Skills Transfer Across Dissimilar Robots by Learning Context-Dependent Rewards

Milad S. Malekzadeh[1], Danilo Bruno[1], Sylvain Calinon[1], Thrishantha Nanayakkara[2] and Darwin G. Caldwell[1]

Abstract— **Robot programming by demonstration encompasses a wide range of learning strategies, from simple mimicking of the demonstrator's actions to the higher level extraction of the underlying intent. By focusing on this last form, we study the problem of extracting the reward function explaining the demonstrations from a set of candidate reward functions, and using this information for self-refinement of the skill. This definition of the problem has links with inverse reinforcement learning problems in which the robot autonomously extracts an optimal reward function that defines the goal of the task. By relying on Gaussian mixture models, the proposed approach learns how the different candidate reward functions are combined, and in which contexts or phases of the task they are relevant for explaining the user's demonstrations. The extracted reward profile is then exploited to improve the skill with a self-refinement approach based on expectation-maximization, allowing the imitator to reach a skill level that goes beyond the demonstrations. The approach can be used to reproduce a skill in different ways or to transfer tasks across robots of different structures. The proposed approach is tested in simulation with a new type of continuum robot (STIFF-FLOP), using kinesthetic demonstrations from a Barrett WAM manipulator.**

I. INTRODUCTION

Imitation encompasses various forms of learning, ranging from simple actions mimicry to the extraction and reproduction of the intent behind these actions. This last form of high-level imitation is analogous to *inverse reinforcement learning* (IRL), where the aim is to extract from demonstrations the unknown reward function that underlies the executed actions [8], [12], [13], [15], [16]. This problem can be studied in various settings, ranging from discrete state and action spaces to continuous domains describing the actions or states of the system. We concentrate here on this last form, by optimizing the robot skills directly in the policy parameters space, which has been revealed to be a well suited strategy to study learning by exploration problems on real robotic platforms [14], [17], [18].

A perspective that we started to study in [11] concerns the use of multidimensional rewards, with the motivation of exploiting richer feedback information in *reinforcement learning* (RL) about the result of a trial. RL traditionally works with a scalar reward function, composed of a weighted sum of different subfunctions, that needs to be carefully designed by the experimenter.

The objective of our work is to extend the representation of rewards to a vector formulation, which would allow the experimenter to decompose the total reward into a set of standard basis reward functions that can be relevant for the domain of application. The goal is to let the robot determine from initial demonstrations in which phase of the task or in which context the different reward components are useful, as well as in which proportions they contribute to the overall evaluation. This multi-objective representation draws potential connections with computational models that consider the role of dopamine-releasing neurons in learning behaviors controlled by reward [7]. In these models, the response types are important for distinct rewarding aspects of environmental stimuli (e.g. food, predator, reproduction).

The use of multidimensional rewards is a subclass of multi-objective RL approaches, whose aim is to consider separate objective functions in the search process [1], [10], [19]. Several of these approaches transform the multiple rewards into a scalar reward at given steps of the optimization, e.g., by switching on/off the reward components or by computing a weighted sum of reward components with adaptive weights (see [19] for a review).

In this paper, we propose to address the IRL problem by using a context-dependent form of rewards vector. By providing a set of candidate reward primitives to the robot and a set of demonstrations of the skill to acquire, we consider the problem of autonomously extracting, from the demonstrations, in which manner and in which context the various reward components are used throughout the task. The intent of the user is thus estimated in the form of high-level combination of a set of reward functions. The robot can then refine the skill by self-exploration, which can, if required, differ from the actions used by the user to fulfill the task. This approach can potentially lead to controllers that perform better that the demonstrated skill. Namely, producing higher rewards than the demonstration.

Such skill transfer mechanism is particularly advantageous in settings for which robots with different structures are used, where the mapping and generalization of the demonstrated actions can be too complex to transfer the skill at an action or movement level without exploration and refinement.

The representation of the final reward as a weighted function of reward primitives brings a meta-level learning

[1] M.S. Malekzadeh, D. Bruno, S. Calinon and D.G. Caldwell are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. name.surname@iit.it.

[2] T. Nanayakkara is with the Division of Engineering, King's College, University of London, Strand, London WC2R 2LS. thrish.antha@kcl.ac.uk.

This work was supported by the STIFF-FLOP European project (FP7-ICT-287728).

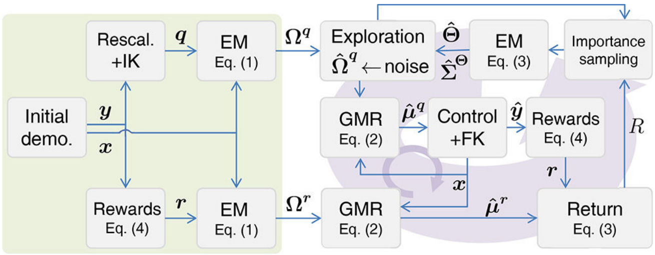


Fig. 1. Workflow of the proposed approach. The left part corresponds to the initialization of the policy and to the extraction of a context-reward mapping from the demonstration. The right part contains two loops depicting iterations at the level of the controller (small circular arrow), and iterations at the level of the exploration algorithm (big circular arrow).

problem that consists of determining the importance (and potential conflicts) of the different sources of the reward, together with the situations and phases of the task in which the reward primitives are relevant. In other words, it consists of extracting from initial demonstrations how to combine different reward sub-functions, without requiring the experimenter to predetermine and fine-tune these weights or artificially create sets of reward profiles active at different parts of the task.

The rest of the paper proceeds as follows. The proposed approach is presented in Section II. Section III describes the kinematic simulator of the STIFF-FLOP continuum robot used in the experiment. Two learning experiments are presented in Section IV. Sections V and VI present discussion, future work and conclusion.

II. PROPOSED APPROACH

The approach learns a context-dependent reward profile that is then used to refine an action profile in configuration space. Different platforms can be used for rewards saliency extraction and for exploration. In this paper, the skill is demonstrated on a gravity-compensated robot with stiff links (acting as a teleoperating device), and a flexible robot inspired by the octopus is used for reproduction.

We define $\boldsymbol{x}$, $\boldsymbol{q}$ and $\boldsymbol{r}$ as context, action and reward variables (all these variables can be multidimensional). $\boldsymbol{y}$ is the Cartesian position of the robot at configuration $\boldsymbol{q}$. A context-reward mapping is extracted from the demonstration and used to refine a context-action mapping by stochastic optimization. The initial context-action mapping can be initialized from the demonstration, or randomly. Therefore, $\boldsymbol{\Omega^r}$ and $\boldsymbol{\Omega^q}$ encode the joint distributions $\mathcal{P}(\boldsymbol{x},\boldsymbol{r})$ and $\mathcal{P}(\boldsymbol{x},\boldsymbol{q})$, respectively.

At each iteration of the self-exploration algorithm, the context variable $\boldsymbol{x}$ is evaluated, and a probabilistic estimate of the expected reward activations $\mathcal{P}(\boldsymbol{r}|\boldsymbol{x})$ is computed. The robot is controlled in configuration space by retrieving a command with $\mathcal{P}(\boldsymbol{q}|\boldsymbol{x})$, which is associated with a resulting position in Cartesian space $\boldsymbol{y}$ with a reward $\boldsymbol{r}$. The rewarding mechanism is used to iteratively modify $\boldsymbol{\Omega^q}$ by taking into account the previous attempts tested so far.

Fig. 1 illustrates the workflow of the approach. Time will be used in the experiments as a simple example of variable driving the changes of context (namely, $\boldsymbol{x}=t$). The approach is not limited to this type of input, and can be driven by other forms of inputs, such as position of external objects, state of the system, etc.

A. Multivariate reward and policy encoding

When demonstrating the task, P candidate functions $\boldsymbol{r}_{i,j} = [r_{i,1},\ldots,r_{i,P}]$ are evaluated at each iteration i, and are associated to input variables $\boldsymbol{x}$ representing the contexts/phases of the task. The profile of these activations is encoded in a *Gaussian mixture model* (GMM) with parameters $\boldsymbol{\Omega^r} = \{\pi_i^r,\boldsymbol{\mu}_i^r,\boldsymbol{\Sigma}_i^r\}_{i=1}^{K^r}$ representing respectively mixing coefficients, centers and covariance matrices. Similarly, the context-action mapping is encoded in a GMM with parameters $\boldsymbol{\Omega^q} = \{\pi_i^{\boldsymbol{q}},\boldsymbol{\mu}_i^{\boldsymbol{q}},\boldsymbol{\Sigma}_i^{\boldsymbol{q}}\}_{i=1}^{K^{\boldsymbol{q}}}$.

After initializing of the parameters with *k-means* clustering, an *expectation-maximization* (EM) algorithm is used to fit a GMM on the augmented dataset $\boldsymbol{\xi}_j^{\boldsymbol{r}} = [\boldsymbol{x}_j,\boldsymbol{r}_j]^\top$ and $\boldsymbol{\xi}_j^{\boldsymbol{q}} = [\boldsymbol{x}_j,\boldsymbol{q}_j]^\top$, by iteratively performing the following steps until convergence (the superscript * represents either $^{\boldsymbol{r}}$ or $^{\boldsymbol{q}}$)

$$\begin{aligned}
\textit{E-step:}\quad h_i(\boldsymbol{\xi}_j^*) &= \frac{\pi_i^*\,\mathcal{N}(\boldsymbol{\xi}_j^*|\,\boldsymbol{\mu}_i^*,\boldsymbol{\Sigma}_i^*)}{\sum_{k=1}^{K^*}\pi_k^*\,\mathcal{N}(\boldsymbol{\xi}_j^*|\,\boldsymbol{\mu}_k^*,\boldsymbol{\Sigma}_k^*)},\\
\textit{M-step:}\quad \pi_i^* &\leftarrow \frac{\sum_{j=1}^N h_i(\boldsymbol{\xi}_j^*)}{\sum_{k=1}^{K^*}\sum_{j=1}^N h_k(\boldsymbol{\xi}_j^*)},\\
\boldsymbol{\mu}_i^* &\leftarrow \frac{\sum_{j=1}^N h_i(\boldsymbol{\xi}_j^*)\;\boldsymbol{\xi}_j^*}{\sum_{j=1}^N h_i(\boldsymbol{\xi}_j^*)},\\
\boldsymbol{\Sigma}_i^* &\leftarrow \frac{\sum_{j=1}^N h_i(\boldsymbol{\xi}_j^*)\;(\boldsymbol{\xi}_j^*-\boldsymbol{\mu}_i^*)(\boldsymbol{\xi}_j^*-\boldsymbol{\mu}_i^*)^\top}{\sum_{j=1}^N h_i(\boldsymbol{\xi}_j^*)},
\end{aligned} \tag{1}$$

where K^* is the number of components in the GMM and N is the number of datapoints. The centers and covariances of the Gaussians can be decomposed as block matrices with input $^{\mathcal{I}}$ and output $^{\mathcal{O}}$ partitions

$$\boldsymbol{\mu}_i^* = \begin{bmatrix}\boldsymbol{\mu}_i^{*\mathcal{I}}\\ \boldsymbol{\mu}_i^{*\mathcal{O}}\end{bmatrix},\quad \boldsymbol{\Sigma}_i^* = \begin{bmatrix}\boldsymbol{\Sigma}_i^{*\mathcal{I}} & \boldsymbol{\Sigma}_i^{*\mathcal{IO}}\\ \boldsymbol{\Sigma}_i^{*\mathcal{OI}} & \boldsymbol{\Sigma}_i^{*\mathcal{O}}\end{bmatrix},$$

where the marginal distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^{*\mathcal{I}},\boldsymbol{\Sigma}_i^{*\mathcal{I}})$ describes the i-th context/phase.

Gaussian mixture regression (GMR) is used to estimate the conditional distributions $\mathcal{P}(\boldsymbol{r}|\boldsymbol{x})$ and $\mathcal{P}(\boldsymbol{q}|\boldsymbol{x})$ at each iteration [3], modeled by a Gaussian distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}^*,\hat{\boldsymbol{\Sigma}}^*)$ with parameters (the superscript * represents either $^{\boldsymbol{r}}$ or $^{\boldsymbol{q}}$)

$$\begin{aligned}
\hat{\boldsymbol{\mu}}^* &= \sum_{i=1}^{K^*} h_i(\boldsymbol{x})\left[\boldsymbol{\mu}_i^{*\mathcal{O}} + \boldsymbol{\Sigma}_i^{*\mathcal{OI}}\boldsymbol{\Sigma}_i^{*\mathcal{I}^{-1}}(\boldsymbol{x}-\boldsymbol{\mu}_i^{*\mathcal{I}})\right],\\
\text{and}\quad \hat{\boldsymbol{\Sigma}}^* &= \sum_{i=1}^{K^*} h_i^2(\boldsymbol{x})\left[\boldsymbol{\Sigma}_i^{*\mathcal{O}} - \boldsymbol{\Sigma}_i^{*\mathcal{OI}}\boldsymbol{\Sigma}_i^{*\mathcal{I}^{-1}}\boldsymbol{\Sigma}_i^{*\mathcal{IO}}\right],
\end{aligned} \tag{2}$$

$$\text{where}\quad h_i(\boldsymbol{x}) = \frac{\pi_i^*\mathcal{N}(\boldsymbol{x}|\,\boldsymbol{\mu}_i^{*\mathcal{I}},\boldsymbol{\Sigma}_i^{*\mathcal{I}})}{\sum_{k=1}^{K^*}\pi_k^*\mathcal{N}(\boldsymbol{x}|\,\boldsymbol{\mu}_k^{*\mathcal{I}},\boldsymbol{\Sigma}_k^{*\mathcal{I}})}.$$

B. Self-refinement with reward-weighted EM algorithm

Another EM strategy is used to iteratively optimize the parameters $\boldsymbol{\Omega^q}$, by treating immediate rewards as probabilities of a fictitious event, in which case probabilistic inference techniques can be used for optimization. The core idea, originally suggested by Dayan and Hinton to avoid gradient computation [6], is nowadays popular in various fields of research, due to the flexibility in the way skills can be represented, which fits well with the new developments targeting a compact and flexible encoding of movement behaviors [4].

Reward-weighted learning is employed to estimate a new policy $\hat{\boldsymbol{\Theta}}$ and an exploration noise $\hat{\boldsymbol{\Sigma}}^{\boldsymbol{\Theta}}$, by following the update rule

$$\hat{\boldsymbol{\Theta}} \leftarrow \frac{\sum_{m=1}^{M} R(\boldsymbol{\Theta}_m)\ \boldsymbol{\Theta}_m}{\sum_{m=1}^{M} R(\boldsymbol{\Theta}_m)},\ R(\boldsymbol{\Theta}_m) = \sum_{i=1}^{N} \sum_{j=1}^{P} \hat{\mu}_{i,j}^{\boldsymbol{r}}\ r_{i,j}(\boldsymbol{\Theta}_m),$$

$$\hat{\boldsymbol{\Sigma}}^{\boldsymbol{\Theta}} \leftarrow \frac{\sum_{m=1}^{M} R(\boldsymbol{\Theta}_m)\ (\boldsymbol{\Theta}_m - \hat{\boldsymbol{\Theta}})(\boldsymbol{\Theta}_m - \hat{\boldsymbol{\Theta}})^{\top}}{\sum_{m=1}^{M} R(\boldsymbol{\Theta}_m)} + \boldsymbol{\Sigma}_0, \qquad (3)$$

where $\boldsymbol{\Sigma}_0$ defines a minimum exploration noise. The ordered set of the best policies $\{\boldsymbol{\Theta}_m\}_{m=1}^{M}$ obtained so far with $R(\boldsymbol{\Theta}_1) \geqslant R(\boldsymbol{\Theta}_2) \geqslant \ldots \geqslant R(\boldsymbol{\Theta}_M)$ is used as a form of importance sampling [9]. The sum of weighted reward profiles over the N datapoints of the trajectory is used as return R, where P is the number of reward candidates and $\hat{\boldsymbol{\mu}}^{\boldsymbol{r}}$ are the reward profiles extracted in Eq. (2). At each iteration, a new policy is generated by random sampling from the distribution $\mathcal{N}(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Sigma}}^{\boldsymbol{\Theta}})$. As demonstrated in [2], [4], [5], the process can easily be extended to multi-optima policy search.

In our case, $\boldsymbol{\Theta}$ contains information about the parameters of $\boldsymbol{\Omega}^{\boldsymbol{q}}$. To guarantee that the covariance matrices remain symmetric positive semi-definite during exploration, the exploration on the covariance is performed on the first eigencomponent $\boldsymbol{a}_{i,1}$ of the ordered eigendecomposition $\boldsymbol{\Sigma}_i^{\boldsymbol{q}} = \boldsymbol{A}_i \boldsymbol{A}_i^{\top}$, with $\boldsymbol{A}_i = [\boldsymbol{a}_{i,1}, \boldsymbol{a}_{i,2}, \ldots, \boldsymbol{a}_{i,D}]$. The parameters of the policy are thus $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_i^{\boldsymbol{q}}, \boldsymbol{a}_{i,1}\}_{i=1}^{K^{\boldsymbol{q}}}$.

III. STIFF-FLOP ROBOT KINEMATICS

In minimally invasive surgery, tools go through narrow openings and manipulate soft organs to perform surgical tasks. There are limitations in current robot-assisted surgical systems due to the rigidity of robot tools. The aim of the STIFF-FLOP European project is to develop a soft robotic arm to perform surgical tasks by controlling the impedance characteristics of selected body parts of the robot. The flexibility of the robot allows the surgeon to move within organs to reach remote areas inside the body and perform challenging procedures in laparoscopy.

The first prototype of the robot, currently under development, will be composed of 3 cylindrical sections (links). Each link of the robot will consist of a soft cylinder with three chambers disposed concentrically around the axis, where air is inflated to bend the link in the desired orientation. A central chamber filled with hard grain-shaped particles is used to stiffen the link at a desired orientation by air suction.

The first measurements on a single module revealed that it can be modeled as a constant curvature section of a circle, see Fig. 2-*a*. In its local frame, the rest position (no chamber is inflated) corresponds to the module aligned along the vertical axis $\boldsymbol{e}_3$, with a rest length L_0.

The position of the tip is a function of the angle α, the arc length β and the curvature radius ρ. The orientation of the tip frame only depends on the position of the tip, evaluated by rotating the base frame to make $\boldsymbol{e}_3$ tangent to the module at the tip, keeping the other axes rigidly displaced along the manipulator. The tip position and orientation of the i-th module in the $(i{-}1)$-th tip frame are respectively defined by $\boldsymbol{Q}_i$ and

$$\boldsymbol{R}_{(i-1)i} = \frac{1}{\boldsymbol{Q}_i^{\top}\boldsymbol{Q}_i} \begin{bmatrix} -Q_{i,1}^2+Q_{i,2}^2+Q_{i,3}^2 & -2Q_{i,2}Q_{i,1} & 2Q_{i,1}Q_{i,3} \\ -2Q_{i,2}Q_{i,1} & Q_{i,1}^2+Q_{i,3}^2-Q_{i,2}^2 & 2Q_{i,2}Q_{i,3} \\ -2Q_{i,1}Q_{i,3} & -2Q_{i,2}Q_{i,3} & Q_{i,3}^2-Q_{i,1}^2-Q_{i,2}^2 \end{bmatrix}.$$

This setup allows an easy integration of multiple robot links, since any additional module can be thought as a constant curvature model applied on the previous. The position and orientation of the tip of the 3-link robot are

$$\boldsymbol{y} = \boldsymbol{Q}_1 + \boldsymbol{R}_{01}\boldsymbol{Q}_2 + \boldsymbol{R}_{01}\boldsymbol{R}_{12}\boldsymbol{Q}_3, \quad \boldsymbol{R}_{03} = \boldsymbol{R}_{01}\boldsymbol{R}_{12}\boldsymbol{R}_{23}.$$

The task parameters for the manipulator is the position $\boldsymbol{y}$ of the tip and its orientation specified by 2 angles $\boldsymbol{\theta} = [\theta_1, \theta_2]^{\top}$ (the rotation around the direction vector of the tip is not considered because the tools mounted at the end-effectors will be provided with this degree of freedom). The direct kinematics is represented by the function $\boldsymbol{W} = \boldsymbol{W}(\boldsymbol{q})$, where $\boldsymbol{W} = [\boldsymbol{y}, \boldsymbol{\theta}]^{\top}$ represents the 5-dimensional task vector and $\boldsymbol{q} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2, \boldsymbol{Q}_3]^{\top}$ the internal parameters.

The inverse differential kinematics is considered, by evaluating the Jacobian $\boldsymbol{J}$ of the direct kinematics and using standard robotics techniques, with the internal variables replacing the role of joints. The standard least-squares solution is used in the experiments. Since no workspace analysis and joint limit measurements have been performed on the hardware so far, these limits have not been considered in the simulation.

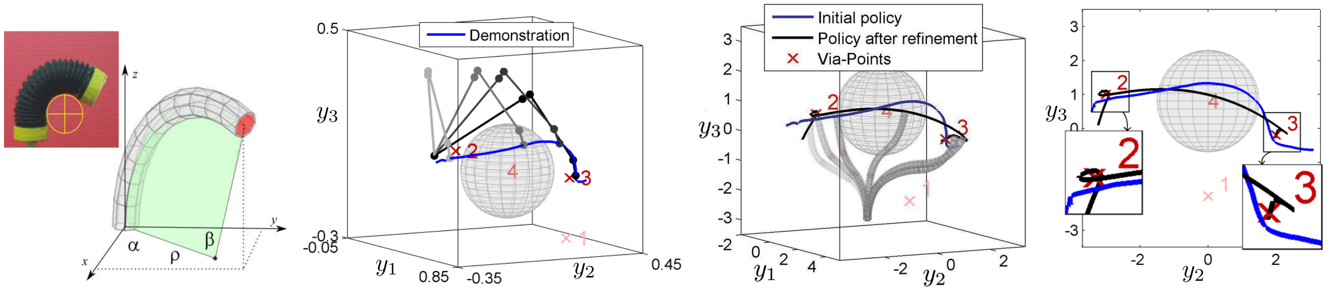
IV. EXPERIMENTS

To demonstrate the skill transfer capability of the approach, a simulated 9-DOF STIFF-FLOP robot is used, with kinesthetic demonstrations from a real 7-DOF Barrett WAM manipulator. The learned reward profiles, extracted from the demonstration, are exploited by the STIFF-FLOP robot together with a crude initialization of the policy by scaling down the observed Cartesian trajectory (with the ratio between the total lengths of the two robots), and computing a least-squares estimate of the inverse kinematics to set initial trajectories for the internal variables $\boldsymbol{q}$, with a fixed orientation of the end-effector. This initial policy then requires self-refinement to adapt to the new morphology and capability of the robot.

Fig. 2 presents a first experiment in which the aim is to pass the end-effector through two via-points, while another via-point, being part of the reward candidates, is irrelevant for the task. The gray sphere depicts an obstacle to be avoided by the arm. We can see that the demonstration (blue line) is not optimal in the sense that even though it passes close to the via-points, it does not exactly pass through them.

A. *Reward extraction phase*

During demonstration with the Barrett WAM, the Cartesian positions of the end-effector are recorded to estimate the reward candidates profile using Eq. (2). Three reward



(a) (b) (c) (d)

Fig. 2. *(a)* Single module description as a constant curvature model. The pose of the tip is a function of the angles α, β and the curvature radius ρ. The red disc depicts the end of the module. *(b)* Demonstration with the 7-DOF Barrett WAM, with three via-points (dark-red crosses: important via-points, light-red cross: irrelevant via-point), the obstacle (gray sphere) and a sub-optimal demonstration (blue line). *(c-d)* Different views of the reproduction with the STIFF-FLOP robot, after self-refinement (black line). Note that the measurements in (b) and (c-d) are performed with different robots, and that the scales thus differ.

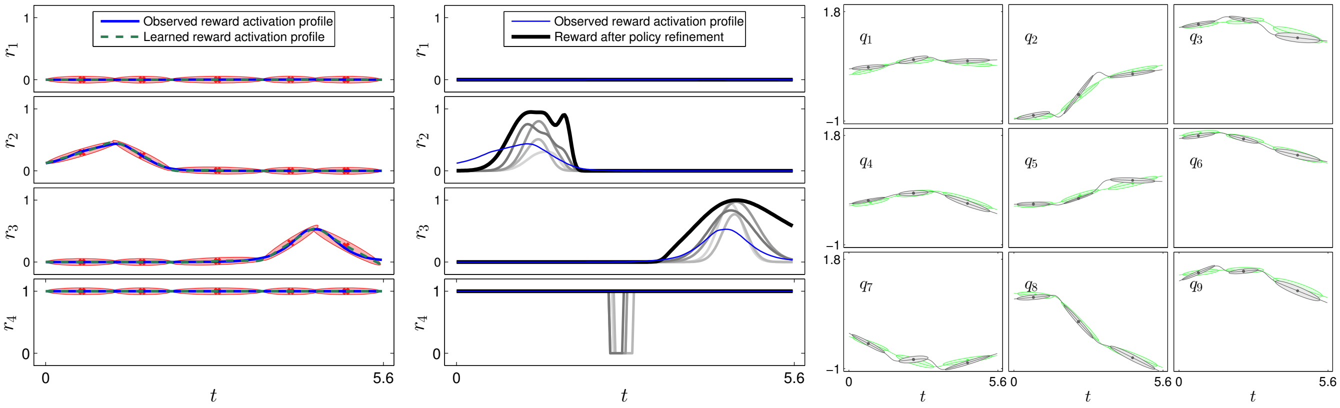


Fig. 3. *Left*: Reward candidates activation profiles demonstrated by using the Barrett WAM, with associated GMM $\boldsymbol{\Omega}^{\boldsymbol{r}}$ with $K^{\boldsymbol{r}}=5$ (light-red ellipsoids). $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are via-point passing rewards, and $\boldsymbol{r}_4$ is related to obstacle avoidance. $\boldsymbol{r}_1$ remains null because the robot does not pass near this via-point irrelevant for the task. The generalized reward profiles, calculated by Eq. (2), are shown with green dash-dotted lines. *Center*: Reward profiles optimized in the STIFF-FLOP robot, depicting the rewards profile before (blue line) and after refinement (black line). The gray lines of increasing intensity correspond to intermediate exploration trials. *Right*: Internal variables $\boldsymbol{q}$ of the STIFF-FLOP robot and associated GMM $\boldsymbol{\Omega}^{\boldsymbol{q}}$ with $K^{\boldsymbol{q}}=3$. The green line and ellipsoids correspond to the initial model. The gray ellipsoids show the Gaussians after self-refinement.

candidates were defined based on the Cartesian distance between the end-effector and the via-points at each time-step. Considering the position of the end-effector $\boldsymbol{y}_i$ at time step i, the j-th reward candidate $r_{i,j}$ is calculated as

$$r_{i,j} = \exp\big(-\alpha||\boldsymbol{y}_i - \boldsymbol{y}^v_j||\big), \quad \forall j \in \{1,2,3\}, \tag{4}$$

where $\boldsymbol{y}^v_j$ is the position of the j-th via-point, and α is a bandwidth coefficient set experimentally. The fourth reward function is binary, defined so that at each iteration, it is 0 if the robot is in contact with the obstacle, and 1 otherwise.

Fig. 3 shows the observed reward activations from the demonstration with the Barrett WAM robot. We can see that $\boldsymbol{r}_1=[r_{1,1},\ldots,r_{N,1}]^\top$, related to the irrelevant via-point, is always very small during the demonstration, while $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are high when the end-effector passes close by. We can see in Fig. 3-*center* that the initial policy parameters result in the STIFF-FLOP robot's arm touching the obstacle for a short time, making $\boldsymbol{r}_4$ drop to 0 for a couple of time steps.

B. Self-refinement of the policy

The policy vector $\boldsymbol{\Theta}$ is constructed from the parameters of $\boldsymbol{\Omega}^{\boldsymbol{q}}$, after initializing the STIFF-FLOP internal variables with inverse kinematics, as described in Section III. At each iteration, new estimated parameters $\hat{\boldsymbol{\Omega}}^{\boldsymbol{q}}$ are used to generate the internal variables trajectory with GMR, and the associated end-effector trajectory in Cartesian space. The obtained rewards are computed at each iteration, with the cumulated return R in Eq. (3) calculated for evaluation purpose.

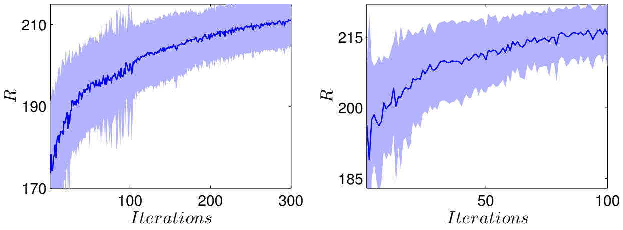


Fig. 4. Evolution of the cumulated returns R in Eq. (3) for the robot tip (*left*) and mid-point (*right*) experiments. The shaded area represents standard deviation.

Fig. 3-*center* shows the reward profile after convergence (black line). As expected, the exploration does not focus on

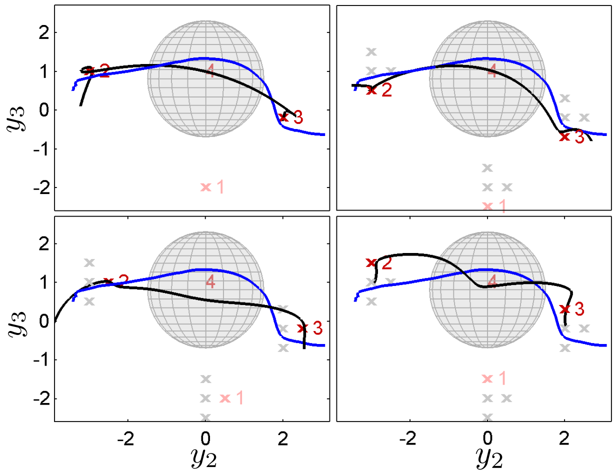


Fig. 5. Generalization capability of the proposed approach. The *top-left* graph shows the initial policy (blue line) and the original via-points. The other graphs show the refined trajectories for new positions of via-points (red-crosses).

the non-important via-point $\boldsymbol{r}_1$ (passing close to this point is not a desired characteristic of the task). The inclusion of this reward candidate $\boldsymbol{r}_1$ does not impact the self-refinement performance of the system. We can see with $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$ that the robot could refine the policy to pass closer to the via-points, thus improving the skill compared to the initial demonstration of the task. We can see with $\boldsymbol{r}_4$ that in the early exploration trials, the robot touches the obstacle. The robot then learns how to avoid it with its body. When exploring in the parameters space, the learned reward profiles penalize the trajectories in which the robot is in contact with the obstacle. The robot then learns to pass through the via-points at correct timing, while avoiding the obstacle by considering its own embodiment.

Fig. 3-*right*, shows the internal variables of the robot before and after refinement (see also Fig. 2). The experiment was run 30 times, with 300 iterations at each run (Fig. 4-*left*).

C. Generalization capability

To show the generalization capability of the approach, the via-points were displaced to see if the system could adapt the movement to these new changes. Fig. 5 depicts the results for 3 new positions. The refinement was done with the same number of iterations and parameters as in Section IV-B. We can see that the algorithm successfully refined the policy parameters to pass through the new positions of via-points while avoiding the obstacle.

D. Extension to nullspace control

In this experiment, the aim is to keep the tip of the robot at a desired fixed position and avoid the obstacle, while a given point on the robot's body should pass through the via-points, see Fig. 6. $\boldsymbol{y}$ refers in this experiment to the position of the mid-point instead of the tip. The same reward functions are used (the irrelevant via-point was not included as reward candidates), and the policy is initialized with the same demonstration. The internal velocities $\dot{\boldsymbol{q}}$ resulting from the exploration are projected into the nullspace of the Jacobian $\boldsymbol{J}$, in order to keep the tip at a desired position and orientation. This experiment is closer to the requirements of a real surgical application, in which the surgeon will control the tip of the robot by teleoperation, while the hyper-redundancy of the body is exploited to avoid areas close to organs and by passing through key points relevant for the operation and provided by the surgeon.

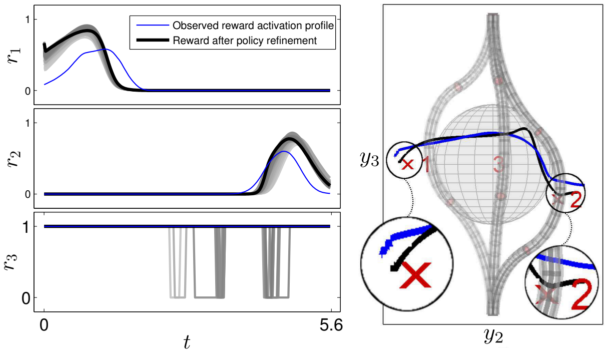


Fig. 6. Via-point experiment with the mid-point of the robot's body. *Left*: Demonstrated reward activation profiles (blue), rewards during self-refinement (gray levels of increasing intensity), and rewards after refinement (black). After refinement, $\boldsymbol{r}_3$ is always 1. *Right*: The black line is the refined trajectory for the robot mid-point and the blue line is the initial trajectory. The via-points are represented by red crosses. The light-red discs within the robot's body depict the connections between the three modules.

Fig. 6-*right* shows the refined trajectory of the robot's mid-point (in black). This trajectory passes closer to the via-points than for the initial demonstration (in blue). The mid-point via-point experiment was run 20 times, with 100 iterations at each run. Fig. 4-*right* presents the evolution of the cumulated rewards for this experiment.

V. Discussion and Future Work

In contrast to inverse reinforcement learning strategies that attempt to explain the observations with reward functions defined for the entire task (or a set of pre-defined reward profiles active for different parts of the task), the proposed approach is based on context-dependent reward-weighted learning, where the robot can learn the relevance of candidate reward functions with respect to time or situation. The robot then applies the learned reward profiles as an activation mask to rank exploration trials performed in the policy parameters space.

This process is useful when the reward function is not *a priori* evident for the end-user, or is changing during the task with respect to the ongoing situation. This aspect shall be of crucial importance in the complex surgical scenario of the STIFF-FLOP project, in which pre-defining a single reward function would be very difficult. Moreover, in the proposed approach, the context or situation are associated to the reward, and not to a controller, such that each situation can be associated with a desired set of goals, rather than the specific way that was used to obtain them. As demonstrated in the experiment, this makes the approach capable of transferring skills across dissimilar robots.

This capability might be a crucial element for future robot learning applications, in which robots will not be only numerous, but will also differ in shapes and capability. This ecosystem of robots will at some point require that robots also teach each others new skills (robot-robot skill transfer). Due to the large variety of robots and large spectrum of possible embodiments, the correspondence problem will become a bottleneck for the transfer of skills only based on action-level representations. Instead, the procedure will likely require (but not be limited to) higher-level forms of imitation capable of extracting and reproducing the intent underlying demonstrated actions, with an appropriate mix of mimicry and goal emulation strategies.

With the proposed approach, the robot can search for its own ways of fulfilling the underlying goals of the task, by considering its own body characteristics and sensorimotor capability. It also permits the transfer of skills that are difficult to achieve or demonstrate (e.g., due to the limits of a teleoperating device). In such case, even if the model is initialized with sub-optimal demonstrations, the rough shape or trend of reward activations can still be exploited by the self-refinement mechanism to improve the return, possibly surpassing the quality of the provided demonstrations.

Another interesting feature of the proposed approach, that will need to be analyzed in future work, is the possibility of discovering *hidden* aims that the user is not necessarily aware of. This can be useful in surgical tasks where surgeons can perform very complicated operations without requiring to explicitly describe what function they are actually optimizing. In some circumstances, it is indeed difficult to exactly determine which cost functions should be included in the learning phase, and in which proportion. In such situations, it is easier to provide a list of possible rewards and let the demonstration determines when, where and how these rewards candidates are relevant for the completion of the task. We also plan in future work to study the problem of learning the required number of contexts/phases from the data with Bayesian nonparametric approaches [2].

VI. CONCLUSION

We have developed a learning strategy relying on context-dependent candidate rewards to extract, from the user demonstrations, how different cost functions are relevant for different parts of the task. After a crude initialization of a policy based on the demonstration(s), the robot finds its own strategy to reproduce the learned goals of the task. It improves the observed skill by using a stochastic reward-weighted EM strategy, with exploration in the policy parameters space.

The proposed algorithm allows the system to retrieve reward profiles from actions demonstrated by user. This amounts to extracting what the underlying aims of the task are, and to weighting them by importance along the task. We showed that this combination of rewards could be formulated as context/time dependent, by using a rewards-weighted Gaussian mixture regression formalism, so that different goals for different situations/phases can be learned.

We demonstrated the generalization capability of the approach in several via-points experiments. In particular, we showed that once the different contexts and the corresponding weights are learned from one robot platform, a drastically different robot can use this information to reproduce the learned skill not only in new situations, but also with a new embodiment.

REFERENCES

[1] L. Barrett and S. Narayanan, "Learning all optimal policies with multiple criteria," in *Proc. Intl Conf. on Machine Learning (ICML)*, Helsinki, Finland, 2008, pp. 41–47.

[2] D. Bruno, S. Calinon, and D. G. Caldwell, "Bayesian nonparametric multi-optima policy search in reinforcement learning," in *Proc. AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA, 2013, pp. 1374–1380.

[3] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[4] S. Calinon, P. Kormushev, and D. G. Caldwell, "Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 369–379, April 2013.

[5] S. Calinon, A. Pervez, and D. G. Caldwell, "Multi-optima exploration with adaptive Gaussian mixture model," in *Proc. Intl Conf. on Development and Learning (ICDL-EpiRob)*, San Diego, USA, 2012, pp. 1–6.

[6] P. Dayan and G. E. Hinton, "Using expectation-maximization for reinforcement learning," *Neural Comput.*, vol. 9, no. 2, pp. 271–278, 1997.

[7] K. Gurney, T. J. Prescott, J. R. Wickens, and P. Redgrave, "Computational models of the basal ganglia: from robots to membranes," *Trends in Neurosciences*, vol. 27, no. 8, pp. 453–459, 2004.

[8] M. Howard, D. Braun, and S. Vijayakumar, "Transferring human impedance behaviour to heterogeneous variable impedance actuators," *IEEE Transactions on Robotics*, vol. 29, no. 4, 2013.

[9] J. Kober and J. Peters, "Imitation and reinforcement learning: Practical algorithms for motor primitives in robotics," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 55–62, 2010.

[10] G. D. Konidaris and A. G. Barto, "An Adaptive Robot Motivational System," in *Proc. Intl Conf. on Simulation of Adaptive Behavior, Animals to Animats 9*, September 2006.

[11] P. Kormushev, S. Calinon, R. Saegusa, and G. Metta, "Learning the skill of archery by a humanoid robot iCub," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Nashville, TN, USA, December 2010, pp. 417–423.

[12] M. Lopes, F. Melo, and L. Montesano, "Active learning for reward estimation in inverse reinforcement learning," in *Proc. European Conf. on Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 31–46.

[13] B. Michini and J. P. How, "Improving the efficiency of Bayesian inverse reinforcement learning," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2012, pp. 3651–3656.

[14] J. Peters and S. Schaal, "Using reward-weighted regression for reinforcement learning of task space control," in *Proc. IEEE Intl Symp. on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2007, pp. 262–267.

[15] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *Proc. Intl Joint Conf. on Artifical Intelligence (IJCAI)*, 2007, pp. 2586–2591.

[16] N. Ratliff, D. Bradley, J. A. Bagnell, and J. Chestnutt, "Boosting structured prediction for imitation learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[17] T. Rueckstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber, "Exploring parameter space in reinforcement learning," *Paladyn. Journal of Behavioral Robotics*, vol. 1, no. 1, pp. 14–24, 2010.

[18] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *Proc. Intl Conf. on Machine Learning (ICML)*, 2012, pp. 1–8.

[19] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker, "Empirical evaluation methods for multiobjective reinforcement learning algorithms," *Machine Learning*, vol. 84, no. 1-2, pp. 51–80, 2010.