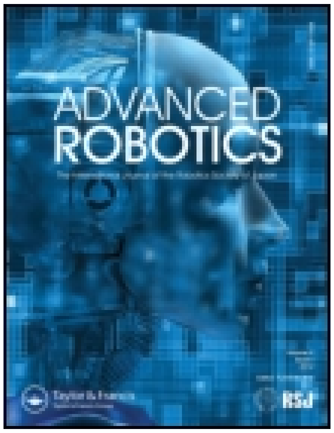

Advanced Robotics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tadr20

# Behavior sequencing based on demonstrations: a case of a humanoid opening a door while walking

Miguel González-Fierro[a], Daniel Hernández-García[a], Thrishantha Nanayakkara[b] & Carlos Balaguer[b]

[a] Robotics Lab, Department of System and Automation, Universidad Carlos III de Madrid, Madrid, Spain.

[b] Center for Robotics Research, Department of Informatics, King's College London, London, UK.

Published online: 17 Feb 2015.

*Advanced Robotics*, 2015
Vol. 29, No. 5, 315–329, http://dx.doi.org/10.1080/01691864.2014.992955

Taylor & Francis
Taylor & Francis Group

FULL PAPER

# Behavior sequencing based on demonstrations: a case of a humanoid opening a door while walking

Miguel González-Fierro[a*], Daniel Hernández-García[a], Thrishantha Nanayakkara[b] and Carlos Balaguer[b]

[a] *Robotics Lab, Department of System and Automation, Universidad Carlos III de Madrid, Madrid, Spain;* [b] *Center for Robotics Research, Department of Informatics, King's College London, London, UK*

(*Received 7 March 2014; revised 30 October 2014; accepted 17 November 2014*)

There is neuroscientific evidence to suggest that imitation between humans is goal-directed. Therefore, when performing multiple tasks, we internally define an unknown optimal policy to satisfy multiple goals. This work presents a method to transfer a complex behavior composed by a sequence of multiple tasks from a human demonstrator to a humanoid robot. We defined a multi-objective reward function as a measurement of the goal optimality for both human and robot, which is defined in each subtask of the global behavior. We optimize a sequential policy to generate whole-body movements for the robot that produces a reward profile which is compared and matched with the human reward profile, producing an imitative behavior. Furthermore, we can search in the proximity of the solution space to improve the reward profile and innovate a new solution, which is more beneficial for the humanoid. Experiments were carried out in a real humanoid robot.

**Keywords:** Learning from demonstration; humanoid robot; skill innovation; postural control

## 1. Introduction

When a human performs a high-level task like 'open the door and leave the room,' there are a sequence of behaviors that takes place to optimally perform the task. Like approaching the door in a manner that the location of the body makes it easier to reach the knob, grasping the knob, performing the movement that activates the mechanism of opening the door, going backwards while holding the knob, detecting that the door is open in a way that it can be overpassed, and finally, going through the opened doorway. All these behaviors are automatically selected to optimize, in some manner, the high-level strategy of performing this task. Figure 1 shows a detail of the high-level task of opening the door.

Recent neuroscientists studies suggest that when a human reproduces a learned task, he understands the consequences of this behavior and try to emulate the overall goal [1]. Even recent studies demonstrate that the main difference between apes and humans is our capability to *over-imitate*, or find newer and better solutions to accomplish optimal actions [2,3]. In that sense, innovation is an essential feature of the human behavior.

Minsky suggested that the way to create a machine that imitates the human behavior is not by constructing a unified compact theory of artificial intelligence [4]. On the contrary, he argues that our brain contains resources that compete between each other to satisfy different goals at the same moment. A similar view is shared by [5,6]. Starting from that idea, our approach is based on computing different reward profiles for different behaviors, which sequentially optimizes different goals.

Robots need to be able to handle similar situations, finding an optimal way to successfully complete these tasks, while maintaining the balance and moving in a safe and smooth manner. In recent years, researchers have taken a significant effort to cope with this problem and Learning from Demonstration (LfD) [7–13] has became one of the most popular ways to create motor skills in a robot. One of the key questions to be solved is *what to imitate* [7,14].

In this paper, we present a sequential method to learn concurrent behaviors from a human demonstrator, adapt them to the robot embodiment, and refine these behaviors to successfully accomplish the desired task.

We collected data from several human demonstrators performing a complex task composed by a set of sequential behaviors. Extracting determined features of every behavior, like Center of Mass (COM) position, human orientation, hand trajectory, etc., and encoding them using Gaussian Mixture Models (GMM), we define a multi-objective reward function identified as the overall goal, which is used as a basis of comparison between the human and the robot. The reward is used to solve the *correspondence problem*, which

[*] Corresponding author. Email: mgpalaci@ing.uc3m.es

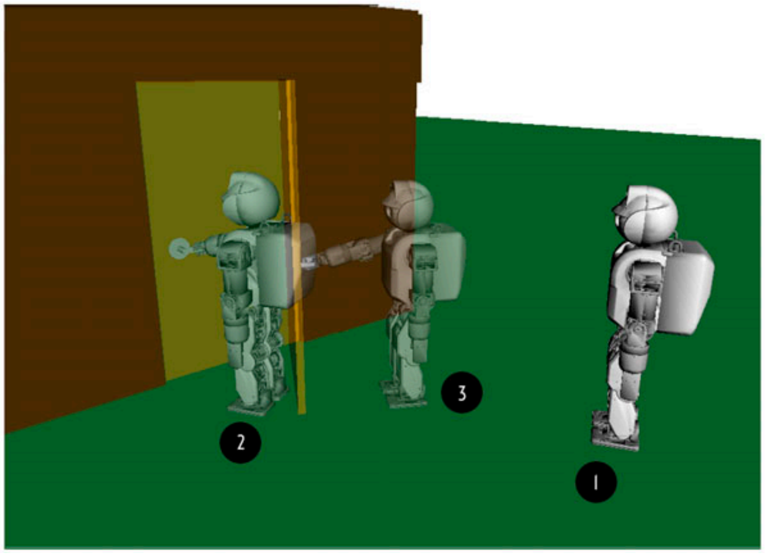

Figure 1. Behavior sequence detail of the high-level task of opening a door by a simulated HOAP-3 robot. The robot starts at point 1, it approaches to point 2 near the door in a way it can reach the knob, after grasping the door, it pulls back the door to point 3. Finally, it releases the knob.

is defined as the action mapping between the demonstrator and the imitator [14]. In this regard, we mapped movements performed in a different kinematic domain and at a different scale to a common domain, defined as the goal domain and expressed mathematically as a reward profile, formed by a multimodal landscape of movement features.

In a previous work, we addressed the problem of mapping a behavior from a group of unexperienced workers to match and even surpass the expert behavior of an elite individual [15]. Using that idea, we proposed a method of imitation learning of a single behavior in a small humanoid robot using the reward as a common space of comparison [16] and later, we improved that idea by making a robot imitate a single human behavior and also innovate a new one which better fits its internal constraints and kinematic structure [17]. Starting from there, we extend the work by proposing in this paper a new human–robot LfD framework where a complex sequence of behaviors, which involves manipulation and locomotion, takes place.

We define a sequential policy for the robot that allows to find in which behavior the robot is and computes a constrained whole-body movement pattern that optimizes the reward in order to be as close as possible to the human's reward. Then, we refine the policy by innovating a new solution which improves the current robot reward.

### *1.1. Overview of the method*

Figures 2 and 3 show the complete architecture of both sequential imitation learning and sequential innovation learning. There are two optimizations in every architecture. A local optimization between behavior episodes and a global optimization of the complete behavior. Therefore, the system not only obtain a local stable movement but it takes into consideration the complete shape of the action movement.

Figure 2 shows the imitation learning process. The human data are acquired using a MOCAP system, which in our experiments is a Kinect camera. These data are used to obtain a model of the human, which generates a joint trajectory $q_i$. This trajectory is used to compute the behavior selector matrix and the human reward profile. The behavior selector matrix indicates the probability distribution of being in a determined behavior given a state. The human reward is compared globally and locally with the robot reward.

The robot imitation process begins by knowing its initial joint values $q_i(0)$. At this point, a new episode $e_i$ begins. An episode is a transition between a pair of initial and final states $\mathbf{X} = (\xi_{\text{ini}}, \xi_{\text{final}})$, which depends on the current behavior $b_i$, the generated trajectory $q_i$, the controller PD and the episodic reward $r_{\text{Rep}}$. Then a local optimization takes place.

The robot reward $r_{\text{Rep}}$ is compared with the human reward $r_{\text{Hep}}$, when its difference $\Delta_1$ is a small number or the maximum number of iterations have passed, the robot satisfactorily imitates the human and the reward candidate for this episode is saved. This process is repeated until all episodes have been computed.

When this loop finishes it means that all episodes for all behaviors have been computed and a candidate complete movement is available. At this point, a global optimization process takes place to minimize the difference between the total robot reward $r_{\text{RTOT}}$ and the total human reward $r_{\text{HTOT}}$,

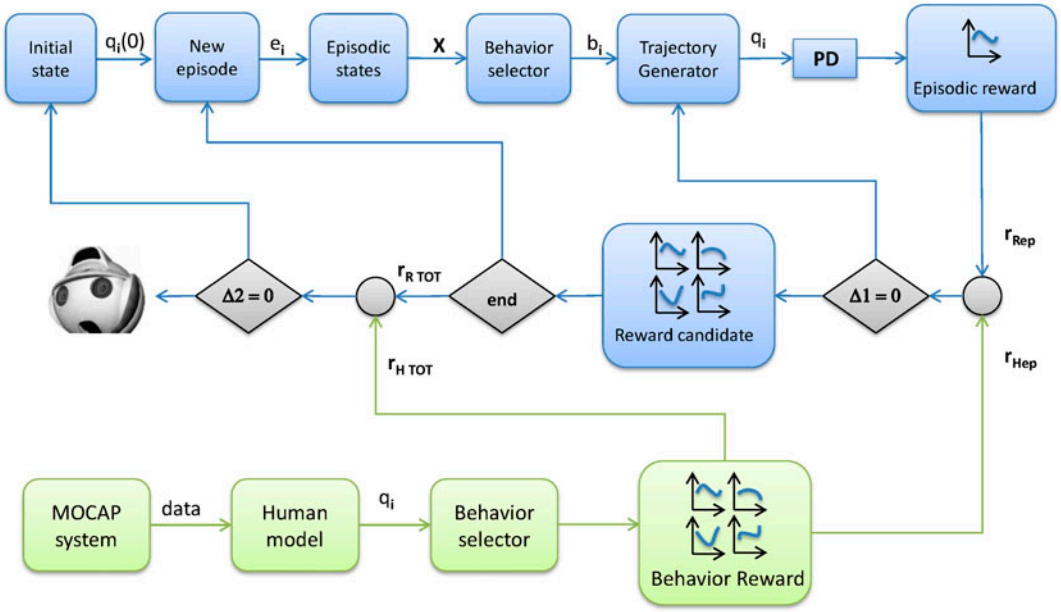

Figure 2. Overview of the imitation system. Using a MOCAP system, the movement of the human demonstrator is obtained and a reward profile for every behavior is computed. On the other hand, the robot starts in an initial state $q_i(0)$. A new episode $e_i$ is defined, which is a pair of initial and final states $\mathbf{X} = (\xi_{\text{ini}}, \xi_{\text{final}})$. Then the behavior selector decides in which behavior the humanoid is. The trajectory generator produces a stable trajectory within the pair of states. The episodic reward is optimized until the difference $\Delta_1$ between the robot reward $r_{\text{Rep}}$ and the human reward $r_{\text{Hep}}$ is small or it reaches a number of iterations. This process is repeated until all behaviors are completed. Then, there is a comparison between the complete reward profile of the robot $r_{\text{RTOT}}$ and the complete reward profile of the human $r_{\text{HTOT}}$, which is the index $\Delta_2 = J$. If this index is close to zero, it means that the imitation is completed.

denoted by $\Delta_2$. If they are similar, the process stops and we conclude that the imitation process is not only successfully achieved locally but globally, taking into account the complete movement.

Figure 3 shows the innovation learning process. It is very similar to the imitation process but this time, instead of comparing with the human reward, it compares with the best reward of the imitations process. Therefore, in this case,

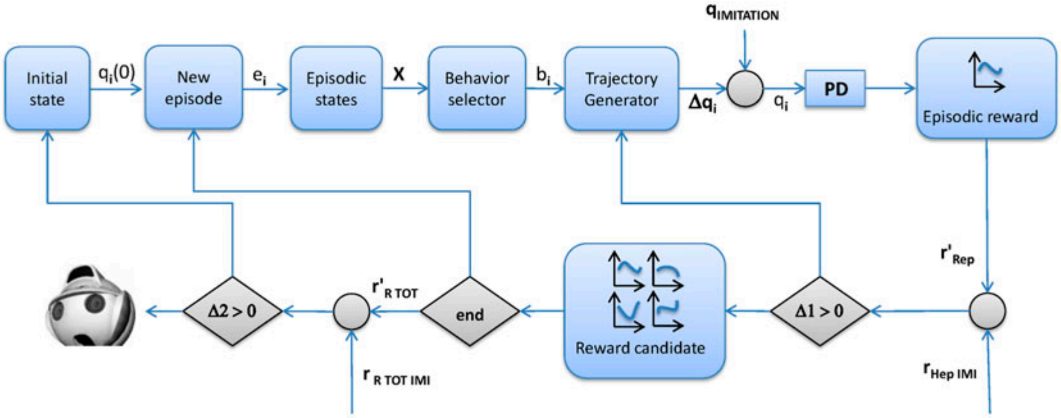

Figure 3. Overview of the innovation system. The process is very similar to Figure 2. The main difference appears in the trajectory generation. The generator perturbs the imitation trajectory $q_{\text{IMITATION}}$ in an amount $\Delta q_i$, to generate a new trajectory $q_i$ which is evaluated in terms of the episodic reward. The other difference is in the reward comparison $\Delta_1$ and $\Delta_2$. In this case, the objective is to maximize the difference. If the robot gets a better episodic reward $r'_{Rep}$ than the reward obtained in the imitation $r_{\text{RepIMI}}$, then $\Delta_1 > 0$ and the local optimization ends. If the robot gets a better reward $r'_{\text{RTOT}}$ than the reward obtained in the imitation $r_{\text{RTOTimi}}$, then $\Delta_2 > 0$ and the innovation is completed.

the robot is not just imitating the human but generating an innovative behavior which is better, since the performance can be measurement with the reward profile.

The document is ordered as follows. In Section 2, the sequential policy search method is presented with a brief overview of GMM and Gaussian Mixture Regression (GMR) and the explanation of the postural primitives used to generate the whole body motion. In Section 3, the behavior acquisition and transfer is studied, then the experiments with the real robot are defined, implemented, and discussed. In 4, the related work is discussed. Finally, in Section 5, the conclusions are presented.

## 2. Sequential policy definition

A learning process that considers the Markov property to predict actions and states is called a Markov Decision Process (MDP) [18].

The learning process is defined as a sequence of finite states $s \in S$ and actions $a \in A$ pairs that produce an associate reward $r \in R$. The agent, starting from a state $s(t)$ will compute an action $a(t)$ to reach a future state $s(t+1)$, obtaining a reward $r(t)$, which can be defined as a set of values or as a mathematical function, it is usually called *the reward function*.

Let $b \in B$ be a set of behaviors that compose the full high-level strategy of performing a task. An example of behavior can be approaching the door in a manner that the location of the body allow to reach the knob, grasping the knob, performing the movement that activate the mechanism to open the door, going backwards while holding the knob, realizing that the door is open in a way that it can be overpassed, and finally, passing through the doorway.

The goal is to determine a policy $\pi(a|s)$ in the form

$$\pi(a|s) = \sum_{b} \pi(a|s,b)\pi(b|s) \tag{1}$$

where $\pi(b|s)$ is the selector of behavior $b$ given a state $s$, and the policy $\pi(a|s,b)$ to select the action $a$, given a behavior $b$.

We consider an episodic learning strategy to generate a policy inside every behavior. At the beginning of the episode, starting from a state $s$, we compute a parameterized postural primitive that takes into account the whole body movement, while maintaining the stability. The parameterized postural primitive can be defined in several ways, in some works like [19], the movement is computed as a dynamic movement primitive [20]. For more complex trajectories that implies displacement and manipulation at the same time, it is easier to define trajectories in the task space [21].

In each episode, we consider an action $a$ that determines the parameters of the postural primitive, which for instance defines the movement plan for the complete episode. The states are defined as the via points of the primitive and the reward profile is computed from the reward function $r^{\pi}(s,a,b)$. The reward is constructed as a metrics to measure the overall goal performance and it depends on the behavior, the initial state of the episode and the action that takes place in this episode.

Figure 4 represents a diagram of an episode. It shows how situations branch off to behaviors and then to actions. Given a situation in the state space, there can be many behaviors according to human demonstrations. Given a behavior, the action to change over time. We choose a branch in the tree using a probability distribution derived from the demonstrations.

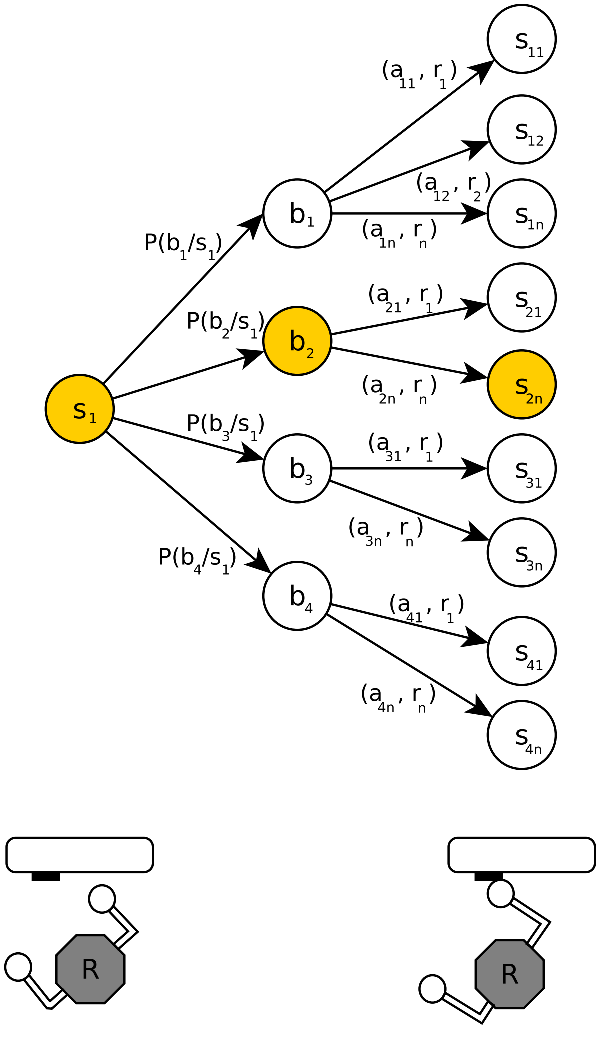

Figure 4. Diagram explaining one episode. The robot, represented in the lower part of the diagram, performs a transition from state $s_1$ to $s_2$. In the upper part of the diagram, there is a tree representing the complete process. Given a state $s_1$, the behavior selector $\pi(b|s)$ computes the probability of being in a behavior $P(b_i/s_1)$. Then $\pi(a|s,b)$ generates an action $a_{ij}$, which retrieves a reward $r_j$. The generated action takes the robot to a state $s_{ij}$. The selection of one branch, in yellow, is determined by the most probable behavior and by the best reward.

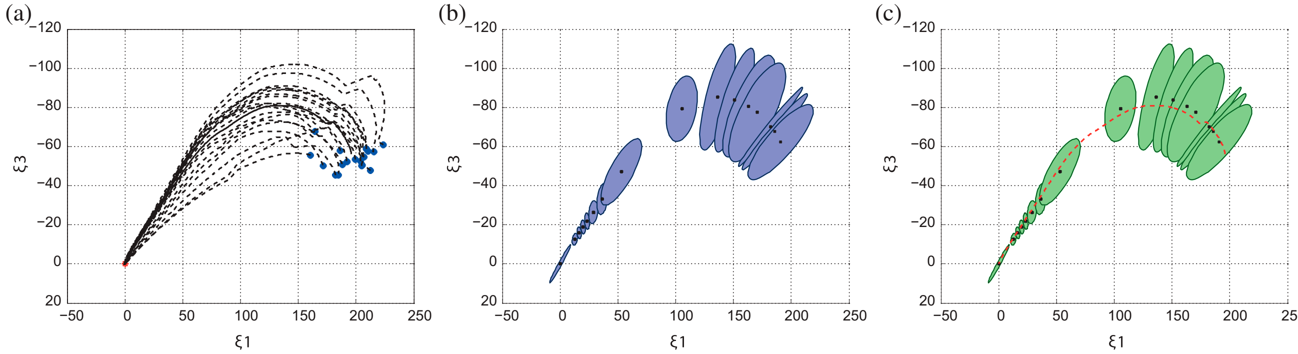

Figure 5. Illustration of the learning process of grasping a door knob from a top view. (a) Training data of the task. (b) GMM of the learned motion. (c) Reproduction of the GMR.

### 2.1. *Encoding and generalizing demonstrations*

The probability distribution space of the human demonstrations is approximated using GMM. A time independent model of the motion dynamics is estimated through a set of first-order non-linear multivariate dynamical systems. [22] propose an approach to imitation learning, and online trajectory modification, by representing movement plans based on a set of non-linear differential equations with well-defined attractor dynamics. We follow a framework presented on [8] allowing learning non-linear dynamics of motions and generating dynamical laws for control.

A variable $\xi$ is defined describing the state of the robot. Let the set **M** of N-dimensional demonstrate data points $\{\xi_i, \dot{\xi}_i\}_{i=0}^{M}$ be instances of a global motion governed by a first-order autonomous ordinary differential equation:

$$\dot{\xi}(t)^M = f(\xi(t)^M), \tag{2}$$

where $\xi^M \in R^n$, and its time derivative $\dot{\xi}^M \in R^n$ are vectors that describe the robot motion. The problem then consists in building a stable estimate $\hat{f}$ of $f$ based on the set of demonstrations.

To build the estimate $\hat{f}$ from the set of demonstrated data points $\{\xi_i, \dot{\xi}_i\}_{i=0}^{M}$, we follow a statistical approach and define $\hat{f}$ through a Gaussian Mixture Model.

#### 2.1.1. *Gaussian mixture models*

The GMMs define a probability distribution $p(\xi^i, \dot{\xi}^i)$ of the training set of demonstrated trajectories as a mixture of the ***K*** Gaussian multivariate distributions $\mathbf{N^k}$

$$p(\xi^i, \dot{\xi}^i) = \frac{1}{K}\sum_{k=1}^{K} \pi^k N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) \tag{3}$$

where $\pi^k$ is the prior probability; $\mu^k = \{\mu^k_{\xi}; \mu^k_{\dot{\xi}}\}$ is the mean value; and $\Sigma^k = \begin{bmatrix} \Sigma^k_{\xi} & \Sigma^k_{\xi\dot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} \end{bmatrix}$ is the covariance matrix of a Gaussian distribution $\mathbf{N^k}$.

The probability density function of the model $N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k)$ is then given by:

$$N^k(\xi^i, \dot{\xi}^i; \mu^k, \Sigma^k) = \frac{1}{\sqrt{(2\pi)^{2d}|\Sigma^k|}} e^{\frac{-1}{2}([\xi^i, \dot{\xi}^i] - \mu^k)^T (\Sigma^k)^{-1} ([\xi^i, \dot{\xi}^i] - \mu^k)} \tag{4}$$

By considering an adequate number of Guassians, and adjusting their means and covariance matrix parameters, almost any continuous density can be approximated to arbitrary accuracy. The form of the Gaussian mixture distribution is governed by the parameters $\pi^k$, $\mu^k$, $\Sigma^k$. The model is initialized using the k-means clustering algorithm starting from a uniform mesh and is refined iteratively through *Expectation-Maximization* for finding the maximum likelihood function of (3).

$$\ln p(\xi^i, \dot{\xi}^i) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi^k N\left(\xi^i_n, \dot{\xi}^i_n | \mu^k, \Sigma^k\right) \right\} \tag{5}$$

Figure 5(a) illustrates the encoding of a training dataset $\{\xi_i, \dot{\xi}_i\}_{i=0}^{M}$ into a model of mixtures of Guassians, Figure 5(b). In this work, we used the Binary Merging (BM) algorithm, [23], to build the GMM. BM determines an optimal minimum number of Gaussian functions to employ, while satisfying the stability criteria and also keeping the error of the estimates under a threshold. To generate a new trajectory from the GMM, one then can sample from the probability distribution function $p(\xi^i, \dot{\xi}^i)$, this process is called GMR.

#### 2.1.2. *Gaussian mixture regression*

The GMM computes a joint probability density function for the input and the output so that the probability of the output conditioned on the input is a Mixture of Gaussians. So it is possible after training, to recover the expected output variable $\hat{\dot{\xi}}$, given the observed input $\xi$. Taking the conditional mean estimate of $p(\dot{\xi}|\xi)$, the estimate of our function $\hat{\dot{\xi}} = \hat{f}(\xi)$ can be expressed as a non-linear sum of linear dynamical systems, given by:

$$\hat{\dot{\xi}} = \sum_{k=1}^{K} h_k(\xi) \left( \Sigma^k_{\dot{\xi}\xi} \left( \Sigma^k_{\xi} \right)^{-1} \left( \xi - \mu^k_{\xi} \right) + \mu^k_{\dot{\xi}} \right) \tag{6}$$

where

$$h_k(\xi) = \frac{p(\xi; \mu^k_{\xi}, \Sigma^k_{\xi})}{\sum_{k=1}^{K} P(\xi; \mu^k_{\xi}, \Sigma^k_{\xi})}, \quad h_k(\xi) > 0 \tag{7}$$

and $\sum_{k=1}^{K} h_k(\xi) = 1$

Figure 5(c) illustrates the GMR as a reproduction of the learned motions. To learn the model of the trajectories, first several demonstrations of the task are presented and them the trajectory is encoded as a mixture of Gaussian distributions. To reproduce the trajectories, one sample from the probability distribution of the GMM trough the GMR process. The GMR approximates the dynamical systems through a non-linear weighted sum of local linear models.

### 2.2. *Parameterized postural primitives*

Based on the demonstrations encoded as GMM, we can compute a parameterized postural primitive for each episode, which defines a complete motion of the humanoid in the task space. To simplify the process of generating a whole body motion taking into account contacts and stability, we decouple the robot in two modules or tasks, the locomotion task and the grasping task. This means that when the robot is performing a locomotion task, the module in charge of computing the grasping task is stopped. In a similar way, when the robot is performing a grasping operation the locomotion module is stopped.

There is a moment when the robot is moving backwards and at the same time is grasping the knob. At this moment, the only active module is the locomotion one. The robot arm is idle to decouple the robot from the door dynamics. We assume that the door weight is small in comparison with the robot weight and the resistive torque of the hinge is negligible.

For the locomotion task of the humanoid, the postural primitive can easily be computed using the cart-table model [24]. This model is based on ZMP, a preview control scheme to obtain the COG trajectory from a defined ZMP trajectory. This method generates a dynamically stable gait trajectory using the Inverted Pendulum Model to approximate the dynamics of the humanoid.

The relationship between ZMP trajectory and COG trajectory is defined by the following equations:

$$p_x = x - \frac{\ddot{x}}{g} z_c \tag{8}$$

$$p_y = y - \frac{\ddot{y}}{g} z_c \tag{9}$$

where $p_x$ is the ZMP reference, $x$ is the COG trajectory, $\ddot{x}$ the COG acceleration, $z_c$ is the COG height, and $g$ is the gravity. In cart-table model (Figure 6), the cart mass corresponds to the center of mass of the robot. If the cart accelerates at a proper rate, the table can be upright for a while. At this moment, the moment around $p_x$ is equal to zero, so the ZMP exists.

$$\tau_{ZMP} = mg(x - p_x) - m\ddot{x}z_c = 0 \tag{10}$$

The solution of (8) and (9) produces the COM trajectory of the whole episode for the lower part of the robot's body.

Regarding the grasping, we can use GMR to define a desired trajectory for the hands and add a modulation term that improves the reward index, similarly to [9].

### 2.3. *Sequential policy search*

We define the sequential policy search problem as an optimization problem where we use the reward framework as a basis of comparison between the human and the robot. The objective is to find a policy for the robot that, in an initial moment, imitates the behavior of the human, by producing a similar reward profile, and later improve the robot performance, by auto exploring new solutions that return a better reward. Taking that into account, we can define an imitation index $J$, which is defined as the optimization problem of minimizing the episodic difference of the human and robot reward profile (11), and the innovation index $J'$, which is defined as the optimization problem of maximizing the positive difference between the episodic imitation reward profile and the new innovation profile (12). To compare

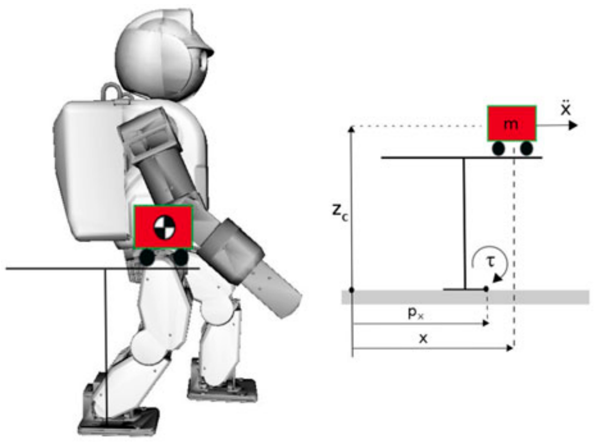

Figure 6. Cart-table model in sagittal plane.

between reward profiles, we make use of the Kullback–Liebler divergence, which can be stated as a directional information transfer.

$$\min J = \sum_b \sum_e r^h(s,a,b) \log \frac{r^h(s,a,b)}{r^r(s,a,b)} \tag{11}$$

where $e \in E$ is the episode, $r^h$ is the human reward profile, and $r^r$ is the robot reward profile.

$$\max J' = \sum_b \sum_e r_i^r(s,a,b) \log \frac{r_i^r(s,a,b)}{r^r(s,a,b)} \tag{12}$$

subject to

$$\mu_i^r(e) \geq \mu^r(e) \tag{13}$$

where $r_i^r$ is the innovation reward profile of the robot, $\mu^r(e)$ is the mean of the imitation reward profile in episode $e$, and $\mu_i^r(e)$ is the mean of the innovation reward profile in episode $e$. The optimization process is performed using Differential Evolution optimizer [25].

## 3. Experiments

The task chosen for testing our method is to make a humanoid robot approach a door, grasp the knob, and open the door while maintaining the balance. The robot used is the middle-size humanoid HOAP-3 of Fujitsu.

### 3.1. *Acquiring behaviors from human demonstrations*

The experimental setup consist of a Kinect camera recording nine human participants opening a door 10 times each (see Figure 7). The API of the Kinect allows to perform an accurate tracking of the human body, which is improved using a Kalman Filter.

The complete task is segmented into several behaviors $b \in B$. The first behavior $b_1$ consists on approaching the door to a place where the knob can be reached, then grasping the knob $b_2$, going backwards leaving the arm passive, but without releasing the knob $b_3$, and finally, releasing the knob $b_4$.

The selected states for the task are position and orientation of the COM, $\xi_{\text{com}} = \{x_{\text{com}}, y_{\text{com}}, \theta_{\text{com}}\}$ and the position of the grasping hand, $\xi_{\text{hand}} = \{x_{\text{hand}}, y_{\text{hand}}, z_{\text{hand}}\}$. All states are measured with respect to the Kinect position.

Let it be noted that the identification, and therefore, the segmentation, of a behavior depends on the perspective of the observer [4,6]. We divided the task of opening a door into four behaviors; however, another observer could define a different set of behaviors or it can be done techniques like in [26–28].

For each human demonstration, a temporal state trajectory $\xi = [\xi_{\text{com}}, \xi_{\text{hand}}]$ is obtained using the Kinect API. After a filtering, the trajectory is automatically classified into the four behaviors. For $b_1$, approaching the door, $\xi_{\text{com}}$ approaches to the door, whose position with respect to the Kinect reference system is known. In $b_2$, grasping the knob, $\xi_{\text{hand}}$ goes up until it touches the knob, whose position with respect to the Kinect reference system is also known. $b_3$ starts when the hand grasps the knob and $\xi_{\text{com}}$ moves backwards. Finally, in $b_4$, the hand release the knob and $\xi_{\text{hand}}$ goes down to a rest position.

Let it be noted that the demonstrations performed by all subjects are in some sense artificial. In order to make the automatic behavior segmentation easier, the subjects are told to perform each behavior separately, i.e. they first approach the door, then move his hand to grasp the knob, then pull the door, and finally, release the knob. A human opening a door in a real environment would perform several of these behaviors at the same time, smoothly and elegantly.

### 3.2. *Learning the behavior selector from human demonstrations*

By observing the human demonstrations, we can construct the behavior selector $\pi(b|s)$ in (1), by obtaining the probability of being in a determined behavior given a combination of states.

Figure 8 represents the mean and standard deviation of all human demonstrations segmented by behaviors.

In order to compute the behavior selector matrix of Figure 9, we first divide each state length into $z$ substates, where the length is $l_i = s_{i\,\max} - s_{i\,\min}$ and the step is $\Delta s_i = l_1/z$. Therefore, each state is composed of a number of substates $[s_{ia}, s_{ib}, s_{ic}, \ldots, s_{iz}]$. Next, for each human demonstration in each behavior, we do a mapping from trajectories to substates, obtaining the probability matrix of Figure 9.

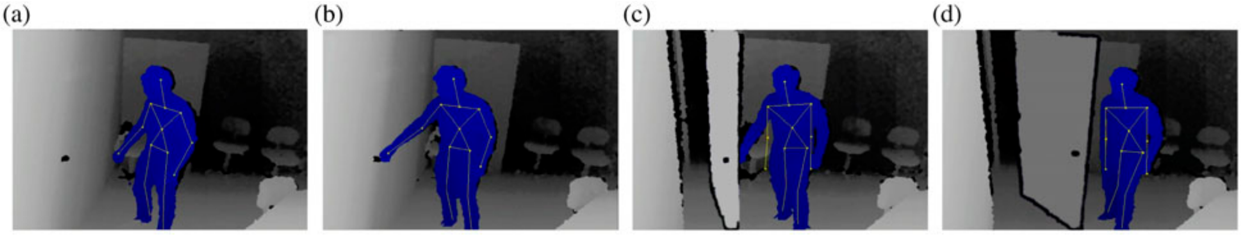

Figure 7. Snapshots of one human demonstrator performing the task of opening the door using the Kinect camera. Each snapshot corresponds to a different behavior. (a) Behavior $b_1$: approaching the door, (b) Behavior $b_2$: grasping the knob, (c) Behavior $b_3$: pulling back the door, and (d) Behavior $b_4$: releasing the knob.

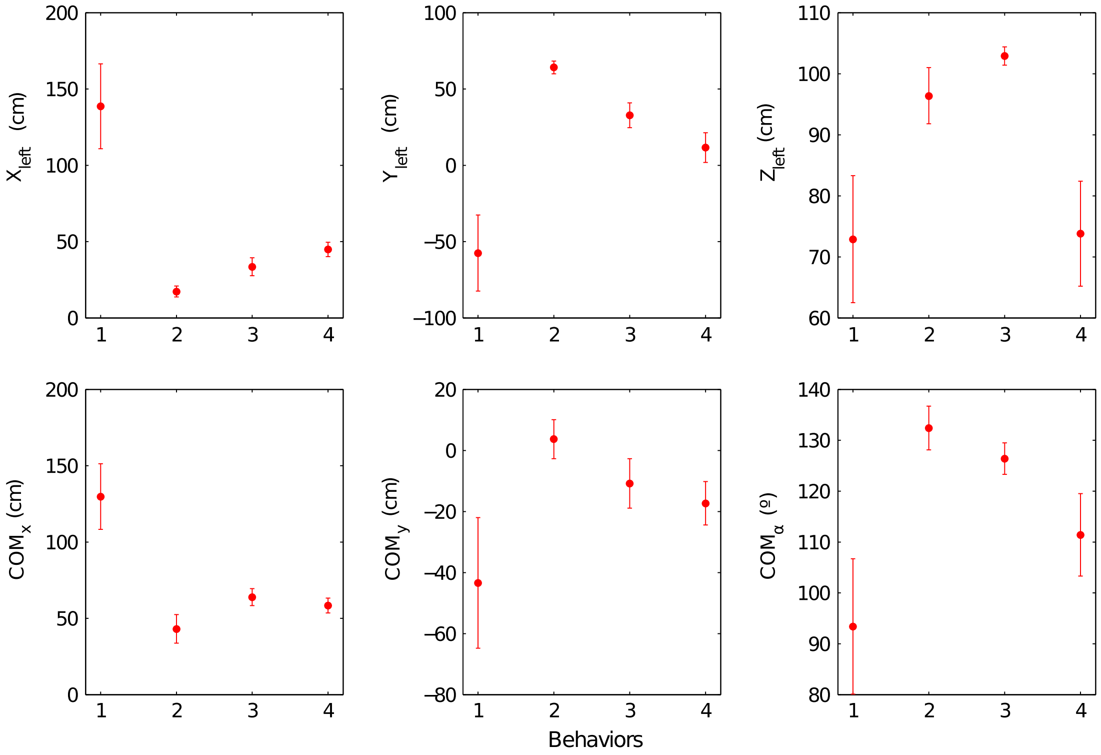

Figure 8. Mean and standard deviation of the human demonstrated states.

In order to compute the probability of being in behavior $b_i$ given a combination of states $\hat{s} = \{s_{1,a}, s_{2,b}, \ldots, s_{n,z}, \}$, where $n$ is the number of states and $a, b, \ldots, z$ corresponds to an arbitrary substate inside a state:

$$P(b_i|\hat{s}) = P(b_i|s_{1,a}).P(b_i|s_{2,b}) \ldots P(b_i|s_{n,z}) \qquad (14)$$

Finally, the selector of behavior can be stated as:

$$\pi(b|s) = b_i \text{ with } b_i = \text{argmax}_i(P(b_i|\hat{s})) \quad i \text{ from 1 to } m \qquad (15)$$

where $m$ is the total number of behaviors.

Once the behavior selector matrix is obtained, it can be used to predict the current robot behavior, given a combination of states. In the case of the robot, we applied a scale factor $\rho$ to obtain the length $l_i' = l_i/\rho$ and the step $\Delta s_i' = l_1'/z$.

### 3.3. *Definition of reward profile*

The reward function $r^\pi(s, a, b)$ varies depending on what behavior is being performed. Let define $d_i$ as the quadratic difference of the actual state $\xi_i$ and $\xi_i^*$, defined as the GMR of the human demonstrations (6) in the case of the human and an adapted trajectory for the robot based on the GMR human trajectory.

$$d_i = (\xi_i - \xi_i^*)^T W(\xi_i - \xi_i^*) \qquad (16)$$

with $W$ a weight matrix.

We also define the reward $i$ as a Cauchy distribution in the form

$$r_i = \frac{1}{\epsilon + d_i} \qquad (17)$$

with a small $\epsilon$.

Let be defined the reward for each behavior.

$$r^\pi(s, a, b_1) = \frac{1}{2}\sum_e r_{\text{com}} + r_{\text{door}} \qquad (18)$$

$$r^\pi(s, a, b_2) = \frac{1}{2}\sum_e r_{\text{hand}} + r_{\text{knob}} \qquad (19)$$

$$r^\pi(s, a, b_3) = \frac{1}{2}\sum_e r_{\text{com}} + \hat{r}_\alpha \qquad (20)$$

$$r^\pi(s, a, b_4) = \frac{1}{2}\sum_e r_{\text{hand}} + r_{\text{antiknob}} \qquad (21)$$

and

$$\hat{r}_\alpha = \frac{\alpha}{\alpha_{\max}} \qquad (22)$$

where $r_{\text{hand}}$ and $r_{\text{com}}$ represent the reward when the hand and COM trajectory of robot and human are close to the trajectory defined by the GMR of the human demonstrations. Both terms represent a direct imitative behavior. The closer the actual trajectory is to the desired trajectory, the higher the reward. The term $r_{\text{door}}$ is the reward obtained for locating in a point near the door where the knob can be reached, the closer the point the higher the reward. $r_{\text{knob}}$

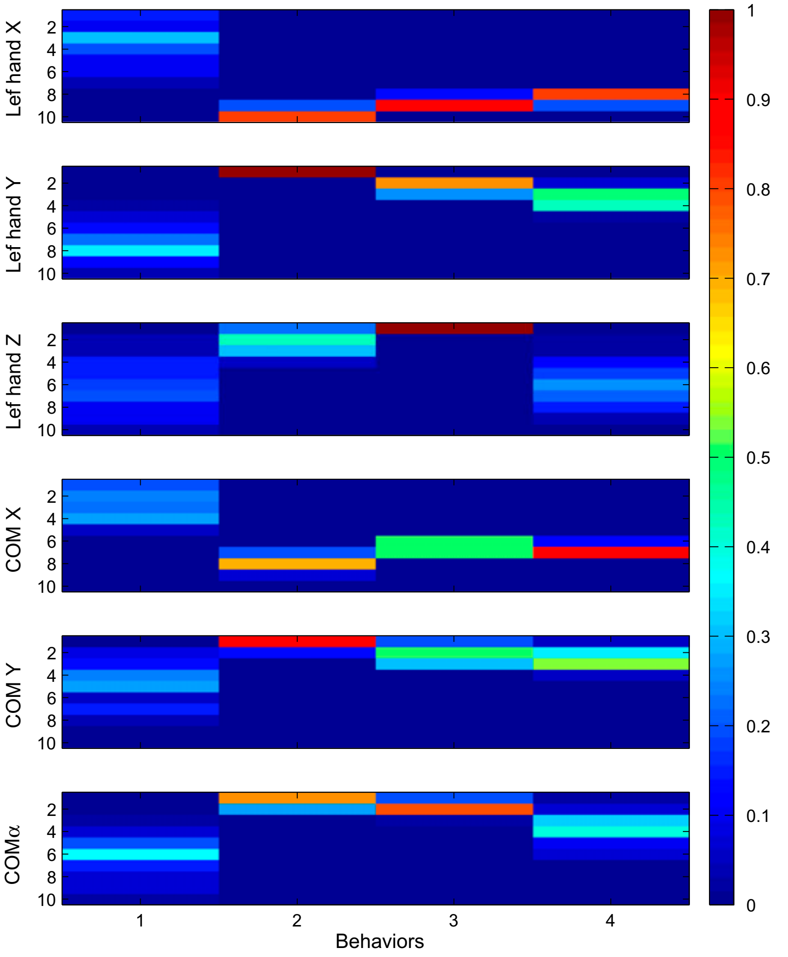

Figure 9. Behavior selector matrix. The columns represent the behaviors and the rows represent the substates in a state. For each substate, the color represents the probability of being in a behavior.

is the reward obtained by the difference between the hand and the knob position, the closer the hand to the knob, the higher the reward. The term $\hat{r}_\alpha$ represents the reward given for the achievement of the high-level task, which is to open the door. Finally, $r_{\text{antiknob}}$ is a reward that penalizes to have the hand close to the knob and follows a sigmoid function that starts on zero and finishes on 1. $\alpha_{\max}$ is the maximum angle that the door opens and $\alpha$ is the actual door angle computed as:

$$\alpha = \frac{1}{2}(\alpha_x + \alpha_y) \tag{23}$$

and

$$\alpha_x = \arccos \frac{x_{\text{hand}}}{l} \tag{24}$$

$$\alpha_y = \arcsin \frac{y_{\text{hand}}}{l} \tag{25}$$

where $l$is the door length.

The robot can find a way to obtain a better total reward than the human if it is able to improve $r_{\text{door}}$, $r_{\text{knob}}$, $\hat{r}_\alpha$, and $r_{\text{antiknob}}$. Those terms represent the possibility of innovation. Please note that all $r^\pi(s, a, b)$ functions have to be normalized so its integration sums to 1 in order to be used with the Kulback–Leibler distance in (11) and (12). In Figure 10, the resulting rewards are plotted.

### 3.3.1. *Discussion on the reward profile*

As [2] suggested, both children and chimpanzees try to emulate the goal of the action when imitating a behavior. Furthermore, some recent studies suggested that the main difference between humans and chimpanzees is the ability of over-imitation [3]. Our proposal of using a reward profile to solve the correspondence problem in order to transfer a complex behavior from a human to a humanoid is based on these previous neuroscience works and previous experiments performed in a real humanoid standing up from a chair [17]. However, we are not sure of what is the internal objective function that the brain is optimizing

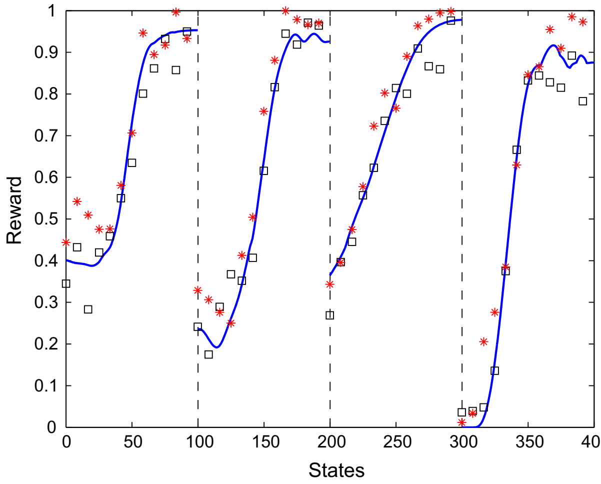

Figure 10. Reward profiles of the complete action of opening a door. The blue line represents the reward profile for the human demonstrator. Black boxes represent the means of the reward for the imitative behavior. The red crosses represent the means of the reward profile for the innovative behavior. The dotted vertical lines represents the changes between behaviors. The vertical axis represents the reward value and the horizontal axis represents the states in which is divided each behavior. As it can be appreciated in the figure, the imitative behavior produces rewards similar to the human's and the innovative behavior produces rewards slightly higher.

when performing a complex task sequence. We proposed a compound reward function that takes into account the position. Position in terms of closeness to the door, position in terms of distance from the hand to the door knob, position in terms of COM trajectory. However, the human brain may use also velocity, acceleration, jerk, or even other factors we are not taking into account.

Although at first sight, the proposed model of imitation and innovation may seem task dependent it is not. The generality comes from the definition of the reward profile. In fact, any behavior can be modeled, from simple ones as in [17] to complex behaviors. The preference in the selection of a predefined reward function over a learned function like in inverse reinforcement learning [29,30] does not affect the general idea of comparing the behavior of a human and a robot in a common domain, which is the reward domain.

Moreover, it can be noted from Figure 10 that the innovative process does not improve the performance radically. The importance of the innovative behavior is not in the improvement quantity. It lies in the fact that the reward profile represents the behavior goal and, at the same time, a metrics to measure its performance. Therefore, since it is a behavior metrics, we can generate different movements, not only imitating the human but innovating a new behavior, which is better than the behavior demonstrated by the human.

As it can be seen from Equations (18)–(21), the reward functions always have an imitative component and an innovative component. We made a mathematical framework where the two main components of learning are present, one is learning from others and the other is learning by self-exploration. The human behavior might be not optimal for specific tasks; however, it is undeniable that a human being is able to adapt to a huge range of situations and behave elegantly and efficiently. That is why we used LfD as a starting point, furthermore, it has some advantages such as the simplification of communicating a complex behavior through demonstrations, the absence of the need to have complex mathematical models of the dynamical system to learn an optimal behavior, and the fact that it does not require an expert teacher to perform the demonstrations, which simplifies the information gathering process [31].

### 3.4. *Trajectory generation and optimization*

Given a behavior and a state, a candidate state space trajectory $\xi_i = [\xi_{\text{com}}, \xi_{\text{hand}}]$ is computed as a cubic spline. A generalized cubic spline is defined as a piecewise polynomial fitted to a set of via points.

$$(t_0, \xi_0^*), (t_1, \xi_1^*)...(t_k, \xi_k^*) \tag{26}$$

where $\xi_i^* \in \mathbb{R}^N$ is the joint via points at time $t_i \in \mathbb{R}$.

Given these via points, there is a cubic trajectory that passes through these points and satisfy a smooth criteria.

$$\xi_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \tag{27}$$

where $a_i, b_i, c_i, d_i$ are the polynomial coefficients optimized. The complete joint trajectory $q(t) \in \mathbb{R}^N$ is a concatenation of (27) over the time intervals.

$$q(t) = \begin{cases} \xi_0(t) & \text{if } t_0 \le t < t_1 \\ \vdots & \\ \xi_k(t) & \text{if } t_{k-1} \le t < t_k \end{cases} \tag{28}$$

Once the candidate trajectory is generated and the behavior that the robot should use is known using (15), the associated reward is computed using (17). The optimization process is performed using Differential Evolution algorithm [25] with (11) and (12) as cost functions.

From the candidate state space trajectory, both locomotion and grasping pattern are obtained using the parameterized postural primitives.

For the locomotion pattern, $\xi_{\text{com}}$ is used to calculate the ZMP reference, which is the input of the cart-table algorithm [24]. For each episode, $\xi_{\text{com}}$ is in fact a spline that connects two states, that in the case of the locomotion pattern, corresponds to one step. The location of this step is the ZMP reference. Therefore, the original trajectory $\xi_{\text{com}}$ is not the one followed by the robot COM. The real COM trajectory is generated by cart-table algorithm, and later, a kinematic inversion is used to compute the joint trajectory.

An example of the behavior 3 computation -going backwards- is presented in Figure 11. The first step is the computation of the GMM given an adaptation of the human demonstrations, which is shown in Figure 11(a). The adaptation ratio is calculated heuristically. From the GMM, the learned state trajectory $\xi^*_{\text{com}}$ is computed as it is showed in Figure 11(b). The current state trajectory $\xi_{\text{com}}$, which is the desired trajectory used to calculate the ZMP reference, is presented in Figure 11(c). The imitation trajectory is drew in black and the innovation trajectory is drew in red. The black squares in the imitation trajectory and the red crosses in the innovation trajectory are the episodes which, in the case of the locomotion, correspond to one step.

The imitation trajectory is closer the demonstrated trajectory, since the objective function is a minimization of the difference between the demonstrated reward and the current reward. The innovation trajectory is away from the demonstrated trajectory, because in this case, the objective function is the maximization of the reward.

The grasping pattern is much more easy to implement in the robot. The desired trajectory $\xi_{\text{hand}}$ corresponds to the robot end effector. The joint trajectory is computed using the humanoid Jacobian.

The humanoid initially detects the three-dimensional position of the knob using the stereo cameras integrated in the robot. The knob is located using a simple color filter. Since the initial position of the robot and the door position is known, we compute the optimization process and generate the desired state space trajectories. This process is computed offline since the genetic algorithm consumes substantial computing resources. Once the desired trajectory is known, both locomotion and grasping trajectories are computed for the robot. The door angle is estimated by knowing the location of the robot with respect to the door hinge.

Some snapshots of the implementation with the real robot are shown in Figure 12.

### 3.4.1. Limitations and considerations

Regarding the implementation of our method in the humanoid robot, some considerations and limitations have to be taken into account. The first difference between the human and the robot performance is the smoothness of the walking pattern. In the case of the human, the COM barely swings when going backwards and the GMR output of the COM is almost a straight line. However, in the robot, the swing is much greater. This produces undesirable effects. The swing may produce a crash of the robot body with the door. Furthermore, it produces a back and forth movement of the door while the robot is moving backwards. To solve this problem, we simplify the computation by allowing the robot to decouple itself from the momentum of the door by relaxing the arm stiffness and having compliance along the plane of the door, meaning the hand can passively move along the plane of the door. For instance, when defining the behaviors, we select $b_3$ to be the moment when the robot is opening the door with the movement of its body, turning off the arm motors. For simplicity, we do not consider for the robot the case when the human is moving backward and pulling the door at the same time.

## 4. Related work

There are many studies conducted in the area of Learning from Demonstration using GMM and GMR to encode kinesthetic trajectories and generalize them to perform a robot movement [8–10,32] or based on Hidden Markov Models (HMM) to encode the human demonstrations so they can be transferred to the robot [11–13]. In our work, instead of learning the robot movement, we learn a reward function, which is the basis of comparison between the human and the robot.

Our work takes some ideas from [9] where a humanoid robot use LfD to initially learn a pick and place task. Later, if an obstacle interrupts the movement, a new movement is computed using reinforcement learning to avoid the obstacle. This gave us the idea of not only using the reward profile as the space of imitation but as a metrics of the behavior's performance. Therefore, improving the imitation reward by searching in the neighborhood of the reward space, the robot can obtain a better reward which by innovating new behaviors which are not those learned from imitation.

The authors in [33] address the problem of *what to imitate* in a similar way to our proposal. They defined several possibilities of task space methods to imitate, what they called task space pool. Next, they define several criteria, like an attention criterion or an effort criterion, to choose

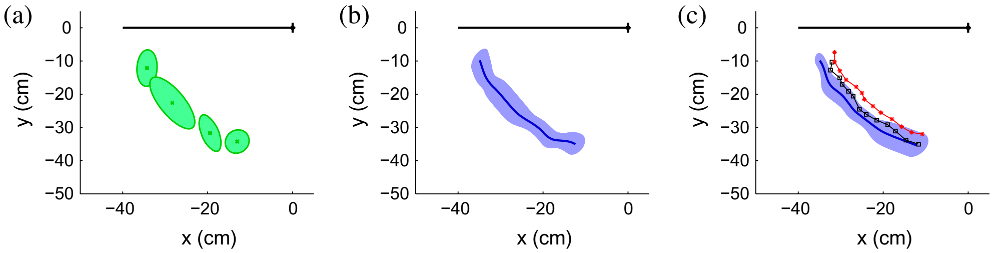

Figure 11. Illustration of the behavior 3 generation: going backwards while opening the door. The door is represented in the initial position as a horizontal black line. (a) GMM of the movement. (b) GMR of the learned motion. (c) Computation of the imitative behavior points (in black) and innovative behavior points (in red). The squares and crosses correspond to each episode.

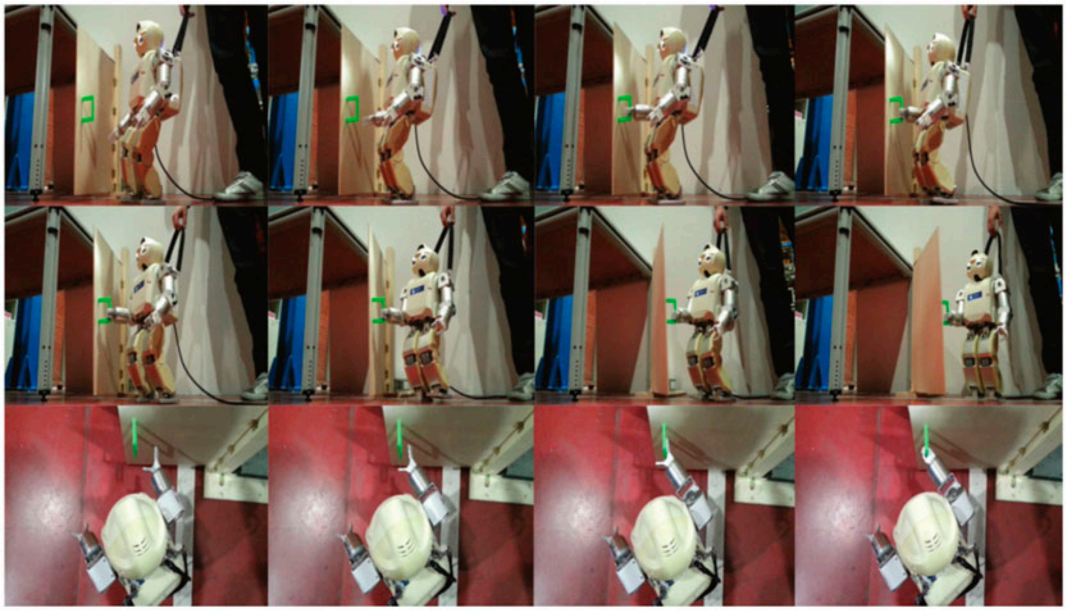

Figure 12. Snapshots of the humanoid robot performing the task of opening a door from different views.

the optimal task space to imitate. In our case, instead of defining a pool of criteria, we learn a probabilistic behavior selector matrix from human demonstrations. It defines the probability of being in a behavior given a set of states. Another interesting approach is presented in [34], where instead of the usual learning from demonstration approach, the robot learns from failed demonstrations.

There are many works related to policy learning like [35–37]. In [19], a policy search method is used to optimally select between several solutions of the same task, initially learned from demonstrations. Our work starts from a similar idea, but instead of selecting solutions of the same task, we select between sub-behaviors of a complex task and try to find an optimal policy that produces a similar reward to that of the human.

The problem of skill transfer and whole body motion transfer has been an interesting area of research in recent years. Some studies addressed the problem manipulating the angular momentum of the COM [35,38], using graphs and Markov chains [39], imitation of movement using neural networks [40] or bayesian networks [26], sequencing multi-contact postures [41], or encoding and organizing learned skills [42].

The framework called *incremental learning* uses a few demonstrations to perform a task which is incrementally improved with the aid of verbal or non-verbal guidance. In [43], a human guides a robot to sequentially construct memory models of the desired task. This incremental learning method, inspired on the behavior of social animals, allows to combine different competences to create complex tasks. Some approaches like [44] are based on constructing a task graph that leads to more general behaviors. Kulic et al. [39,45] generates whole-body motion using factorial HMM that encodes and clusters a set of incrementally learned movement primitives that can be combined to generate different behaviors.

Our work has points in common with [46] in the sense that they propose a reinforcement learning algorithm for robot manipulation that simultaneously optimizes the shape of the movement and the sequential subgoals between motion primitives. In contrast, we define a set of behaviors, each of them has a different goal and so a reward profile that represent that goal.

Our approach shares many similarities with *inverse reinforcement learning* (IRL) [29,30,47,48] and *inverse optimal control* (IOC) [49–51]. IRL is initially presented in [29,30] as the problem of extracting a reward function given an optimal behavior. The reward is extracted as a linear combination of basis features of the behavior. It can be obtained using support vector machines [29], methods based on maximum entropy [47] or active learning [48]. Similarly, IOC aims to determine the optimization criterion that produced a demonstrated dynamic process. It was successfully applied to locomotion [50], pedestrian detection [49], and manipulation [51]. In contrast with the commented approaches that attempt to explain the observations with rewards functions defined for the complete behavior, our method relies on context-dependent goal-oriented reward functions that are selected depending on which task the robot is executing.

One of the co-authors of this paper also co-authored [52,53] that use a similar approach. In their work, a surgical robot learns several tasks demonstrated by a surgeon, who selects a set of critical points that the robot's end effector has to touch. They proposed a LfD and skill innovation method based on the reward. One of the main differences with the work proposed in this paper is the way they select the basis reward functions and how they relate to each other. Instead of defining a fixed reward function for each task or goal, the robot is provided with a set of candidate reward functions. The optimal combination of these basis functions and in which proportion they are relevant to different parts of the task are learned by demonstrations. In our work, a fixed reward function is defined for each part of the behavior. The use of a specific reward function is decided by a selector matrix, learned from the human, that predicts the current state of the robot behavior and allow to apply the associated reward function.

This work is an improvement of previous papers [16,17]. In [16], we proposed to imitate a simple behavior like standing up from a chair using human demonstrations. The comparison is made in the reward domain which is a measurement of the goodness of the behavior goal. Later, in [17], we extended this work by not only imitating but innovating new behaviors using a Markov Transition Matrix to encode the reward variability and represent the behavior strategy when performing an action.

## 5. Discussion and conclusions

The presented work addresses one of the biggest questions of LfD, what to imitate [7]. As some studies reveal [1], the human brain understands the final goal of the action and reproduces it by optimizing some kind of metrics, allowing to successfully and elegantly reach the goal. Our proposal is to define this metrics as a reward profile which can be used as a basis of comparison between the human demonstrator and the robot. But an important feature of us humans is the ability to innovate new behaviors [2,3]. Therefore, we propose a reward base optimization process where the robot explores the neighbor solution space to come up with new behaviors which produce a better reward. Our framework allows a robot to create complex sequential behaviors taking into account the whole body movement.

We define a sequential multi-objective reward function for every sub-behavior of the complete task. The optimization problem consist on generating a policy for the robot to obtain an episodic reward similar to the human's, achieving an imitative behavior. Refining this policy, we can generate new solutions which improves the reward profile to achieve an innovative behavior, more relevant to the robot circumstances. The result is a framework to generate whole-body motions for the robot which can be generalized to any movement that can be learned from demonstrations.

We carried out experiments in a real humanoid robot performing the task of opening a door to test our method.

### 5.1. *Key contribution*

The main contribution of this work is the solution to the correspondence problem between a human and a robot in a common space, which represents a metrics to achieve the task goal, the reward space, and its application in a complex behavior formed by a sequence of actions. The reward space is formed by different components, depending on the objective of the action in every moment. This agrees with the theory of Minsky that proposes that our brain manages different resources that compete between each other to fulfill different goals [4].

### 5.2. *Future work*

In future works, we will investigate the adequacy and performance of different reward functions, which involves different movement features. They can be also learned by means of techniques such as inverse reinforcement learning [29]. We will also apply a Markov theory to the human demonstrations to obtain a Reward Transition Matrix that encode the variability of each behavior to make predictions as we did in a previous work [17].

## Funding

The research leading to these results has received funding from the ARCADIA project [grant number DPI2010-21047-C02-01], funded by CICYT project grant on behalf of Spanish Ministry of Economy and Competitiveness, from the RoboCity2030-II-CM project [grant number S2009/DPI-1559], funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU. This work was partially funded by the European Commission's Seventh Framework Programme [grant number FP7/2007–2013] DARWIN project [grant number FP7-270138].

## Supplemental data

Supplemental data for this article can be accessed at http://dx.doi.org/10.1080/01691864.2014.992955.

## Notes on contributors

**Miguel González-Fierro** received his BSc and MSc degrees in Electrical Engineering in 2008 from the University Carlos III of Madrid, Spain. In 2009, he received his MSc degree in Robotics and in 2014, his PhD degree in Robotics. His research is related to humanoid robot control, postural control, kinematics, dynamics, machine learning, and learning from demonstration.

**Daniel Hernández García** graduated in Electronic Engineering from the Universidad Simón Bolívar, Venezuela in 2007. In 2010, he received his MSc degreein Robotics and in 2014, his PhD degree in Robotics from University Carlos III of Madrid, Spain. His research interest include imitation learning, artificial intelligence, knowledge representation, control and teleoperation of humanoid robots, human–robot interaction, and robot perception.

**Thrishantha Nanayakkara** received his BSc degree in Electrical Engineering from University of Moratuwa, Sri Lanka, in 1996, his MSc degree in Electrical Engineering in 1998 and his PhD degree in 2001 from Saga University, Japan. Subsequently, he has been a postdoctoral research fellow at Johns Hopkins University, a Radcliffe fellow at Harvard University, and a research affiliate at MIT. At present, he is a senior lecturer at King's College London. His research is related to learning from demonstration, impedance control, adaptive control, passive dynamics, and medical robotics.

**Carlos Balaguer** received his PhD degree in Automation from the Polytechnic University of Madrid (UPM), Spain, in 1983. From 1983 to 1994, he was with the Department of Systems Engineering and Automation of the UPM as an associated professor. Since 1996, he has been a full professor of the Robotics Lab at the University Carlos III of Madrid. He is currently the vice chancellor for Research and Knowdledge of the University. His research interest are robots' design and development, robot control, path & task planning, force–torque control, assistive and service robots, climbing robots, legged and humanoid robots, and human–robot interaction.

## References

[1] Gergely G, Bekkering H, Király I, et al. Rational imitation in preverbal infants. Nature. 2002;415:755.

[2] Whiten A, McGuigan N, Marshall-Pescini S, Hopper L. Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee. Phil. Trans. Royal Soc. B: Biol. Sci. 2009;364:2417–2428.

[3] Nielsen M, Mushin I, Tomaselli K, Whiten A. Where culture takes hold: "overimitation" and its flexible deployment in western, aboriginal, and bushmen children. Child Dev. 2014;85:2169–2184.

[4] Minsky M. The emotion machine. New York (NY): Pantheon; 2006.

[5] Brooks RA. Elephants don't play chess. Robot. Autonom. Syst. 1990;6:3–15.

[6] Brooks RA. Intelligence without representation. Artif. Intell. 1991;47:139–159.

[7] Argall B, Chernova S, Veloso M, Browning B. A survey of robot learning from demonstration. Robot. Autonom. Syst. 2009;57:469–483.

[8] Gribovskaya E, Zadeh K, Mohammad S, Billard A. Learning nonlinear multivariate dynamics of motion in robotic manipulators. Int. J. Robot. Res. 2010;30:80–117.

[9] Guenter F, Hersch M, Calinon S, Billard A. Reinforcement learning for imitating constrained reaching movements. Adv. Robot. 2007;21:1521–1544.

[10] Khansari-Zadeh S, Billard A. Learning stable nonlinear dynamical systems with gaussian mixture models. IEEE Trans. Robot. 2011;27:943–957.

[11] Calinon S, D'halluin F, Sauser E, Caldwell D, Billard A. Learning and reproduction of gestures by imitation. IEEE Robot. Autom. Mag. 2010;17:44–54.

[12] Ariki Y, Hyon S-H, Morimoto J. Extraction of primitive representation from captured human movements and measured ground reaction force to generate physically consistent imitated behaviors. Neural Network. 2013;40:32–43.

[13] Billard A, Calinon S, Guenter F. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. Robot. Autonom. Syst. 2006;54:370–384.

[14] Alissandrakis A, Nehaniv C, Dautenhahn K. Imitation with ALICE: learning to imitate corresponding actions across dissimilar embodiments. IEEE Trans. Syst. Man Cybern. Syst. Hum. 2002;32:482–496.

[15] Nanayakkara T, Piyathilaka C, Subasingha A, Jamshidi M. Development of advanced motor skills in a group of humans through an elitist visual feedback mechanism. In: IEEE International Conference on Systems of Systems Engineering; San Antonio; 2007.

[16] Gonzlez-Fierro M, Balaguer C, Swann N, Nanayakkara T. A humanoid robot standing up through learning from demonstration using a multimodal reward function. In: IEEE-RAS International Conference on Humanoid Robots, Humanoids 2013; IEEE; Atlanta, USA; 2013.

[17] González-Fierro M, Balaguer C, Swann N, Nanayakkara T. Full-body postural control of a humanoid robot with both imitation learning and skill innovation. Int. J. Humanoid Robot. 2014;11:1450012.

[18] Nanayakkara T, Sahin F, Jamshidi M. Intelligent control systems with an introduction to system of systems engineering; Boca Raton: CRC Press; 2009.

[19] Daniel C, Neumann G. Learning concurrent motor skills in versatile solution spaces. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Algarve, Portugal; 2012; p. 3591–3597.

[20] Schaal S, Peters J, Nakanishi J, Ijspeert A. Learning movement primitives. Robot. Res. 2005;:561–572.

[21] Sentis L, Park J, Khatib O. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. IEEE Trans. Robot. 2010;26:483–501.

[22] Schaal S, Mohajerian P, Ijspeert A. Dynamics systems vs. optimal control – a unifying view. In: Paul Cisek TD, Kalaska, JF, editors, Computational neuroscience: theoretical insights into brain function. Vol. 165, Progress in brain research; Elsevier; 2007; p. 425–445.

[23] Khansari-Zadeh SM, Billard A. BM: an iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models. In: Proceeding of the International Conference on Robotics and Automation (ICRA); Algarve, Alaska; 2010; p. 2381–2388.

[24] Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K. Yokoi K., Hirukawa H. Biped walking pattern generation by using preview control of zero-moment point. In: Proceedings. ICRA'03. IEEE International Conference on Robotics and Automation. Vol. 2, IEEE; Taipei, China; 2003; p. 1620–1626.

[25] Storn R, Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global. Optim. 1997;11:341–359.

[26] Grimes DB, Chalodhorn R, Rao RP. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In: Robotics: Science and Systems; Cambridge (MA); 2006; p. 199–206.

[27] Chalodhorn R, MacDorman KF, Asada M. Humanoid robot motion recognition and reproduction. Adv. Robot. 2009;23:349–366.

[28] Aleotti J, Caselli S. Learning manipulation tasks from human demonstration and 3d shape segmentation. Adv. Robot. 2012;26:1863-1884.

[29] Abbeel P, Ng A. Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the Twenty-

First International Conference on Machine Learning; ACM; Banff, Alberta, Canada; 2004; p. 1.

[30] Ng A, Russell S. Algorithms for inverse reinforcement learning. In: Proceedings of the Seventeenth International Conference on Machine Learning; Stanford, CA, USA; 2000; p. 663–670.

[31] Schaal S. Is imitation learning the route to humanoid robots? Trends. Cognit. Sci. 1999;3:233–242.

[32] Calinon S. Robot programming by demonstration: a probabilistic approach. Lausanne: EPFL Press; 2009.

[33] Muhlig M, Gienger M, Steil JJ, Goerick C. Automatic selection of task spaces for imitation learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems; St. Louis, MO, USA; 2009; p. 4996–5002.

[34] Grollman DH, Billard A. Donut as I do: learning from failed demonstrations. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; Shanghai, China; 2011; p. 3804–3809.

[35] Matsubara T, Morimoto J, Nakanishi J, Hyon S-H, Hale JG, Cheng G. Learning to acquire whole-body humanoid center of mass movements to achieve dynamic tasks. Adv. Robot. 2008;22:1125–1142.

[36] Yamaguchi A, Hyon S, Ogasawara T. Reinforcement learning for balancer embedded humanoid locomotion. In: 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids); IEEE; Nashville, TN, USA; 2010; p. 308–313.

[37] Yi S-J, Zhang B-T, Hong D, Lee DD. Online learning of a full body push recovery controller for omnidirectional walking. In: 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids); IEEE; Bled, Slovenia; 2011; p. 1–6.

[38] Naksuk N, Lee C, Rietdyk S. Whole-body human-to-humanoid motion transfer. In: 5th IEEE-RAS International Conference on Humanoid Robots; IEEE; Tsukuba, Japan; 2005; p. 104–109.

[39] Kulić D, Takano W, Nakamura Y. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. Int. J. Robot. Res. 2008;27:761–784.

[40] Yokoya R, Ogata T, Tani J, Komatani K, Okuno HG. Experience based imitation using rnnpb. In: IEEE/RSJ International Conference on Intelligent Robots and Systems; IEEE; Beijing, China; 2006; p. 3669–3674.

[41] Bouyarmane K, Kheddar A. Humanoid robot locomotion and manipulation step planning. Adv. Robot. 2012;26:1099–1126.

[42] Lin H, Lee C. Self-organizing skill synthesis. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008 IROS; IEEE; Nice, France; 2008; p. 828–833.

[43] Saunders J, Nehaniv C, Dautenhahn K. Teaching robots by moulding behavior and scaffolding the environment. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction; ACM; Salt Lake City, UT, USA; 2006; p. 118–125.

[44] Pardowitz M, Zöllner R, Dillmann R. Incremental learning of task sequences with information-theoretic metrics. In: European Robotics Symposium; Springer; Palermo, Italy; 2006; p. 51–63.

[45] Kulic D, Nakamura Y. Comparative study of representations for segmentation of whole body human motion data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 IROS; IEEE; St. Louis, MO, USA; 2009; p. 4300–4305.

[46] Stulp F, Evangelos T, Stefan S, et al. Reinforcement learning with sequences of motion primitives for robust manipulation. IEEE Trans. Robot. 2012;28:1360–1370.

[47] Ziebart BD, Maas AL, Bagnell JA, Dey AK. Maximum entropy inverse reinforcement learning. In: AAAI; Chicago, IL, USA; 2008; p. 1433–1438.

[48] Lopes M, Melo F, Montesano L. Active learning for reward estimation in inverse reinforcement learning. In: Machine Learning and Knowledge Discovery in Databases; Springer; 2009; p. 31–46.

[49] Ratliff N, Ziebart B, Peterson K, Bagnell JA, Hebert M. Dey AK, Srinivasa S. Inverse optimal heuristic control for imitation learning. In: International Conference on Artificial Intelligence and Statistics; AISTATS; Florida, USA; 2009.

[50] Mombaur K, Truong A, Laumond J. From human to humanoid locomotion–an inverse optimal control approach. Autonom. Robot. 2010;28:369–383.

[51] Kalakrishnan M, Pastor P, Righetti L, Schaal S. Learning objective functions for manipulation. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; Karlsruhe, Germany; 2013; p. 1331–1336.

[52] Malekzadeh MS, Bruno D, Calinon S, Nanayakkara T, Caldwell DG. Skills transfer across dissimilar robots by learning context-dependent rewards. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); IEEE; Tokyo, Japan; 2013; p. 1746–1751.

[53] Calinon S, Bruno D, Malekzadeh MS, Nanayakkara T, Caldwell DG. Human-robot skills transfer interfaces for a flexible surgical robot. Biomed: Comput. Meth. Programs; 2014;116:81–96.