Malekzadeh *et al. Robotics and Biomimetics* 2014, **1**:13
http://www.jrobio.com/content/1/1/13

Robotics and Biomimetics
a SpringerOpen Journal

RESEARCH Open Access

Learning by imitation with the STIFF-FLOP surgical robot: a biomimetic approach inspired by octopus movements

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

Abstract

Transferring skills from a biological organism to a hyper-redundant system is a challenging task, especially when the two agents have very different structure/embodiment and evolve in different environments. In this article, we propose to address this problem by designing motion primitives in the form of probabilistic dynamical systems. We take inspiration from invertebrate systems in nature to seek for versatile representations of motion/behavior primitives in continuum robots. We take the perspective that the incredibly varied skills achieved by the octopus can guide roboticists toward the design of robust motor skill encoding schemes and present our ongoing work that aims at combining statistical machine learning, dynamical systems, and stochastic optimization to study the problem of transferring movement patterns from an octopus arm to a flexible surgical robot (STIFF-FLOP) composed of two modules with constant curvatures. The approach is tested in simulation by imitation and self-refinement of an octopus reaching motion.

Keywords: Learning from demonstration; Robot-assisted surgery; Continuum robots

Background

Time allowed biological organisms to refine and find smart ways to behave and survive in their environment. Hence, imitation of the models, systems, and elements of nature for the purpose of solving complex human problems (biomimetics) has given rise to new technologies in robotics, both at the level of hardware and software.

In minimally invasive surgery (MIS), widely used in the abdominal operations in the last 30 years [1], tools go through narrow openings and manipulate soft organs to perform surgical tasks. However, due to the limited flexibility and maneuverability of the available tools in MIS, up to four or five trocar accesses are required during operation. To cope with this issue, novel robotic surgical instruments have been developed to give higher flexibility and dexterity to the surgeons [2]. Such systems are actuated by motors moving rods [3], gears [4], or a combination of cables and rods [5]. Yet, stiffness control of such rigid mechanical structures remains a largely unexplored area. The actuation techniques presented in [3-5] are not able to have different level of stiffness in different parts of the robot. In [3], the robot consists of two snake-like robots, one covering the other. They use cable-driven mechanism to stiff the inner robot while the outer longer robot can be compliant at the tip. By using this technique, it is not possible to have a robot compliant at the proximal part and stiff at the distal part. A motor-driven robot has been designed in [4] with two arms and the motors embedded on the arms. This robot provides only stiff configuration at the tip suitable to perform surgical tasks in single-port laparoscopy (SPL).

The STIFF-FLOP project aims at developing biologically inspired manipulators and soft robotic arms to manipulate objects while controlling the stiffness of selected body part [6]. They are inspired by *biological systems* such as an octopus arm or elephant trunk. There are several examples of this kind of robots with high degrees of freedom (DOF) such as the OCTARM continuum robot [7] or the Clemson University elephant trunk [8]. In spite of being inherently compliant when interacting with objects, their rigid nature imposes limitations in their usability in the

*Correspondence: milad.malekzadeh@iit.it
[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genoa, Italy
Full list of author information is available at the end of the article

medical field. Hence, some other manipulators with no rigid structures were developed, e.g., the Air-Octor [9], which combines pneumatic tendon actuation or the arm developed within the OCTOPUS project [10].

None of the mentioned systems answer to the MIS requirements since flexibility and softness come generally with limited force; while high forces are difficult to be achieved without any rigid supports [11]. A different approach has been followed in the STIFF-FLOP project [12] based on variable and controllable stiffness techniques using composite granular jamming, membrane coupling, and pneumatic actuation [13].

There is a wide range of bioimitation mechanisms, ranging from the blind copy of actions (mimicry) to the understanding of the intent underlying an observed action (goal-level imitation), associated with self-training capability required to match the extracted intent. Several approaches in machine learning exist to cover this span of imitative behaviors.

The core idea of acquiring skills by imitation has been labeled over the years with various names such as *teaching by showing* [14], *robot coaching* [15], *programming by demonstration* [16], or *learning from demonstration* (LfD) [17]. A survey of LfD techniques addressing robotics control problems has been presented in [17]. In [16], Billard et al. present an overview of different machine learning approaches in robot programming by demonstration.

In programming by demonstration, providing *good demonstration* plays a key role to speed up the learning process. This is not always easy, specially when the contexts for the demonstration and the reproduction are not sufficiently similar. This may happen, for example, when the structure of the demonstrator agent and the robot are very different. The generalization capability is also often limited to a certain range of situations, which shows the importance of designing representations of skills and tasks that can be trained by various learning strategies. In particular, one key challenge is to allow the imitation system to continuously adapt the learned model when new demonstrations are available, by also exploiting in parallel exploration, auto-calibration, and adaptation strategies. In some situations, the design of efficient *self-refinement* algorithms is crucial for the robot to learn how to perform a task in new situations [16]. A robust and flexible representation of skills and movements is one of the keys to enable robots to jointly exploit several learning methodologies. In this article, we study this problem in the case of biological or robotic systems characterized by a hyper-redundant kinematic chain.

In [18,19], we introduced a high-level skill transfer approach by learning context-dependent reward functions in the same vein as in inverse reinforcement learning (IRL) [20] problems. We first used a statistical approach to extract the underlying intents of a demonstrated action (by kinesthetic teaching with a 7-DOF Barrett WAM), in the form of context-dependent objective functions, and then transferred these high-level goals to the STIFF-FLOP robot by using a stochastic reward-weighted self-refinement strategy so that the robot could find a policy matching the extracted objectives.

In this article, the skill transfer problem from an invertebrate system in nature (octopus) to a STIFF-FLOP robot with two modules is studied, see Figure 1. In particular, we introduce a novel representation of dynamical systems dedicated to continuum structures and study the problem in the case of octopus reaching movements. The proposed approach presents a new form of dynamic motion primitives for continuous bodies that can cope with the hyper-redundancy of the system and the perturbations in the environment.

We use a representation of the octopus arm movement which enables us to transfer motion skills to a STIFF-FLOP robot with two modules. The representation, based on Euler angles and offset values following the standard Denavit-Hartenberg convention in robotics, is invariant to rotation and translation in Cartesian coordinate system.

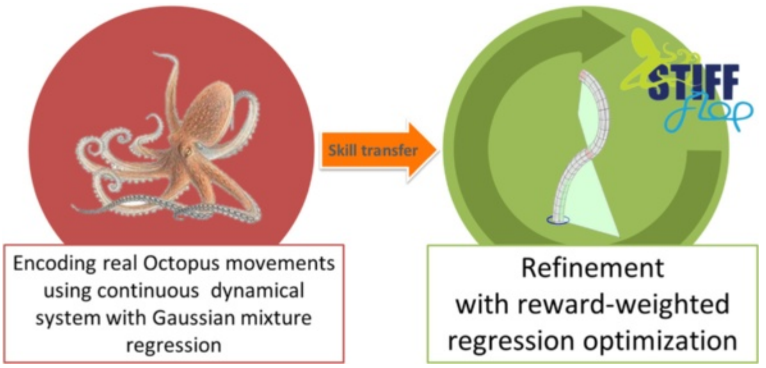
Skill transfer
STIFF
flop
Encoding real Octopus movements using continuous dynamical system with Gaussian mixture regression
Refinement with reward-weighted regression optimization

Figure 1 The STIFF-FLOP robot mimics the octopus reaching motion after skill transfer by using a probabilistic representation based on dynamical systems.

The use of rotation-translation invariant representations is indeed highly preferable if we want to exploit biological motion data in skill transfer applications. Otherwise, the extracted features can hardly be generalized to other situations.

Our approach then exploits the interaction between statistical models and continuous dynamical systems. In particular, the use of statistics allows the system to discover relevant invariants and correlations among the control variables in the form of joint probability distributions, information that is later exploited in the reproduction phase to reproduce appropriate movements when confronted with perturbations.

In this article, we present a first set of experiments to validate that the proposed approach can effectively enable the STIFF-FLOP robot to mimic a motion recorded from an octopus arm.

Octopus reaching motion

Octopuses are considered to be among the most developed and intelligent animals in the invertebrate kingdom, with impressive skills that can be attributed to the high maneuverability of their arms and the capacity of the peripheral nervous system to process sensory information and control arm movements [21]. The octopus can use its arms for various tasks such as locomotion, food gathering, hunting, and sophisticated object manipulation.

Kinematic analyses of octopus reaching and fetching movements have already revealed remarkable control principles underlying movement generation [21-24].

Motion primitives can be regarded as a minimal set of movements, which can be combined in different ways to give rise to rich movement repertoires and facilitating the acquisition of skills [25]. In [21], Zelman et al. demonstrated that the kinematics of octopus arm movements can be described by a reduced set of motion primitives, which required the analysis of different types of arm movements. They showed that a weighted combination of 2D Gaussian functions, in the form of a Gaussian mixture model (GMM), could be used to span arm behavior in 3D space.

The encoding approach that we propose takes inspiration from this work but provides two important novelties. It first extends the approach to a dynamical system to handle perturbations in the environment. It then exploits the GMM in a different way. In [21], each Gaussian is used as a parametric function to fit the local shape of a surface (as bell-shaped components), while it is used in our application to fit the joint distribution of the data. Combined with a regression approach, this allows us to exploit the statistical properties of the model if multiple demonstrations are available. Our work also differs in the sense that the analysis and encoding are not the final targets. Namely, we do not aim at classifying octopus movements or studying movements from a physiological perspective. Instead, we study the problem of designing technical approaches that could be used to transfer movement skills from biological systems to the STIFF-FLOP robot. This goal is reflected by the proposed encoding approach that has closer links to robotics systems, with a reversible representation with three Euler angles and a scalar offset, depending on two input variables (time and arm-index). The representation facilitates retargeting to the robot by representing the two agents as kinematic chains with Denavit-Hartenberg parameters.

In [26], a motion capture system for 3D tracking of octopus arm movements was developed to create a database containing several types of arm movements from octopuses of different ages and sizes. The problem is very challenging. Indeed, the soft property of the octopus arm and its flexible ability to change shape does not allow the use of conventional shape recognition algorithms. In contrast to human motion analysis, it is also not possible to attach markers on the octopus arms as it immediately rejects them [26].

In this paper, we exploited one of the most representative reaching movement from this database, gratefully provided by Prof. Binyamin Hochner, Shlomi Hanassy, and Alexander Botvinnik from the Department of Neurobiology, Hebrew University, Jerusalem, Israel, and Prof. Tamar Flash from the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel.

Recording and storing such data require discrete sampling in both time and space. The raw data consists of discrete Cartesian position data with (on average) 100 points along the octopus arm and (on average) 100 time steps. To reduce the computational time, we re-sampled the data in both arm-index and time with 50 instances (surface of 2,500 points).

Methods

Spatiotemporal representation of octopus movements

In order to generate a mathematical model of the octopus movements, the arm was approximated as a robot with a high number of links. The continuous arm is approximated by a kinematic chain with a high number of revolute joints, describing the local curvature and torsion, alternated with prismatic joints describing the local elongation. The forward kinematics of such system can thus be described by a set of three Euler angles and a scalar offset. The *ZYX* decomposition of the Euler angles resulted in the most appropriate selection for the motion range in our dataset. This representation is reversible; i.e., by having the angles and offset, the points in Cartesian space can be retrieved and vice-versa.

We will define as continuous *arm-index* $s \in [0, 1]$ the position of any point along an arm, with $s = 0$ and $s = 1$

representing respectively the base and the tip. We will rescale the duration of a movement so that it can be represented by a continuous *time index* $t \in [0, 1]$. We thus have three Euler angles θ_x, θ_y, and θ_z and an offset ΔL for each t and s that can be represented as a set of surfaces as shown in Figure 2.

The original raw data (noisy Cartesian positions) are first resampled and smoothed through a two-dimensional polynomial fitting (surface fitting) with a 7-degree polynomial. The degree for the polynomial is set experimentally by testing different orders. The arm movement is translated to keep the base of the octopus at the origin. The data is then globally rescaled to match the size of the robot.

Dynamical system with evolution over time and arm-index

Dynamical movement primitives (DMP) is a popular tool in robotics to learn and reproduce movements while being robust to perturbations [27]. It consists of a spring-damper system modulated by a non-linear force profile encoded as a series of predefined basis functions associated with local force behaviors (or alternatively, by the path of a virtual spring-damper system producing a non-linear force profile). We introduced in [28] a probabilistic formulation of DMP by jointly learning the basis functions and the local force behaviors. The trajectory of the attractor point is then encoded by means of a statistical model (as explained later), whence the trajectory can be later extracted by regression.

The novelty in the current work is the use of surface attractors instead of trajectory attractors (generalization of the generic spring-damper system to a spatiotemporal dynamical system). We use both arm-index s and time t (rather than only time) as input variables, which enables the approach to encode the movement of invertebrate arm movements with a compact model (namely, by encoding the movement of all points along the arm).

The motion is reconstructed from the dynamical systems with

$$\ddot{\boldsymbol{x}} = \kappa^{\mathcal{P}} \left[\boldsymbol{y} - \boldsymbol{x}\right] - \kappa^{\mathcal{V}} \dot{\boldsymbol{x}} \quad \Rightarrow \quad \boldsymbol{y} = \frac{1}{\kappa^{\mathcal{P}}}\ddot{\boldsymbol{x}} + \frac{\kappa^{\mathcal{V}}}{\kappa^{\mathcal{P}}}\dot{\boldsymbol{x}} + \boldsymbol{x}, \quad (1)$$

where $\kappa^{\mathcal{P}}$ and $\kappa^{\mathcal{V}}$ are respectively the stiffness of the spring and the damping ratio of the damper. $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ are the velocity and acceleration calculated by differentiating $\boldsymbol{x}$ for each s with respect to t. In other words, the observations $\boldsymbol{x}$ of the motion are converted into the trajectory of virtual attractors $\boldsymbol{y}$ for each arm-index. Similarly as $\boldsymbol{x} = \left[\theta_z, \theta_y, \theta_x, \Delta L\right]$ in Figure 2, $\boldsymbol{y}$ can thus be represented as a set of attractor surfaces.

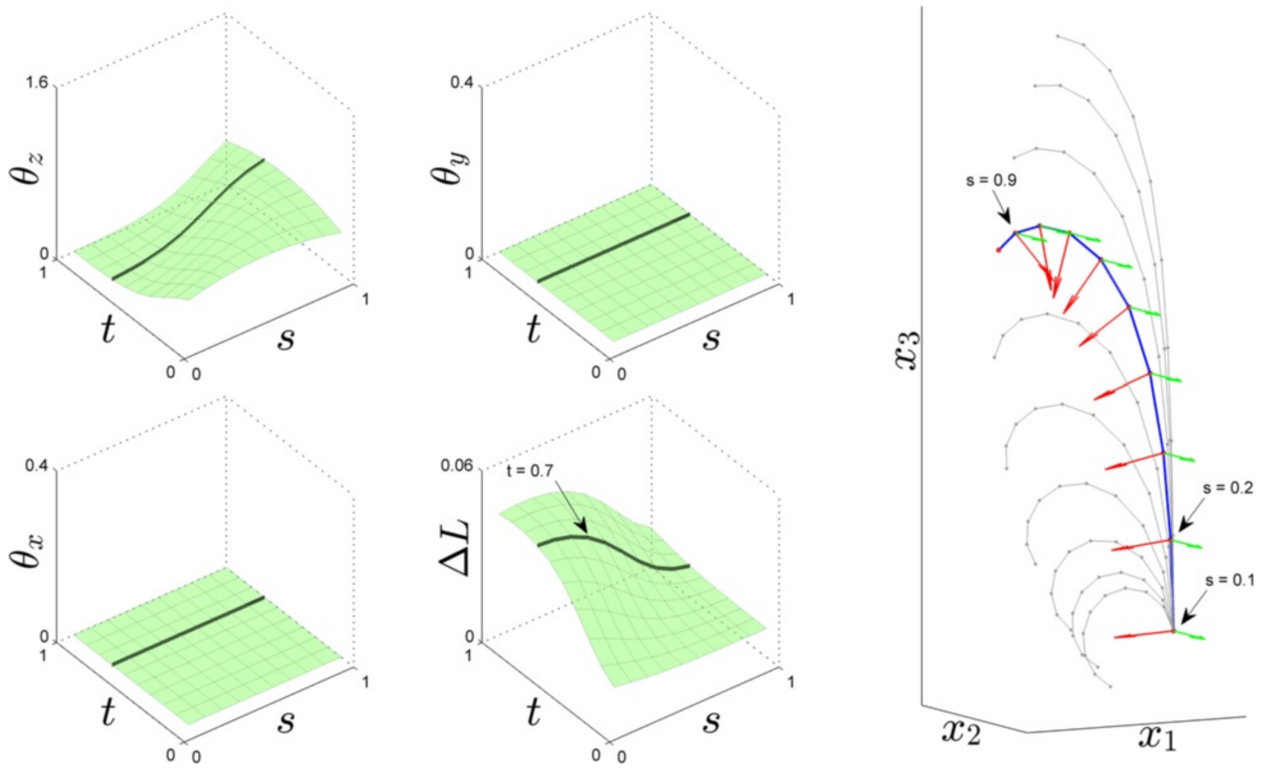

Figure 2 Spatiotemporal representation. *Left:* The black lines represent slices in the Euler angle surfaces $\boldsymbol{\theta}$ and offset surface $\boldsymbol{\Delta L}$, corresponding to a static pose described by all the links ($0 \leq s \leq 1$) at $t = 0.7$. *Right:* For the same time frame, some of the corresponding Frenet frames along the arm are depicted in a 3D Cartesian space.

An illustration of a continuum arm is presented in Figure 3 by using the proposed dynamical system with evolution over time and arm-index[a].

A probabilistic model of continuous movements

An *expectation-maximization* (EM) algorithm [29] is used to fit a Gaussian mixture model (GMM) to the augmented attractor dataset $\boldsymbol{\xi} = [t, s, \boldsymbol{y}]$, which is described as

$$\mathcal{P}(\boldsymbol{\xi}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

$$\text{with} \quad \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \times \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\xi} - \boldsymbol{\mu}_k)\right),$$

where D is the dimension of variables and K is the number of Gaussian components.

The parameters of the GMM including the priors π_k, centers $\boldsymbol{\mu}_k$, and covariance matrices $\boldsymbol{\Sigma}_k$ are learned by iteratively performing the following steps until convergence

$$\begin{aligned}
\textit{E-step:} \quad h_i\left(\boldsymbol{\xi}_j\right) &= \frac{\pi_i\,\mathcal{N}\left(\boldsymbol{\xi}_j|\,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)}{\sum_{k=1}^{K}\pi_k\,\mathcal{N}\left(\boldsymbol{\xi}_j|\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)},\\
\textit{M-step:} \quad \pi_i &\leftarrow \frac{\sum_{j=1}^{N} h_i\left(\boldsymbol{\xi}_j\right)}{\sum_{k=1}^{K}\sum_{j=1}^{N} h_k\left(\boldsymbol{\xi}_j\right)},\\
\boldsymbol{\mu}_i &\leftarrow \frac{\sum_{j=1}^{N} h_i\left(\boldsymbol{\xi}_j\right)\boldsymbol{\xi}_j}{\sum_{j=1}^{N} h_i\left(\boldsymbol{\xi}_j\right)},\\
\boldsymbol{\Sigma}_i &\leftarrow \frac{\sum_{j=1}^{N} h_i\left(\boldsymbol{\xi}_j\right)\left(\boldsymbol{\xi}_j-\boldsymbol{\mu}_i\right)\left(\boldsymbol{\xi}_j-\boldsymbol{\mu}_i\right)^\top}{\sum_{j=1}^{N} h_i\left(\boldsymbol{\xi}_j\right)},
\end{aligned} \tag{2}$$

where N is the number of datapoints. In the *E-step*, $h_i\left(\boldsymbol{\xi}_j\right)$ is the posterior probability $\mathcal{P}\left(k|\boldsymbol{\xi}_j\right)$, computed by the

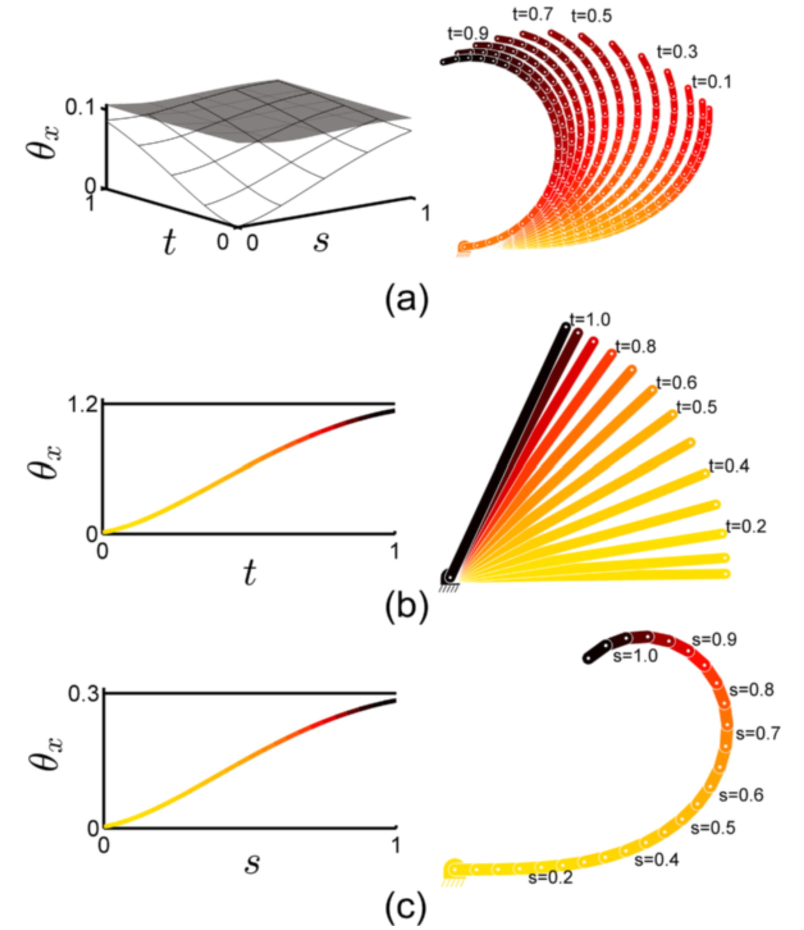

Figure 3 A dynamical system with surface attractor (evolution over time and arm-index). (a) The grey surface on the left represents the attractor surface corresponding to the observed motion of the continuum arm, while the white surface is the reproduced motion of the arm. The right figure shows the arm configurations in 2D Cartesian space at different time steps. **(b)** The left figure shows the evolution in time of a link with a given arm-index *s* along the kinematic chain and the corresponding configurations (on the right). **(c)** The figures show the pose of the continuum arm for a given time step *t*.

Bayes theorem. This is the probability that the Gaussian component k is responsible for $\boldsymbol{\xi}_j$. EM guarantees the improvement of the likelihood at each iteration, converging to a local optimum [30]. The EM algorithm is initialized with *k-means* clustering[b] [31] to start the iterative procedure from a good initial estimate. Here, K is determined empirically.

Gaussian mixture regression (GMR) is then used to estimate the conditional probability $\mathcal{P}\left(\boldsymbol{y}|s,t\right)$ with a Gaussian distribution $\mathcal{N}\left(\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right)$, with a fast computation independent from the number of datapoints used to train the model [32]. In our case,

$$\hat{\boldsymbol{\mu}} = \sum_{i=1}^{K} h_i\left(\boldsymbol{\xi}^{\boldsymbol{I}}\right)\left[\boldsymbol{\mu}_i^O + \boldsymbol{\Sigma}_i^{OI}\boldsymbol{\Sigma}_i^{I^{-1}}\left(\boldsymbol{\xi}^{\boldsymbol{I}} - \boldsymbol{\mu}_i^I\right)\right],$$
$$\text{and}\quad \hat{\boldsymbol{\Sigma}} = \sum_{i=1}^{K} h_i^2\left(\boldsymbol{\xi}^{\boldsymbol{I}}\right)\left[\boldsymbol{\Sigma}_i^O - \boldsymbol{\Sigma}_i^{OI}\boldsymbol{\Sigma}_i^{I^{-1}}\boldsymbol{\Sigma}_i^{IO}\right], \tag{3}$$

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

$$\boldsymbol{\mu}_i = \begin{bmatrix}\boldsymbol{\mu}_i^I\\ \boldsymbol{\mu}_i^O\end{bmatrix},\qquad \boldsymbol{\Sigma}_i = \begin{bmatrix}\boldsymbol{\Sigma}_i^I & \boldsymbol{\Sigma}_i^{IO}\\ \boldsymbol{\Sigma}_i^{OI} & \boldsymbol{\Sigma}_i^O\end{bmatrix},$$
$$\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{\xi}^{\boldsymbol{I}}\\ \boldsymbol{\xi}^{\boldsymbol{O}}\end{bmatrix},\quad \boldsymbol{\xi}^I = \begin{bmatrix}s\\ t\end{bmatrix},\quad \boldsymbol{\xi}^O = \boldsymbol{y},$$

where the superscripts ^{I} and ^{O} represent input and output variables.

In contrast to other regression methods, GMR does not model the regression function directly, but models the joint probability of the data. It then derives the regression function from the joint probability model. Density estimation can thus be learned in an off-line phase (with the GMM encoding approach presented above) and extracted in real time at reproduction, since GMR has a low computation time independent from the number of datapoints used to train the model.

Figure 4 illustrates the encoding and retrieval process. For regression, any subset of multivariate input and output dimensions can be selected, which can change in the course of the reproduction. Expectations on the remaining dimensions can be computed in real time, corresponding to a convex sum of linear approximations (with weights varying non-linearly). GMR can thus handle different sources of missing information, since the system is able to consider any combination of input/output mappings during reproduction.

Given the attractor surfaces $\boldsymbol{y}$, the whole motion can then be directly reconstructed by a double integration over t as

$$\dot{\boldsymbol{x}} = \int_t \left(\kappa^{\mathcal{P}}\left[\boldsymbol{y}-\boldsymbol{x}\right] - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}\right)dt \quad , \quad \boldsymbol{x} = \int_t \dot{\boldsymbol{x}}\,dt\,. \tag{4}$$

We will use the term DS-GMR to refer to this combination of dynamical system and Gaussian mixture regression.

Self-refinement with reward-weighted regression

The model parameters above can then be optimized through a self-refinement algorithm with an objective or reward function defined according to the different objectives of the task. The self-refinement process consists of searching for new solutions in the policy parameter space, associate a reward to each trial, and then sample from the exploration space by a weighted combination of the most successful trials obtained so far.

Dayan and Hinton originally suggested that a self-refinement problem can be tackled by EM to avoid gradient computation [33]. They introduced the idea of

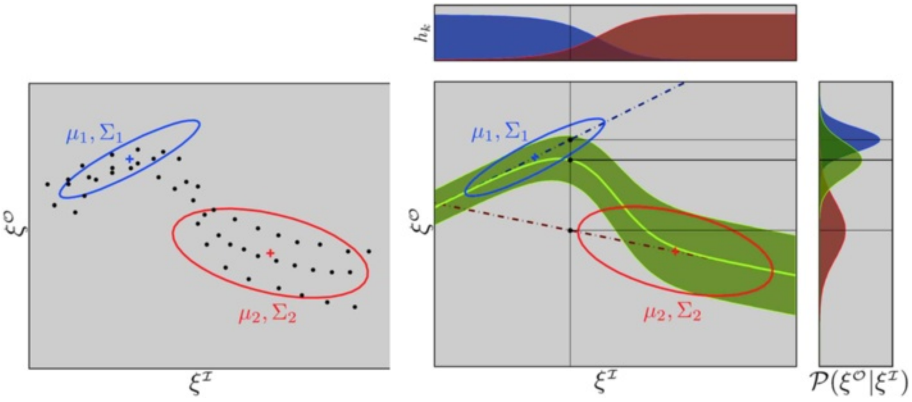

Figure 4 Illustration of the mapping representation as a mixture model and retrieval process. *Left:* Encoding of the input-output observations in a GMM. *Right:* Probabilistic estimation of output data from given input data with GMR.

treating immediate rewards as probabilities of a fictitious event, in which case probabilistic inference techniques can be used for optimization. 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 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 [34-38]. Indeed, several research fields converged to similar algorithmic solutions, with approaches such as the *cross-entropy method* (CEM) [39] or the *covariance matrix adaptation evolution strategy* (CMA-ES) [40].

One option for implementing such self-refinement is to use an EM-based stochastic optimization algorithm to refine the GMM parameters encoding the movement, similarly as in [18]. The procedure corresponds to an EM algorithm in which the reward signal is treated as a pseudo-likelihood, which can easily be extended to multi-optima policy search [41].

Another line of research is to explore methodologies that could move toward more structured techniques of exploration. In [42], it was proposed to speed up the search process by redefining it as an iterative reward-weighted regression problem. This is particularly relevant for the subclass of problems in which we have access to the goal or to the highest value that a reward can take (e.g., reaching the center of a target, be as close as possible to a reference trajectory, etc.).

In this case, the reward (or objective) is expressed in the form of a vector, and the aim of the optimization is treated as an iterative reward-weighted regression problem in the augmented space formed by the policy parameters and the achieved goals. At each iteration, a Gaussian distribution showing the joint probability is fit to the augmented data containing the best policies and the corresponding goals obtained so far. The input of the regression problem is represented by the desired goal $\boldsymbol{\zeta}^{I*}$. The output is a candidate optimal policy that is tested on the system and that will be associated with the corresponding outcome.

An illustrative example of the reward-weighted regression algorithm with 1D policy and goal is shown in Figure 5.

The augmented data is defined as $\boldsymbol{\zeta} = \left[\boldsymbol{\zeta}^{I\top}, \boldsymbol{\zeta}^{O\top}\right]^\top$. At each iteration, the ordered set of datapoints $\left\{\boldsymbol{\zeta}_m\right\}_{m=1}^{M}$ with $r\left(\boldsymbol{\zeta}_1\right) \geqslant r\left(\boldsymbol{\zeta}_2\right) \geqslant \ldots \geqslant r\left(\boldsymbol{\zeta}_M\right)$ is used as a form of importance sampling [37] to estimate a Gaussian distribution with parameters

$$\begin{aligned} \tilde{\boldsymbol{\mu}} &\leftarrow \frac{\sum_{m=1}^{M} r\left(\boldsymbol{\zeta}_m\right)\ \boldsymbol{\zeta}_m}{\sum_{m=1}^{M} r\left(\boldsymbol{\zeta}_m\right)}, \\ \tilde{\boldsymbol{\Sigma}} &\leftarrow \frac{\sum_{m=1}^{M} r\left(\boldsymbol{\zeta}_m\right)\ \left(\boldsymbol{\zeta}_m - \tilde{\boldsymbol{\mu}}\right)\left(\boldsymbol{\zeta}_m - \tilde{\boldsymbol{\mu}}\right)^\top}{\sum_{m=1}^{M} r\left(\boldsymbol{\zeta}_m\right)} + \boldsymbol{\Sigma}_0. \end{aligned} \tag{5}$$

where $\boldsymbol{\Sigma}_0$ is a predefined minimal exploration noise. With the joint probability $\mathcal{P}\left(\boldsymbol{\zeta}\right) \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}\right)$ in Eq. (5), the conditional probability $\mathcal{P}\left(\boldsymbol{\zeta}^O|\boldsymbol{\zeta}^{I*}\right)$ can then be computed to obtain a subsequent policy (see Eq. (3)), by stochastic sampling from the retrieved Gaussian distribution in the policy parameter space.

Thus, at each iteration of the algorithm, a new policy $\boldsymbol{\zeta}^O$ is evaluated and associated with the achieved goal $\boldsymbol{\zeta}^I$ and the corresponding scalar reward r (calculated through a reward function of exponential form). This iterative process continues until convergence or a maximum number of iterations is attained.

Similarly to EM-based stochastic optimization, the process can be extended to a GMM/GMR representation for multi-policy search [41].

In the proposed experiment, the process starts from a set of randomly generated policies, where the initial number has been determined empirically. The objective function is also explicitly defined.

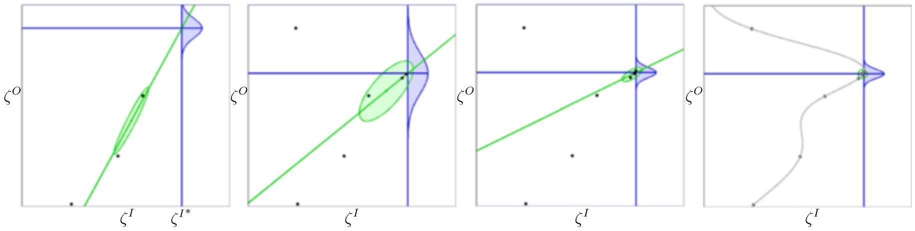

Figure 5 An example of reward-weighted regression. *(From left to right)* Subset of iterations during the refinement algorithm. At each iteration, a new Gaussian distribution, depicted by the green ellipse, is fit to the most promising augmented dataset (policy parameters and goal). A regression-based exploration strategy is then used in the augmented-space $\boldsymbol{\zeta}$ to iteratively find better policies to achieve the goal. For the regression, we assume that we know the desired goal (blue vertical bar $\boldsymbol{\zeta}^{I*}$) but we do not know how to achieve it (namely, the input of the regression is the desired goal).

STIFF-FLOP robot kinematics

The first prototype of the STIFF-FLOP robot, currently under development, will be composed of two cylindrical modules. Each section 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.

In this section, we briefly summarize the forward and inverse kinematics of the STIFF-FLOP robot (see [18] for more details). We will use a constant curvature model to derive the forward kinematics along the whole kinematic chain (see Figure 6). The inverse kinematics is developed in a modular way, so that any number of links can be used within the model. We use a two-link robot for the experiment but the approach can be scaled to a higher number of modules.

In the local frame of each module, 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, see Figure 6-*left*.

The position $\boldsymbol{Q}_i$ of the tip of the i-th module can be written as a function of the angle α_i, the arc length β_i, and the curvature radius ρ_i (see Figure 6-*center*) as

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

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

$$\rho_i = \frac{Q_{i,1}^2 + Q_{i,2}^2 + Q_{i,3}^2}{2\sqrt{Q_{i,1}^2 + Q_{i,2}^2}}, \quad \alpha_i = \arctan\frac{Q_{i,2}}{Q_{i,1}},$$

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

The constant curvature coordinates allow us to obtain the position of any point along the single module. 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 module are then given by

$$\boldsymbol{F}_{\rho_i,\alpha_i,\beta_i}(\gamma) = [\rho_i (1 - \cos(\beta_i\gamma)) \cos(\alpha_i), \\ \rho_i (1 - \cos(\beta_i\gamma)) \sin(\alpha_i), \; \rho_i \sin(\beta_i\gamma)]. \tag{6}$$

Here, γ corresponds to all possible points from the base of the module 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 angle commonly used as internal variables in kinematic models of standard manipulators.

The constant curvature model also allows us to evaluate the orientation of the tip that 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 orientation of the i-th module in the $(i-1)$-th tip frame is defined by

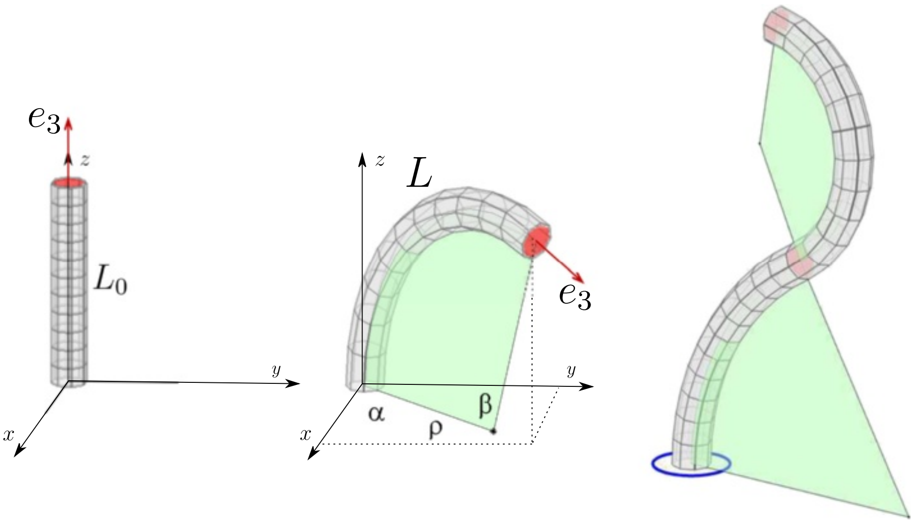

Figure 6 Model of the STIFF-FLOP robot. *(Left)* Single module of a constant curvature model at rest position with the rest length L_0, *(center)* with the pose of the tip described as a function of α, β, and L. *(Right)* STIFF-FLOP robot with two modules.

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

The use of the 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 module 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, a 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 joint 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 joint 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 two links, this results into $\boldsymbol{y} = \boldsymbol{Q}_1 + \boldsymbol{R}_{01}\boldsymbol{Q}_2$ and $\boldsymbol{R}_{03} = \boldsymbol{R}_{01}\boldsymbol{R}_{12}$.

Results and discussion

The model is first tested to emulate two typical octopus reaching motion primitives by manually setting the model parameters. It is then tested if these parameters could be autonomously learned from real octopus motions and transferred to the STIFF-FLOP robot.

Emulation of octopus motion primitives

The model is first employed to emulate typical movements used by octopus to reach for food. These movements involve bend propagation and elongation of the proximal part of the arm. Note here that the reaching motion differs from the *fetching* of the grasped food to the mouth, where the octopus arm is configured into a quasi-articulated structure with three segments and three rotary joints [22,24,43]. In contrast, *reaching* movements are usually generated by a combination of two motor primitives: the propagation of a bend along the arm and an elongation of the arm [25,44]. These two motor primitives may be combined with different weights to create a broader spectrum of reaching movements.

The advantage of the DS-GMR representation to encode such kind of patterns is that it provides not only information about individual bending/elongation patterns but also information about their local correlations throughout the movement, in the form of a full covariance matrix.

The planar bend propagation movement from the base to the tip can easily be simulated by setting non-zero off-diagonal elements in a full covariance matrices (tilted elongated ellipsoids) acting as attractors in a curvature-elongation space. For example, for planar movements, one of the Euler angles and offset (e.g. θ_x and ΔL) can represent the curvature-elongation space, see Figure 7a-c. An elongation primitive can easily be obtained by translating one of the Gaussians in the elongation space, see Figure 7d-f.

Imitation of octopus motion by the STIFF-FLOP robot

In this section, we test if the proposed encoding technique can be used to autonomously transfer skills from real octopus movements to the STIFF-FLOP robot.

Figure 8 shows the recording of the representative octopus reaching movement that was selected from the database (see '*Octopus reaching motion*' section for details). We can observe that the angle θ_x has a wider range than the others, suggesting that the octopus reaching movement is mostly achieved in a plane.

The DS-GMR model is exploited to transfer the movement to a two-link STIFF-FLOP robot. The constant curvature constraint of each link results in a segmentation of the continuous surfaces into a given number of subsurfaces, corresponding to the number of modules (here, two).

In order to map the movement to the STIFF-FLOP robot, we consider the Frenet frames at the base of the robot ($\boldsymbol{R}_1$), in the middle point between the two modules $\left(\boldsymbol{R}_{n/2}\right)$, and at the tip $\left(\boldsymbol{R}_{n/2}\right)$, where n is the number of segments used for the discretization of the continuous kinematic chain.

These frames can be easily calculated from the Euler angle representation at each time step t. The elongation of each module can be calculated by summing the elongation of the $n/2$ segments composing each module. The frames

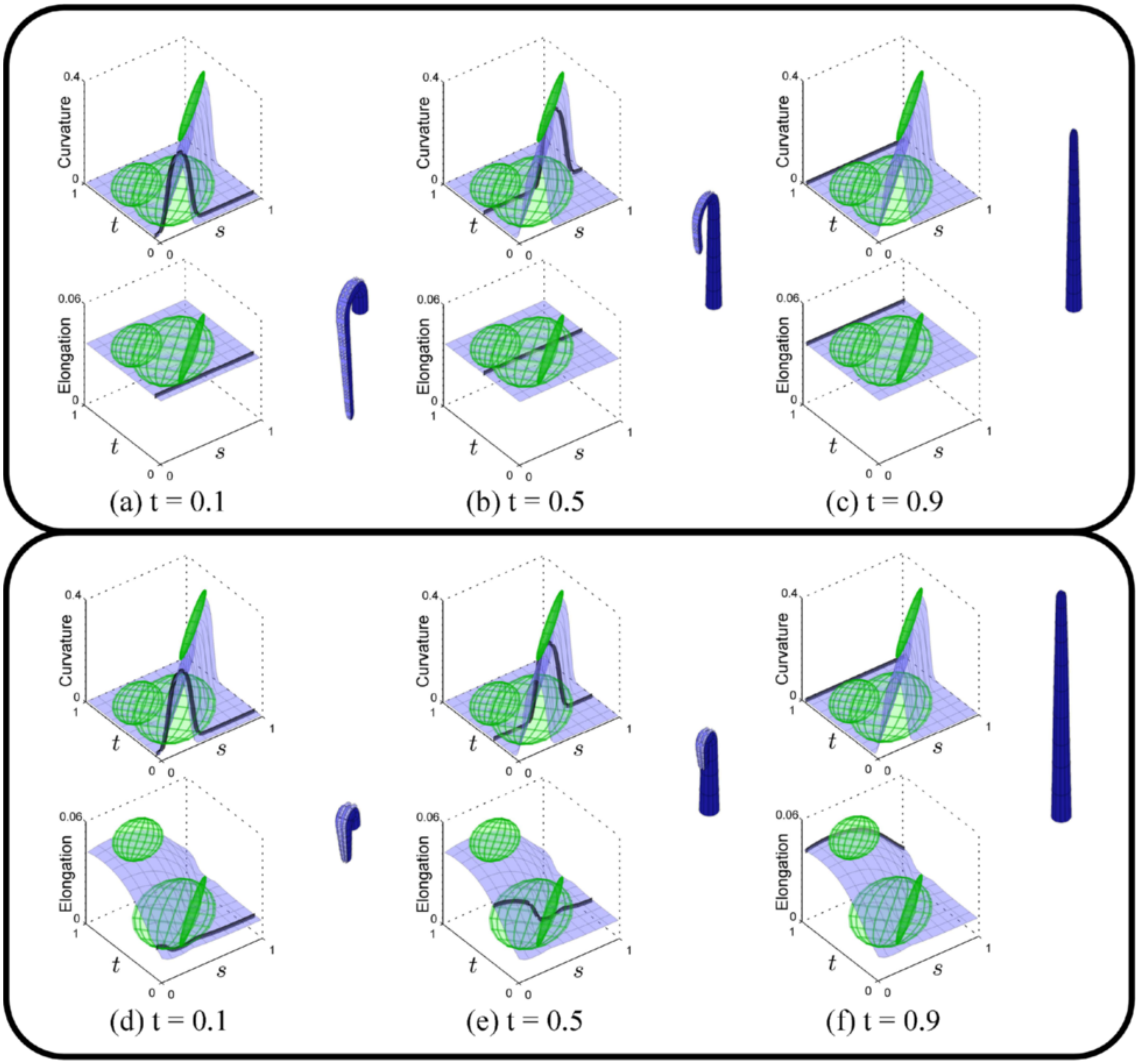

Figure 7 Emulation of reaching movements. Emulation of reaching movement with bend propagation **(a-c)** and with bend elongation **(d-f)**.

$\boldsymbol{R}_1$, $\boldsymbol{R}_{n/2}$, and $\boldsymbol{R}_n$ are then used to approximate the corresponding STIFF-FLOP module variables $\left[L_j, \alpha_j, \beta_j\right]_{j=1}^{2}$, see Figure 6. This is achieved by considering the rotation matrix $\boldsymbol{R}$, corresponding to the frame at the tip of each module, defined as

$$\text{For the first module :} \quad \boldsymbol{\mathcal{R}} = \boldsymbol{R}_1^\top \boldsymbol{R}_{n/2} \,,$$

$$\text{For the second module :} \quad \boldsymbol{\mathcal{R}} = \boldsymbol{R}_{n/2}^\top \boldsymbol{R}_n \,,$$

and calculating the constant curvature variables for each module as

$$\alpha = \arctan\left(\mathcal{R}_{23}/\mathcal{R}_{13}\right), \quad \beta = \arccos\left(\mathcal{R}_{33}\right) \quad \text{and} \quad L = \sum_{i=1}^{n/2} \Delta L_i \,,$$

where, for example, $\mathcal{R}_{23}$ is the scalar in the second row and third column of matrix $\boldsymbol{\mathcal{R}}$, and ΔL is the offset.

The result of this procedure is shown in Figure 9, where Euler angles and offsets corresponding to the octopus (light gray) and STIFF-FLOP (green and red for the first and second modules, respectively) are presented.

This approach for the transfer of skills from a continuous arm to an arm with piecewise constant curvatures can provide a good initial estimate but does not guarantee the best fit. Indeed, even though the movement is quite similar, the different structure of the STIFF-FLOP robot can cause some dissimilarity in the learned movement.

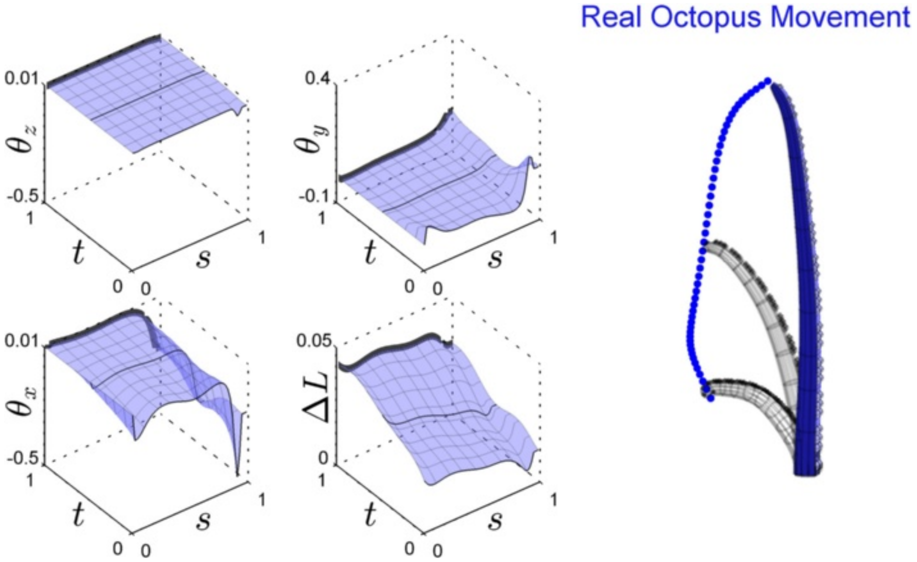

Figure 8 An example of octopus reaching movement after pre-processing of the data. The black lines on the surfaces show configurations at three different time steps (initial, mid, and final configurations), with corresponding octopus arm pose in a 3D Cartesian space depicted in the right graph. The blue dots show the position of the tip during the movement.

The self-refinement approach is thus used to refine this initial estimate, by using as policy parameters the centers of the GMM, as well as the first eigencomponents of ordered eigendecompositions of the covariances (directions of the main axes of the ellipsoids).

Figure 10 shows a GMM model with three Gaussians fitted to the attractor surfaces representing the movement. By considering the centers and the first eigencomponent of the covariance matrix for these three Gaussians, the policy has 36 dimensions. The stiffness and damping

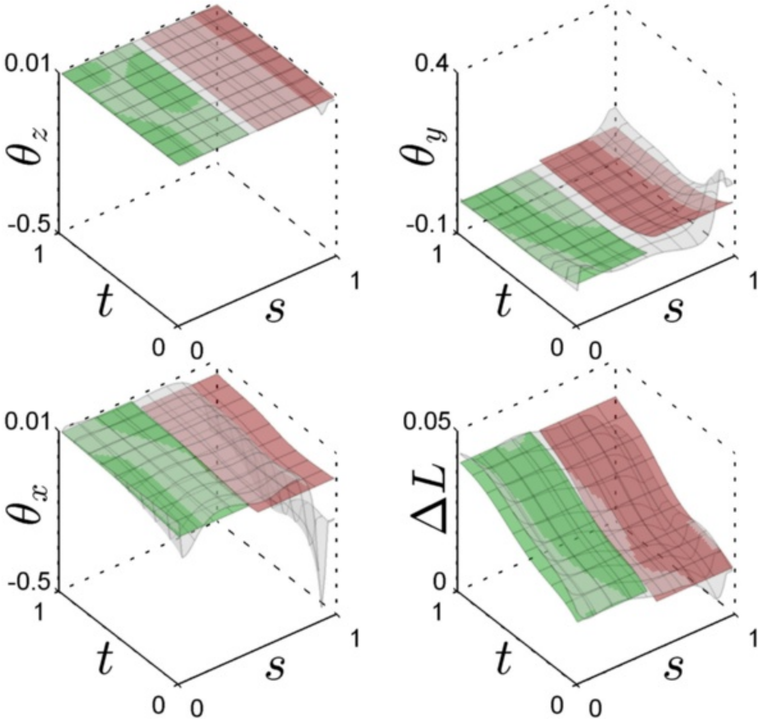

Figure 9 Mapping from the octopus to the STIFF-FLOP robot. The Euler angles and offsets related to the octopus movement (light gray) and the corresponding STIFF-FLOP robot with two modules (green and red surfaces).

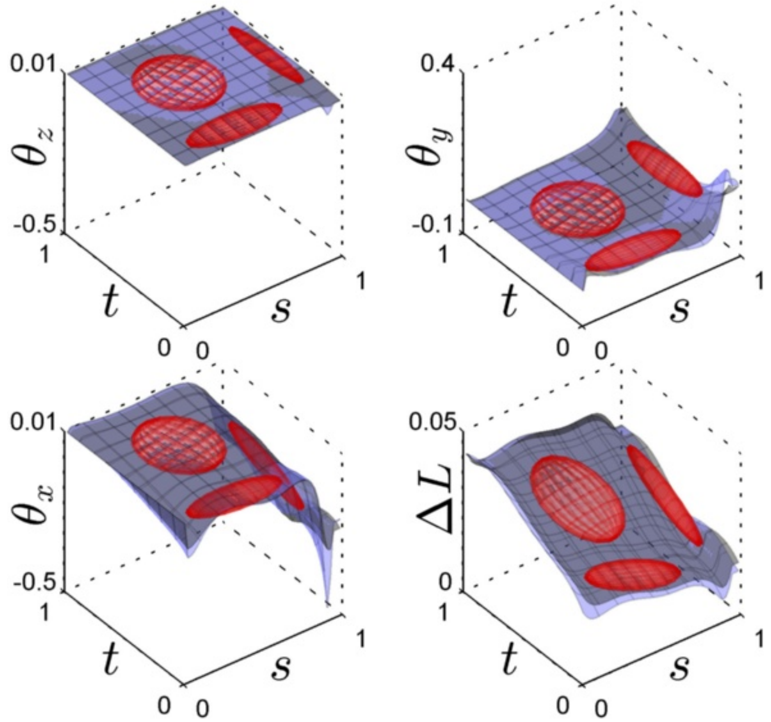

Figure 10 Modeling of surface attractors. The complete set of variable $\boldsymbol{x}$ and the attractor $\boldsymbol{y}$ are plotted in light blue and gray colors, respectively. The Gaussian components of the GMM model are the red ellipsoids fitting the attractor surfaces.

parameters are set empirically with a critical damping constraint.

A reward function based on the Cartesian distance of the STIFF-FLOP end effector ($\boldsymbol{x}^r$) and the octopus tip ($\boldsymbol{x}^o$) is defined at time instances $t = 0.2$, $t = 0.5$, and $t = 1$ as

$$r = \frac{1}{3} \sum_{t=0.2,0.5,1} \exp\left(-\delta||\boldsymbol{x}^r(t) - \boldsymbol{x}^o(t)||\right),$$

where δ is a bandwidth coefficient set experimentally. With three 3D points at different time instances, the goal has nine dimensions.

In order to keep the robot parameters within the real hardware limits, we additionally considered hard constraints in the search process for each of the two modules as

$$L_0 \leq L \leq 1.8L_0 \quad \text{and} \quad \beta \leq 2\pi/3,$$

where L_0, L, and β are the module minimum length, length, and curvature, respectively (see Figure 6).

The results of this experiment are shown in Figures 11 and 12, averaged over 30 runs of the same experiment. Figure 13 shows the convergence of the reward after 10-20 self-refinement iterations, by starting from an initial set of 20 randomly generated policies based on the initial demonstration.

Conclusions

We presented an approach based on statistical dynamical systems to encode movements in biological or robotic systems with a continuous kinematic structure. We showed that the approach could be employed to learn skills from demonstration, and that it could be combined with a self-refinement strategy based on iterative reward-weighted regression.

The aim is to extract relevant motion primitives from biological systems and map those to the STIFF-FLOP robot, while creating a set of natural motion patterns as building blocks. These building blocks could later be combined and reorganized differently to build new types of movements and skills.

The case study with octopus reaching movements, of course, does not target as a final goal to replicate reaching movements in surgery. But such typical motion repertoire allows us to test the efficiency of the proposed encoding approach and to exploit existing databases of biological movements. Even if the same exact movements will not be used in the surgical application, it provides us with a very important starting point to the design of robust models capable of encoding various skills in flexible continuum robots. In other words, the presented work is primarily aimed at exploiting typical behaviors observed in invertebrate systems to test an encoding strategy and guide us toward the design of new modeling techniques that

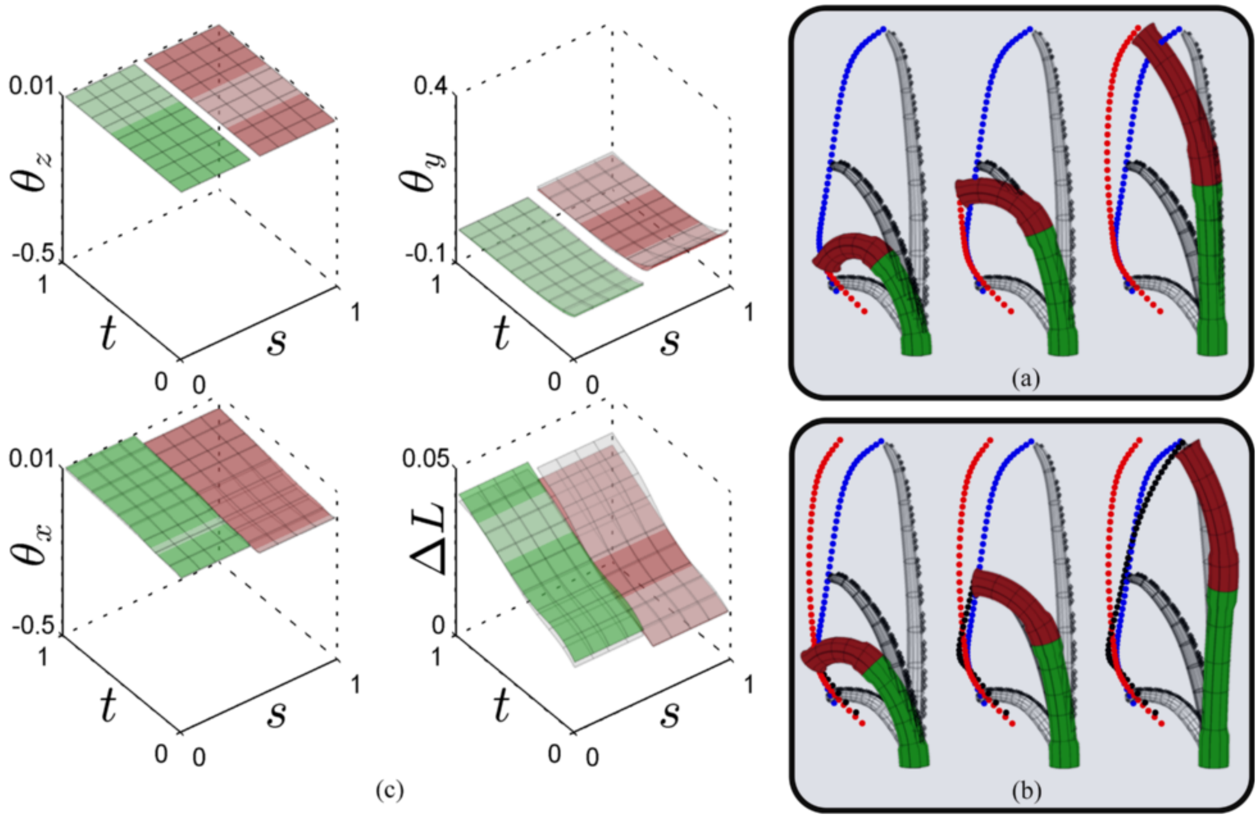

Figure 11 Qualitative comparison of the octopus and STIFF-FLOP robot movements, before and after self-refinement. (a) Comparison of the octopus and STIFF-FLOP robot movements based on the initial learning from demonstration approach (before self-refinement) at time instances $t = 0.2$, $t = 0.5$ and $t = 1$. The red dots represent the tip trajectory. **(b)** Reproduction on the robot after self-refinement. The black dots depict the refined trajectory of the tip. **(c)** The initial (light gray) and refined (green and blue surfaces) control variables.

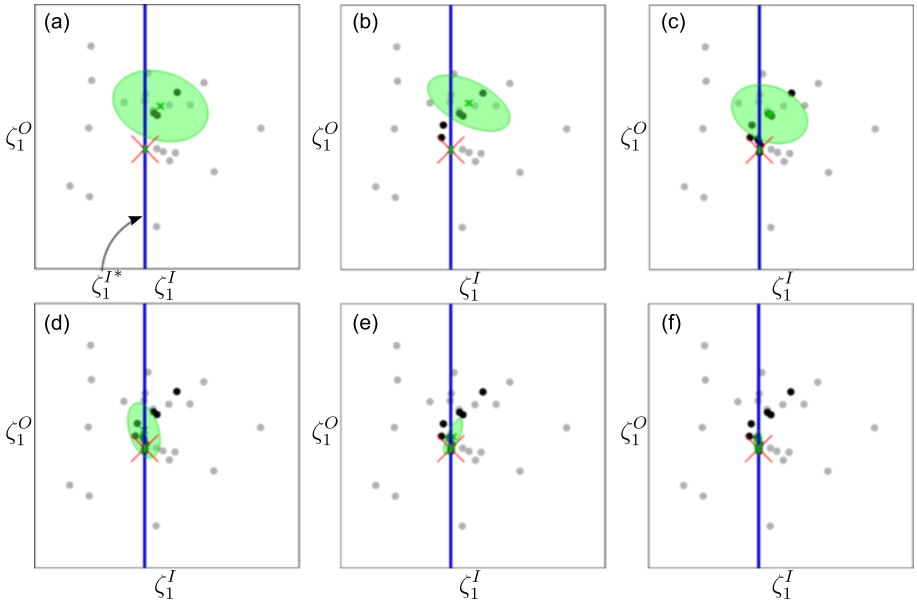

Figure 12 Convergence of the GMR search process. (a-f) A sample search in the augmented goal-policy space $\boldsymbol{\zeta}$ is plotted for six selected self-refinement iterations. Here, only one dimension for $\boldsymbol{\zeta}^O$ and $\boldsymbol{\zeta}^I$ is depicted, but we have $\boldsymbol{\zeta}^O \in \mathbb{R}^{36}$ and $\boldsymbol{\zeta}^I \in \mathbb{R}^9$. The gray dots show the initial set of iterations and the black dots are the new estimated policy. The blue line represents the known goal of the task with highest reward on which the red cross shows the best achieved policy after convergence.

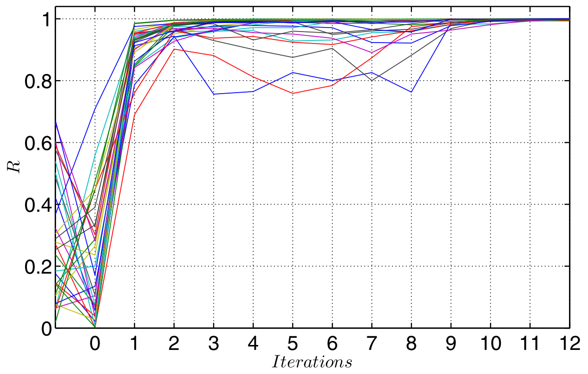

Figure 13 Reward profile for 30 runs, plotted individually.

could be used to encode a broader range of skills and patterns, including new skills that will be relevant to surgical applications.

In future work, we aim at applying this learning approach to the real STIFF-FLOP platform, by considering diverse motion/feedback skill primitives that could be used to assist doctors in surgical operations. We plan to exploit the context-dependent learning interface that we proposed in [18] as a high-level imitation approach to extract and exploit the most relevant cost functions that could explain the observation of octopus movements. In particular, we plan to study how the octopus exploits the degrees of freedom of its flexible arm to perform various tasks, in the form of objective functions that could vary depending on the ongoing situation. The variability information could then be exploited to regulate the stiffness and damping of the robot, with appropriate synergies among curvature, torsion, and elongation variables.

Endnotes

[a] Note that in this implementation, the evolution is only semi bi-dimensional in the sense that the evolution over s with $t = 0$ is first computed, followed by evolutions over t for $s \in [0, 1]$, providing a continuous surface.

[b] k-means is a method of clustering that aims to partition observations into K clusters in which each observation belongs to the cluster with the nearest mean. The center of each cluster after applying k-means, has the minimum sum of distances from all other points in that cluster.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MM participated in the development and implementation of the methods and performed the experiments. SC developed the models for the spatiotemporal probabilistic dynamical system representation and the stochastic optimization strategy with GMR and supervised the whole process of implementations and experiments. DB developed the models for the forward and inverse kinematics of the robot, with associated codes, and helped with the implementation of the methods. MM and SC wrote the manuscript and developed the codes for the learning methods and experiments. DC provided advices for the writing of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We gratefully acknowledge Prof. Binyamin Hochner, Shlomi Hanassy, and Alexander Botvinnik from the Department of Neurobiology, Hebrew University, Jerusalem, Israel, and Prof. Tamar Flash from the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, who kindly provided us with the 3D octopus motion data processed from vision sensing and used in this article. We would also like to thank them for their truly inspiring work and for their recommendation concerning the interpretation of the collected octopus behaviors. This work was supported by the STIFF-FLOP European project (FP7-ICT-287728).

Author details

[1]Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genoa, Italy. [2]Idiap Research Institute, Rue Marconi 19, CH-1920 Martigny, Switzerland.

Received: 30 June 2014 Accepted: 25 September 2014
Published online: 16 October 2014

References

1. Reynolds W (2001) The first laparoscopic cholecystectomy. JSLS 5(1):89–94
2. Vitiello V, Lee S, Cundy TP, Yang G (2013) Emerging robotic platforms for minimally invasive surgery. IEEE Rev Biomed Eng 6:111–126
3. Degani A, Choset H, Wolf A, Zenati MA (2006) Highly articulated robotic probe for minimally invasive surgery. In: Proc. IEEE Intl Conf. on Robotics and Automation (ICRA), Orlando, FL, USA. pp 4167–4172
4. Piccigallo M, Scarfogliero U, Quaglia C, Petroni G, Valdastri P, Menciassi A, Dario P (2010) Design of a novel bimanual robotic system for single-port laparoscopy. IEEE/ASME Trans Mechatronics 15(6):871–878
5. Bajo A, Goldman RE, Wang L, Fowler D, Simaan N (2012) Integration and preliminary evaluation of an insertable robotic effectors platform for single port access surgery. In: Proc. IEEE Intl Conf. on Robotics and Automation (ICRA), Saint Paul, MN, USA. pp 3381–3387
6. Smith KK (1989) Trunks, tongues, and tentacles: moving with skeletons of muscle. Am Sci 77:29–35
7. McMahan W, Chitrakaran V, Csencsits M, Dawson D, Walker ID, Jones BA, Pritts M, Dienno D, Grissom M, Rahn CD (2006) Field trials and testing of the OctArm continuum manipulator. In: Proc. IEEE Intl Conf. on Robotics and Automation (ICRA), Orlando, FL, USA. pp 2336–2341
8. Walker ID (2000) Some issues in creating "invertebrate" robots. In: Proc. intl symp. on adaptive motion of animals and machines, Montreal, Canada
9. McMahan W, Jones BA, Walker ID (2005) Design and implementation of a multi-section continuum robot: air-octor. In: Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), Edmonton, Canada. pp 2578–2585
10. Martinez RV, Branch JL, Fish CR, Jin L, Shepherd RF, Nunes R, Suo Z, Whitesides GM (2013) Robotic tentacles with three-dimensional mobility based on flexible elastomers. Adv Mater 25(2):205–212
11. Cianchetti M, Ranzani T, Gerboni G, De Falco I, Laschi C, Menciassi A (2013) STIFF-FLOP surgical manipulator: Mechanical design and experimental characterization of the single module. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan. pp 3567–3581
12. STIFF-FLOP European project official website, (2014). http://www.stiff-flop.eu/
13. Jiang A, Xynogalas G, Dasgupta P, Althoefer K, Nanayakkara T (2012) 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), Vilamoura, Portugal. pp 2922–2927
14. Kuniyoshi Y, Inaba M, Inoue H (1989) Teaching by showing: Generating robot programs by visual observation of human performance. In: Proc. intl symposium of industrial robots, Tokyo, Japan. pp 119–126
15. Riley M, Ude A, Atkeson C, Cheng G (2006) Coaching: An approach to efficiently and intuitively create humanoid robot behaviors. In: Proc. IEEE-RAS intl conf. on humanoid robots (humanoids), Genova, Italy. pp 567–574

16. Billard A, Calinon S, Dillmann R, Schaal S (2008) Robot programming by demonstration. In: Siciliano B, Khatib O (eds). Handbook of robotics. Springer, Secaucus. pp 1371–1394
17. Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. Robot Auton Syst 57(5):469–483
18. Calinon S, Bruno D, Malekzadeh MS, Nanayakkara T, Caldwell DG (2014) Human-robot skills transfer interfaces for a flexible surgical robot. Comput Methods Programs Biomed 116(2):81–96. Special issue on new methods of human-robot interaction in medical practice
19. Malekzadeh MS, Bruno D, Calinon S, Nanayakkara T, Caldwell DG (2013) Skills transfer across dissimilar robots by learning context-dependent rewards. In: Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), Tokyo, Japan. pp 1746–1751
20. Ng AY, Russell SJ (2000) Algorithms for inverse reinforcement learning. In: Intl conf. on machine learning, Stanford, CA, USA. pp 663–670
21. Zelman I, Titon M, Yekutieli Y, Hanassy S, Hochner B, Flash T (2013) Kinematic decomposition and classification of octopus arm movements. Front Comput Neurosci 7:60
22. Yekutieli Y, Sagiv-Zohar R, Aharonov R, Engel Y, Hochner B, Flash T (2005) Dynamic model of the octopus arm. I. Biomechanics of the octopus reaching movement. J Neurophysiol 94(2):1443–1458
23. Gutfreund Y, Flash T, Fiorito G, Hochner B (1998) Patterns of arm muscle activation involved in octopus reaching movements. J Neurosci 18(15):5976–5987
24. Gutfreund Y, Flash T, Yarom Y, Fiorito G, Segev In, Hochner B (1996) Organization of octopus arm movements: a model system for studying the control of flexible arms. J Neurosci 16(22):7297–7307
25. Flash T, Hochner B (2005) Motor primitives in vertebrates and invertebrates. Curr Opin Neurobiol 15(6):660–666
26. Zelman I, Galun M, Akselrod-Ballin A, Yekutieli Y, Hochner B, Flash T (2009) Nearly automatic motion capture system for tracking octopus arm movements in 3d space. J Neurosci Methods 182(1):97–109
27. Ijspeert A, Nakanishi J, Pastor P, Hoffmann H, Schaal S (2013) Dynamical movement primitives: Learning attractor models for motor behaviors. Neural Comput 25(2):328–373
28. Calinon S, Li Z, Alizadeh T, Tsagarakis NG, Caldwell DG (2012) Statistical dynamical systems for skills acquisition in humanoids. In: Proc. IEEE intl conf. on humanoid robots (humanoids), Osaka, Japan. pp 323–329
29. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc 39(1):1–38
30. Wu C (1983) On the convergence properties of the EM algorithm. Ann Stat 11:95–103
31. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Proc. of the 5th Berkeley symp. on mathematical statistics and probability, Berkeley, CA, USA. pp 281–297
32. Ghahramani Z, Jordan MI (1994) Supervised learning from incomplete data via an EM approach. In: Cowan JD, Tesauro G, Alspector J (eds). Advances in Neural Information Processing Systems. Morgan Kaufmann Publishers, Inc., Burlington Vol. 6. pp 120–127
33. Dayan P, Hinton GE (1997) Using expectation-maximization for reinforcement learning. Neural Comput 9(2):271–278
34. Peters J, Schaal S (2007) Using reward-weighted regression for reinforcement learning of task space control. In: Proc. IEEE intl symp. on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), Honolulu, HI, USA. pp 262–267
35. Ruecksties T, Sehnke F, Schaul T, Wierstra D, Sun Y, Schmidhuber J (2010) Exploring parameter space in reinforcement learning. Paladyn J Behav Robot 1(1):14–24
36. Theodorou E, Buchli J, Schaal S (2010) A generalized path integral control approach to reinforcement learning. J Mach Learn Res 11:3137–3181
37. Kober J, Peters J (2010) Imitation and reinforcement learning: Practical algorithms for motor primitives in robotics. IEEE Robot Autom Mag 17(2):55–62
38. Stulp F, Sigaud O (2012) Path integral policy improvement with covariance matrix adaptation. In: Proc. Intl Conf. on Machine Learning (ICML), Edinburgh, Scotland. pp 1–8
39. Kroese DP, Rubinstein RY (2004) The cross-entropy method: A unified approach to combinatorial optimization. In: Monte-Carlo simulation and machine learning. Springer, New York
40. Hansen N (2006) The CMA evolution strategy: A comparing review. In: Lozano J, Larranaga P, Inza I, Bengoetxea E (eds). Towards a new evolutionary computation. Studies in Fuzziness and Soft Computing. Springer, Berlin-Heidelberg Vol. 192. pp 75–102
41. Calinon S, Kormushev P, Caldwell DG (2013) Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning. Robot and Autonom Syst, Elsevier 61(4):369–379
42. Kormushev P, Calinon S, Saegusa R Metta G (2010) Learning the skill of archery by a Humanoid robot iCub, Proc. IEEE intl conf. on humanoid robots (humanoids), Nashville. TN, USA. pp 417–423
43. Sumbre G, Fiorito G, Flash T, Hochner B (2005) Neurobiology: Motor control of flexible octopus arms. Nature, Nature Publishing Group, New York City, NY, USA 433(7026):595–596
44. Hanassy S (2008) Reaching movements of the octopus involve both bend propagation and arm elongation. MSc thesis, Medical Neurobiology Department, Jerusalem, Israel: The Hebrew University

doi:10.1186/s40638-014-0013-4
Cite this article as: Malekzadeh *et al.*: Learning by imitation with the STIFF-FLOP surgical robot: a biomimetic approach inspired by octopus movements. *Robotics and Biomimetics* 2014 **1**:13.