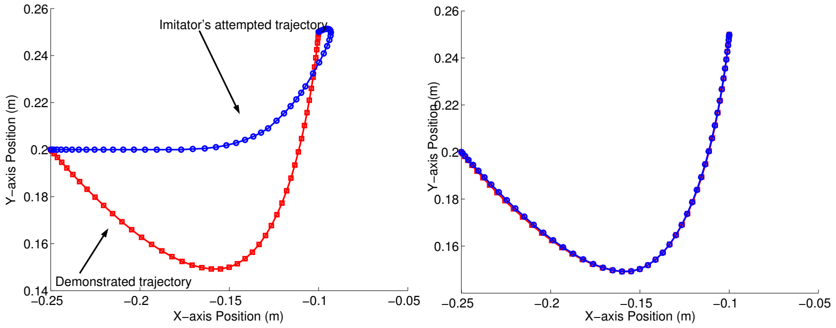

Figure 10. In these examples, the imitation system has a known "cost-to-go" module for the target of the demonstration. The plot on the left shows that this goal is achieved, but that the via-point goal is not. A new "cost-to-go" module is therefore learnt, providing the imitation system with the ability to follow the new trajectory.

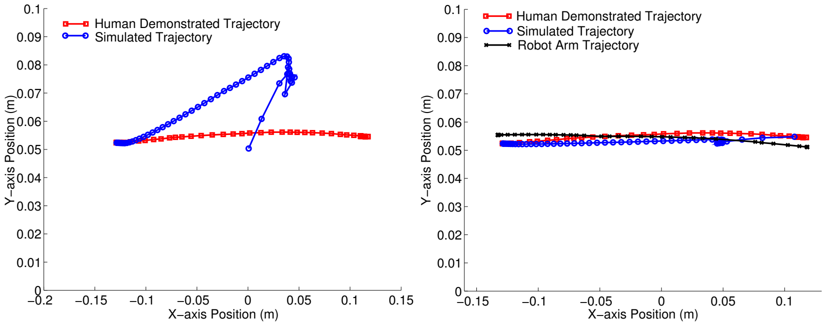

Figure 11. The plot on the left shows a human demonstration (red) and the imitation system's first attempt to replicate the movement (blue). The plot on the right shows the same demonstration (red), the system's trajectory after learning a new module (blue), and the trajectory produced by the robot arm (black).

a different time. Our further work will look at the imitation of tasks involving very specific timing, such as tapping a sequence. The temporal aspects of movement are as important as the spatial aspects, but are not as well understood. Following our biologically-inspired philosophy, we intend to look at neuroscientific models of timing and time representation. How the human brain deals with timing and coordination on different time scales is an interesting issue, one that our present framework is well suited to investigate.

The remaining figure shows the application of the system to real human movements. Fig. 11 shows a typical human point-to-point reaching demonstration and the system's attempts to replicate the movement. In the first plot, running the existing modules through the internal forward kinematics model does not result in the goal being achieved, so a new module is learnt but no movement of the robot arm takes place. In the second plot, the new module has been learnt and the system is able to replicate the trajectory. The simulated trajectory is then used to drive the robot arm.

6. Conclusions and Further Work

Since we intended to integrate the perception system with the motor system to recognise the movements of a human demonstrator, the first step was to start by implementing a human-like motor system. The most biologically realistic of the theories that seeks to explain the computations of the human motor system is the minimum variance model set out by Harris and Wolpert. The results given in the previous section demonstrate an optimal control scheme, based on this model, that produces the required features of human-like movement on a robot arm.

To apply this scheme as the underlying motor representation for a robotic imitation learning scheme requires the system to be extended to include modularity. The human brain does not have a single controller for all movements and contexts, but is thought to have multiple controllers that can be modified through learning. Part of the optimal control scheme to implement the minimum variance model is a "cost-to-go" for a given target position, calculated "off-line" from the movement itself, providing a convenient modular system. The "cost-to-go" modules can then be combined in a system for recognising actions from demonstration, such as that described in Demiris and Hayes. Since the controller produces human-like movement, the modular structure is appropriate for recognising human movements from demonstration. When the system observes the demonstration, it runs each controller module in an internal simulation and tests the trajectory against the observed demonstrated trajectory. If the demonstration is known, it will be able to use the appropriate controller module to imitate the movement on a robot arm. If it is unknown, the system will be able to extract features from the demonstration, such as the target position and the location of any via-points and use them to build a new module. The optimal control scheme for producing human-like movement is therefore a key component of the system for robotic imitation learning.

Acknowledgements

This work has been supported through a Doctoral Training Award from the UK's Engineering and Physical Sciences Research Council (EPSRC). The authors wish to thank Bassam Khadhouri and Paschalis Veskos of the Biologically-inspired Autonomous Robots Team (BioART) at Imperial College for their advice and encouragement. We also wish to thank Anthony Dearden and Matthew Johnson, also of BioART, for their work with the computer vision tracking system.

Bibliography

1. Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Science*, 3(6):233–242, June 1999.
2. Yiannis Demiris and Gillian Hayes. Imitation as a dual-route process featuring predictive and learning components: A biologically-plausible computational model. In K. Dautenhahn and C. Nehaniv, editors, *Imitation in Animals and Artifacts*, chapter 13. MIT Press, 2002.
3. Christopher G. Atkeson, Joshua G. Hale, Frank Pollick, Marcia Riley, Shinya Kotosaka, Stefan Schaal, Tomohiro Shibata, Gaurav Tevatia, Ales Ude, Sethu Vijayakumar, and Mitsao Kawato. Using humanoid robots to study human behviour. *IEEE Intelligent Systems*, pages 46–56, 2000.
4. Giacomo Rizzolatti, Luciano Fadiga, Vittorio Gallese, and Leonardo Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.
5. Giacomo Rizzolatti, Laila Craighero, and Luciano Fadiga. The mirror system in humans. In Maxim I. Stamenov and Vittorio Gallese, editors, *Mirror Neurons and the Evolution of Brain and Language*, volume 42 of *Advances in Consciousness Research.* John Benjamins Publishing Company, 2002.
6. Vittorio Gallese and Alvin Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, December 1998.

7. Robert Gordon and Joe Cruz. Simulation theory. In Lynn Nadel, editor, *Encyclopedia of Cognitive Science.* December 2002.
8. Yiannis Demiris and Matthew Johnson. Distributed, predictive perception of actions: A biologically inspired robotics architecture for imitation and learning. *Connection Science Journal*, 15(4):231–243, December 2003.
9. P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954.
10. P. Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42:223–227, 1981.
11. Tamar Flash and Neville Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, July 1985.
12. Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement: Minimum torque-change model. *Biological Cybernetics*, 61:89–101, 1989.
13. Christopher M. Harris and Daniel M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, August 1998.
14. Hiroyuki Miyamoto, Daniel M. Wolpert, and Mitsuo Kawato. Computing the optimal trajectory of arm movement: The TOPS (Task Optimization in the Presence of Signal-dependent noise) model. In R.J.Duro, J. Santos, and M. Grana, editors, *Biologically Inspired Robot Behaviour Engineering*, volume 109 of *Studies In Fuzziness And Soft Computing*, chapter 14, pages 395–416. Springer-Verlag, 2003.
15. John Demiris and Gillian Hayes. Do robots ape? In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, pages 28–31. MIT Press, 1997.
16. Aude Billard. Imitation (a review). In M. A. Arbib, editor, *Handbook of Brain Theory and Neural Networks*, pages 566–569. MIT Press, 2002.
17. Narender Ramnani and R. Christopher Miall. A system in the human brain for predicting the actions of others. *Nature Neuroscience*, 7(1):85–90, January 2004.
18. J. Randall Flanagan and Roland S. Johansson. Action plans used in action observation. *Nature*, 424:769–771, August 2003.
19. Giacomo Rizzolatti, Leonardo Fogassi, and Vittorio Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9):661–670, September 2001.
20. J. Demiris, S. Rougeaux, G. M. Hayes, L. Berthouze, and Y. Kuniyoshi. Deferred imitation of human head movements by an active stereo vision head. In *Proceedings of the 6th IEEE International Workshop on Robot Human Communication*, pages 88–93. IEEE Press, September 1997.
21. Cynthia Breazeal and Brain Scassellati. Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11):481–487, November 2002.
22. Emanuel Todorov and Micheal I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, November 2002.
23. Jianfeng Feng, Kewei Zhang, and Gang Wei. Towards a mathematical foundation of minimum-variance theory. *Journal of Physics A: Mathematical and General*, 35:7287–7304, 2002.
24. Bruno Siciliano and Lorenzo Sciavicco. *Modelling and Control of Robot Manipulators.* McGraw-Hill Education, April 1996.
25. Mark Billinghurst. Real world teleconferencing. In *Proceedings of CHI '99, Conference Companion.* May 1999.
26. Mark Billinghurst and Hirokazu Kato. Collaborative mixed reality. In *Proceedings of the First International Symposium on Mixed Reality*, pages 261–284. Springer-Verlag, 1999.