Human-robot skills transfer interfaces for a flexible surgical robot

Sylvain Calinon[a], Danilo Bruno[a], Milad S. Malekzadeh[a], Thrishantha Nanayakkara[b], Darwin G. Caldwell[a]

[a]*Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy.*

[b]*Department of Informatics, King's College London, Strand, London, WC2R 2LS, United Kingdom.*

Abstract

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. 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. This article addresses the problem of designing learning interfaces enabling the transfer of skills from human demonstration. Robot programming by demonstration encompasses a wide range of learning strategies, from simple mimicking of the demonstrator's actions to the higher level imitation of the underlying intent extracted from the demonstrations. By focusing on this last form, we study the problem of extracting an objective function explaining the demonstrations from an over-specified set of candidate reward functions, and using this information for self-refinement of the skill. 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 objective functions with respect to the current phase of the task or encountered situation. The robot then exploits this information for skills refinement in the policy parameters space. The proposed approach is tested in simulation with a cutting task performed by the STIFF-FLOP flexible robot, using kinesthetic demonstrations from a Barrett WAM manipulator.

Keywords: Robot-assisted surgery, soft robotics, learning from demonstration, skills transfer, inverse reinforcement learning, stochastic optimization

1. INTRODUCTION

The rapid development of robotics and sensors technologies brings new exciting challenges to human-robot interaction and machine learning. The idea of robots confined only to large manufacturing environments is vanishing, and the range of robotic applications, scales and morphologies is increasing rapidly. The future generation of surgical robots is a representative example emphasizing the urgent needs of developing new types of user-friendly human-robot learning interfaces.

Current programming solutions used by the leading commercial robotics companies require the knowledge of a dedicated computer language and/or the use of a clumsy teaching pendant. Re-programming these robots requires expertise in computer programming and/or in robotics. The current software solutions are not acceptable because they do not match with the new requirements of re-employing robots to achieve different tasks, to function in various environments and to interact and collaborate with multiple users. This is reminiscent of the pre personal-computer age when people needed to be expert in computer programming to make the computer achieve a desired task. As with personal computers, the development of robots and the reduction of cost are now reaching a point where more natural and user-friendly interfaces are required to re-program the robot. The aim is to enable users who are expert in their respective fields, but who are not expert in robotics or programming, to teach robots new skills according to their needs, rather than the predetermined expectations of the robot manufacturers. Surgical robotics, and associated user interfaces, constitute a formidable area to study such interaction.

A promising approach to the problem of transferring skills to robots is to mirror the way humans learn by imitation and practice [1, 2]. Robot programming by demonstration seeks to make robot learning more like human learning, by exploiting both imitation and optimization/self-calibration in an intertwined and interactive manner, by considering the social context, task, situation and environment. The current limitation of most learning-by-imitation techniques is that they require the mapping between the demonstrator and imitator variables to be set explicitly (e.g., same number of degrees of freedom or limbs). While this might not be critical for robots looking like humans, this issue is problematic for other forms of robots. In medical applications, the diversity of robots, their miniaturization, as well as the ongoing research in material science and actuator technologies will lead to systems with very different structures, showing superhuman capability, hyper-redundancy, and drastically different ways of sensing and moving in the world.

A source of inspiration to build the next generation of learning interfaces is to reflect upon the diversity of imitation mechanisms in animals, observing how the mechanisms evolve both during the lifespan of the individuals and across generations [3]. Imitation covers a wide spectrum of strategies ranging from the blind copying of observed actions (action-level imitation, mimicry) to more elaborate forms of cognition (goal-directed imitation, social interaction, intent understanding). Thus far, research in robot learning interfaces mainly covered either one side of the spectrum (bottom-up extraction of statistical patterns from multivariate time series, continuous signal processing), or the other (top-down decomposition of tasks, symbolic reasoning, high-level planning). The middle range of the spectrum leaves plenty of room for further improvement.

A potential way of linking the two sides is to consider learning strategies reminiscent of *inverse optimal control* (IOC) and *inverse reinforcement learning* (IRL) [4, 5, 6, 7, 8, 9]. These two problems relate to imitation in the sense that they seek to extract the objective functions underlying a set of demonstrations. This knowledge can then be used by the robot to reproduce the task by minimizing a cost function, or maximizing a reward function, thus replicating the underlying intent instead of the specific actions.

We are studying this problem within the STIFF-FLOP European project, with the aim of transferring skills from a surgeon teleoperator to a flexible robot that can selectively stiffen its body to navigate within the patient through a trocar port. This form of *continuum robot* is inspired by the way the octopus makes use of its embodiment to achieve skillful movements [10, 11].

This article presents the various interfaces required by the STIFF-FLOP robot in a surgery application, by introducing the kinematics of the STIFF-FLOP robot with a focus on the learning challenges. The example of a cutting movement will be described and discussed throughout the paper.

The article is organized as follows. Section 2 presents the envisaged surgical applications with the STIFF-FLOP robot, with an emphasis on different interfacing requirements, with respect to training and testing (Section 2.1), control and teleoperation (Section 2.2), and self-refinement (Section 2.3). The proposed learning approach is presented in Section 3. Section 4 describes the kinematic simulator of the STIFF-FLOP robot used in the experiment. Several learning experiments are then presented in Section 5. Finally, Section 6 concludes the paper and discusses future work.

2. Surgical applications with the STIFF-FLOP robot

The clinical target of STIFF-FLOP is laparoscopic surgery (single or multi-port, and natural orifices translumenal endoscopic surgery). Andersen *et al.* studied the benefit of robotic-assisted surgery over open surgery and laparoscopic surgery for various medical procedures, revealing that robotic surgery can reduce the post-operation length of stay and risk of death [12]. A number of procedures and conditions were considered for potential applications of the STIFF-FLOP robot, such as laparoscopic operations required to treat gall bladder disease, reflux-disease syndrome and rectal cancer. In particular, colorectal surgery is a challenging procedure, due to the difficulty of reaching the pelvis, and retracting the organs with the right tension during dissection.

The range of motion given to the surgical instruments inside the abdomen is of critical importance for surgical performance and safety. The robotic systems currently available have a limited range of movement, often relying on the end-effector endowrist internal articulation. This limited flexibility increases the fulcrum effect on the abdominal wall port-sites, and makes the system poorly manoeuvrable. It requires the surgeon to move the arm outside the abdomen to change its surgical target, thus complexifying the positioning of the system in the operating room.

In contrast to existing systems, the surgical scenario envisioned in STIFF-FLOP is to provide the robot with the capability of turning around organs in the thoracic and abdominal cavities. Moreover, since the robot cannot completely avoid contact between the robotic arm and intra-abdominal organs, its structure is developed such that it can locally adopt various degrees of stiffness/compliance.

The first prototype of the robot, currently under development, will be composed of 3 cylindrical sections (links). Each link 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.

This new type of robots requires the development of several dedicated interfaces that are described in the next three sections.

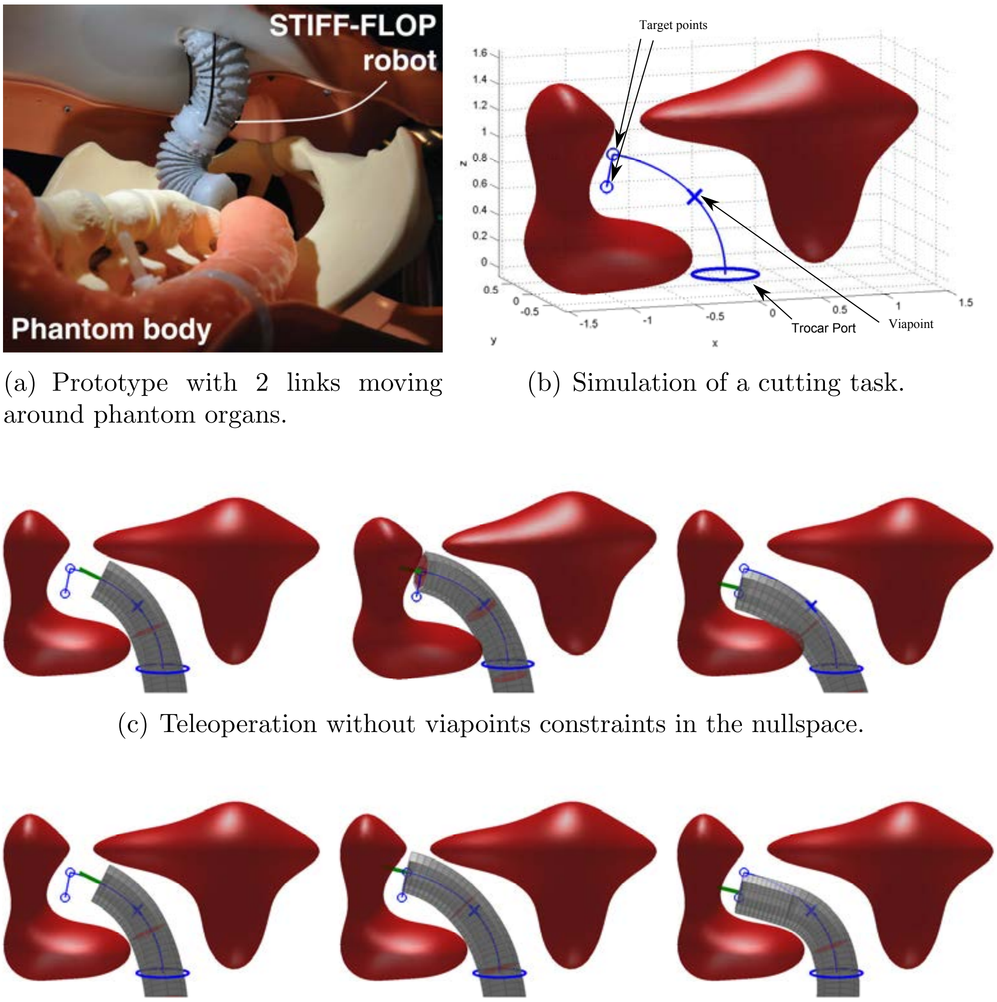
2.1. Design of training and testing interfaces

This section describes the test rigs currently under development to analyze and train the robot. Robot skills transfer and refinement will be performed in two steps: with a realistic simulation environment and with the real robot. The simulation will be based on SOFA, an open-source framework primarily targeting real-time medical simulation with deformable tissues [13].

A realistic operating environment will be considered to train and test the behavior of the real robotic arm. The abdominal surgery scenario is characterized by narrow spaces with difficult access. A transparent plastic chamber will be constructed with inflatable balloons representing organs, that can be pseudo-randomly inflated and deflated to mimic pulsating tissues, with adjustable distances between the ports and the manipulated organs.

A test environment will also be created to resemble the human anatomy of the pelvis, gastroesophageal junction, and adrenal glands. This will be used for several purposes: 1) to measure the benefit of the STIFF-FLOP robot over existing technology; 2) to train surgeons to interact with the STIFF-FLOP robot; and 3) to train the robot controller through stochastic optimization. The present article focuses on this third way of employing the sensorized test environment.

Phantom organs and structures will be included in the test rig to simulate real organs (pelvis, rectum, colon, spleen, and liver). They will be manufactured with a soft and elastic material such as silicone or urethane rubber to realistically emulate human cavities. The test rig will be equipped with sensorized plates to evaluate the interaction between the robot and the phantom organs. The design of this system will be modular to facilitate its reconfiguration. When the system will be available (a first prototype is shown in Fig. 1-*(a)*), it will be exploited for self-refinement of the controller in several situations.



(a) Prototype with 2 links moving around phantom organs.

(b) Simulation of a cutting task.

(c) Teleoperation without viapoints constraints in the nullspace.

(d) Teleoperation with a learned controller in the nullspace.

Figure 1: *(a)* STIFF-FLOP robot. *(b)* Cutting task used as an example of movement of the tip controlled by the surgeon. *(c-d)* Illustration of the importance of considering passing-through viapoints as a second constraint during teleoperation. Without constraints, the control of the tip results in the robot's body touching the organs.

2.2. Design of control and teleoperation interfaces

With the emergence of hyper-redundant and flexible robots in surgery applications, the mapping between the teleoperation devices and the robots becomes more complex, because the two have very different structures. New human-robot interaction challenges arise, for which new learning interfaces need to be developed.

The most common control scenarios we will consider are those in which the surgeon controls the end-effector of the robot with a teleoperating interface (or another robotic/sensing device), leaving the global motion of the STIFF-FLOP arm as a learning and control problem. The considered movement can include tasks such as tissue grasping, multi-tool operations (e.g. holding a tool and ablating), and motion patterns related to suturing operations. Input motions provided by the surgeon (e.g., through joysticks) is directly translated into the corresponding movements of the tip, overcoming the kinematic problems of standard handheld laparoscopic tools moving through a fulcrum point. While the surgeon controls the motion of the end-effector, the robot provides assistance by navigating safely in the proximity of organs. The redundancy of the robot is exploited to passively move in-between body tissues to avoid tension or potential damage.

This is achieved by letting the surgeon define critical viapoints during the teleoperation, or by detecting previously observed situations requiring to pass the robot's body through specific positions (e.g., because of fragile surrounding areas). In the first case, the definition of the viapoints can be made before the procedure or when the tip passes through these points. As recommended by the members of the STIFF-FLOP consortium with practical expertise in surgery, the operator needs to be sure that not only the tip (that is teleoperated) but also the body of the robot passes through these desired viapoints.

An example of the problem is depicted in Fig. 1-*(b)*, where the goal is to introduce the STIFF-FLOP robot between two organs (represented as red blobs) and to perform a cutting movement at a given place. A set of critical points are determined by the surgeon when entering into the patient's body through the trocar port, defining the important areas where the robot should pass through.

All the calculations for the inverse kinematics of the robot and the movement of the intermediate points are performed online during the teleoperation. The robot is controlled with a Jacobian-based approach to inverse kinematics, by learning movement behaviors in the nullspace of the Jacobian. In Fig. 1-*(d)*, by controlling the motion of the robot in the nullspace, two additional constraints are fulfilled in parallel:

- The body of the robot is kept inside the trocar port;
- Once the tip has passed through the viapoint and continues its movement toward the target point, the nearest point on the robot's body is always kept as close as possible to the viapoint (perpendicularly to the trajectory).

2.3. Design of learning and self-refinement interfaces

Transferring skills to the STIFF-FLOP robot poses many correspondence challenges due to the drastically different structure of the teleoperating device and the flexible robot, which makes it difficult to transfer the skill directly at an action or movement level. It is thus proposed to consider higher-level imitation strategies, such as emulating the goal of the task through extraction of the user's intent from demonstrations/teleoperations.

This *inverse reinforcement learning* (IRL) 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, 15, 16]. We will consider the case in which multiple objectives can contribute to the search process, see e.g., [17, 18, 19].

The robot is provided with an over-specified set of candidate objective functions, and extracts how to select and weight these functions to best explain the observed demonstrations, providing a score in the form of a scalar return. To cover the cases in which the objective functions are relevant only for some parts of the task (instead of being relevant over the entire task), this list is often provided as a set of basis functions active for different situations or for different phases of the overall behavior. The robot then needs to keep only the most prominent objective functions from these candidates, where sparse regression techniques have been considered for feature selection, e.g. by using l_1 regularization [8].

We instead propose to learn the joint distribution between a variable $\boldsymbol{x}$ describing the context of the task (possibly multidimensional) and the returns of candidate objective functions $\boldsymbol{r}$. By exploiting Bayes' rule, regression is then used to estimate the relevance of each objective function for a newly encountered situation (new input $\boldsymbol{x}$). This estimate is used as a mask to regulate the importance of the different objectives to evaluate the overall return of the new behavior that is currently evaluated. This window has the effect of sparsely selecting which are the most important objective functions to consider at each iteration, and how to weigh those according to their respective importance. The estimate of the weighting terms is provided in the form of a Gaussian distribution, providing a local estimate of the covariations (e.q., to determine the correlations among the objectives or extract the most prominent functions by spectral analysis).

The robot thus determines from the initial demonstrations in which phase of the task (or in which context) the different reward components are useful, and in which proportions they should contribute to the overall evaluation of the task. This type of context-based multi-objective representation has a similar role as computational models of dopamine-releasing neurons in learning behaviors controlled by reward [20]. In these models, the response types are indeed relevant for distinct rewarding aspects of environmental stimuli (e.g. food, predator, reproduction).

In our case, such skill transfer mechanism is particularly advantageous to transfer skills across robots with different embodiments. In these situations, the mapping and generalization of the demonstrated actions can indeed be too complex to transfer skills at an action or movement level, and instead requires an efficient combination of imitation with exploration and self-refinement strategies. The proposed context-dependent reward-weighted learning strategy offers us the flexibility to learn the relevance of candidate reward functions with respect to time or situation. The nature of the approach leaves the robot with the freedom to exploit its own body characteristics for exploration of new solutions replicating the task, with the possibility of reaching a level of skill that goes beyond that of the demonstration (learning from suboptimal demonstrations).

3. Proposed learning model

The proposed approach consists of learning a context-reward mapping from the demonstration, that is then used to refine a context-action mapping (with actions represented in configuration space) by stochastic optimization. Different platforms can be used for the two phases. Here, the skill is demonstrated on a gravity-compensated robot with stiff links (acting as

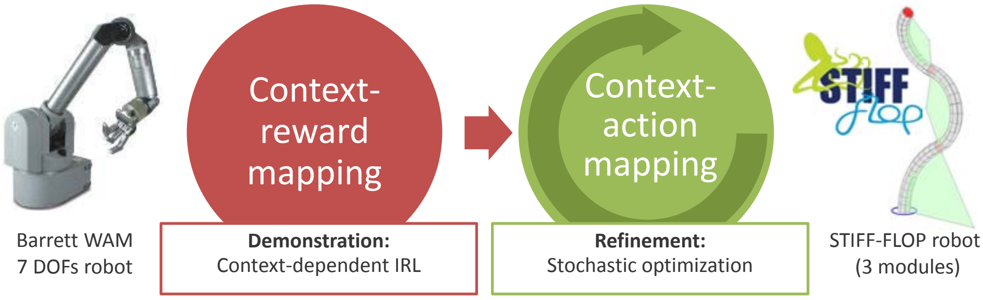


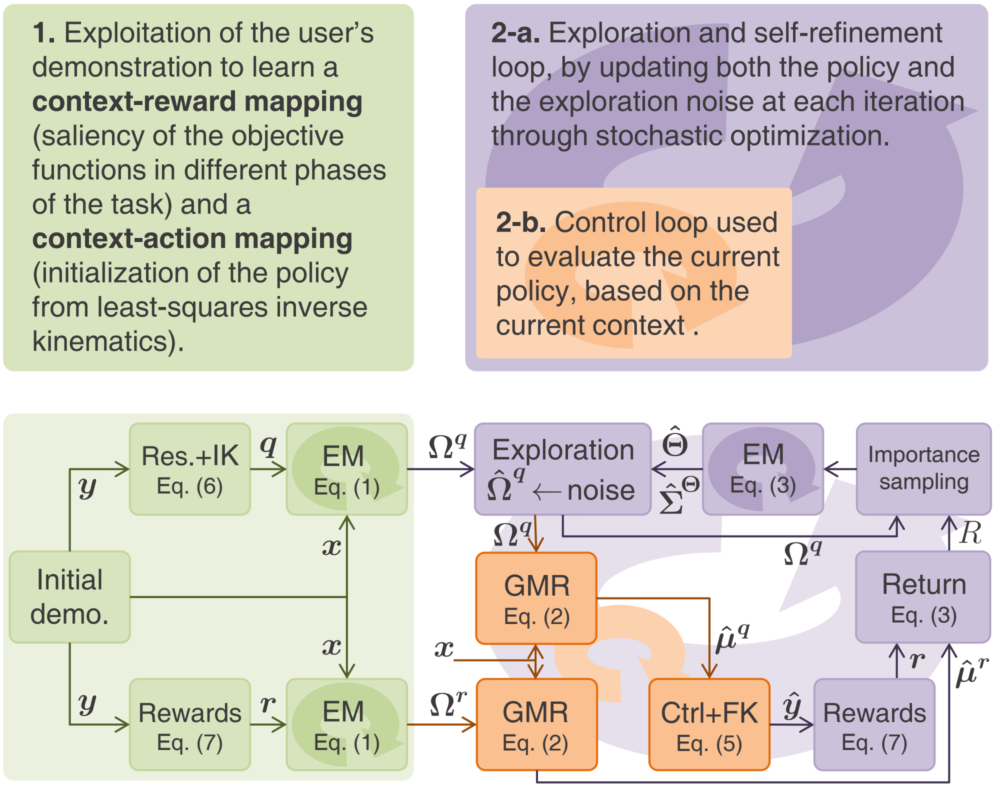
Figure 2: Transfer of skills from a 7 degrees of freedom manipulator with stiff links to the STIFF-FLOP continuum robot. The manipulator is controlled with torque commands used to compensate for the gravity, making the robot lightweight such that it can be used as an interface to demonstrate a movement by kinesthetic teaching. The demonstrations are used to extract when these objectives are relevant for the completion of the task. This is achieved by learning a context-reward mapping in the form of a multivariate probability distribution. The model is then used to reproduce the task on the STIFF-FLOP continuum robot, with a stochastic optimization process performed in the policy parameters space, representing the mapping between context variables and actions (internal control variables) in a compact form. Each new trial is evaluated with a mask estimated by regression on the learned context-reward joint distribution.

a teleoperating device), and reproduced on a flexible robot inspired by the octopus, see Fig. 2.

This process is advantageous when the objective function is not *a priori* evident for the end-user, or is changing during the task with respect to the current situation. This aspect shall be of crucial importance in the complex surgical scenario of the STIFF-FLOP project, where pre-specifying a single objective function would be very difficult. In the proposed approach, the context or situation are associated with the reward, such that each situation can be associated with a desired set of goals, rather than the specific way that was used to obtain them.

Throughout the paper, we will use $\boldsymbol{x}$, $\boldsymbol{q}$ and $\boldsymbol{r}$ as context, action and reward variables (all these variables can be multidimensional). $\boldsymbol{y}$ is the end-effector position of the robot (in Cartesian space) after action $\boldsymbol{q}$ (in configuration space). $\boldsymbol{\Omega}^{\boldsymbol{r}}$ and $\boldsymbol{\Omega}^{\boldsymbol{q}}$ encode the joint distributions $\mathcal{P}(\boldsymbol{x},\boldsymbol{r})$ and $\mathcal{P}(\boldsymbol{x},\boldsymbol{q})$, respectively. The initial context-action mapping can be initialized from the demonstration, or randomly.

At each iteration, given the current context variable $\boldsymbol{x}$, the robot is con-



1. Exploitation of the user's demonstration to learn a **context-reward mapping** (saliency of the objective functions in different phases of the task) and a **context-action mapping** (initialization of the policy from least-squares inverse kinematics).

2-a. Exploration and self-refinement loop, by updating both the policy and the exploration noise at each iteration through stochastic optimization.

2-b. Control loop used to evaluate the current policy, based on the current context .

Figure 3: Workflow of the proposed approach. *1.* Extraction of a context-reward mapping from the demonstration, and initialization of the policy as a context-action mapping. *2-a.* Iterations loop at the level of the search process. *2-b.* Iterations loop at the level of the robot controller.

trolled in configuration space by retrieving a command with $\mathcal{P}(\boldsymbol{q}|\boldsymbol{x})$, which is associated with a resulting position in Cartesian space $\boldsymbol{y}$. We used the mean of the distribution $\mathcal{P}(\boldsymbol{r}|\boldsymbol{x})$ as a mask on the relevant objective functions, which is used to evaluate the current reward. After a given number of iterations with this control scheme, a final return score R is assigned to the current policy, and a new $\boldsymbol{\Omega}^{\boldsymbol{q}}$ is generated by taking into account the previously tested policies. Fig. 3 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, $\boldsymbol{x}=t$). Note that 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.

3.1. Multivariate reward and policy encoding

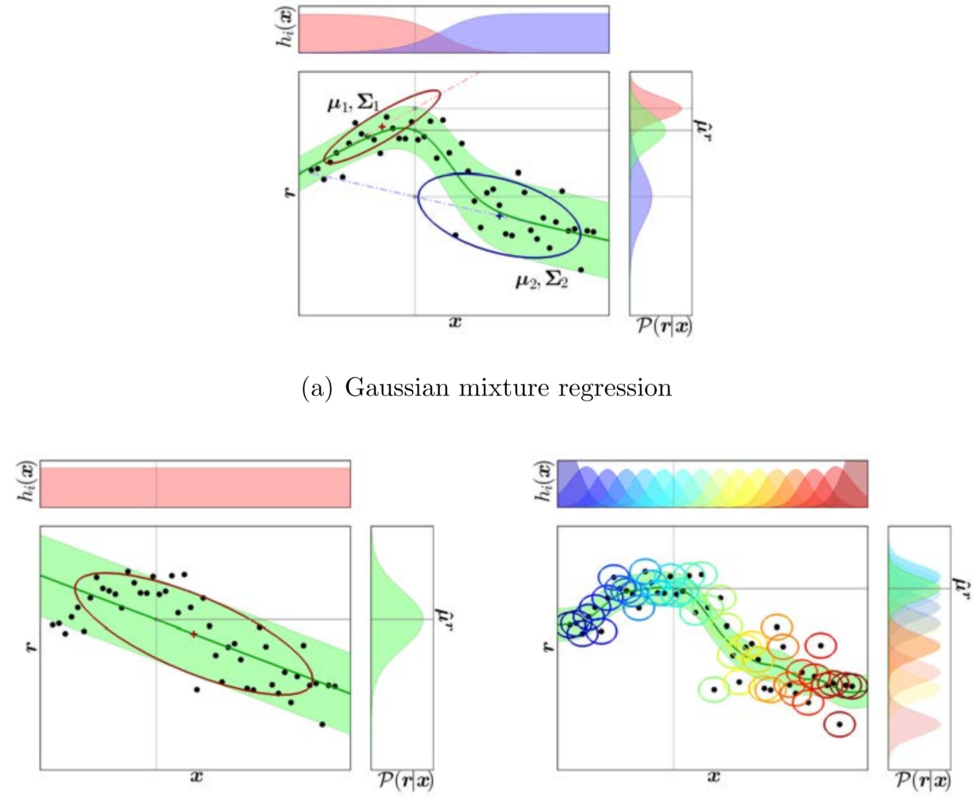
The mapping problem described above is implemented with *Gaussian mixture regression* (GMR) [21, 22], an approach that has also been used in other surgery scenarios [23, 24].

For each demonstration of the task, P candidate objective functions $[r_{1,n},\ldots,r_{P,n}]$ are evaluated at each iteration $n\in\{1,\ldots,N\}$. They are associated to input variables $\boldsymbol{x}$ representing the context/phase of the task. The mapping is encoded in a *Gaussian mixture model* (GMM) with parameters $\boldsymbol{\Omega}^{\boldsymbol{r}}=\{\pi_i^{\boldsymbol{r}},\boldsymbol{\mu}_i^{\boldsymbol{r}},\boldsymbol{\Sigma}_i^{\boldsymbol{r}}\}_{i=1}^{K^{\boldsymbol{r}}}$ representing respectively mixing coefficients (priors), centers and covariance matrices [25]. Similarly, the context-action mapping is encoded in a GMM with parameters $\boldsymbol{\Omega}^{\boldsymbol{q}}=\{\pi_i^{\boldsymbol{q}},\boldsymbol{\mu}_i^{\boldsymbol{q}},\boldsymbol{\Sigma}_i^{\boldsymbol{q}}\}_{i=1}^{K^{\boldsymbol{q}}}$.

An *expectation-maximization* (EM) algorithm [26] is used to fit a GMM on the augmented dataset $\boldsymbol{\xi}_n^{\boldsymbol{r}}=[\boldsymbol{x}_n,\boldsymbol{r}_n]^{\top}$ and $\boldsymbol{\xi}_n^{\boldsymbol{q}}=[\boldsymbol{x}_n,\boldsymbol{q}_n]^{\top}$, by iteratively performing the following steps until convergence (the superscript * represents either $^{\boldsymbol{r}}$ or $^{\boldsymbol{q}}$)

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

where K^* is the number of components in the GMM and N is the number of datapoints. The two steps above are iteratively computed until a stopping



(a) Gaussian mixture regression

(b) Least-squares linear regression

(c) Nonparametric kernel regression

Figure 4: *(a)* Illustration of the encoding of $\mathcal{P}(\boldsymbol{x}, \boldsymbol{r})$ as a Gaussian mixture model (GMM) with two components, and estimation of $\mathcal{P}(\boldsymbol{r}|\boldsymbol{x})$ with Gaussian mixture regression (GMR). Both $\boldsymbol{x}$ and $\boldsymbol{r}$ can be multidimensional, and the same procedure is applied to $\boldsymbol{x}$ and $\boldsymbol{q}$. *(b)* Analogy with standard linear regression when a single component is used ($K^{\boldsymbol{r}} = 1$). *(c)* Analogy with Nadaraya-Watson kernel regression when a Gaussian is centered on each datapoint ($K^{\boldsymbol{r}} = N$).

criterion is satisfied. EM guarantees the improvement of the likelihood at each iteration, converging to a local optimum [27]. The EM algorithm is initialized with *k-means* clustering [28] to start the iterative procedure from a good initial estimate.

By defining which variables span for input and output parts (noted respectively by $^{\mathcal{I}}$ and $^{\mathcal{O}}$ superscripts), a block decomposition of the vectors $\boldsymbol{\mu}_i^*$ and matrices $\boldsymbol{\Sigma}_i^*$ can be written as

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

Gaussian mixture regression (GMR) is then used to estimate the conditional expectation of the reward and the action conditioned on the context [21, 22], by estimating the conditional density $\mathcal{P}(\boldsymbol{r}|\boldsymbol{x})$ and $\mathcal{P}(\boldsymbol{q}|\boldsymbol{x})$ at each iteration in the form of a Gaussian distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}^*, \hat{\boldsymbol{\Sigma}}^*)$ with parameters (the superscript * represents either $^{\boldsymbol{r}}$ or $^{\boldsymbol{q}}$)

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

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

By defining $|\boldsymbol{\Sigma}_i^{*\mathcal{I}}|$ as the determinant of $\boldsymbol{\Sigma}_i^{*\mathcal{I}}$, and $D^{\boldsymbol{x}}$ as the dimension of the context variable, the above mixture weights are computed with

$$\text{and} \quad \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^{*\mathcal{I}}, \boldsymbol{\Sigma}_i^{*\mathcal{I}}) = \frac{1}{(2\pi)^{\frac{D^{\boldsymbol{x}}}{2}} |\boldsymbol{\Sigma}_i^{*\mathcal{I}}|^{\frac{1}{2}}} \exp\!\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_i^{*\mathcal{I}})^{\top} (\boldsymbol{\Sigma}_i^{*\mathcal{I}})^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i^{*\mathcal{I}}) \right).$$

In contrast to other regression methods such as *Locally Weighted Regression* (LWR) [29], *Locally Weighted Projection Regression* (LWPR) [30], or *Gaussian Process Regression* (GPR) [31, 32], GMR does not model the regression function directly, but models a joint probability density function of the data. It then derives the regression function from the joint density model. Density estimation can be learned in an off-line phase (with the EM process presented above). For regression, any subset of multivariate input and output dimensions can be selected, which can change on-the-fly during reproduction. Expectations on the remaining dimensions can be computed in an online manner, corresponding to a convex sum of linear approximations (with weights varying non-linearly). GMR can thus handle different sources of missing information for the context variables, since the system is able to consider any combination of input/output mappings during reproduction. In terms of computation, learning the model depends linearly on the number of datapoints, while prediction is independent on this number. The regression estimate can thus be computed very rapidly, and provides a probabilistic estimate of the output signal as a full Gaussian distribution. Depending on the number of Gaussians being used, GMR can cover a wide spectrum of regression mechanisms, from standard linear regression to non-parametric kernel regression,[1] see Fig. 4.

3.2. Self-refinement with reward-weighted EM algorithm

This section presents the stochastic optimization algorithm used to refine the parameters $\boldsymbol{\Omega^q}$. The process consists of stochastically exploring for new solutions in the policy parameters space, associate a reward signal to each trial, and reshape the exploration space by a weighted combination of the most successful trials obtained so far. The procedure corresponds to an expectation-maximization (EM) algorithm in which the reward signal is treated as a likelihood.

Dayan and Hinton originally suggested that a RL problem can be tackled by EM to avoid gradient computation [34]. They introduced the core idea of treating immediate rewards as probabilities of a fictitious event, in which case probabilistic inference techniques can be used for optimization. They showed that in some circumstances, it is possible to make large well-founded changes to the policy parameters without explicitly estimating the curvature of the space of expected payoffs, by a mapping onto a maximum likelihood probability density estimation problem. In effect, they maximize the reward by solving a sequence of probability matching problems, where the task parameters are chosen at each step to match a fictitious distribution determined by the average rewards experienced on the previous steps. Although there can be large changes in the task parameters from one step

[1] In contrast to GPR, the mixture weights $h_i(\boldsymbol{x})$ are not determined by the local structure of the training data, but by the components of the global GMM [33].

to the next, there is a guarantee that the average reward is monotonically increasing. From this simple idea, various reward-weighted policy learning approaches emerged [15, 14, 35, 36, 16].

Interestingly, such trend brings a cross-disciplinary flavour to robot learning by exploration, by making links with other research fields such as stochastic optimization and evolutionary computation. Indeed, several research fields converged to similar algorithmic solutions (and conclusions about the robustness of the approach), with approaches such as the *cross-entropy method* (CEM) [37] or the *covariance matrix adaptation evolution strategy* (CMA-ES) [38].

In our approach, 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

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

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

where $\boldsymbol{\Sigma}_0$ defines a predetermined minimum exploration noise (corresponding to a regularization term in EM). 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 [36]. 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 $\hat{\boldsymbol{\mu}}^{\boldsymbol{r}}$ is the estimated mask on the relevant objective functions computed 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 shown in [39, 40, 41], the above process can easily be extended to multi-optima policy search.

In our case, $\boldsymbol{\Theta}$ will contain parts of the parameters $\boldsymbol{\Omega}^{\boldsymbol{q}}$. Namely, $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_i^{\boldsymbol{q}}, \boldsymbol{a}_{i,1}\}_{i=1}^{K^{\boldsymbol{q}}}$, where $\boldsymbol{a}_{i,1}$ is the first eigencomponent of the ordered eigendecomposition $\boldsymbol{\Sigma}_i^{\boldsymbol{q}} = \boldsymbol{A}_i \boldsymbol{A}_i^{\top}$, with $\boldsymbol{A}_i = [\boldsymbol{a}_{i,1}, \boldsymbol{a}_{i,2}, \ldots, \boldsymbol{a}_{i,D}]$. This parameterization reduces the number of parameters to explore and guarantees that the covariance matrices remain symmetric positive semi-definite.

The use of an EM-based reward-weighted learning strategy in our surgery scenario has three advantages:

1. The flexibility it offers in the way skills can be represented. The exploration can be conducted directly in the policy parameters space, rep-

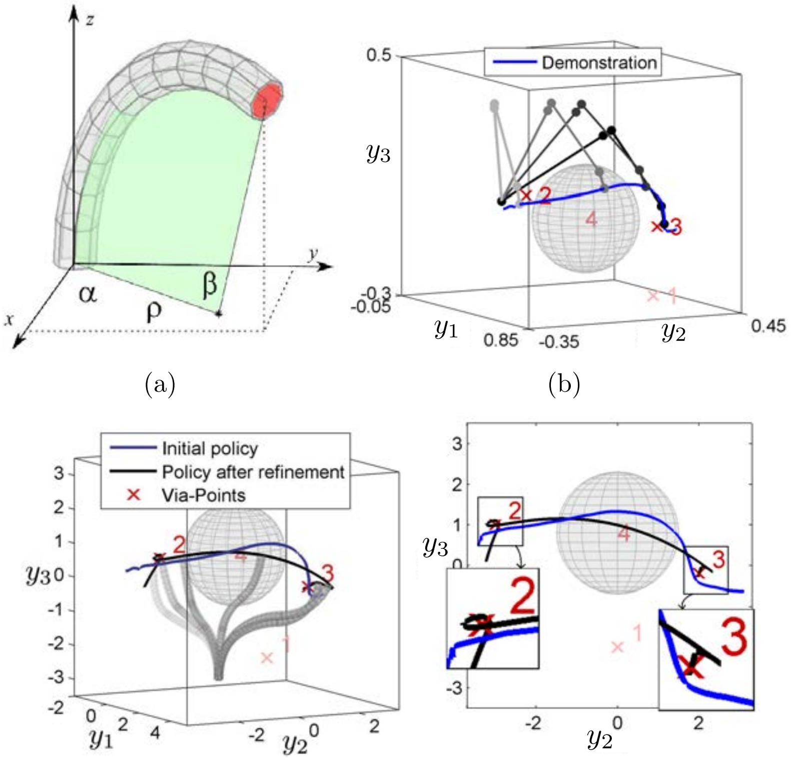
resented by a compact GMM. This parsimony drastically reduces the search space, with GMR naturally smoothing out the space of possible control solutions, leading to natural movement behaviors even if the random sampling process results in a poor selection of the parameters.

2. The convergence properties of the underlying EM mechanism driving the search, which is, interestingly, the same core EM mechanism adopted for the extraction of context-reward mappings from the demonstration. EM has a number of properties that make it a simple and attractive algorithm over gradient-ascent approaches. It constitutes a good convergence compromise for an exploration problem, being conservative in complex solution spaces, but still guaranteeing linear convergence. The convergence toward optimal parameters has sometimes been reported to be slow while the convergence in likelihood was rapid, see discussion in [42]. Such rapid convergence in likelihood is desirable, because the predictive aspect of data modeling is more important in the considered scenario than the *true* value of the parameters.
3. The possibility to estimate not only the most promising policy parameters, but also to reshape the exploration term in the form of a full covariance matrix in the policy parameters space (second order search mechanism, easily extensible to more complex exploration terms with a mixture of Gaussians).

4. STIFF-FLOP robot kinematics

The first prototype of the robot, currently under development, will be composed of 3 cylindrical sections (links) [43]. Each link 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 direction. A central chamber filled with hard grain-shaped particles is used to stiffen the link at a desired pose by air suction.

The first measurements on a single link revealed that it can be modeled as the constant curvature section of a circle, see Fig. 5-a. In its local frame, the rest position (no chamber is inflated) corresponds to the link aligned along the vertical axis $\boldsymbol{e}_3$, with a rest length L_0. The current prototype of the single link is 50 cm long in the rest position and has a diameter of 40 cm. When totally inflated, it can elongate by 80%. Moreover, each link can bend at approximatively 180°.



(a)

(b)

(c)

(d)

Figure 5: *(a)* Single link 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 link. *(b)* Sub-optimal demonstration with the 7-DOF Barrett WAM (depicted in blue line), with candidate objective functions evaluated based on the distances to three via-points (dark-red crosses: important via-points, light-red cross: irrelevant via-point), and a proscribed area delimited by a gray sphere. *(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.

The position $\boldsymbol{Q}_i$ of the tip of the i-th link can be written as a function of the angle α_i, the arc length β_i and the curvature radius ρ_i (see Fig. 5-a) as

$$\boldsymbol{Q}_i = \Big[\rho_i\big(1-\cos(\beta_i)\big)\cos(\alpha_i), \quad \rho_i\big(1-\cos(\beta_i)\big)\sin(\alpha_i), \quad \rho_i\sin(\beta_i)\Big].$$

Both variables $\boldsymbol{Q}_i$ or $\{\rho_i, \alpha_i, \beta_i\}$ can be used to describe the kinematics of the link. The constant curvature coordinates of the single link can be obtained from the position $\boldsymbol{Q}_i$ of the tip by using the inverse relations

$$\rho_i = \frac{Q_{i,1}^2 + Q_{i,2}^2 + Q_{i,3}^2}{2\sqrt{Q_{i,1}^2 + Q_{i,2}^2}}, \quad \alpha_i = \arctan\frac{Q_{i,2}}{Q_{i,1}}, \quad \beta_i = \arccos\left(1 - \frac{\sqrt{Q_{i,1}^2 + Q_{i,2}^2}}{\rho_i}\right).$$

The constant curvature coordinates allow us to obtain the position of any point along the single link. Given the position of the tip, the constant curvature coordinates are obtained by the equation above. The Cartesian coordinates of a point positioned at a fractional position $\gamma \in [0, 1]$ of the link are then given by

$$\boldsymbol{F}_{\rho_i,\alpha_i,\beta_i}(\gamma) = \Big[\rho_i\big(1-\cos(\beta_i\gamma)\big)\cos(\alpha_i), \; \rho_i\big(1-\cos(\beta_i\gamma)\big)\sin(\alpha_i), \; \rho_i\sin(\beta_i\gamma)\Big]. \tag{4}$$

Here γ corresponds to all possible points from the base of the link to the tip ($\gamma = 0$ corresponds to the base).

We will use the Cartesian position $\boldsymbol{Q}_i$ of the tip in the rest frame of the base as internal variables. They will replace the role of joint angles commonly used as internal variables in kinematic models of standard manipulators.

The constant curvature model allows us to evaluate the orientation of the tip, which only depends on the position of the tip evaluated by rotating the base frame to make $\boldsymbol{e}_3$ tangent to the link at the tip, keeping the other axes rigidly displaced along the manipulator. The tip orientation of the i-th link in the $(i-1)$-th tip frame is defined by

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

The use of constant curvature coordinates allows us to evaluate the possible limits that the hardware possesses. The only limitation on the possible configurations are placed in the fact that the inflation mechanism only allows a limited range of elongation, depending on the bending of the link and its orientation in space. The length of the robot can be obtained as a function of the constant curvature coordinates as $L_i = \rho_i \beta_i$.

Once the robot is bent in a given direction, the range of possible elongations can be obtained by fixing the curvature radius ρ_i and varying the arclength β_i, which is achieved by inflating the chambers. The geometry of the system suggests that this elongation also depends on the bending direction (i.e., which chamber is inflated to get that curvature). As a result, we will have limitations such as $\beta_{\min}(\alpha_i, \rho_i) < \beta_i < \beta_{\max}(\alpha_i, \rho_i)$, which will need to be obtained experimentally (no workspace analysis and limit measurements have been performed on the prototype so far). As a starting hypothesis, we will consider limitations corresponding to the nominal elongation when all the chambers are inflated (80%), thus limiting the possible lengths of the robot to $L_0 < L_i < L_0 + 0.8L_0$.

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 robot can be recursively evaluated, for any possible number of links K, by computing

$$\boldsymbol{y} = \sum_{i=1}^{K} \boldsymbol{R}_{0(i-1)} \boldsymbol{Q}_i, \quad \boldsymbol{R}_{0K} = \prod_{i=1}^{K} \boldsymbol{R}_{(i-1)i},$$

with $\boldsymbol{R}_{00} = \boldsymbol{I}$. For 3 links, this results into

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

The full control of the orientation of the tip is not needed, because the rotation around its main axis will be performed during surgery by the tool mounted at the end-effector. This degree of freedom is essential for most surgical tasks, and this rotation could not be efficiently controlled by reorienting the whole robot, because it would highly limit the movement of the other links, possibly passing near vulnerable organs in winding configurations.

For this reason, only the direction of the main axis of the tip frame is controlled. This can be easily obtained in the frame of reference of the tip. The rotation bringing the vertical axis $\boldsymbol{e}_3$ of the tip to a generic direction vector $\boldsymbol{V}$ is represented by the quaternion

$$\boldsymbol{z} = \big[\cos\frac{\omega}{2},\ \sin\frac{\omega}{2}\boldsymbol{u}\big]^{\top},$$

where ω is the angle between $\boldsymbol{e}_3$ and $\boldsymbol{V}$, and $\boldsymbol{u} = \boldsymbol{e}_3 \times \boldsymbol{V}$. By evaluating the above in the frame of reference of the tip ($\boldsymbol{u} \perp \boldsymbol{e}_3$), only the components $\boldsymbol{\theta} = [z_2, z_3]^\top$ are sufficient to describe the rotation. As a result, the task parameters for the manipulator are the position $\boldsymbol{y}$ of the tip and its orientation $\boldsymbol{\theta}$, collected into a 5-dimensional task vector $\boldsymbol{W} = [\boldsymbol{y}, \boldsymbol{\theta}]^\top$.

The kinematics of the link is finally complemented by controlling the base of the manipulator, corresponding to 6 additional DOFs. For example, the position $\boldsymbol{Q}_0 = [y_{0,1}, y_{0,2}, y_{0,3}]^\top$ controlling the translation of the base, and the Euler angles $\boldsymbol{\eta}_0 = [\psi_0, \theta_0, \phi_0]^\top$ controlling its orientation (with a corresponding rotation matrix denoted by $\boldsymbol{R}_0$).

In the initial phase of the operation, the base is moved to insert the robot inside the patient's body, maintaining it within the trocar port during the motion. Notice that the range of rotations remains quite small during the surgical task, which facilitates the avoidance of singularities due to the Euler angles representation.

This provides the whole system with a total number of $3K+6$ degrees of freedom, with K the number of links. The total internal variables will be denoted by

$$\hat{\boldsymbol{q}} = [\boldsymbol{\eta}_0, \boldsymbol{Q}_0, \boldsymbol{Q}_1, \dots, \boldsymbol{Q}_K]^\top,$$

with the position and orientation of the tip computed as

$$\boldsymbol{y} = \boldsymbol{Q}_0 + \boldsymbol{R}_0 \left(\sum_{i=1}^{K} \boldsymbol{R}_{0(i-1)} \boldsymbol{Q}_i \right), \quad \boldsymbol{R}_{0K} = \boldsymbol{R}_0 \prod_{i=1}^{K} \boldsymbol{R}_{(i-1)i}. \tag{5}$$

The direct kinematics is represented by the function $\boldsymbol{W} = \boldsymbol{W}(\hat{\boldsymbol{q}})$. An 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 joint angles in standard manipulators. Namely,

$$\frac{d\boldsymbol{W}}{dt} = \frac{\partial \boldsymbol{W}}{\partial \hat{\boldsymbol{q}}} \frac{d\hat{\boldsymbol{q}}}{dt} = \boldsymbol{J} \frac{d\hat{\boldsymbol{q}}}{dt}.$$

Given a starting position for the robot, corresponding to a choice of internal parameters $\hat{\boldsymbol{q}}_0$ and task parameters $\boldsymbol{W}_0$, the final configuration can be computed as a minimum norm solution with

$$\hat{\boldsymbol{q}}(t) = \int_0^t \boldsymbol{J}^\dagger \frac{d\boldsymbol{W}}{dt} dt + \hat{\boldsymbol{q}}(0), \tag{6}$$

where $\frac{d\boldsymbol{W}}{dt}$ is the speed of the movement in task space and $\boldsymbol{J}^{\dagger}$ is the pseudoinverse of the Jacobian matrix. The speed of movement is composed by a linear speed and an angular velocity, specifying the rotation of the tip orientation.

A 3-links manipulator will be endowed with 15 DOF. Given the 5-dimensional task space, 10 DOFs of redundancy can be exploited to control the body when the surgeon moves the tip to perform the surgical task.

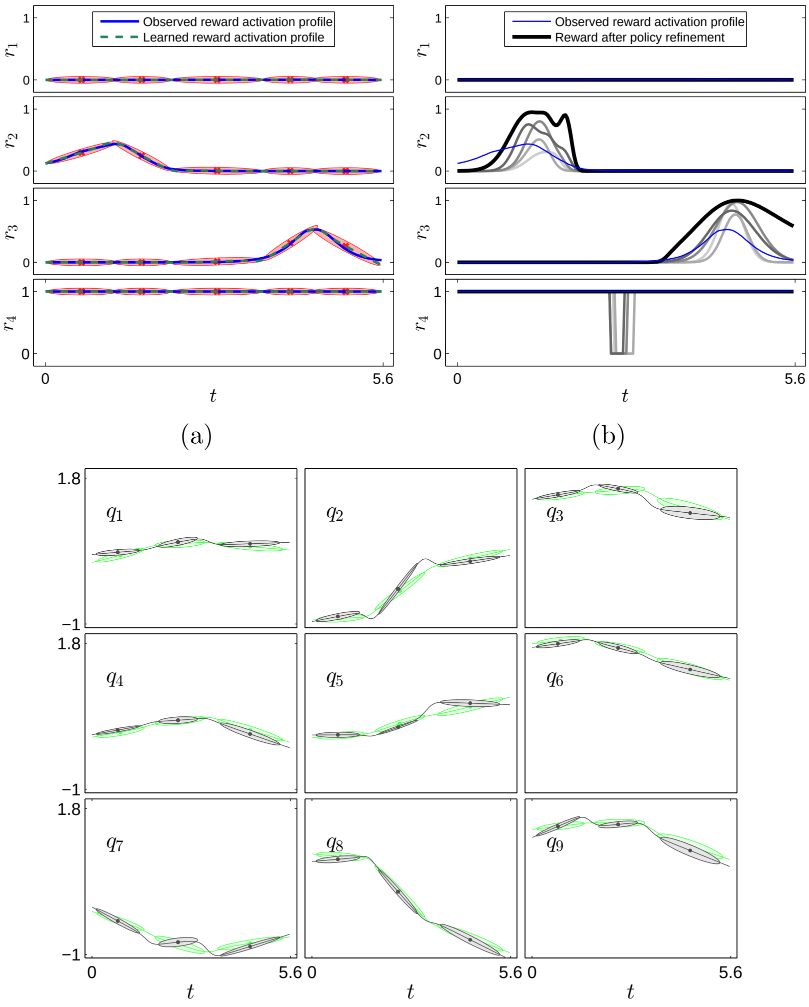
The above Jacobian-based formalism allows us to exploit various techniques originally developed for standard stiff robots (e.g., limits and singularity avoidance, weighted inverse kinematics, constrained optimization). In particular, the nullspace of the Jacobian allows the control of the robot's body without affecting the task kinematics.

5. Experiments

To demonstrate the skill transfer capability of the approach, a 9-DOF STIFF-FLOP robot (3 links) is used in simulation, with kinesthetic demonstrations from a real 7-DOF Barrett WAM manipulator. The learned context-reward mapping extracted from the demonstration is exploited by the STIFF-FLOP robot to refine a context-action mapping initialized from a crude rescaling of the observed trajectory in Cartesian space (with the ratio between the total lengths of the two robots), and a least-squares estimate of the inverse kinematics to create a profile for the internal variables $\boldsymbol{q}$, with a fixed orientation of the end-effector. This initial policy requires self-refinement to adapt to the new morphology and capability of the robot.

In the experiments, time is used as the context variable, and the number of Gaussians corresponding to the number of phases in the task is selected based on a *Bayesian information criterion* (BIC) [44]. Another option is to select the number of Gaussians in the model with a Bayesian nonparametric approach, as described in [41].

An earlier version of this experiment is also reported in [45]. We extend here the previous results with the additional experiment of a cutting movement, closer to the envisaged surgical scenario. The first part of the experiment presents the simpler case in which the robot can freely control its links (Section 5.1). It is then extended to the case in which both the teleoperator and the robot can collaboratively move the robot (Section 5.2).



(a)

(b)

(c)

Figure 6: *(a)* Activation of the objective functions from the demonstrations with 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 related to via-point distance, and $\boldsymbol{r}_4$ is related to the avoidance of a spherical area. $\boldsymbol{r}_1$ remains null because the robot does not pass through this via-point, considered as unimportant for the task (irrelevant objective function candidate). The generalized context-reward mapping calculated with Eq. (2) are shown with green dash-dotted lines. Time is depicted in seconds. *(b)* Activation of the objective functions when the policy is optimized with the STIFF-FLOP robot. The profiles before and after refinement are respectively depicted in blue and black line, with the gray lines of increasing intensity corresponding to intermediate trials. *(c)* Internal control variables $\boldsymbol{q}$ of the STIFF-FLOP robot and associated GMM $\boldsymbol{\Omega}^{\boldsymbol{q}}$ with $K^{\boldsymbol{q}} = 3$. The ellipsoids in green correspond to the initial model. The ellipsoids in gray show the Gaussians after self-refinement. The scales of $\boldsymbol{q}$ correspond to robot links of unit length.

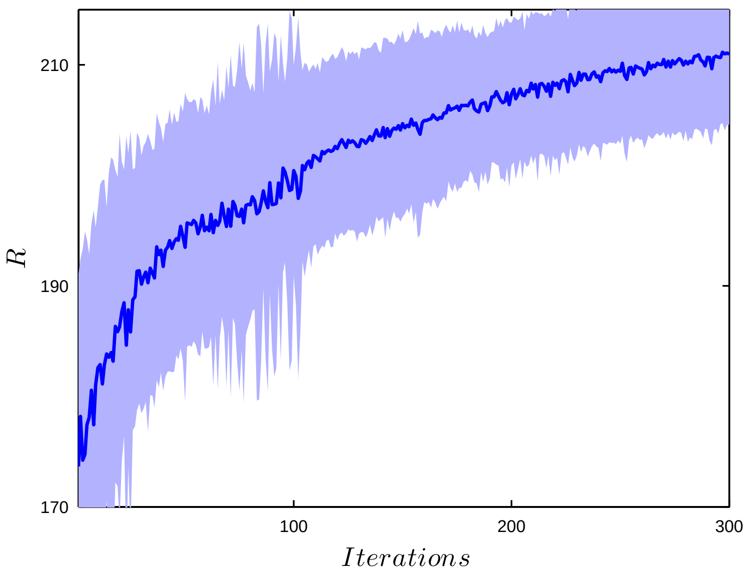


Figure 7: Evolution of the cumulated returns R in Eq. (3) for the robot tip control experiment. The shaded area represents standard deviations.

5.1. Robot tip control experiment

The task of the first experiment consists of learning to pass the end-effector through two via-points while avoiding a spherical area, without external intervention from the teleoperator. An additional via-point, which is not part of the task constraints, is added to the set of candidate objectives to test the robustness of the system to cope with irrelevant reward candidates.

Fig. 5 presents the experiment. The gray sphere depicts the area to be avoided by the arm. We can see that the demonstration (blue line) is sub-optimal: the robot passes close to the via-points but does not pass through them.

In this first experiment, $\boldsymbol{y}_n$ refers to the position of the robot's end-effector at time step n. The first three objective functions are defined based on the Cartesian distance between the end-effector and the via-points. The j-th objective function candidate $r_{j,n}$ is calculated as

$$r_{j,n} = \exp\big(-\alpha||\boldsymbol{y}_n - \boldsymbol{y}_j^v||\big), \quad \forall j \in \{1,2,3\}, \tag{7}$$

where $\boldsymbol{y}_j^v$ is the position of the j-th via-point, and α is a bandwidth coefficient set experimentally. The fourth reward function is binary, defined as 0 if the robot is in contact with the spherical area, and 1 otherwise.

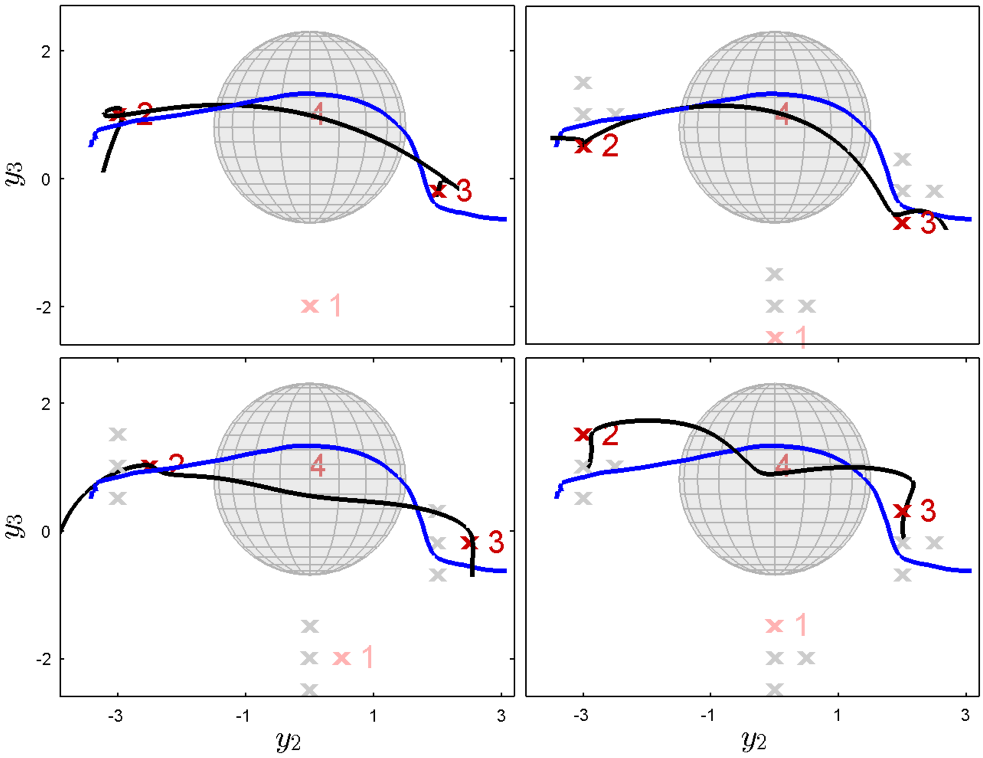


Figure 8: Generalization capability of the proposed approach. The *top-left* graph shows the initial policy (blue line) and the original via-points. The other graphs show the refined trajectories for new positions of via-points (red-crosses).

Fig. 6-*(a)* shows the activation of the objective functions with the demonstration from the Barrett WAM robot. We can see that $\boldsymbol{r}_1 = [r_{1,1}, \ldots, r_{1,N}]^\top$, related to the irrelevant via-point, is close to zero 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. 6-*(b)* that the initial policy parameters result in the STIFF-FLOP robot's arm entering into the dangerous area for a short time, making $\boldsymbol{r}_4$ drop to zero for a couple of time steps.

Fig. 6-*(b)* shows the reward profile after convergence (black line). 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 also see with $\boldsymbol{r}_4$ that in the early exploration trials, the robot enters the proscribed spherical area. The robot then learns how to avoid it with its continuum body. Indeed, when exploring in the parameters space, the learned mask on the objective functions penalizes the trajectories in which the robot enters the proscribed spherical area. The robot then learns to pass through the via-points at correct timing, while avoiding the forbidden region with its own embodiment. As expected, the exploration does not focus on the irrelevant via-point $\boldsymbol{r}_1$ (passing close to this point is not part of the task constraints). The results show that the inclusion of this reward candidate $\boldsymbol{r}_1$ does not impact the self-refinement performance of the system.

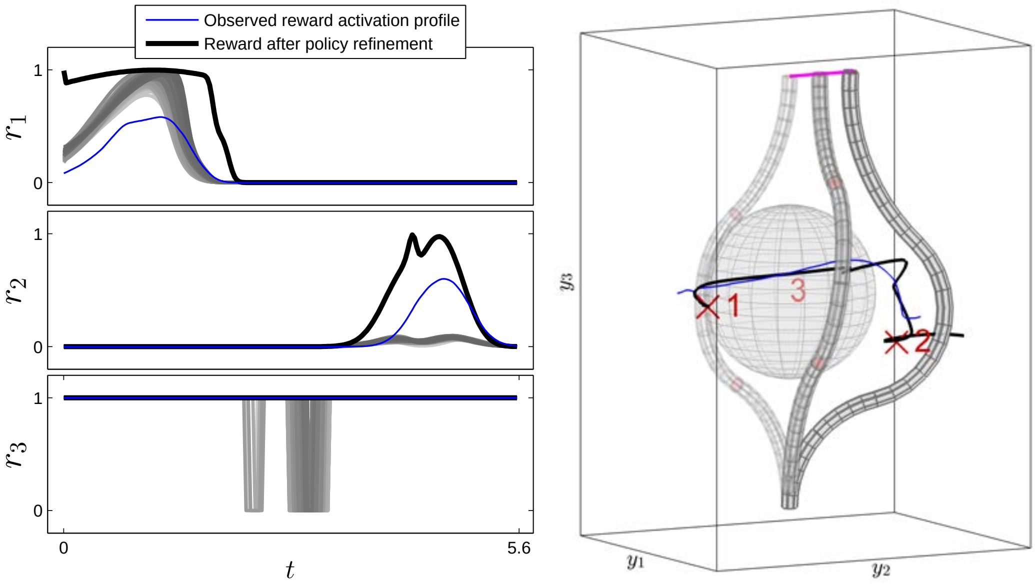
Fig. 6-*(c)*, shows the internal variables of the robot before and after refinement (see also Fig. 5).

Fig. 7 presents average results over 30 runs of the same experiment, with 300 self-refinement iterations at each run (the number of iterations was determined based on convergence).

In order to highlight the generalization capability of the approach, the via-points were displaced to see if the system could find new movements that could adapt to these changes (the refinement was done with the same conditions and parameters as in the original experiment). Fig. 8 presents the results for 3 new positions. We can see that the proposed approach successfully refined the policy parameters to pass through the new positions of via-points while avoiding the spherical region.

5.2. Robot mid-point control experiment

The experiment is then extended to the case in which both the teleoperator and the robot collaboratively move the robot. This version of the



(a)

(b)

Figure 9: Via-point experiment with the robot's mid-point while the surgeon controls the robot's tip to achieve a cutting task. *(a)* Activation of the objective functions during demonstration (blue), during self-refinement (gray levels of increasing intensity), and rewards after refinement (black). After refinement, $\boldsymbol{r}_3$ is always 1 (the robot fully avoids the proscribed area). *(b)* Refinement of the mid-point trajectory (black line). The pink line at the robot's tip shows the cutting motion controlled by the teleoperator. The blue line depicts the initial trajectory. The via-points are represented by red crosses. The light-red discs within the robot's body shows the connections between the three links.

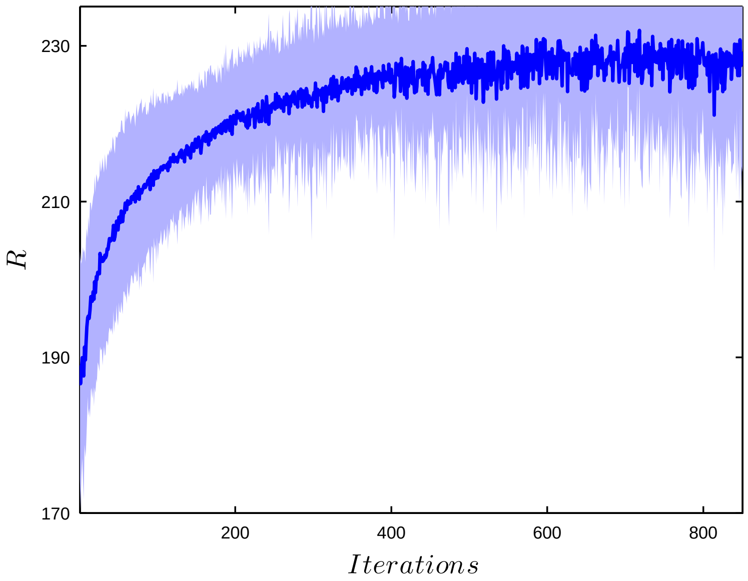


Figure 10: Evolution of the cumulated returns R in Eq. (3) for the cutting task experiment. The shaded area represents standard deviations.

experiment is closer to the requirements of the envisaged surgical application, in which the surgeon control the tip of the robot (here, with a tool used to achieve a cutting motion), while the robot controls a mid-point along the robot's body without interfering with the teleoperation.

The robot learns how to exploit its kinematic redundancy to pass its body through key points of potential relevance for the operation and predefined by the teleoperator, while avoiding a spherical area also defined by the surgeon (e.g., delimiting vital organs), see Fig. 9 (and Fig. 1 for the illustration of the challenges).

In this experiment, $\boldsymbol{y}_n$ now refers to the position of the robot's mid-point at time step n. The policy is initialized with the same demonstration as in the previous experiment. A similar set of candidate objective functions is also employed (apart from the irrelevant via-point that was this time not included).

By exploiting the kinematic model presented in Section 4, the control of intermediate points on the robot can be achieved by calculating the Jacobians at these points and projecting the inverse differential kinematics on the nullspace. This is achieved by assigning an internal coordinate $s \in [0, K]$ to the robot, specifying the rest position of all the points of the back bone of the manipulator ($s=0$ and $s=K$ correspond respectively to the base and end-effector). The Cartesian position can then be obtained with Eqs (4) and (5) as

$$\boldsymbol{P}(s) = \boldsymbol{Q}_0 + \boldsymbol{R}_0 \left(\sum_{i=0}^{\lfloor s \rfloor} \boldsymbol{R}_{0(i+1)} \boldsymbol{Q}_{i+1} + \boldsymbol{F}_{\rho_{i+1},\alpha_{i+1},\beta_{i+1}}(s - \lfloor s \rfloor) \right),$$

where $\lfloor s \rfloor$ is the floor function applied to s.

The intermediate Jacobians can be evaluated as $\boldsymbol{J}(s) = \frac{\partial \boldsymbol{P}(s)}{\partial \hat{\boldsymbol{q}}}$. This allows the evaluation of internal variables trajectories corresponding to (nullspace) movements of the intermediate point $\boldsymbol{P}(s)$ as

$$\hat{\boldsymbol{q}}_0(t) = \int_0^t \left[\left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J} \right) \boldsymbol{J}^{\dagger}(s) \frac{d\boldsymbol{P}}{dt} \right] dt + \hat{\boldsymbol{q}}_0(0),$$

where $\boldsymbol{J}$ is the Jacobian of the task space.

The sum of two motions at each time step describes a movement where the end-effector tracks a desired position and orientation (controlled by teleoperation), while any selected point along the robot's body (here, the mid-point with $s = \frac{K}{2}$) is displaced to fulfill other objectives.

Figs 9 and 10 present the results of the experiment. Fig. 9-*(b)* shows the refined trajectory of the robot's mid-point (in black). We can see that this trajectory passes closer to the via-points than for the initial demonstration (in blue). Fig. 10 presents the evolution of the cumulated rewards after repeating the same experiment 20 times, with 850 self-refinement iterations for each run of the experiment (the number of iterations was determined based on convergence).

6. Conclusion and perspectives for future work

We developed a learning strategy relying on context-dependent rewards to extract, from expert demonstrations, which objective functions are relevant for different parts of the task, in which proportion they are relevant, as well as the synergies among those (by estimation of a full covariance matrix), so that different goals for different situations can be learned. This information is then used to estimate how the different candidate objectives should interact to evaluate a new policy, in a possibly different context.

We proposed an implementation of this approach by using Gaussian mixture model (GMM) to learn the context-reward mapping, which is then used with Gaussian mixture regression (GMR) to estimate a mask on the candidate objectives functions, employed to evaluate new exploration trials.

The paper focused on the special case in which the context variable represents different phases of the task, where time is used as a special case of context variable. This amounts to extracting what the underlying aims of the task are, and to weighting them by importance along the task for the evaluation of new reproduction attempts. We also concentrated on the special case in which the skill is explored and refined in the policy parameters space, by using a stochastic reward-weighted EM strategy to re-estimate the next policy and the next exploration noise at each iteration. After a crude initialization of a policy based on the demonstration(s), the robot then finds its own strategy to reproduce the learned objectives.

We demonstrated the generalization capability of the approach in two experiments depicting an envisaged surgical scenario, and showed that the proposed approach can be used to transfer skills among different robot embodiments. The skill was demonstrated with a robot manipulator used as teleoperating device, and was then transferred to a completely different octopus-inspired robot. This was achieved by combining several learning strategies based on imitation (inverse optimal control) and self-refinement (stochastic optimization). With the proposed approach, the robot can search for its own ways to match the discovered objectives of the task, by considering its own body characteristics and sensorimotor system.

The proposed learning approach provides a teaching interface to transfer skills that are difficult to achieve or demonstrate (e.g., due to the limits of the teleoperating device). The experiments showed that, even if the model was initialized with sub-optimal demonstrations, the general shape of the candidate objectives activations could still be exploited by the self-refinement mechanism to improve the final score, surpassing the quality of the initially provided demonstrations.

Our plan for future work is to test the proposed approach with larger sets of objective functions, with candidate rewards that are not directly related to the achievement of the task (e.g., smoothness, energy consumption, manipulability), or that are *hidden* objectives that the demonstrator might not be aware of. We will explore how to exploit the retrieved covariance information in the predicted activation of the objectives to address this challenge.

From a broader perspective, the future of medical robotics applications will be characterized by numerous types of robots differing in shapes and capability. We foresee that this ecosystem will require that the robots can teach each others new skills by their own, instead of relying each time on an expert user to re-program each single robot separately. The current research trend in human-robot teaching interaction will thus progressively require to be extended to robot-robot teaching interactions, and to a simultaneous transfer of skills to multiple platforms from the same set of demonstrations (one-to-many instead of one-to-one teaching interaction).

Due to the large variety of robots and to the large spectrum of possible embodiments, the correspondence problem will also likely become a bottleneck for the transfer of skills based only on action-level representations. This might require the development of higher-level forms of imitation capable of extracting and reproducing the intent underlying the demonstrated actions, with an appropriate combination of action mimicry and goal emulation strategies, and a sparse and incremental involvement of human teachers.

Acknowledgements

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

[1] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics, Springer, Secaucus, NJ, USA, 2008, pp. 1371–1394.

[2] B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robot. Auton. Syst. 57 (5) (2009) 469–483.

[3] C. L. Nehaniv, K. Dautenhahn (Eds.), Imitation and social learning in robots, humans, and animals: behavioural, social and communicative dimensions, Cambridge University Press, Cambridge, UK, 2007.

[4] P. Abbeel, A. Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: Proc. Intl Conf. on Machine Learning (ICML), 2004.

[5] N. Ratliff, B. D. Ziebart, K. Peterson, J. A. Bagnell, M. Hebert, A. Dey, S. Srinivasa, Inverse optimal heuristic control for imitation learning, in: Intl Conf. on Artificial Intelligence and Statistics (AIStats), 2009.

[6] M. Lopes, F. Melo, 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.

[7] K. Mombaur, J.-P. Laumond, A. Truong, An inverse optimal control approach to human motion modeling, in: Robotics Research, Vol. 70 of Springer Tracts in Advanced Robotics, 2011, pp. 451–468.

[8] M. Kalakrishnan, P. Pastor, L. Righetti, S. Schaal, Learning objective functions for manipulation, in: IEEE Intl Conf. on Robotics and Automation (ICRA), 2013, pp. 1331–1336.

[9] M. Howard, D. Braun, S. Vijayakumar, Transferring human impedance behaviour to heterogeneous variable impedance actuators, IEEE Transactions on Robotics 29 (4).

[10] A. Jiang, G. Xynogalas, P. Dasgupta, K. Althoefer, T. Nanayakkara, Design of a variable stiffness flexible manipulator with composite granular jamming and membrane coupling, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 2922–2927.

[11] A. Jiang, A. Ataollahi, K. Althoefer, P. Dasgupta, T. Nanayakkara, A variable stiffness joint by granular jamming, in: ASME Intl Design Engineering Technical Conf. & Computers and Information in Engineering Conf. (IDETC/CIE), 2012, pp. 267–275.

[12] J. E. Anderson, D. C. Chang, J. K. Parsons, M. A. Talamini, The first national examination of outcomes and trends in robotic surgery in the united states, Journal of the American College of Surgeons 215 (1) (2012) 107–114.

[13] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, L. Grisoni, SOFA - An Open Source Framework for Medical Simulation, in: Medicine Meets Virtual Reality (MMVR), Palm Beach, FL, USA, 2007.

[14] T. Rueckstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, J. Schmidhuber, Exploring parameter space in reinforcement learning, Paladyn. Journal of Behavioral Robotics 1 (1) (2010) 14–24.

[15] J. Peters, 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.

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

[17] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, E. Dekker, Empirical evaluation methods for multiobjective reinforcement learning algorithms, Machine Learning 84 (1-2) (2010) 51–80.

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

[19] G. D. Konidaris, A. G. Barto, An Adaptive Robot Motivational System, in: Proc. Intl Conf. on Simulation of Adaptive Behavior, Animals to Animats 9, 2006.

[20] K. Gurney, T. J. Prescott, J. R. Wickens, P. Redgrave, Computational models of the basal ganglia: from robots to membranes, Trends in Neurosciences 27 (8) (2004) 453–459.

[21] Z. Ghahramani, M. I. Jordan, Supervised learning from incomplete data via an EM approach, in: J. D. Cowan, G. Tesauro, J. Alspector (Eds.), Advances in Neural Information Processing Systems, Vol. 6, Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1994, pp. 120–127.

[22] S. Calinon, F. Guenter, A. Billard, On learning, representing and generalizing a task in a humanoid robot, IEEE Trans. on Systems, Man and Cybernetics, Part B 37 (2) (2007) 286–298.

[23] C. E. Reiley, E. Plaku, G. D. Hager, Motion generation of robotic surgical tasks: Learning from expert demonstrations, in: Intl Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC), 2010, pp. 967–970.

[24] P. Giataganas, V. Vitiello, V. Simaiaki, E. Lopez, G. Yang, Cooperative in situ microscopic scanning and simultaneous tissue surface reconstruction using a compliant robotic manipulator, in: Proc. IEEE Intl Conf. on Robotics and Automation (ICRA), 2013.

[25] G. J. McLachlan, D. Peel, Finite Mixture Models, Wiley-Interscience, New York, USA, 2000.

[26] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B 39 (1) (1977) 1–38.

[27] C. Wu, On the convergence properties of the EM algorithm, Annals of Statistics 11 (1983) 95–103.

[28] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. of the 5th Berkeley Symp. on mathematical statistics and probability, 1967, pp. 281–297.

[29] S. Schaal, C. G. Atkeson, Constructive incremental learning from only local information, Neural Computation 10 (8) (1998) 2047–2084.

[30] S. Vijayakumar, A. D'souza, S. Schaal, Incremental online learning in high dimensions, Neural Computation 17 (12) (2005) 2602–2634.

[31] D. Nguyen-Tuong, J. Peters, Local Gaussian process regression for real-time model-based robot control, in: IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), 2008, pp. 380–385.

[32] D. B. Grimes, R. Chalodhorn, R. P. N. Rao, Dynamic imitation in a humanoid robot through nonparametric probabilistic inference, in: Proc. Robotics: Science and Systems (RSS), 2006, pp. 1–8.

[33] Y. Tian, L. Sigal, F. De la Torre, Y. Jia, Canonical locality preserving latent variable model for discriminative pose inference, Image and Vision Computing 31 (3) (2013) 223–230.

[34] P. Dayan, G. E. Hinton, Using expectation-maximization for reinforcement learning, Neural Comput. 9 (2) (1997) 271–278.

[35] E. Theodorou, J. Buchli, S. Schaal, A generalized path integral control approach to reinforcement learning, J. Mach. Learn. Res. 11 (2010) 3137–3181.

[36] J. Kober, J. Peters, Imitation and reinforcement learning: Practical algorithms for motor primitives in robotics, IEEE Robotics and Automation Magazine 17 (2) (2010) 55–62.

[37] D. P. Kroese, R. Y. Rubinstein, The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning, Springer New York, 2004.

[38] N. Hansen, The CMA evolution strategy: A comparing review, in: J. Lozano, P. Larranaga, I. Inza, E. Bengoetxea (Eds.), Towards a New Evolutionary Computation, Vol. 192 of Studies in Fuzziness and Soft Computing, Springer Berlin / Heidelberg, 2006, pp. 75–102.

[39] S. Calinon, P. Kormushev, D. G. Caldwell, Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning, Robotics and Autonomous Systems 61 (4) (2013) 369–379.

[40] S. Calinon, A. Pervez, 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.

[41] D. Bruno, S. Calinon, 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.

[42] L. Xu, M. I. Jordan, On convergence properties of the EM algorithm for Gaussian mixtures, Neural Comput. 8 (1) (1996) 129–151.

[43] M. Cianchetti, T. Ranzani, G. Gerboni, I. De Falco, L. C., M. A., STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 3567–3581.

[44] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (2) (1978) 461–464.

[45] M. S. Malekzadeh, D. Bruno, S. Calinon, T. Nanayakkara, D. G. Caldwell, Skills transfer across dissimilar robots by learning context-dependent rewards, in: Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013, pp. 1746–1751.