A Skill Transfer Approach for Continuum Robots - Imitation of Octopus Reaching Motion with the STIFF-FLOP Robot

Milad S. Malekzadeh[1] and **Sylvain Calinon**[1,2] and **Danilo Bruno**[1] and **Darwin G. Caldwell**[1]
[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy
[2] Idiap Research Institute, Rue Marconi 19, CH-1920 Martigny, Switzerland

Abstract

The problem of transferring skills to hyper-redundant system requires the design of new motion primitive representations that can cope with multiple sources of noise and redundancy, and that can dynamically handle perturbations in the environment. One way is to take inspiration from invertebrate systems in nature to seek for new versatile representations of motion/behavior primitives for continuum robots. In particular, the incredibly varied skills achieved by the octopus can guide us toward the design of such robust encoding scheme. This abstract presents our ongoing work that aims at combining statistical machine learning, dynamical systems and stochastic optimization to study the problem of transferring skills to a flexible surgical robot (STIFF-FLOP) composed of 2 modules with constant curvatures. The approach is tested in simulation by imitation and self-refinement of an octopus reaching motion.

Introduction

There is a wide range of skills transfer 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, but it remains very challenging to make these approaches co-exist and benefit from each others. 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 abstract, we study the problem in the case of biological or robotic systems characterized by a hyper-redundant kinematic chain.

Spatiotemporal Representation of Octopus Movement

In this paper, we exploited one representative sample from recordings of various octopus reaching movements (Zelman et al. 2013), gratefully provided by Prof. Binyamin Hochner, Shlomi Hanassy and Alexander Botvinnik (Department of Neurobiology, Hebrew University), and Prof. Tamar Flash (Department of Computer Science and Applied Mathematics, Weizmann Institute of Science). The raw data consists of Cartesian positions of points along the octopus arm at different time steps during the movement. A pre-processing step was used to smooth and resample the data (with 50 points in both time and space) with a two-dimensional polynomial fit (with a 7-degree polynomial, set experimentally by testing different orders).

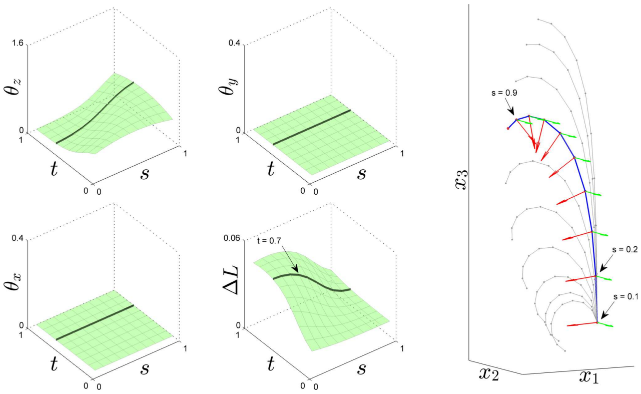
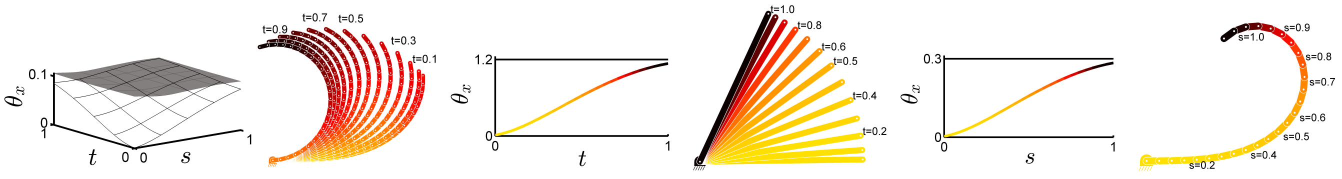


Figure 1: *Left:* Four surfaces representing the Euler angles $\boldsymbol{\theta}$ and offset $\boldsymbol{\Delta L}$ for the whole movement. The black lines show a certain time slice, corresponding to a static pose described by all the links $(0 \leq s \leq 1)$ at $t = 0.7$. *Right:* The arm configuration and Frenet frames along the arm for the same time frame.

In order to generate a model of the Octopus movements, the arm was approximated as a robot with a high number of links. Each link can for example be described by a set of 3 Euler angles (ZYX decomposition order) and one offset (link length).

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. The duration of a movement will be represented by a continuous *time index* $t \in [0, 1]$. We thus have a set of 3 Euler angles $\theta_x, \theta_y, \theta_z$ and an offset ΔL for each t and s, that can be represented as a set of surfaces, see Fig. 1.



(a) (b) (c)

Figure 2: A dynamical system with surface attractor (evolution over time and arm-index). *(a)* The grey surface represents this attractor for the motion of a continuum arm, while the white surface is the resulting motion of the arm. *(b)* For a given arm index s along the kinematic chain, the corresponding trajectory of the link at point s is represented. *(c)* For a given time index t, we obtain a virtual spring-damper system in the s space, describing a static pose of the robot.

Continuum Arm Motion Encoding with Gaussian Mixture Regression

Dynamical movement primitives (DMP) is a popular tool in robotics to encode and reproduce movements while being robust to perturbations (Ijspeert et al. 2013). It consists of a spring-damper system modulated by a non-linear force profile encoded as a series of pre-defined basis functions associated with local force behaviors. We introduced in (Calinon et al. 2012) a probabilistic formulation of DMP by jointly learning the basis functions and the local force behaviors with *Gaussian mixture regression* (GMR). We also introduced an alternative representation by assuming that the non-linear force can be represented by a moving spring-damper system pulling the system, allowing us to encode the path of the attractor point instead of the force.

With this DS-GMR model, after observation of a motion $\boldsymbol{x}$, the trajectory of the virtual attractor $\boldsymbol{y}$ that generates the motion is described by

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

where $\kappa^{\mathcal{P}}$ and $\kappa^{\mathcal{V}}$ are respectively the stiffness of the spring and the viscosity of the damper. $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ are the velocity and acceleration.

In order to employ the above system with octopus movements, we use as input variable $[s, t]$, and as output variable an attractor $\boldsymbol{y}$ to reproduce $\boldsymbol{x} = [\theta_z, \theta_y, \theta_x, \Delta L]$. $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ are 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 in Fig. 1, $\boldsymbol{y}$ can be represented as a set of attractor surfaces. An illustration of a continuum arm is presented in Fig. 2.[1].

The joint distribution $[s, t, \boldsymbol{y}]$ is encoded in a *Gaussian mixture model* (GMM) of K components, trained by *expectation-maximization* (EM) after initialization of the parameters with *k-means* clustering. GMR is then used to estimate $\mathcal{P}(\boldsymbol{y}|s, t)$, within a fast computation time independent of the number of datapoints used to train the model. Given $\boldsymbol{y}$, the whole motion can then be reconstructed by double integration.

Self-Refinement with Reward-Weighted Regression

The compact GMM representation presented above can be refined by other learning strategies such as self-refinement. One option for implementing such self-refinement is to use an EM-based stochastic optimization algorithm to refine the GMM parameters, similarly as in (Calinon et al. 2014). The procedure corresponds to an EM algorithm in which a reward signal of exponential form is treated as a likelihood, which can easily be extended to multi-optima policy search (Calinon, Kormushev, and Caldwell 2013).

In this paper, we pursue another approach with the aim of providing a more structured exploration strategy. In (Kormushev et al. 2010), it was proposed to exploit the structure of the optimization problem to speed up the search process by redefining the self-refinement problem as an iterative reward-weighted regression, which is 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, staying as close as possible to a reference trajectory, etc.).

In this paper, a Gaussian distribution is fit to the augmented data $\boldsymbol{\xi} = [\boldsymbol{\xi}^{I^{\top}}, \boldsymbol{\xi}^{O^{\top}}]^{\top}$ formed by the policy parameters $\boldsymbol{\xi}^{O}$ (the GMM parameters of the model described in the previous section) and the associated goals $\boldsymbol{\xi}^{I}$ (the movement to imitate). The ordered set of datapoints $\{\boldsymbol{\xi}_m\}_{m=1}^{M}$ with rewards $r(\boldsymbol{\xi}_1) \geqslant r(\boldsymbol{\xi}_2) \geqslant \ldots \geqslant r(\boldsymbol{\xi}_M)$ is used as a form of importance sampling (Kober and Peters 2010) to estimate iteratively a new Gaussian distribution. A new policy $\boldsymbol{\xi}^{O}$ is then estimated trough conditional probability by giving the desired goal as input and by evaluating the retrieved model to create a new point in the dataset.

Applications

Manual Design of Motion Primitives

The DS-GMR model is first employed to emulate typical movements used by octopus to reach for food, namely bend propagation and bend elongation (Zelman et al. 2013). These reaching movements involve bend propagation and elongation of the proximal part of the arm (Flash and

[1] Note that in this implementation, the evolution is only semi bidimensional 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.

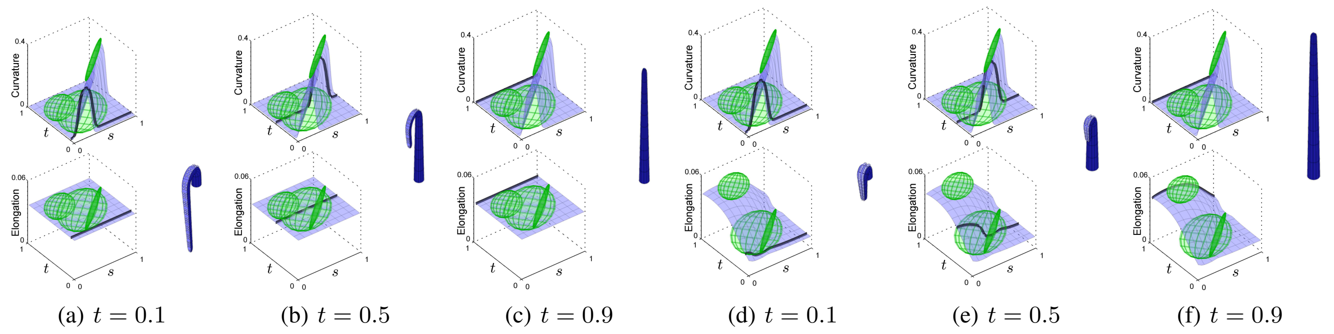


Figure 3: Emulation of reaching movement with bend propagation *(a-c)* and bend elongation *(d-f)*.

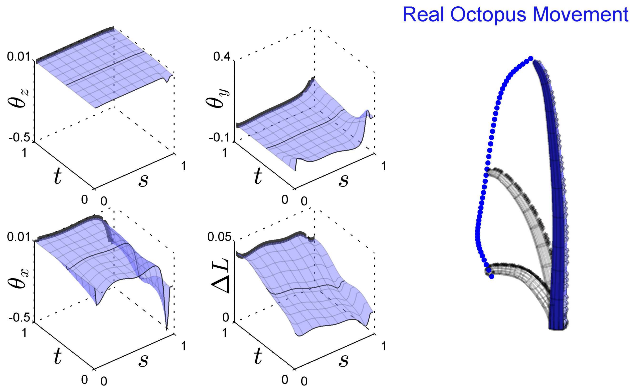


Figure 4: An example of octopus reaching movement (after data pre-processing). The black lines on the surfaces show configurations at three different time steps (initial, mid and final configurations), with corresponding octopus arm pose in a 3D Cartesian space (the tip trajectory is represented with blue dots).

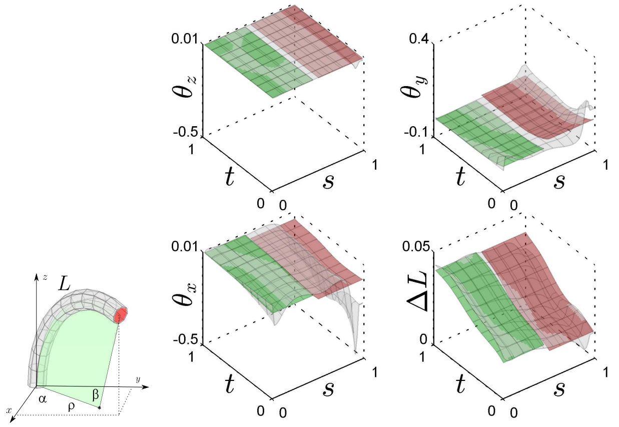


Figure 5: *Left:* Single module of a constant curvature model, with the pose of the tip described as a function of α, β and L. *Right:* The Euler angles and offsets related to the octopus movement (light gray) and the corresponding STIFF-FLOP robot with 2 modules (green and red surfaces).

Hochner 2005). The bend propagation movement from the base to the tip can be simulated by setting a full covariance matrix with non-zero off-diagonal elements (in the form of a tilted ellipsoid), acting as an attractor in a curvature-elongation space (e.g., θ_x and ΔL), see Fig. 3(a-c). An elongation primitive can also be obtained by translating one of the Gaussians in the elongation space, see Fig. 3(d-f).

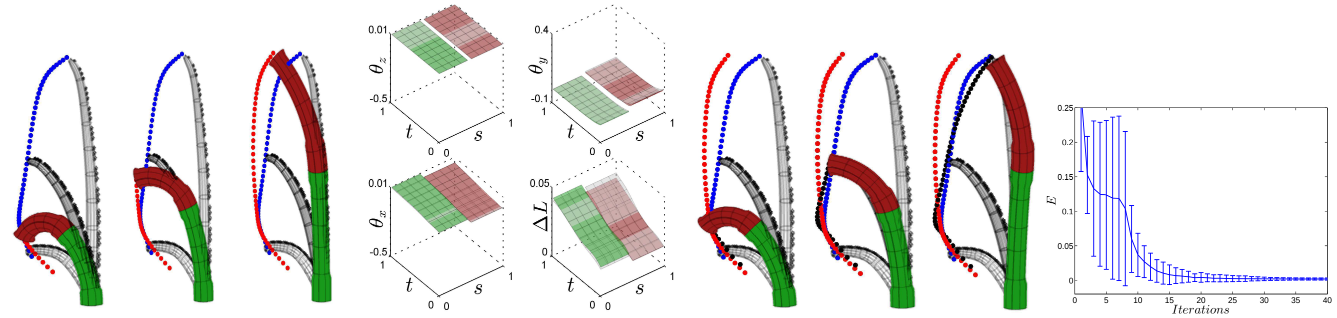
Reaching movements are usually generated by a combination of these two motor primitives: propagation of a bend along the arm and arm elongation. They may be combined with different weights to create a broader spectrum of reaching movements. Instead of encoding individual information about bending/elongation patterns, the advantage of DS-GMR is that it provides local correlations throughout the movement in the form of a full covariance matrix on the control commands retrieved by GMR.

Transferring Skills from the Octopus to the STIFF-FLOP Robot

Fig. 4 shows an example of octopus reaching movement. 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 can be exploited to transfer the movement to a STIFF-FLOP robot composed of several modules juxtaposed in series (Cianchetti et al. 2014). We employ here a simplified representation for each STIFF-FLOP module by approximating the shape of each module by a constant curvature model. This constraint requires us to divide the set of continuous surfaces into a given number of subsurfaces corresponding to the number of modules (here, two).

This transfer of motion skills from a continuous arm to an arm with piecewise constant curvatures can provide a good initial estimate but does not necessarily guarantee the best fit. Indeed, even though the movement is quite similar, the different structure of the STIFF-FLOP robot can cause dissimilarity in the learnt movement. The self-refinement approach presented before is thus used to refine this initial estimate, by using the centers and the first eigencomponent of the covariance matrices of the GMM as policy parameters. A reward function based on the Cartesian distance of the



(a) $t = 0.2$ (b) $t = 0.5$ (c) $t = 1$ (d) (e) $t = 0.2$ (f) $t = 0.5$ (g) $t = 1$ (h)

Figure 6: *(a-c)* Comparison of the tip positions of the octopus arm (blue dots) and STIFF-FLOP robot (red dots) based on the skill transfer imitation approach (before self-refinement) at time instances $t=0.2$, $t=0.5$ and $t=1$. *(d)* The initial (light gray) and refined variables (green and red surfaces). *(e-g)* The comparison after self-refinement (the black dots are the new tip trajectory). *(h)* The Error over 30 runs.

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

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

where γ is a bandwidth coefficient set experimentally and E is the position error.

In order to keep the robot parameters within the real hardware limits, we also considered hard constraints in the search process for each of the 2 modules as $L_0 \leq L \leq 1.8L_0$ and $\beta \leq 2\pi/3$, where L_0, L and β are the module minimum length, length and curvature (see Fig. 5-*left*).

The results of the skill transfer and self-refinement processes are shown in Figs 5-*right* and 6. Fig. 6(h) shows the average and standard deviation of the error (E) for 30 runs of the same experiment. We can observe a convergence of the error (equivalently the reward) after 40 self-refinement iterations, by starting from an initial set of 20 randomly generated policies based on the initial demonstration.

Conclusion

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 transfer skills from demonstration, and that it can be combined with a self-refinement strategy based on iterative reward-weighted regression. Our future work aims at applying this learning approach to the real STIFF-FLOP platform, by considering a wider range of motion/feedback skills that could be used to assist doctors in surgical operations.

Acknowledgment

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

References

Calinon, S.; Li, Z.; Alizadeh, T.; Tsagarakis, N. G.; and Caldwell, D. G. 2012. Statistical dynamical systems for skills acquisition in humanoids. In *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, 323–329.

Calinon, S.; Bruno, D.; Malekzadeh, M. S.; Nanayakkara, T.; and Caldwell, D. G. 2014. Human-robot skills transfer interfaces for a flexible surgical robot. *Computer Methods and Programs in Biomedicine* 116(2):81–96. Special issue on new methods of human-robot interaction in medical practice.

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

Cianchetti, M.; Ranzani, T.; Gerboni, G.; Nanayakkara, T.; Althoefer, K.; Dasgupta, P.; and Menciassi, A. 2014. Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-flop approach. *Soft Robotics* 1(2):122–131.

Flash, T., and Hochner, B. 2005. Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology* 15(6):660–666.

Ijspeert, A.; Nakanishi, J.; Pastor, P.; Hoffmann, H.; and Schaal, S. 2013. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation* 25(2):328–373.

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

Kormushev, P.; Calinon, S.; Saegusa, R.; and Metta, G. 2010. Learning the skill of archery by a humanoid robot iCub. In *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, 417–423.

Zelman, I.; Titon, M.; Yekutieli, Y.; Hanassy, S.; Hochner, B.; and Flash, T. 2013. Kinematic decomposition and classification of octopus arm movements. *Frontiers in Computational Neuroscience* 7(60).