Active Incremental Learning of Robot Movement Primitives

Guilherme Maeda[1], **Marco Ewerton**[1], **Takayuki Osa**[2], **Baptiste Busch**[3], **Jan Peters**[1,4]
[1]Department of Computer Science, Technische Universität Darmstadt, Germany
{maeda,ewerton,peters}@ias.tu-darmstadt.de
[2]Department of Complexity Science and Engineering, The University of Tokyo, Japan
osa@ms.k.u-tokyo.ac.jp
[3]Flowers Laboratory, Centre de Recherche Inria Bordeaux, France
baptiste.busch@inria.fr
[4]Department for Empirical Inference and Machine Learning, Max-Planck Institute, Germany

Abstract: Robots that can learn over time by interacting with non-technical users must be capable of acquiring new motor skills incrementally. The problem then is deciding when to teach the robot a new skill or when to rely on the robot generalizing its actions. This decision should be made by the robot and not by the human. Thus, the robot must quantify the suitability of its own skill for an unseen task. To this end, we present an algorithm for active requests to incrementally learn movement primitives. A movement primitive is learned on a trajectory output by a Gaussian Process. The latter is used as a library of demonstrations that can be extrapolated with confidence margins. This combination not only allows the robot to generalize using as few as a single demonstration but more importantly, to indicate when such generalization can be executed with confidence or not. In experiments, a real robot arm indicates to the user which demonstrations should be provided to increase its repertoire of reaching skills. Experiments will also show that the robot becomes confident in reaching objects for whose demonstrations were never provided, by incrementally learning from the neighboring demonstrations.

Keywords: Gaussian process, movement primitives, active learning

1 Introduction

Robots will eventually make part of our daily lives, helping us at home, taking care of the elderly, and collaborating at work. Such robots cannot be pre-programmed prior to deployment, and will most likely start with an empty set of skills. One challenge that must be addressed is that of creating, augmenting and tailoring a robot's repertoire of skills as needed, over time. Motion planning (MP) can be very efficient to generate robot motions. However, in the scenarios of interest in this paper, MP is too stringent in terms of requirements: the geometry of the environment is usually unknown, sensing/perception is prone to occlusions due to clutter, and non-technical users are not supposed to worry about cost functions or heuristics. Imitation learning can avoid such drawbacks by relying on the presence of a human teacher. Imitation learning takes advantage of parametric movement representations, referred here as movement primitives, to encode and generalize human demonstrations. However, imitation learning has primarily addressed *how* to endow and refine robots with motor skills. It does not address *when* the learning should take place. Reasoning *when* improvement is actually needed is, nevertheless, an essential and difficult problem to be solved. This paper focuses on an algorithm for such reasoning and its connection with the learning of movement primitives.

The usual practice in imitation learning is to rely on human judgment to decide when a skill must be added or corrected. Human judgment in robot skills, however, presents a number of issues. For example, it requires constant human attention: as the robot moves, the user must observe and decide if a correction or addition of a new skill is needed, which is not always obvious. Also, the human cannot judge if a generalized skill is suitable unless the robot actually executes it, which can lead to undesirable/dangerous situations. On the other hand, if the robot has means to quantify the quality

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

of its own skills for a given task, it can then decide how confident it is to execute it. We frame this problem as an active learning problem where the human provides demonstrations rather than labels.

The principal contribution of this paper is an active learning algorithm that allows a robot to reason about the confidence of its movement primitives. The robot can decide *when* a demonstration is required, making active requests to a human depending on its confidence. A secondary contribution is a methodology to train Dynamical Movement Primitives (DMPs) [1] with contextualized demonstrations encoded by Gaussian Processes (GPs) where the uncertainty of the latter acts as a surrogate to measure the amount of required extrapolation.

2 Related Work

Active learning addresses the problem of which labels to ask in order to improve the sampling efficiency of a learning agent. This idea strongly connects with the way a physical robot should improve its skills by asking a human for demonstrations. Not surprisingly, active learning has been explored in different contexts within robotics, particularly when associated with imitation learning [2] and reinforcement learning. For example, an agent has been shown to learn navigation policies much more efficiently by using both autonomous execution and demonstrations when compared to a pure reinforcement learning agent [3, 4]. Active learning also plays an important role in developmental robotics for the lifelong acquisition of robot skills, either by self-exploration and self-improvement [5], or by requesting human assistance via demonstration [6].

The interplay between active learning and skill learning on physical robots presents many open problems. One of them regards the appropriate metric used to trigger a request for assistance, such as novelty and uncertainty reduction [7], or the confidence in executing an action [4]. Other problems relate to the frequency of requests [8], the modality of the input [9], and the learning of the threshold that triggers active requests [3]. In lifelong learning, one problem is that of defining in which space (motor or goal) a learning agent learns more efficiently [10]. In this paper, uncertainty reduction will be used as the metric of choice, the threshold will be manually set, and skills will be learned in the robot's goal space. These practical decisions will allow us to focus on the algorithmic problem of programming movement primitives for a robot under active requests.

While incremental learning has been addressed by a number of authors, the active learning aspect has not been explored to the same level. Calinon and Billard [11] addressed the problem of incrementally refining robot primitives with Gaussian Mixture Models (GMMs). In a similar fashion, Lee and Ott [12] proposed a refinement tube defined by impedance control where trajectories are encoded with Hidden Markov Models. Ewerton et al. [13] introduce a method to incrementally improve policies by interleaving feedback control with corrective interventions in open-loop. Farraj et al. [14] uses a measure of information gain to decide when new data should be added to the robot's skill repertoire. Ahmadzadeh et al. [15] use a generalized cylinder representation to encode multiple demonstrations and refine the parts that need correction. In common is the fact that *the human is in charge of deciding* when to teach and refine the robot skills. Here, our goal is to allow a robot to *actively take such decisions*.

To achieve the desired confidence reasoning, the choice of the representation used to encode the training data is crucial. Probabilistic modeling seems, therefore, the natural choice for enabling such reasoning. Probabilistic Movement Primitives (ProMPs) [16] treat multiple demonstrations as multi-variate Gaussians, but present the issue that an unknown number of initial demonstrations are required to create an informative prior. The same problem is also found with GMMs [11, 17]. Additionally, as models based on mixtures of linear components, such methods rely on heuristics to define the number of components. Moreover, an issue of the linearity is that consistency of the generalization is only guaranteed at the vicinity of the training data. In this respect, GPs have been used to represent robot primitives with great generalization capabilities [19, 20] at the expense of a higher computational cost.

Dynamical Movement Primitives (DMPs) have been known to generalize well around the vicinity of a demonstration and its capabilities have been extended in many ways with GPs. Ude et al. [21] use a combination of local weighted regression and GPs to predict the parameters of DMPs in a database. In [22], GPs are used to learn DMPs parameters in joint space while querying states in the input space of the GP as the final desired Cartesian position of the movement. Matsubara et al. [23] use GPs to learn the parameters of DMPs conditioned by the style of the movement while Bitzer

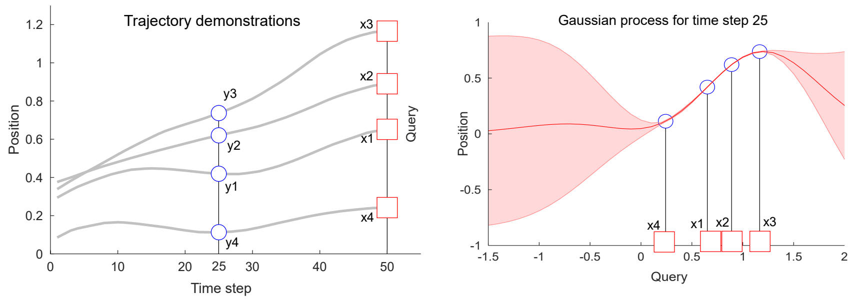


Figure 1: A sketch of the GP approach. Given four trajectory demonstrations (left plot), Gaussian processes are trained for each time slice to predict positions and uncertainties given new queries (contexts). In this paper, queries are defined as the final trajectory position. The plot at the right shows the prediction of a trained GP for the time step 25. In this example, the input space (the trajectory goal or context) is one dimensional.

and Vijayakumar [24] use GPs as a dimensionality reduction approach to encode DMPs in a latent space. These prior works show that the combination of GPs and DMPs are useful to contextualize parameters of the movement but do not exploit the use of uncertainty for *active learning*. In this paper, we focus on the uncertainty provided by the GPs. For this reason, we use GPs and DMPs in two separate steps. The idea is to obtain confidence bounds in the same space in which demonstrations are provided: as trajectories (as opposed to the parameters of DMPs) so that the value of the uncertainty is physically meaningful.

3 A Movement Primitive with Uncertainty for Active Request

GPs have been known to extrapolate with an indication of uncertainty. However, extrapolations inevitably lead to errors, which for many fine motor skills (such as reaching a grasping position) will turn into execution failures. On the other hand, a DMP guarantees that any goal can be reached beyond the training set, but it cannot indicate how far a generalization to an unseen input can be made before a new demonstration is required. When combined in tandem, DMPs will learn trajectories extrapolated with confidence bounds by the GPs. Each component will be explained in the following subsections. To avoid confusion, hereinafter, a query made in the input space of a GP will be referred to as a context. A query that the robot makes to the human to obtain a demonstration will be referred to as a demonstration request.

3.1 Generalizing the Training Data with Gaussian Processes

Define the vector $\mathbf{x}_{1:N}$ as a trajectory comprised of N time steps. To provide a concrete, practical explanation, we focus on modeling trajectories in the Cartesian[1] space, $\mathbf{x}_n = [x_n, y_n, z_n]^\intercal$. For each time step, independently trained Gaussian Processes (GPs) are used to predict the triple $(f_{x,n}(\mathbf{x}_*), f_{y,n}(\mathbf{x}_*), f_{z,n}(\mathbf{x}_*))$, where $f_{(\cdot),n}(\mathbf{x}_*)$ is a scalar, and $\mathbf{x}_*$ is a desired location on the input space of the model. We focus on reaching specific positions of the robot workspace, such that $\mathbf{x}_* = [x_N, y_N, z_N]^\intercal_{\text{goal}}$ is the context. Since each Cartesian dimension will be treated independently, the subscripts x, y, z will be dropped and $f_n(\mathbf{x})$ will be used to represent a generic dimension.

The goal of the GPs is to predict a trajectory as a sequence $f_{1:N}(\mathbf{x}_*)$ for an unseen context $\mathbf{x}_*$ by learning a map from context to trajectories $\mathcal{M} : \mathbf{x}_* \rightarrow f_{1:N}(\mathbf{x}_*)$. For active requests of demonstrations, the prediction to be in the form of a distribution $p(f_{1:N}|\mathbf{x}^*)$. Assume that the positions of D demonstrated trajectories at each n-th step $\mathbf{t}_n = [t_n^1, ..., t_n^D]^\intercal$ and the final positions (i.e. contexts) achieved by their respective trajectories $\mathbf{X} = [\mathbf{x}^1, ..., \mathbf{x}^D]^\intercal$ are jointly Gaussian. The joint

[1]Orientations are not addressed in this paper, for simplicity and due to the following two assumptions. First, the initial and final orientations of the robot hand are given, the former by the robot's initial position, and the latter by the reference frame of the object to be reached. Second, experiments show that interpolation along the trajectory between these two orientations is sufficient for satisfying the particular tasks proposed in this paper.

distribution is given as

$$\left[\begin{array}{c}\mathbf{t}_n\\ f_n(\mathbf{x}_*)\end{array}\right]\sim\mathcal{N}\left(\mathbf{0},\left[\begin{array}{cc}K(\mathbf{X},\mathbf{X})+\sigma_s^2\mathbf{I} & k(\mathbf{X},\mathbf{x}_*)\\ k(\mathbf{x}_*,\mathbf{X}) & k(\mathbf{x}_*,\mathbf{x}_*)\end{array}\right]\right),$$

where σ_s^2 is a noise variance to be trained as a hyper-parameter. For a single context query, $k(\mathbf{x}_*,\mathbf{x}_*)$ is a scalar, and the covariance matrices $K(\mathbf{X},\mathbf{X})$, $k(\mathbf{X},\mathbf{x}_*)$ have dimensions $D\times D$, $D\times 1$, respectively. Each element $K_{i,j}=k(\mathbf{x}_i,\mathbf{x}_j)$ is computed with the squared exponential covariance function

$$k(\mathbf{x}_i,\mathbf{x}_j)=\sigma_f^2\exp\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_j)^{\intercal}\mathbf{W}(\mathbf{x}_i-\mathbf{x}_j)\right), \tag{1}$$

where σ_f^2 and $\mathbf{W}$ are hyper-parameters. The proposed GP encoding is illustrated in Figure 1.

The GP prediction is

$$f_n(\mathbf{x}_*)\sim\mathcal{N}(\mu_n,\sigma_n^2), \tag{2}$$

where

$$\mu_n=k(\mathbf{x}_*,\mathbf{x})[K(\mathbf{x},\mathbf{x})+\sigma_s^2\mathbf{I}]^{-1}\mathbf{t}_n, \tag{3}$$

and

$$\sigma_n^2=k(\mathbf{x}_*,\mathbf{x}_*)-k(\mathbf{x}_*,\mathbf{x})[K(\mathbf{x},\mathbf{x})+\sigma_s^2\mathbf{I}]^{-1}k(\mathbf{x},\mathbf{x}_*). \tag{4}$$

The variance σ_n^2 is the essential component for active requests. In this paper, the uncertainty is computed as the average magnitude over the whole trajectory

$$u=\frac{1}{N}\sum_{n=1}^{N}\sqrt{(\sigma_{x,n}+\sigma_{y,n}+\sigma_{z,n})^2}. \tag{5}$$

The computation in the form of (5) was chosen solely due to its simplicity. More sophisticated ways to compute the uncertainty can potentially improve the performance of the active learning algorithm but such investigation is beyond the scope of this work.

3.2 Accurate Executions with Dynamical Movement Primitives

The predictions of the GPs do not guarantee that the resulting trajectory will accurately achieve a desired unseen context $\mathbf{x}_*$. This is particularly true when the number of demonstrations is insufficient (one or two), and the GPs have to generalize significantly beyond its training set. Figure 2 illustrates two cases of generalization where a context $\mathbf{x}_*^1$ is closer to the training set than the context $\mathbf{x}_*^2$. Although none of the extrapolations exactly satisfy the context, only a small correction is required for $\mathbf{x}_*^1$ while for $\mathbf{x}_*^2$ the accuracy is quite low. The error in prediction is caused by a lack of observations at the vicinity of the desired context, which is also reflected in terms of the resulting uncertainty.

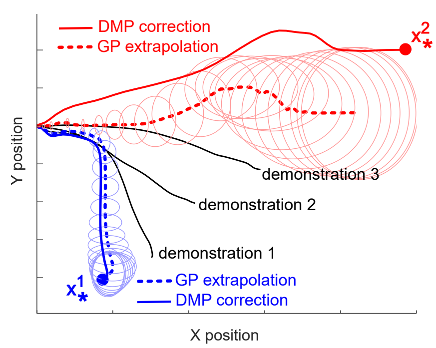


Figure 2: The DMPs are learned on the GP output and can correct for inaccuracies. The large uncertainty for $\mathbf{x}_*^2$ indicates that a demonstration should be requested instead.

We assume that the predicted GP trajectory is valid when its uncertainty (5) is below a specified threshold. This prediction is then encoded in the form of a DMP, guaranteeing that the corrected trajectory can exactly reach the goal. Regarding the input $\mathbf{x}_*^2$, given the large uncertainty of the GPs, it seems prudent to rather request a human demonstration.

Taking y as one of the Cartesian coordinates, a DMP is defined as a dynamical system of the form

$$\tau\ddot{y}=\alpha_y(\beta_y(y_*-y)-\dot{y})+f(z), \tag{6}$$

where τ is a temporal scaling variable that controls the duration of the movement. The parameters α_y, β_y are set to emulate a critically damped, linear spring-damper dynamics with stiffness $\alpha_y\beta_y$ and damping α_y. The context, in the form of the final position of the trajectory, is used as the goal attractor y_*.

An arbitrary force $f(z)$ acts on the linear spring-damper system and captures the nonlinearities of the demonstration. The phase variable z controls the evolution of the system and is by itself defined as a dynamical system of the form $\tau \dot{z} = -\alpha_z z$ (see e.g. [1] for details).

Algorithm 1 GP-DMP($\{\mathbf{x}^d, t^d\}_{1:D}, \mathbf{x}_*, u_{\text{trig}}$)

1: $GP.train(\{\mathbf{x}^d, t^d\}_{1:D})$
2: $(\mu_{1:N}, \sigma^2_{1:N}) \leftarrow GP.predict(\mathbf{x}_*)$
3: **if** TRAJ_UNC($\sigma^2_{1:N}$) < u_{trig} **then**
4: $\quad DMP.train(\mu_{1:N})$
5: $\quad y \leftarrow DMP.reach\_goal(\mathbf{x}_*)$
6: $\quad$ ROBOT_EXECUTION(y)
7: **else**
8: $\quad \{\mathbf{x}^{D+1}, t^{D+1}\} \leftarrow$ ACTIVE_REQUEST
9: $\quad$ go to line 1

Algorithm 1 shows a pseudocode of the proposed method. As inputs, the algorithm requires a training set comprised of D demonstrations, an unseen context $\mathbf{x}_*$, and the trigger of the uncertainty u_{trig}. In line 1, the GPs are initialized. In line 2, the mean and variance of the extrapolated trajectories are computed. In line 4, a DMP is trained on the GP mean and its goal attractor is adjusted (line 5) to accurately reach the desired context. In line 8, the robot requests a new demonstration if the GP variance is beyond the uncertainty trigger.

4 Experiments on Reaching Tasks with a Real Robot

In the following experiments, a 7 degree-of-freedom lightweight arm was used. The threshold on the uncertainty, computed by (5), was empirically set to 9 cm. The GPs were trained on the (x,y,z) coordinates of the Cartesian trajectory of the robot hand. The orientation of the robot hand along the trajectory was given by spherical linear interpolation (slerp) between the initial and final orientations of the robot along the trajectory. The final orientation was defined by the reference frame of the object to be reached. Inverse kinematics was used to execute the joint-space tracking controllers. A video of the experiments reported in this section can be watched in `https://youtu.be/s9kG_IKzqO4`.

4.1 Assembly Scenario

In this first experiment, we considered an industrial environment where a non-technical user had to program a robot collaborator to reach different objects in its workspace. As shown in Figure 3, there were 10 objects to be reached such as different parts of a product on a shelf and tools on a toolbox. To simplify the realization of the experiment, we hand-coded the reaching locations (position and orientation) of each object. Obstacles and the geometry of the environment are unknown to the robot. The plot on the right shows the desired reference frames of the final positions of the hand, that is, the contexts. The robot indicated for which objects the human should provide demonstrations, with the goal of reaching all objects.

As the initial training set was empty, the robot randomly selected one object for the first demonstration. After each demonstration, the GPs were trained and predictions for each of the contexts were made. Their respective uncertainties $\{u_1, ..., u_{10}\}$ were then ranked according to (5). The most uncertain reaching object was then used by the robot to request a human demonstration. This process was repeated until all predictions fell below the specified 9 cm threshold. Figure 4(a) shows the result of this procedure for a particular trial where the robot initial random request was to reach the lower plate on the shelf. The prediction for all contexts given this single demonstration is shown in Figure 4(a). The ellipses represent the projections of one standard deviation along the trajectory on the XY plane. In this particular case, the next requested demonstration was for the ratchet, followed by the screw driver. The predictions are shown in Figure 4(b). Finally, demonstrations were requested for the set of hex keys and the upper plate. The predictions are shown in Figure 4(c). The color of the trajectories for uncertainties above the threshold are shown in red, otherwise in blue.

Note from the Figure 4(c) that after five demonstrations all 10 trajectories were within the confidence limits. At this stage, the robot could execute all reaching primitives confidently. This result shows an important distinction of this work when compared to other probabilistic movement representations such as GMMs and ProMPs [11, 17, 16]. Here, the robot dictates which demonstrations are required to generate a library that satisfies the given task while in prior work the user has to make such decision based on his/her own judgment.

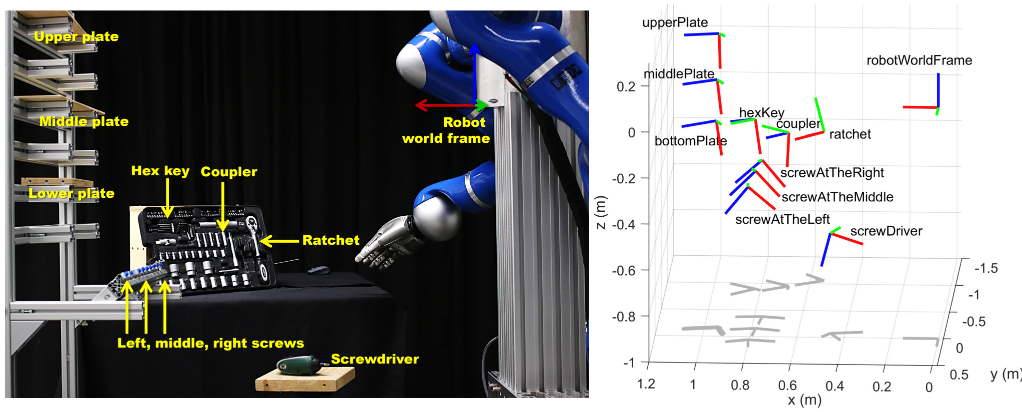


Figure 3: A robot starts with an empty skill set. It must reach 10 objects whose positions and orientations were given (right). The robot uses the GP-DMP predictions to predict reaching trajectories for all objects. Active requests for demonstrations are made to decrease the uncertainty of the reaching trajectories.

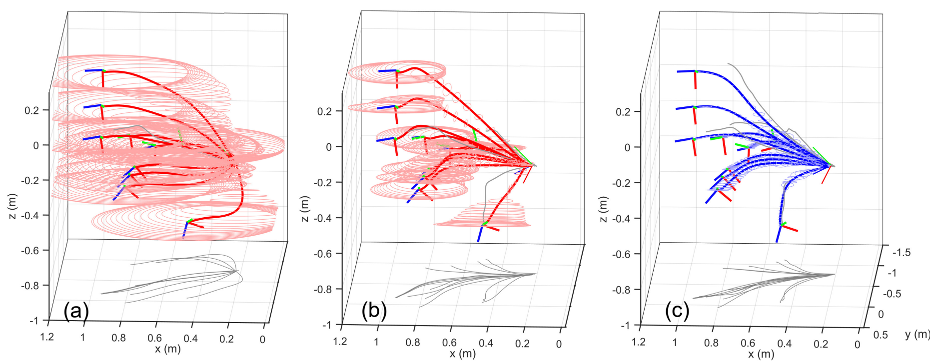


Figure 4: Decrease of the trajectory uncertainties as demonstrations for objects are being requested. The ellipses show a single standard deviation. Uncertainties above the threshold are shown in red. The blue trajectories indicate that the robot is confident in executing the primitives. (a) The uncertainties for all contexts after one demonstration. (b) Predictions after two demonstrations. (c) Predictions after five demonstrations.

A characteristic of the method is that the uncertainty decreases for all objects, and not only for the objects whose demonstrations were requested. This is particularly evident when plotting the decrease of uncertainty for the three screw positions (left, middle, right) as shown in Figure 5. During this particular trial, no demonstrations for any of the screws were requested. Nevertheless, the robot could increase its confidence on how to reach the screws because of the demonstrations provided for the neighboring objects.

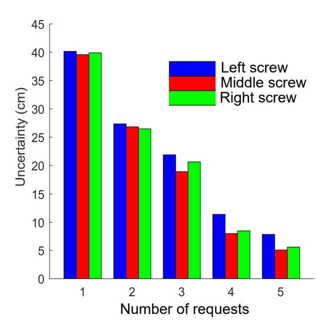


Figure 5: Decrease of the robot's uncertainty in reaching the three screws.

Figure 6(a) shows the average result over 10 different runs where the bars represent the averaged uncertainty with a single standard deviation. The runs differ by the first object whose trajectory demonstration was requested. We systematically used each of the 10 objects to initialize the first skill in each trial. The results show that, regardless of which object a demonstration was first requested—which led to different sequences of demonstrations—in all cases, a total of five requests showed sufficient to achieve high confidence for all trajectories. In Figure 6(b), we compare the uncertainty decrease criterion with uniform random requests. Note that the uncertainty after the third request decreases at a significantly lower rate. After the fifth request, the robot uncertainty still presented a quite large variance.

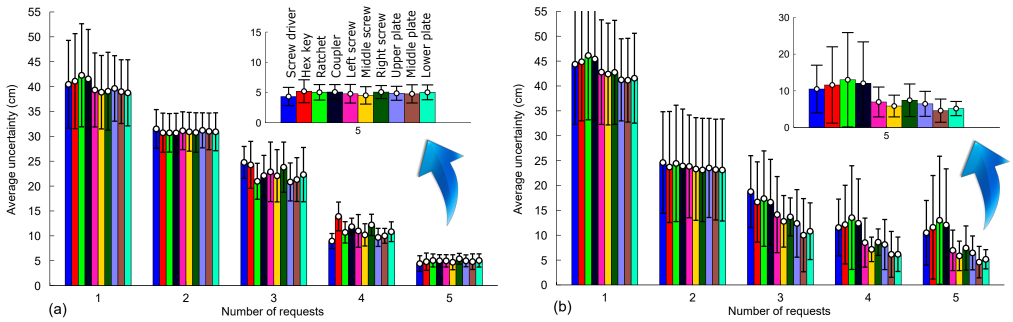


Figure 6: Average result over 10 runs where the bars represent the averaged uncertainty of the robot to reach each object. A single standard deviation is shown on the top of each bar. (a) Requests for the most uncertain object. (b) Uniform random requests.

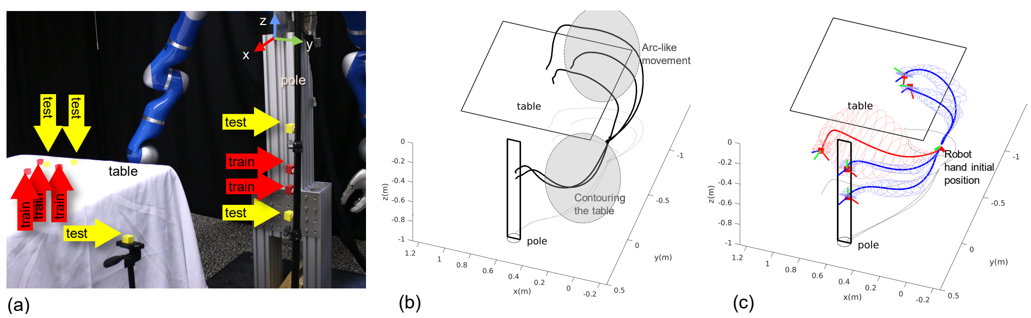


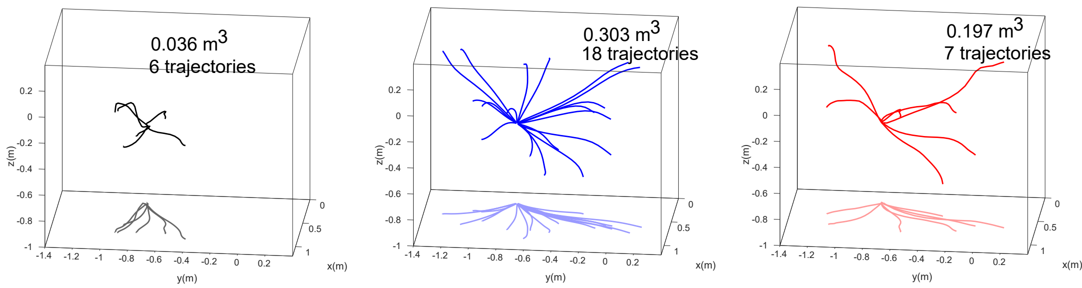
Figure 7: Generating different movements according to the location of the context. (a) Demonstrations were provided for the red blocks. Yellow blocks were used for test. (b) Demonstrated trajectories for the red blocks. (c) Generalization towards the yellow blocks. Note that the robot is uncertain for the block away from the training data.

Although not addressed here, skill transfer is a possible application of the same algorithm. The learned library of skills could be used to address a second scenario, with different objects at different locations. The GPs provide a straightforward way to retrieve the closest skill for a new context in a new setting. The uncertainty provides a way to indicate to the user that a refinement/adaptation of the skill is necessary.

4.2 Contextual Movement Primitives

Contextualization of movement primitives has been explored by many authors (e.g. [21, 23, 20]) as a way to augment the versatility of robot skills. Our method brings the advantage that the uncertainty is used to collect demonstrations for new contexts in an active manner. In the particular case addressed here, the context is the final position of the robot hand. The assumption due to the GPs is that contexts that are close to each other must have trajectories with similar profiles. The goal is not to learn an underlying pattern that governs all demonstrations, but to learn how to generate different motions according to their contexts. Figure 7(a) shows a scenario where the robot has to reach different parts of the workspace in different ways. As the robot hand starts behind the table, to reach the blocks on the table, the hand must move vertically with and arc-like motion. Differently, to reach the blocks on the pole, the hand must contour the back of the table in order to avoid collision. To train the GPs, five demonstrations were provided, one for each red block. These demonstrations are shown in Figure 7(b). Note that the robot does not need to be aware of the table as an obstacle since the human is providing the collision-free demonstration via kinesthetic teaching.

GPs were used to extrapolate trajectories for each of the yellow blocks shown in 7(a). GP output encoded as DMPs are shown in Figure 7(c) together with their uncertainties. The solutions, shown as blue curves, can capture different characteristics for each part of the workspace. An uncertain query is shown by the red trajectory which gives the robot with possibility to execute or not the



(a) Initial training set (b) Generalized trajectories (c) Requested demonstrations

Figure 8: Incrementally adding demonstrations on-demand. The numbers indicate the volume of the convex hull formed by the trajectories. (a) Initial training set. (b) Generalizations satisfying the confidence threshold. (c) Additional requested demonstrations.

solution. The robot could also execute the trajectory slowly while asking the human to supervise its execution, or to request the human for a new demonstration.

4.3 Increasing Coverage On-Demand

In a third set of experiments, the robot was trained on a collision-free workspace with the goal of expanding its skills on-demand. Six demonstrations were initially provided by moving the robot hand from its home position to an arbitrary location as shown in Figure 8(a). The user arbitrarily defined new goal positions using a tracked wand. The robot then tried to generate a trajectory to reach this goal given the current set of demonstrations. If the uncertainty (5) was larger than the specified threshold, the robot requested a new demonstration, which was then added to the training set and used to re-train the GPs.

In total, the user indicated 18 new queries whose generalized trajectories are shown in Figure 8(b). For 11 cases, the trajectories could be generalized with confidence. The remainder seven cases required additional demonstrations as the robot uncertainty was above the 9 cm threshold. These additional demonstrations are shown in Figure 8(c). To quantify the performance we consider the volume of the convex hull defined by the trajectories. The initial demonstration set used to initialize the GPs covered a volume of 0.036 m^3 (Figure 8(a)). The generalized trajectories covered a volume of 0.303 m^3 (Figure 8(b)). The additional set of demonstrations covered a volume of 0.197 m^3 (Figure 8(c)). The increased workspace volume achieved by the robot is eight times larger than the initial volume of the initial demonstration set. This experiment indicates that, under the active learning setting, a robot can increase its initial skill set while being efficient in the number of requests.

The previous experiments show that differently from methods based on mixture of models such as GMMs, here, there is no need for computing the correct number of components as the data grow, thus, avoiding heuristics or cross-validations.

5 Conclusions

This paper presented an algorithm that allows a robot to incrementally increase its skills by reasoning *when* to request demonstrations. The method uses GPs to provide a measure on the confidence in which the training set is being extrapolated, which is used to trigger a new demonstration request. DMPs are then learned on the GP output to guarantee that the desired position can be achieved exactly. The combination also provides a way to contextualize the demonstration, thus, offering DMPs the capability to address different types of demonstrations. A few decision have been made, such as the criterion for queries, the threshold on the uncertainty, and the space in which to encode demonstrations. We are currently working to automate many of such decisions. We focused on the reaching position as a context, but as a future work, we will consider other contexts such as the position of obstacles, the geometry of objects and the ergonomics of different interacting humans.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand), from the project BIMROB of the "Forum für interdisziplinäre Forschung" (FiF) of the TU Darmstadt, from the European Union's Horizon 2020 research and innovation programme under grant agreement 640554 (SKILLS4ROBOTS), and 645582 (RoMaNS). The authors would like to thank Riad Akrour for invaluable feedback.

References

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

[2] A. P. Shon, D. Verma, and R. P. Rao. Active imitation learning. In *AAAI*, pages 756–762, 2007.

[3] S. Chernova and M. Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 233. ACM, 2007.

[4] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1, 2009.

[5] A. Baranes and P.-Y. Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

[6] D. H. Grollman and O. C. Jenkins. Dogged learning for robots. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2483–2488. IEEE, 2007.

[7] D. Silver, J. A. Bagnell, and A. Stentz. Active learning from demonstration for robust autonomous navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 200–207. IEEE, 2012.

[8] M. Cakmak, C. Chao, and A. L. Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010.

[9] M. Cakmak and A. L. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012.

[10] C. Moulin-Frier and P.-Y. Oudeyer. Learning how to reach various goals by autonomous interaction with the environment: unification and comparison of exploration strategies. In *1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM2013), Princeton University, New Jersey*, 2014.

[11] S. Calinon and A. Billard. Active teaching in robot programming by demonstration. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 702–707. IEEE, 2007.

[12] D. Lee and C. Ott. Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31(2-3):115–131, 2011.

[13] M. Ewerton, G. Maeda, G. Kollegger, J. Wiemeyer, and J. Peters. Incremental imitation learning of context-dependent motor skills. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 351–358. IEEE, 2016.

[14] F. B. Farraj, T. Osa, N. Pedemonte, J. Peters, G. Neumann, and P. Giordano. A learning-based shared control architecture for interactive task execution. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[15] R. Ahmadzadeh, M. A. Rana, and S. Chernova. Generalized cylinders for learning, reproduction, generalization, and refinement of robot skills. In *Proceedings of Robotics: Science and Systems*, 2017.

[16] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2616–2624, 2013.

[17] S. Calinon, I. Sardellitti, and D. G. Caldwell. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 249–254. IEEE, 2010.

[18] M. Ewerton, G. Neumann, R. Lioutikov, H. Ben Amor, J. Peters, and G. Maeda. Learning multiple collaborative tasks with a mixture of interaction primitives. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 1535–1542, 2015.

[19] M. Schneider and W. Ertel. Robot learning by demonstration with local gaussian process regression. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 255–260. IEEE, 2010.

[20] T. Osa, N. Sugita, and M. Mitsuishi. Online trajectory planning and force control for automation of surgical tasks. 2017.

[21] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815, 2010.

[22] D. Forte, A. Gams, J. Morimoto, and A. Ude. On-line motion synthesis and adaptation using a trajectory database. *Robotics and Autonomous Systems*, 60(10):1327–1339, 2012.

[23] T. Matsubara, S.-H. Hyon, and J. Morimoto. Learning parametric dynamic movement primitives from multiple demonstrations. *Neural Networks*, 24(5):493–500, 2011.

[24] S. Bitzer and S. Vijayakumar. Latent spaces for dynamic movement primitives. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 574–581. IEEE, 2009.