In Proc. 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia (2011).

Model-Free Apprenticeship Learning for Transfer of Human Impedance Behaviour

Takeshi Mori, Matthew Howard and Sethu Vijayakumar

***Abstract*—We present a method for transferring behaviour from humans to robots via apprenticeship learning. While previous methods have relied on an accurate model of the demonstrator's dynamics, in most practical settings such models fail to capture (i) complex, non-linear dynamics of the human musculoskeletal system, and (ii) inconsistencies between modelling assumptions and the configuration and placement of measurement apparatus. To avoid such issues, we propose a model-free approach to apprenticeship learning, in which off-policy, model-free reinforcement learning techniques are used to extract a model of the objective function optimised in human behaviour. As a key ingredient, we derive a novel formulation of Least Squares Policy Iteration (LSPI) and Least Squares Temporal Difference learning (LSTD) to enable their application in this setting. The robustness of our approach is demonstrated in experiments where human hitting behaviour is transferred to a non-biomorphic robotic device.**

I. Introduction

A promising approach to the acquisition of skilled behaviours, such as ball hitting, racket swinging, etc., by robots is to transfer existing strategies from humans. This transfer is the subject of numerous studies in imitation learning, whereby robots are made to mimic human demonstrative behaviours, either by direct matching of command and action sequences [18], [4] or by matching specific features of the human's behaviour [3]. However, the application of such approaches comes into difficulty when considering the numerous differences between the human musculo-skeletal system and the dynamics and actuation of robots (Fig. 1). Specifically, imitation by trajectory matching relies on the selection of appropriate features of the dynamic behaviour of both the human and robot plants. This is non-trivial since it is often unclear how different features relate to task performance over the duration of the movement, especially for highly dynamic tasks, such as hitting [12] or ball-throwing [8].

To avoid problems such as these, it has recently been proposed to take an inverse optimal control approach for transfer between heterogeneous systems [12]. The basis of this approach is to imitate behaviour on the level of *task objectives*, exploiting recent algorithmic advances in Apprenticeship Learning (AL) or inverse reinforcement learning [2], [17], [19]. Specifically, such approaches extract features of the objective function, that do not depend on the specific dynamic properties of the demonstrator and are thereby suitable for transfer to robotic hardware.

T. Mori, M. Howard and S. Vijayakumar are with the School of Informatics, University of Edinburgh, Scotland, UK. takeshi.mori@ed.ac.uk

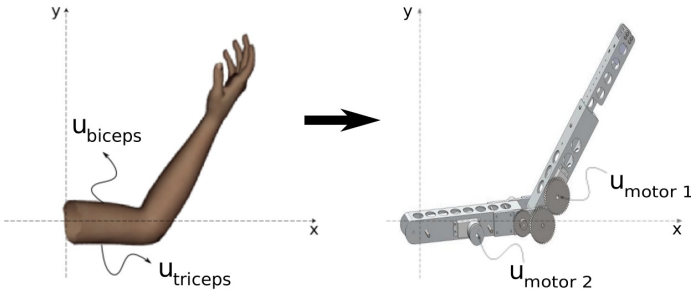


Fig. 1. Human and robotic actuation systems: (left) Humans use muscle activations (e.g., $u_{triceps}$ and u_{biceps}) while (right) robotic systems are controlled with command signals to the motors (e.g., u_{motor1} and u_{motor2}).

AL has been successfully demonstrated in several highly dynamic control problems, such as helicopter control [9] and car navigation [1]. Typically, these have involved learning from data in which the dynamics of the system are well-known, for example in the context of teleoperation of a robot. However, AL has so far found relatively limited application learning directly from human behaviour (e.g., in order to transfer behaviour to anthropomorphic robots or manipulators). In this domain, one of the major obstacles is the problem of modelling the dynamics of the demonstrator, i.e., the human musculo-skeletal system. Problems arise, for example, due to the complexity of human muscle and tendon dynamics, and the difficulty of non-invasive measurement of musculo-skeletal properties (e.g., limb weight, length, inertia and impedance characteristics).

To overcome these difficulties, in this paper we propose a novel, model-free approach to AL, where the effort of modelling human dynamics is totally avoided. Our approach is tailored to the demands of behaviour transfer in a non-invasive fashion. Specifically, we take a model-free Reinforcement Learning (RL) approach whereby information about the dynamics is implicitly encoded in a pre-recorded data set, rather than explicitly defined in a model. One of the key requirements is to avoid the need for action exploration in the RL (which, in our domain, would require execution of exploratory actions on the demonstrator plant - i.e., stimulation of the human's muscles). This is achieved by employing off-policy RL techniques, using modified versions of Least Squares Policy Iteration (LSPI) [14] and Least Squares Temporal Difference learning (LSTD) [7], [6] in finite horizon. We illustrate the practicality of our approach in numerical simulations and in an experiment in which behaviour is transferred from a human to a variable impedance robotic actuator.

II. Problem Definition

We address the problem of transferring human skills, i.e., control strategies of an expert (human) demonstrator (e) to an apprentice (robot) learner (l) assuming that the two have a very different embodiment in terms of their dynamics. Specifically, we assume the expert has state ${}^e\mathbf{x} \in \mathbb{R}^n$, controls movement with commands ${}^e\mathbf{u} \in \mathbb{R}^m$, and has dynamics

$$ {}^e\dot{\mathbf{x}} = {}^e\mathbf{f}({}^e\mathbf{x}, {}^e\mathbf{u}) \quad \in \mathbb{R}^n. \tag{1}$$

Our goal is to transfer behaviour to a heterogeneous learner (robot), with state ${}^l\mathbf{x} \in \mathbb{R}^p$, command signal ${}^l\mathbf{u} \in \mathbb{R}^q$ and dynamics

$$ {}^l\dot{\mathbf{x}} = {}^l\mathbf{f}({}^l\mathbf{x}, {}^l\mathbf{u}) \quad \in \mathbb{R}^p. \tag{2}$$

Note that, in general, the state and action space of the two systems (${}^e\mathbf{x}, {}^e\mathbf{u}$ and ${}^l\mathbf{x}, {}^l\mathbf{u}$) may differ significantly between human and robot. For example, for a human expert, ${}^e\mathbf{u}$ may correspond to muscle activations, whereas for a robot learner ${}^l\mathbf{u}$ may correspond to desired position of a servo-motor.

Crucially, we note the difficulty of obtaining a model of ${}^e\mathbf{f}(\cdot)$ in (1) when learning from human demonstrations since its form is highly specific to a number of factors related to the musculo-skeletal properties and the measurement apparatus. For example, ${}^e\mathbf{f}(\cdot)$ contains information about the complex mass distribution and inertia of the demonstrator's limb and the elasticity of muscles and tendons. It will also depend critically on measurement factors such as, for example, the positioning of surface electromyography (EMG) sensors, and the way in which muscles move under the sensors during motion. All of these factors make it difficult to obtain an accurate model of the dynamics for existing model-based approaches to AL.

A. Transfer via Apprenticeship Learning

Our approach follows closely the framework proposed in [12], a schematic of which is depicted in the lower half of Fig. 2. The key idea involves abstracting out the difference in dynamics by performing behaviour transfer at the level of task objectives. Specifically, we employ AL to extract a model of the objective function from recordings of a human demonstrating some task. From a set of K demonstration trajectories ${}^eD = \left\{({}^e\mathbf{x}_0^k, {}^e\mathbf{u}_0^k), \cdots, ({}^e\mathbf{x}_N^k, {}^e\mathbf{u}_N^k)\right\}_{k=0}^K$ we learn a model of the human objective function in the form

$$ {}^eJ = \sum_{i=1}^{\eta} w_i {}^eh_i({}^e\mathbf{x}(T)) + \sum_{i=\eta+1}^{N} w_i \int_0^T {}^el_i({}^e\mathbf{x}, {}^e\mathbf{u}, t)\, dt $$

or, more compactly

$$ {}^eJ = \mathbf{w}^\top {}^e\boldsymbol{\xi}({}^e\mathbf{x}, {}^e\mathbf{u}) \tag{3}$$

where ${}^eh_i(\cdot), {}^el_i(\cdot)$ are a set of (known) basis functions representing terminal and running costs, respectively, i.e., ${}^e\boldsymbol{\xi} = ({}^eh_1, \cdots, {}^eh_\eta, \int_0^T {}^el_{\eta+1}dt, \cdots, \int_0^T {}^el_N dt)^\top$, $\mathbf{w} = (w_1, \cdots, w_N)^\top$ are parameters to be estimated from data, and we assume (by renormalisation, if necessary) that $w_i > 0$ and $||\mathbf{w}||_1 = 1$. Note that, the basis functions ${}^eh_i(\cdot)$, ${}^el_i(\cdot)$ may be made up of a set of bases for a generic function approximator (e.g., Gaussian radial basis functions), or a set of salient features of the task (e.g., energy or accuracy costs).

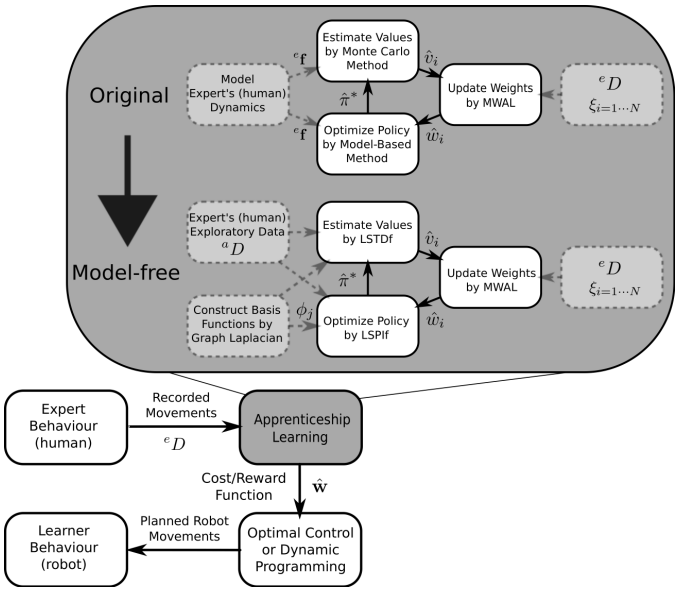


Fig. 2. Transfer via apprenticeship learning (AL). Cost weights $\hat{\mathbf{w}}$ are learnt by AL via expert's (human) demonstration ${}^eD$. Learner (robot) optimises the behaviour under the transferred cost weights $\hat{\mathbf{w}}$. Comparison of original and model-free ALs are shown in the top half.. The original AL employs Monte Carlo method for value estimation and model-based method for policy optimisation, both of which need (human) dynamics model ${}^e\mathbf{f}$. On the other hand, model-free AL collects exploratory data ${}^aD$ and LSTDf estimates values and LSPIf optimises policy without any dynamics models.

For the behaviour transfer, we then take these learnt weight parameters $\hat{\mathbf{w}}$ and use them to optimise behaviour with respect to the dynamics of the robot learner (2). Specifically, we optimise the learner behaviour with respect to the learnt cost

$$ {}^lJ = \hat{\mathbf{w}}^\top {}^l\boldsymbol{\xi}({}^l\mathbf{x}, {}^l\mathbf{u}) \tag{4}$$

where ${}^l\boldsymbol{\xi} = ({}^lh_1, \cdots, {}^lh_\eta, \int_0^T {}^ll_{\eta+1}dt, \cdots, \int_0^T {}^ll_N dt)^\top$, are a set of equivalent basis functions for the learner, taking into account the correspondence with ${}^e\boldsymbol{\xi}({}^e\mathbf{x}, {}^e\mathbf{u})$ [12].

B. Model-based Apprenticeship Learning

While several model-based methods for AL have been proposed [12], [19], [2], in the literature, our framework of choice is Multiplicative Weights AL (MWAL) [19] which has been shown to be robust and efficient in the robotics domain. MWAL is an iterative approach that uses estimates of the expected value of the observed behaviour ${}^eD$ to find an approximation $\hat{\mathbf{w}}$ of the expert's weights. Specifically, the algorithm starts by assigning an initial guess of the weights $\hat{\mathbf{w}}$, and calculating the vector ${}^e\hat{\mathbf{v}}$ of expected values of the demonstration data ${}^eD$ under each of the cost bases ${}^e\xi_i(\cdot,\cdot)$,

$$ {}^e\hat{v}_i = \frac{1}{K} \sum_{k=0}^{K} \left\{ {}^e\xi_i({}^e\mathbf{x}^k, {}^e\mathbf{u}^k) \right\}. \tag{5}$$

It then iterates between three stages, namely, (i) forward optimisation of a control policy under the current weight estimate $\hat{\mathbf{w}}$, (ii) estimation of the vector of expected values $\hat{\mathbf{v}}$ of that policy under each of the cost bases ${}^e\xi_i(\cdot,\cdot)$, and (iii)

Algorithm 1 Model-based MWAL

Input: ${}^eD$: human demonstration data
${}^e\boldsymbol{\xi}$: cost basis vector
${}^e\mathbf{f}$: human muscle dynamics model
Initialise weights $\hat{w}_i = 1/N$ for all i
Estimate ${}^e\hat{\mathbf{v}}$ from ${}^eD$ through (5).
repeat
  Optimise policy $\hat{\boldsymbol{\pi}}$ by model-based method under $\hat{\mathbf{w}}$ with ${}^e\mathbf{f}$ and ${}^e\boldsymbol{\xi}$
  Estimate value vector $\hat{\mathbf{v}}$ through Monte Carlo sampling of $\hat{\boldsymbol{\pi}}^*$ applied to ${}^e\mathbf{f}$
  Update $\hat{\mathbf{w}}$ according to $\hat{w}_i := \hat{w}_i \beta^{-\alpha({}^e\hat{v}_i - \hat{v}_i)}$
  re-normalise: $\hat{\mathbf{w}} := \hat{\mathbf{w}}/||\hat{\mathbf{w}}||_1$
until $\hat{\mathbf{w}}$ is unchanged
Output: $\hat{\mathbf{w}}$

update of the estimated $\hat{\mathbf{w}}$ by reducing the difference of the estimated costs, i.e., minimising the criterion

$$J_{MWAL} = \hat{\mathbf{w}}^\top(\hat{\mathbf{v}} - {}^e\hat{\mathbf{v}}), \tag{6}$$

until convergence. A summary of model-based MWAL is shown in Algorithm 1, where $\beta = (1+\sqrt{(2\log N)/L})^{-1}$ [19], L is number of iterations, and α is learning rate.

Crucially, in existing implementations of this process, a model of the expert dynamics ${}^e\mathbf{f}(\cdot)$ enters at two of the stages (see the top of Fig. 2). First, at stage (i) a model-based approach is used for forward optimisation, for example model-based RL, ILQG [12], differential DP, etc. Second, at stage (ii) the expected values $\hat{\mathbf{v}}$ are computed through Monte Carlo sampling. In other words, the policy found in step (i) is used along with the expert dynamics ${}^e\mathbf{f}(\cdot)$ in order to generate a set of trajectories $\tilde{D}$. These are used in place of ${}^eD$ in (5) to compute the estimated value vector $\hat{\mathbf{v}}$. Clearly, the accuracy of these estimates, and thereby the quality of our estimated $\hat{\mathbf{w}}$, depends heavily on the accuracy of our model of ${}^e\mathbf{f}(\cdot)$.

C. Model-free AL for Behaviour Transfer

An alternative to the model-based approach outlined above, is to take a purely data-driven approach to learning. In other words, rather than trying model the dynamics of the demonstrator, we would like to represent those dynamics implicitly, in terms of data recorded on a per-experiment basis. The advantages of this are that (i) it avoids the need for complex modelling and invasive measurements of the human musculoskeletal system, and (ii) it sidesteps the aforementioned issues related to measurement apparatus (e.g., positioning of EMG sensors). Most importantly, since we avoid bootstrapping estimates of $\hat{\boldsymbol{\pi}}^*$ and $\hat{\mathbf{v}}$ on an (erroneous) model of ${}^e\mathbf{f}(\cdot)$, we should achieve more accurate estimates of the cost parameters $\hat{\mathbf{w}}$.

To realise this *model-free transfer of behaviour*, our proposal is to exploit model-free techniques from RL. In our setting, one of the requirements on such an approach is that learning must be conducted *off policy*. The latter is necessary since, during the estimation of $\hat{\mathbf{w}}$, it is not possible to sample trajectories from the human under a prescribed command sequence (i.e., we cannot directly control the human's muscles according to policies planned during the forward optimisation). We also note that, for the final transfer part (i.e., forward optimisation of behaviour for the apprentice (robot)) we can continue to exploit models of the robot dynamics (2), since these are relatively easy to obtain for artificial systems.

One method for model-free AL uses stochastic gradient descent based on the relative entropy [5], in which gradients are estimated by importance sampling. While the probablistic formulation of the latter is appealing, one of the difficulties lies in its application to problems with long horizons (i.e., long duration demonstrations), since the variance of the importance sampling estimator increases exponentially with the trajectory length [10]. To avoid such problems, here we propose a method based on LSPI [14], a model-free and off-policy RL technique, with efficient sample usage. We turn to the details in the next section.

III. Method

In this section, we outline our approach to model-free AL for behaviour transfer. In contrast to model-based MWAL, our approach works on two data sets, namely the set of task demonstrations, ${}^eD$, and a second, auxiliary data set ${}^aD$ collected during, for example, random movement (e.g., motor babbling). The latter *implicitly represents the dynamics of the human*, and can be used as a proxy for an explicit model of ${}^e\mathbf{f}(\cdot)$. The primary difference in the new approach is in the policy optimisation and value estimation (see Fig. 2), as detailed below.

A. Least Squares Policy Iteration in Finite Horizon

In its standard formulation, LSPI [14] is composed of two steps: (i) policy evaluation and (ii) policy improvement. In the former, the value function Q^π is linearly approximated based on sample data $\{\mathbf{x}_m, \mathbf{u}_m, \bar{\mathbf{x}}_m, j_m\}_{m=1}^M$ where $\bar{\mathbf{x}}_m$ is the integrated state (i.e., the state to which the system transitions when command $\mathbf{u}_m$ is applied in state $\mathbf{x}_m$) and j_m is the instantaneous cost of making that transition. Note that, no assumption is made about the origin of these samples, thereby allowing on- or off-policy estimation of Q^π.

Typically, LSPI is formulated as an infinite horizon problem, however, when learning from demonstration, we are more commonly interested in *discrete tasks with a finite time horizon*, such as reaching, manipulation or throwing. We must therefore derive a formulation of LSPI applicable to such tasks.

The main difference in the finite horizon setting, is that the policy and the value function become non-stationary (i.e., time-dependent). This means that we need to find an approximation of the value function $Q_t^\pi(\mathbf{x}, \mathbf{u})$ *indexed on the time step* t. An efficient way to do this, is to use time-indexing through the parameters: $Q_t^\pi(\mathbf{x}, \mathbf{u}) \approx \hat{Q}_t^\pi(\mathbf{x}, \mathbf{u}) = \boldsymbol{\phi}(\mathbf{x}, \mathbf{u})^\top \boldsymbol{\theta}_t$ where $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u})$ are a stationary set of basis functions. At the tth time step, our approximation should minimise

$$J_t(\boldsymbol{\theta}_t) = \frac{1}{2}\sum_{m=1}^{M}\left(Q_t^\pi(\mathbf{x}_m, \mathbf{u}_m) - \boldsymbol{\phi}(\mathbf{x}_m, \mathbf{u}_m)^\top \boldsymbol{\theta}_t\right)^2. \tag{7}$$

Algorithm 2 Policy Optimisation by LSPIf

1: **Input:** $D \equiv \{\mathbf{x}_m, \mathbf{u}_m, \bar{\mathbf{x}}_m, j_m\}_{m=1}^M$: sample data
T: time horizon, $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u})$: basis functions
2: Set value function $\hat{V}_T^\pi(\mathbf{x}) = h(\mathbf{x})$ and
sufficient statistic $\mathbf{A} = \sum_{m=1}^M \boldsymbol{\phi}_m \boldsymbol{\phi}_m^\top$
3: **for** $t = T-1$ **to** 0 **do**
4: Approximate action value function:
$Q_t^\pi(\mathbf{x}, \mathbf{u}) \approx \boldsymbol{\phi}(\mathbf{x}, \mathbf{u})^\top \boldsymbol{\theta}_t$ where $\boldsymbol{\theta}_t := \mathbf{A}^{-1}\mathbf{b}$
and $\mathbf{b} := \sum_{m=1}^M \boldsymbol{\phi}_m (j_m + \hat{V}_{t+1}^\pi(\bar{\mathbf{x}}_m))$
5: Optimise policy: $\boldsymbol{\pi}_t(\mathbf{x}) = \arg\min_{\mathbf{u}} \boldsymbol{\phi}(\mathbf{x}, \mathbf{u})^\top \boldsymbol{\theta}_t$
6: Set value function: $\hat{V}_t^\pi(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}, \boldsymbol{\pi}_t(\mathbf{x}))^\top \boldsymbol{\theta}_t$
7: **end for**
8: **Output:** $\{\boldsymbol{\pi}_t(\mathbf{x})\}_{t=0}^{T-1}$

This can be achieved by solving

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}_t} J_t(\boldsymbol{\theta}_t) = -\sum_{m=1}^{M} \boldsymbol{\phi}_m \left(Q_t^\pi(\mathbf{x}_m, \mathbf{u}_m) - \boldsymbol{\phi}_m^\top \boldsymbol{\theta}_t \right) = \mathbf{0} \quad (8)$$

for $\boldsymbol{\theta}_t$, where $\boldsymbol{\phi}_m = \boldsymbol{\phi}(\mathbf{x}_m, \mathbf{u}_m)$.

Here, to compute (8), an estimate of $Q_t^\pi(\mathbf{x}_m, \mathbf{u}_m)$ is needed at every time step. This can be obtained through bootstrapping: we first initialise the value function at T as $\hat{V}_T^\pi(\mathbf{x}) = V_T^\pi(\mathbf{x}) = h(\mathbf{x})$, where $h(\mathbf{x})$ is the terminal cost. We then iteratively solve (8) from time step $T-1$ to 0 using the *value function estimate for the next time step* as our estimate of $Q_t^\pi(\mathbf{x}_m, \mathbf{u}_m)$, i.e.,

$$Q_t^\pi(\mathbf{x}_m, \mathbf{u}_m) \approx j_m + \hat{V}_{t+1}^\pi(\bar{\mathbf{x}}_m) \quad (9)$$

where $\hat{V}_{t+1}^\pi(\bar{\mathbf{x}}_m) = \boldsymbol{\phi}(\bar{\mathbf{x}}_m, \boldsymbol{\pi}_{t+1}(\bar{\mathbf{x}}_m))^\top \boldsymbol{\theta}_{t+1}$. The policy is then estimated as

$$\boldsymbol{\pi}_{t+1}(\mathbf{x}) = \arg\min_{\mathbf{u}} \boldsymbol{\phi}(\mathbf{x}, \mathbf{u})^\top \boldsymbol{\theta}_{t+1} \quad (10)$$

and the optimal parameters are retrieved at each time step

$$\boldsymbol{\theta}_t := \mathbf{A}^{-1}\mathbf{b}, \quad (11)$$

where $\mathbf{A} = \sum_{m=1}^M \boldsymbol{\phi}_m \boldsymbol{\phi}_m^\top$ and $\mathbf{b} = \sum_{m=1}^M \boldsymbol{\phi}_m (j_m + \hat{V}_{t+1}(\bar{\mathbf{x}}_m))$. We call this algorithm LSPIf (LSPI in finite horizon). A summary is provided in Algorithm 2.

Finally, we note that for finite horizon *policy evaluation*, we can use the same process, but simply omit the policy improvement step (10) (i.e., step 5 in Algorithm 2). This is effectively the finite horizon version of LSTD [7], [6]. In our setting, this is important since it provides a model-free method for estimating the value vector $\hat{\mathbf{v}}$ that is required for minimising the MWAL objective (6).

B. Model-free MWAL for Behaviour Transfer

We are now in a position to construct a model-free version of the MWAL algorithm. As input to the algorithm, we require (i) task demonstrations ${}^eD$, (ii) a set of auxiliary data ${}^aD$ collected from the demonstrator performing random actions, and (iii) a set of cost basis functions ${}^e\boldsymbol{\xi}(\cdot)$.

We initialise learning as for the model-based algorithm, i.e., by setting an initial guess for the weights $\hat{\mathbf{w}}$ and by estimating the value vector for the demonstration data ${}^e\hat{\mathbf{v}}$ through (5). The algorithm then iterates between the same three steps, i.e. (i) estimating the optimal policy $\hat{\boldsymbol{\pi}}$ under the current estimate of the weights $\hat{\mathbf{w}}$, (ii) estimating the value vector $\hat{\mathbf{v}}$ under $\hat{\boldsymbol{\pi}}$ and (iii) updating $\hat{\mathbf{w}}$ according to the difference between $\hat{\mathbf{v}}$ and ${}^e\hat{\mathbf{v}}$.

Algorithm 3 Model-free MWAL

1: **Input:** ${}^eD$: human demonstration data
${}^aD$: auxiliary data from human randomly
performing actions, ${}^e\boldsymbol{\xi}$: cost basis vector
2: Initialise cost weights $\hat{w}_i = 1/N$ for all i
3: Estimate ${}^e\hat{\mathbf{v}}$ from ${}^eD$ through (5).
4: **repeat**
5: **Optimise** policy $\hat{\boldsymbol{\pi}}$ with LSPIf applied to ${}^aD$
with costs $\{j_m\}_{m=1}^M$ predicted by $\hat{\mathbf{w}}$, ${}^e\boldsymbol{\xi}$
6: **Estimate** value vector $\hat{\mathbf{v}}$ by LSTD with ${}^e\boldsymbol{\xi}$ under $\hat{\boldsymbol{\pi}}^*$
7: **Update** $\hat{\mathbf{w}}$ according to $\hat{w}_i := \hat{w}_i \beta^{-\alpha({}^e\hat{v}_i - \hat{v}_i)}$
re-normalise: $\hat{\mathbf{w}} := \hat{\mathbf{w}} / ||\hat{\mathbf{w}}||_1$
8: **until** $\hat{\mathbf{w}}$ is unchanged
9: **Output:** $\hat{\mathbf{w}}$

The two main differences are, first, rather than using a model-based optimisation scheme in step (i), we use model-free, off-policy LSPIf. This uses the auxiliary data set ${}^aD$ augmented with a set of cost predictions $\{j_m\}_{m=1}^M$ from the current estimate of the cost function (i.e., weights $\hat{\mathbf{w}}$).

Second, in step (ii), we avoid using Monte Carlo sampling of trajectories from the policy $\hat{\boldsymbol{\pi}}$ under the dynamics ${}^e\mathbf{f}$ to estimate the value vector $\hat{\mathbf{v}}$. Instead, we use finite horizon LSTD (LSTDf) to evaluate the learnt policy under each of the cost bases ${}^e\boldsymbol{\xi}(\cdot)$. A comparison of the two approaches is illustrated in Fig. 2, and a summary of the model-free algorithm is provided in Algorithm 3 (where α and β are defined in the same way as for the model-based approach).

C. Selection of Basis Functions

The approach outlined so far makes no structural assumptions on the expert's dynamics ${}^e\mathbf{f}(\cdot)$. However, for its successful application, a design decision must be made on the basis functions $\boldsymbol{\phi}(\cdot)$ used to approximate the action-value function Q^π. In general, this will depend on numerous factors, such as the data dimensionality, density and smoothness, as well as any prior knowledge about its form. In this section, we briefly explore how this choice affects the accuracy and computational cost of learning the cost function.

1) Setup: As a simple test example, we apply model-free MWAL to the finite-horizon, linear quadratic regulator (LQR) problem, and compare learning performance for different choices of basis function $\boldsymbol{\phi}(\cdot)$. The LQR problem is a standard optimal control problem, in which the dynamics are linear

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \quad (12)$$

and the cost quadratic. In our instantiation of the problem, the dynamics represent a 1-D point mass m where $\mathbf{x} = (q, \dot{q})^\top$ is the state (position and velocity), u is the force (controlled

	Model-Based		Model-Free			
	Correct	Incorrect	Polynomial	Graph Laplacian		
Size of ${}^aD$	-	-	6	300	600	900
$\|\hat{\mathbf{v}} - \mathbf{v}^*\|$	0	338.87	0	45.27 $\pm$62.69	19.61 $\pm$28.35	18.84 $\pm$22.43
$\|\hat{\mathbf{w}} - \mathbf{w}^*\|$	4.33 $\times 10^{-4}$	53.02	4.33 $\times 10^{-4}$	10.29 $\pm$12.89	4.47 $\pm$6.22	3.62 $\pm$3.86

TABLE I
ERROR IN ESTIMATED FEATURE VALUE VECTOR $\hat{\mathbf{v}}$ (AVERAGED OVER ITERATIONS OF MWAL) AND FINAL WEIGHT ESTIMATE $\hat{\mathbf{w}}$ FOR DIFFERENT CHOICES OF $\boldsymbol{\phi}(\cdot)$. SHOWN ARE (MEAN $\pm$ S.D.)$\times 10^{-3}$ OVER 50 TRIALS.

at 50Hz), $\mathbf{b} = (0, 1/m)^\top$ and $A_{1,1} = 0$, $A_{2,2} = 0$, $A_{2,1} = 0$, $A_{1,2} = 1$. The cost is

$$J = w_1 \mathbf{x}(T)^\top \mathbf{x}(T) + w_2 \int_0^T u^2 \, dt = \mathbf{w}^\top \boldsymbol{\xi}(\mathbf{x}, u), \qquad (13)$$

where $T = 0.04\,s$ and $\mathbf{w} = (w_1, w_2)^\top = (0.8, 0.2)^\top$.

As demonstrations ${}^eD$, $K = 7$ trajectories are collected from initial positions $q(t = 0) \in \{-3, -2, \ldots, 3\}$ under the optimal policy with respect to (13). We then compare learning with (i) a polynomial basis $\boldsymbol{\phi}(\mathbf{x}, u) = (q^2, \dot{q}^2, u^2, q\dot{q}, qu, \dot{q}u)^\top$, and (ii) a graph Laplacian basis [16], with $L = 125$ randomly generated reference sample sets. An auxiliary data set ${}^aD = \{\mathbf{x}_m, u_m, \bar{\mathbf{x}}_m\}_{m=1}^M$ containing $M = 6$ points is given for the polynomial basis, and three different sets of size M=300, 600 and 900 are given for graph Laplacian basis. To ground our comparison, we also apply model-based MWAL (using ILQG [15] for the forward optimisation) to the same demonstration data (i) with an exact model of the dynamics (i.e., given (12)), and (ii) with incorrectly modelled dynamics. The latter, incorrect model takes the same parametric form as (12), but the vector $\mathbf{b}$ is multiplied by 0.8 (i.e., the mass is increased).

2) Evaluation: We evaluate the effect that the choice of basis function has on (i) accuracy of the final weight estimate (measured as the l_2-norm between the expert weights $\mathbf{w}$ and the estimated $\hat{\mathbf{w}}$) and (ii) accuracy of the estimated feature value vector $\hat{\mathbf{v}}$ used in learning. This gives an indication of the quality of the learning signal $||{}^e\hat{\mathbf{v}} - \hat{\mathbf{v}}||$. Since the best possible learning signal would arise from using the expected feature value vector $\mathbf{v}^*$ of the true optimal policy[1] $\boldsymbol{\pi}^*$, we can assess the quality of the model-free learning signal by looking at $||\hat{\mathbf{v}} - \mathbf{v}^*||_2$. These values are presented in Table I.

3) Estimation Accuracy: As can be seen, the error of ILQG with correct model is zero since, in the problem (12)-(13), ILQG finds the exact solution. Second, since the polynomial basis can exactly represent the true optimal value function (which is quadratic for the LQR problem), it also achieves zero error with just $M = 6$ samples. Finally, the graph Laplacian basis does a little worse than the polynomial basis on this problem, but still far outperforms the model-based method with the incorrect dynamics model. This is to be expected since it is a non-parametric technique, and as such does not have as strong a bias as the polynomial basis. However, for problems where the shape of the value function is not known a priori, it remains a competitive choice.

4) Sample and Computational Complexity: From Table I, it can be seen that the graph Laplacian requires more data than the polynomial basis, with the trend of increasing accuracy as the sample size M increases. In general, the model-free approach requires more data than the model-based approaches (namely, the auxiliary data ${}^aD$), and this comes with some associated increase in computational cost. In practice, however, this may be a small price to pay in order to avoid the difficulty of modelling the expert's dynamics.

[1]These quantities are computed by numerical solution of the Riccati equations to find $\boldsymbol{\pi}^*$, and then use of Monte Carlo to estimate $\mathbf{v}^*$

IV. EXPERIMENTS

In this section, we test the performance of our approach for model-free behaviour transfer across heterogeneous systems. We first investigate the problem in simulation to compare its performance against model-based approaches subject to modelling errors. We then look at its robustness when learning from real human data, given measurements that are *inconsistent with the model*, (e.g., due to poor sensor placement), and illustrate the effects on the learner robot behaviour.

A. Simulation Study

The goal of this investigation is to compare the performance of our model-free approach against that of model-based methods in the face of modelling errors. As a case study for this, we investigate the problem of transferring a hitting task, in which a ball is hit by wrist action of a human, to a non-biomorphic, variable impedance robot.

To simulate human demonstration of this task, ILQG is used to plan a set of optimal trajectories under the dynamics of a human wrist model. The latter consists of a single joint system actuated by two antagonistic muscles, with Kelvin-Voight muscle dynamics [13] (Fig. 3(b)). The torque on the joint depends on muscle tensions $\mathbf{T} \in \mathbb{R}^2$, according to

$$\tau(q, \dot{q}, \mathbf{u}) = -\mathbf{A}^\top \mathbf{T}(q, \dot{q}, \mathbf{u}), \qquad (14)$$

where $\mathbf{A} = (0.025, -0.025)^\top\, m$ represent moment arms, with a quadratic dependence on the muscle activations $\mathbf{u} \in \mathbb{R}^2$

$$\mathbf{T}(q, \dot{q}, \mathbf{u}) = \mathbf{K}(\mathbf{u})\big(\mathbf{l}_r(\mathbf{u}) - \mathbf{l}(q)\big) - \mathbf{B}(\mathbf{u})\dot{\mathbf{l}}(\dot{q}), \qquad (15)$$

where $\mathbf{l}(q) = \mathbf{l}_m - \mathbf{A}q \in \mathbb{R}^2$ are muscle lengths, $\mathbf{l}_m \in \mathbb{R}^2$ is the muscle length at $q = 0$,

$$\mathbf{K}(\mathbf{u}) = diag(\mathbf{k}_0 + g_k\mathbf{u}), \quad \mathbf{B}(\mathbf{u}) = diag(\mathbf{b}_0 + g_b\mathbf{u}) \quad (16)$$

are diagonal muscle stiffness and damping matrices ($\in \mathbb{R}^{2\times 2}$), respectively, and $\mathbf{l}_r(\mathbf{u}) = \mathbf{l}_0 + g_r\mathbf{u} \in \mathbb{R}^2$ is the muscle rest length. The coefficients g_k, g_b, g_r, and offsets $\mathbf{k}_0$, $\mathbf{b}_0$ and $\mathbf{l}_0$ are provided by the muscle model [13].

The task is to hit a target as hard as possible. For this, we model the expert's cost function as

$$J = w_1|q(T) - q^*|_\epsilon - w_2\dot{q}(T) + w_3 \int_0^T |Z\ddot{q}|_\epsilon \, dt \qquad (17)$$

where $q^* = 30^\circ$ is the target position in joint space and Z is a scaling factor, and $|x|_\epsilon$ denotes the ϵ-absolute value[2] of x. The three terms of (17) respectively correspond to (i) minimising the distance to the centre of the target[3] (ball), (ii) maximising the angular velocity at impact ($T = 0.5\,s$), and (iii) minimising effort during movement[4] Our goal is to estimate the weights $\mathbf{w} = (w_1, w_2, w_3)^\top$ that determine the trade-off between these objectives.

As training data ${}^eD$, $K = 5$ trajectories from initial positions $\{-20, -10, 0, 10, 20\}^\circ$ are sampled from the expert's policy (optimised with respect to (17)), and 600 random points in state-action space are used as the auxiliary data ${}^aD$. As the form of the value function in this problem is unknown, the graph Laplacian model is used as a generic basis function.

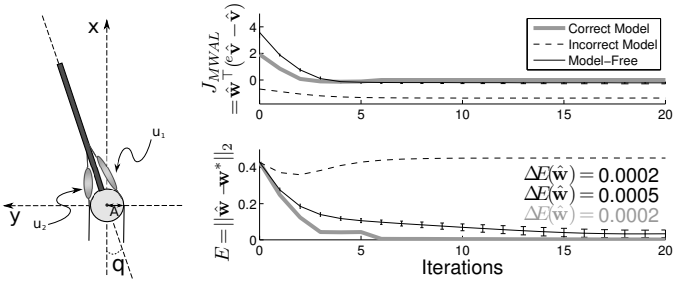
We compare model-free MWAL (MF-MWAL) to model-based MWAL (MB-MWAL, ref. Sec. II-B) given (i) an exact model of the demonstrator's dynamics (14)-(16) and (ii) a dynamics model containing modelling errors. For the latter, we use a model in which the muscle damping is overestimated, i.e., $\mathbf{B}'(\mathbf{u}) = 1.5\mathbf{B}(\mathbf{u})$. Note that, since in real world modelling of human muscle dynamics, the dependency of the damping factor on muscular activation is poorly understood, such discrepancy could be realistic.

Estimates $\hat{\mathbf{w}}$ of the objective function parameters were obtained with the different approaches over 50 trials where, for the model-free approach, ${}^aD$ and the graph Laplacian reference samples were randomly generated in each run. We evaluate performance in terms of the error in the learnt weights (L2 norm between $\mathbf{w}$ and $\hat{\mathbf{w}}$) and the value of J_{MWAL} (see (6)) achieved. We also used the learnt weights to plan optimal trajectories (i) under the true human dynamics (14)-(16) and (ii) under the heterogeneous dynamics of a robotic VIA, namely that of the MACCEPA joint [11]. In the former, good performance is indicated by the ability of the learnt objective function to predict the original behaviour of the demonstrator. In the latter, we test how well behaviour is transferred in terms of matching the characteristic features. The results are summarised in Fig. 3.

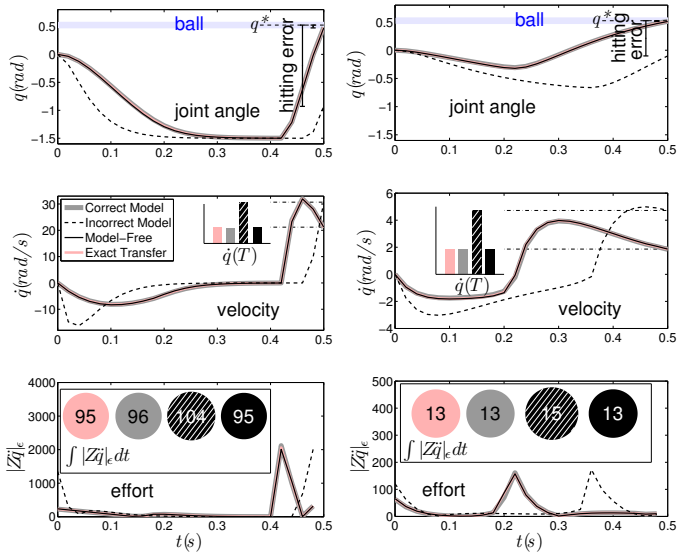
Looking at the learning curves (Fig. 3(c)), we see rapid convergence[5] with the best final accuracy (see Fig. 3(a)), achieved by MB-MWAL using the correct model. As expected, MB-MWAL with the erroneous model performed poorly, converging to an erroneous prediction of the weights. In contrast,

		Exact Transfer	Model-Based		Model -Free
			Correct	Incorrect	
Error in Weights			0.002	0.452	0.033 ± 0.021
Cost	Human	-0.343	-0.343	0.367	-0.322 ± 0.028
	MACCEPA	-0.030	-0.030	0.286	-0.030 ± 0.000

(a) Error in final weights and cost accumulation under expert's true cost.



(b) Model. (c) Learning curves.



(d) Human Simulator. (e) MACCEPA Simulator.

Fig. 3. Simulation study. Shown are: (a) Error in final weights and cost accumulation under true (expert) weights for different simulators (mean $\pm$ s.d. over 50 trials), (b) Forward dynamics model of the human wrist used for MB-MWAL, (c) Learning curves: J_{MWAL} (top) and error in weights (bottom) over iterations of MWAL. Optimal trajectories with respect to the learnt cost functions under (d) the simulated human dynamics, and (e) MACCEPA dynamics. Features to note in (d) and (e) are (i) the hitting error (top row), (ii) the impact velocity (bar chart in second row) and (iii) the integrated effort (represented by the area of the shaded circles, bottom row).

the model-free approach achieved similar accuracy as that of MB-MWAL with the correct model on average (note that the variance arises from the variation in ${}^aD$ across trials).

In terms of behavioural predictions, MF-MWAL and MB-MWAL with the exact model accurately predict the demonstrator behaviour (see Fig. 3(d)), since the trade-off between accuracy, effort and impact velocity preferred by the demonstrator is captured well in the estimated weights $\hat{\mathbf{w}}$. In contrast, when the erroneous dynamics model is used, a different trade-off is learnt, in which accuracy concerns are compromised (ref. Fig. 3(d), 'hitting error') in favour of achieving higher velocity at T (ref. Fig. 3(d), middle panel) at similar effort (represented as the area of the circles in Fig. 3(d), bottom panel). These

[2]Note that, the ϵ-absolute value is used in the place of absolute or squared cost terms to avoid difficulties with outliers. It is defined as

$$|x|_\epsilon = \begin{cases} ax^4 + bx^2, & \text{if} |x| < \epsilon, \\ |x| + c, & \text{otherwise}, \end{cases}$$

where a, b and c are uniquely determined as $a = -1/(8\epsilon^3), b = 3/(4\epsilon)$ and $c = -3\epsilon/8$ under the condition that the first and second derivatives ($\partial|x|_\epsilon/\partial x$ and $\partial^2|x|_\epsilon/\partial x^2$) exist.

[3]Note that, the ball is modelled as having finite diameter (see shaded region in Fig. 3(d) and (e)), so that even if q^* is not exactly achieved, hitting is deemed successful if the system enters this region.

[4]Since $\int_0^T |\tau|_\epsilon \, dt = \int_0^T |I\ddot{q}|_\epsilon \, dt$, the third term corresponds to the torque-constraint during movement, scaled by Z/I.

[5]$\Delta E(\hat{\mathbf{w}})$ denote the error change in the final iteration.

EMG Signals	w_1	w_2	w_3
(u_1, u_2)	0.49 ± 0.03	0.01 ± 0.02	0.50 ± 0.03
(u_2, u_3)	0.49 ± 0.04	0.01 ± 0.02	0.50 ± 0.04
(u_1, u_3)	0.49 ± 0.03	0.01 ± 0.01	0.51 ± 0.04
(u_1, u_2, u_3)	0.49 ± 0.02	0.01 ± 0.02	0.50 ± 0.03
(u_1)	0.59 ± 0.20	0.15 ± 0.27	0.26 ± 0.12
(u_2)	0.58 ± 0.25	0.23 ± 0.30	0.20 ± 0.12
(u_3)	0.51 ± 0.01	0.02 ± 0.02	0.47 ± 0.04

TABLE II
COST WEIGHTS, EACH OF WHICH IS LEARNT BY MODEL-FREE AL WITH EMG SIGNALS AND THEIR COMBINATIONS. SHOWN ARE (MEAN VALUE ± S.D.) OVER 10 TRIALS.

differences in priorities are also reflected after transfer to the robot imitator (Fig. 3(e)), where again, the behaviour learnt with the erroneous model compromises on accuracy in favour of end-time velocity.

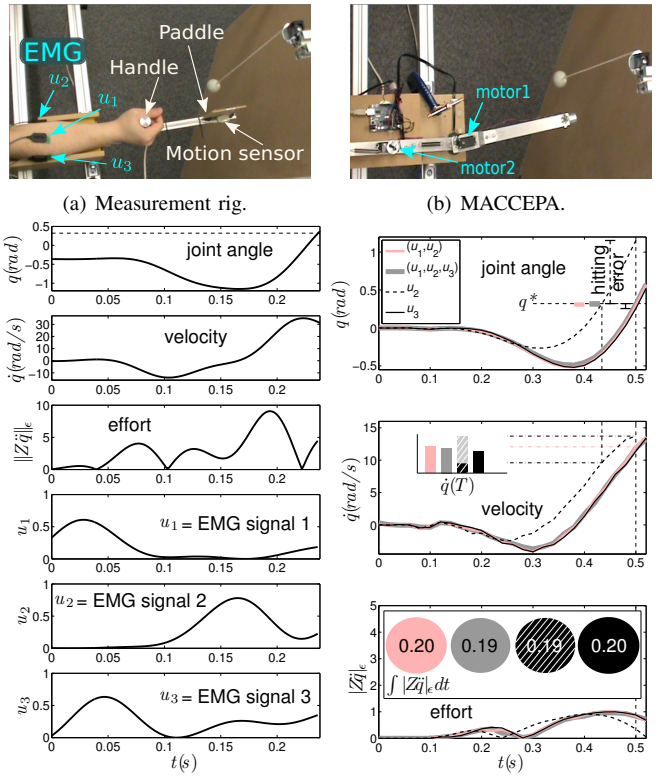
Finally, if we look at the accumulated cost of the trajectories learnt with the different approaches (Fig. 3(a)) (evaluated under the demonstrator's original cost function), the same pattern emerges. The behaviour learnt with the correct model, and the model-free approach incur similar costs as that of the expert. However, the behaviour learnt with the erroneous dynamics model incurs much higher cost, since it optimises for the wrong trade-off in (17).

B. Experiment

In this experiment, we illustrate the feasibility of our approach for learning from real human demonstrations and transferring behaviour to a robotic system in hardware. The goal is to investigate the robustness of our approach in a setting where it would be difficult to apply a model-based approach due to measurement errors.

We again focus on a ball hitting task using wrist-action. For collecting demonstrations, the measurement rig shown in Fig. 4(a) is used. The rig consists of a hinge joint with a paddle attached, that is aligned to a ball suspended from a string. The rig has a handle which the demonstrator grasps to rotate the joint and hit the ball with the paddle. A magnetic motion sensor (Flock of Birds, Ascension Tech. Corp.) is used to measure the angle of the demonstrator's wrist (hinge angle) at a 500Hz sampling rate. Simultaneously, surface EMG sensors (Bagnoli-8, Delsys), placed on the forearm measure the muscle activations of the demonstrator. With this setup, we are able to measure trajectories of the human through state- (modelled as ${}^e\mathbf{x} = (q, \dot{q})^\top$, the instantaneous wrist angle and velocity) and action-space (modelled as the muscle activations ${}^e\mathbf{u}$).

One of the difficulties of applying a model-based approach in this experiment, is the accurate placement of EMG sensors on the subject in a way that is consistent with the model. For example, if a model such as that illustrated in Fig. 3(b) is used, one must carefully align the EMG sensors on appropriate muscles of the subject to ensure correspondence with those in the model. However, with a model-free approach, such careful alignment is not necessary, since the wrist dynamics are *implicitly captured* in the auxiliary data. To investigate this, we use three EMG sensors casually placed at different points on the arm (ref. Fig. 4(a)), and then look at learning performance



(a) Measurement rig. (b) MACCEPA.

(c) Human data profiles. (d) Trajectories on MACCEPA.

Fig. 4. Transferring human skills. Shown are (a) Apparatus for recording human demonstrations of the hitting task, (b) MACCEPA arm which imitates the human hitting motion, (c) Human data profiles, where three EMG signals (u_1, u_2, u_3) are collected from different sensors in a single human motion, (d) Trajectories on MACCEPA, each of which is obtained by applying the motor command sequence optimised under learnt weights $\hat{\mathbf{w}}$. Features to note in (d) are (i) the hitting error (top row), (ii) the impact velocity (bar chart in second row) and (iii) the integrated effort (represented by the area of the shaded circles, bottom row). The trajectory learnt with u_2 is significantly different to those learnt with u_3 and the EMG combinations.

using data from *single sensors* and *different combinations of sensors*, to verify if we get consistent results.

Demonstrations were collected of a human hitting the ball (suspended at $q^* = 18.4°$) with the paddle as hard as possible, given a fixed time duration ($T = 0.24\,s$) in which to complete the movement. To reduce the effects of noise and variability in the demonstrations, 3 trajectories from each of 5 start positions $q = \{10, 0, -10, -20, -30\}°$) were collected, and the data was preprocessed by (i) smoothing the signals with a Butterworth filter and (ii) temporal alignment of trajectories around the time of impact T. The trajectories from each start state were then averaged, and the resultant $K = 5$ mean trajectories used as training data ${}^eD$. Additionally, $15,000$ auxiliary data samples ${}^aD$ were collected in short bursts of random left-right wrist movements, and subsampled into sets of 900 training and 200 test points with roughly uniform distribution in state-action space. The latter were used for cross-validation: the value function approximation was evaluated by the mean squared TD error at the test points and, if this became large, the subsampling was repeated and the forward optimisation and value estimation retried.

For estimating the human objective, we again modelled the cost function with (17), and sought the best estimate of the weights $\hat{\mathbf{w}}$. Note that, in this experiment, since the true human cost function is unknown, we cannot explicitly calculate the error in the estimated weights. Instead, convergence was measured by examining the magnitude of the weight update, and the consistency of the weight prediction between different learning runs (different subsamples and combinations of EMG sensors).

Table II gives the weights learnt using the different combinations of EMG signals. As can be seen, for all combinations (u_1, u_2), (u_2, u_3), (u_1, u_3) and (u_1, u_2, u_3), the learnt weights are approximately similar, which suggests that, given a wide enough coverage of the muscle signals, our approach is largely unaffected by sensor placement.

Examining the weights learnt with a single sensor (u_1, u_2 or u_3), we see that those learnt with u_1 or u_2 have large variance, and their average is quite different from those of the combined measurements: the weight on the velocity term (w_2) is 10 times increased, and those pertaining to the accuracy and effort terms (w_1 and w_3) are also different. It appears that the data from the single sensor does not give sufficient representation of the underlying control strategy used by the human (e.g., information about the action related to the flexor is not captured in the u_1 signal and vice versa, see Fig. 4(c)). Interestingly, the weights learnt with u_3 are similar to those with the combinations, since both features of u_1 and u_2 are roughly represented in u_3 (see Fig. 4(c)), which seems to be a sufficient representation for this task. Overall, the results confirm that a consistent model of the human's cost function can be obtained without the need for careful sensor placement, provided they give sufficient coverage of the arm.

Finally, to evaluate our approach for behaviour transfer, we used ILQG to find the optimal controller for the MACCEPA with respect to the cost function (17) using the weights learnt under the different combinations of sensors. The results are depicted in Fig. 4(d). For the cases where the weight predictions were consistent, we also see consistency in the behaviour transfer in terms of the trade-off between accuracy, impact velocity and effort. In contrast, however, the behaviour learnt with u_2 made impact at earlier time with lower velocity. This indicates that, while the exact placement of sensors is unimportant with the model-free approach, it is still necessary to take some care in placing the sensors in order to ensure appropriate coverage of the arm muscle signals and accurately capture the demonstrated behaviour.

V. Conclusion and Future Work

We presented a model-free approach to apprenticeship learning (AL) that enables transfer of task-oriented skills from humans to robots. Our approach is strongly motivated by the application of AL techniques to learning from human behaviour: while previous model-based approaches have relied on the ability to model the dynamics of the expert, for the complex human musculo-skeletal system this is infeasible, especially in the face of measurement errors (e.g., due to inconsistent placement of sensors). In response to this, we have derived a novel formulation of LSPI and LSTD reinforcement learning methods in finite horizon, such that they can be effectively applied in an AL framework. Simulation and experiment show the effectiveness and robustness of our approach for transfer of hitting behaviour from human recordings to a non-biomorphic variable impedance robot.

In future work, we plan to scale our method up to higher dimensional systems, such as transferring human punching movements to a 2-link MACCEPA system. Other directions of investigation will also include methods for automatic construction of the basis functions of the cost model.

Acknowledgement

This work was funded by the EU Seventh Framework Programme (FP7) as part of the STIFF project and a RAEng-Microsoft Research Fellowship Chair to SV. We thank Jun Nakanishi, David Braun and Konrad Rawlik for useful feedback.

References

[1] P. Abbeel, D. Dolgov, A. Ng, and S. Thrun. Apprenticeship learning for motion planning with application to parking lot navigation. *IROS*, 2008.

[2] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *ICML*, 2004.

[3] A. Alissandrakis, C. Nehaniv, and K. Dautenhahn. Correspondence mapping induced state and action metrics for robotic imitation. *IEEE Trans. Sys., Man, Cybernetics (B)*, 37(2):299–307, 2007.

[4] C. Atkeson and S. Schaal. Robot learning from demonstration. *ICML*, 1997.

[5] A. Boularias, J. Kober, and J. Peters. Relative entropy inverse reinforcement learning. *AISTATS*, 2011.

[6] J. A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49:233–246, 2002.

[7] S. Bradtke and A. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(2):33–57, 1996.

[8] D. J. Braun, M. Howard, and S. Vijayakumar. Exploiting variable stiffness in explosive movement tasks. *RSS*, 2011.

[9] A. Coates, P. Abbeel, and A. Ng. Apprenticeship learning for helicopter control. *Comm. ACM*, 52(7):97–105, 2009.

[10] P. W. Glynn. Importance sampling for markov chains: asymptotics for the variance. *Communication Statistics - Stochastic Models*, 10:701–717, 1994.

[11] R. V. Ham, B. Vanderborght, B. Verrelst, M. V. Damme, and D. Lefeber. Maccepa: the mechanically adjustable compliance and controllable equilibrium position actuator used in the 'controlled passive walking' biped veronica. *Robotics and Autonomous Systems*, 55:761–768, 2007.

[12] M. Howard, D. Mitrovic, and S. Vijayakumar. Transferring impedance control strategies between heterogeneous systems via apprenticeship learning. *Humanoids*, 2010.

[13] M. Katayama and M. Kawato. Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. *Biological Cybernetics*, 69:353–362, 1993.

[14] M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *J. Machine Learning Research*, 4:1107–1149, 2003.

[15] W. Li and E. Todorov. Iterative linearlization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):1439–1453, 2007.

[16] S. Mahadevan and M. Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *J. Machine Learning Research*, 8:2169–2231, 2007.

[17] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. *ICML*, 2000.

[18] B. Price and C. Boutilier. A bayesian approach to imitation in reinforcement learning. *IJCAI*, 2003.

[19] U. Syed and R. Schapire. A game-theoretic approach to apprenticeship learning. *NIPS*, 2008.