3rd HAND
FP7-ICT-2013-10-610878
1 October 2013 (48months)

D4.1:
Scientific report on Modeling and Prediction of Human Intent for Primitive Activation

TUDa

⟨mail@jan-peters.net, maeda@ias.tu-darmstadt.de⟩

Due date of deliverable: M12
Actual submission date: M13
Lead Partner: Inria
Partners: TUDa, Inria, USTT
Revision: draft
Dissemination level: PU

This deliverable reports the modeling of the human-robot collaborative behavior used to recognize actions and activate robot primitives. This report describes a new framework based on probabilistic interaction primitives for creating such a collaborative model. The assumption that human-robot trajectories can be linearly correlated in an appropriate lower dimensional space forms the basis of our approach. We have investigated multi patterns of interaction with a mixture of models, and also how to generalize probabilistic primitives when manipulating objects of different sizes and shapes. We also discuss how this work is currently being integrated with the works in deliverable D3.1.

Executive Summary

This report discusses the first year achievements in human action recognition and activation of the corresponding robot movement primitive for collaborative tasks. Throughout the first year we have introduced Interaction Probabilistic Movement Primitives [MEL+14, ENL+15] (in short, interaction primitives), as a framework aimed at probabilistically modeling human-robot collaborative tasks. Interaction primitives are used to predict the human intent (action recognition), which is then used to activate and coordinate the movement primitive of the robot assistant. To generalize the robot grasping to different objects, we have also investigated how parameters extracted from point-cloud images can be used to warp known objects into new objects [BKP14], which are then used to modify the movement primitives. Currently we are addressing the close connections between the interaction primitives and higher level plans of deliverable D3.1. Specifically, we are considering generalization to new scenarios by making use of relational policies that satisfy an underlying reward with the work of [MPG+15], and the pre-triggering of robot primitives with the recognition of assembly phases as proposed in [BML+15].

So far progress has been evaluated by experiments with Darias, the robotic platform based in TU-Darmstadt, consisting of a 7DoF KUKA lightweight arm with a DLR-HIT five-fingered hand. We have shown proof-of-concept experiments where a toolbox is assembled with the help of Darias acting as a third hand.

Role of Modeling and Prediction of Human Intent for Primitive Activation in 3rdHand

This is an essential part of an assistive robot framework as, ultimately, the action that the robot takes must not only be correlated to the human observations, but must also be represented as a primitive that can be suitably used by a low-level control loop. Action recognition for primitive activation is useful to inform the logical learning method in [MPG+15] in order to improve higher level decisions. Although not addressed during the first year, future work will leverage on the interplay between the detection of pair-wise interaction phases and situational awareness [BML+15, MMRL15], described in D3.1, with action recognition at the primitive level. Combined with D3.1, our role is to provide algorithms for assistive/collaborative robot policies at both relational and movement primitive level of representation.

Contribution to the 3rdHand scenario

In this deliverable we have developed collaborative movement primitives in multi-task scenarios, by using both supervised [MEL+14] and unsupervised [ENL+15] learning; and also proposed the generalization of primitives for grasping different objects based on warping functions [BKP14]. The control accuracy of Darias under the policies generated by the proposed methods using hand over of different objects during the assembly of a toolbox were also evaluated.

1 Tasks, objectives, results

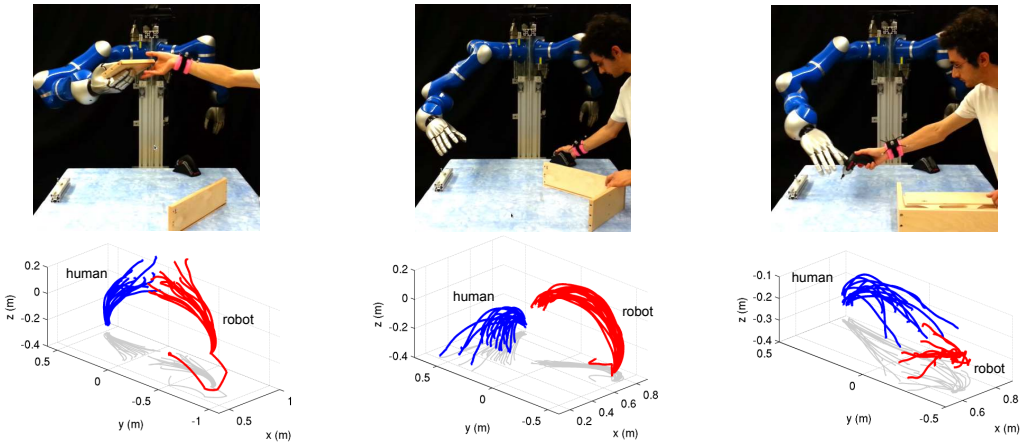
1.1 Planned work

As a deliverable specified within Work Package 4 (WP4) (Learning Primitives for Cooperative Manipulation) the planned work is largely motivated by the following requirements

- Our method must be able to model the mutual dependencies of two agents, particularly the interaction between a human and a robot
- The robot must be proactive, it must predict and execute a task by reasoning about the next, most probable human action
- In the case of unsatisfactory response, the robot skills must be modified on-demand by means of natural human instruction, ideally by simply observing a demonstration from the human coworker, rather than by direct kinesthetic teaching
- The robot must be able to compose complex actions based on simpler collaborative primitives
- The robot must optimize its own primitives in order to increase the efficiency of the interaction

1.2 Actual work performed

During the first year we have been able: (1) to develop and validate the use of interaction primitives for handover of different parts during assembly tasks [MEL+14], (2) to generalize probabilistic primitives during grasping using warping functions [BKP14], and (3) to address multiple interaction patters between the human coworker and the robot assistant [ENL+15] using unlabeled data, (4) we have been able to record trajectories and poses of objects with high-precision magnetic trackers (USTT), motion capture (Inria), and off-the-shelf Kinect and cameras (UIBK, TUDa).



(a) Handing over a plate (b) Handing over a screw (c) Holding the screw driver

Figure 1: Current results where the robot acts as a third hand by handing over a plate or a screw depending on the action recognized. The robot is also able to hold and release a tool such that the coworker can have both hands free.

We evaluated the physical capabilities and control accuracy of the Darias platform by proposing hand over experiments during the assembly of a toolbox. We could accomplish grasping of small and delicate parts, such as the head of a five millimeter screw, by using auxiliary stands for objects.

Figure 1 shows the three subtasks that were used during the experiments. The bottom row shows the demonstrated pairs of human-robot trajectories. Videos of our current results in collaborative interaction primitives are available from the two links:
`http://youtu.be/2Ok6KQQQDNQ` related to the published material [MEL+14], and `http://youtu.be/9XwqW_VObDw` for the submitted publication [ENL+15].

There are many problems to be solve in order to fulfill all requirements of Work Package 4 of the 3rdHand proposal. Great part of future work relates to the integration of several of the works described in D3.1. While we have been able to recognize a task by observing the current human movement, we have not yet implemented preemptive action in the interaction primitive framework. In the short term, we expect to add anticipative/preemptive robot behavior to the interaction primitive framework, in the sense that primitives can be pre-triggered by learning a sequence of co-related chain of tasks. This can be achieved by identifying such a sequence using conditional random fields with the work of temporal segmentation of pair-wise interactions [BML+15].

The results of this deliverable are constrained to a specific scenario, and the assembly of a toolbox can not be generalized since it is based on a pure imitation learning approach. We will incorporate Relational Inverse Reinforcement Learning [MPG+15] reported in D3.1 to provide generalization to interaction primitives such that different tasks can be accomplished from the same demonstrations.

Interaction primitives must also seamlessly integrate with optimal trajectories

provided by the Open Robot Simulator of USTT (described in see D5.1). We have been working on the parsing of whole trajectories into segments encoded in a skill library by using the algorithms of [LMPN15], described in D3.1.

Also, current work is addressing the use of motion capture data from two collaborators to generate feasible trajectories for Darias. The outcome of this work will free the coworker from doing kinesthetic teaching. Instead the worker is expected to simply execute the task by himself, while the vision/motion capture system is used to capture his/her measurements.

1.3 Relation to the state-of-the-art

Action recognition based on data analysis and classification of interactions between multiple persons has long been addressed within the computer vision community, in particular due to interests in visual surveillance tasks, e.g., tracking of multiple pedestrians. Action recognition has motivated several works based on hidden Markov models (HMMs) and graphical models in general, such as Oliver et al. [ORP00], [LON10], and [BAVE+13] for example. Inference on the graphical model allows a robot to anticipate human activity and choose a corresponding, preprogrammed robot response. Wang et al. [WMD+13] propose the intention-driven dynamics model. Koppula et al. [KS13] use a conditional random field with subactivities, human poses, object affordances and object locations over time.

Also motivated by a robot assistant, Tanaka et al. [TKSK12] use a Markov model to predict the positions of a worker in an assembly line. The space in which the worker moves is discretized into different regions. A Gaussian mixture model relates positions to procedures. Using this information a robot, then, delivers tools and parts to a human worker along the assembly line. Llorens et al. [LBA] present hybrid design for a to arm robot to be used on the shoulder. Colored Petri Nets have been used to model high level transitions, e.g., the robot holds the panel in position or the screw is in position, etc. Petri Nets accounts for discrete control transitions while at the motion level, Partial Least Squares Regression has been used to find the best action of the robot at future time steps.

Interaction dynamics need to be specified in a way that allows for robust reproduction of the collaborative task under different external disturbances, and a common approach is based on direct force sensing or emulation. Rozo et al. [RCC+13] proposed a framework for haptic collaboration between a human and a robot manipulator. Given a set of kinesthetic demonstrations, their method learns a mapping between measured forces and the impedance parameters used for actuating the robot, e.g., the stiffness of virtual springs governing the collaborative task. In another force-based approach, Lawitzky et al. [LMLH12] proposed learning physical assistance in a collaborative transportation task. In the early learning phase, the robot uses the measured force values to follow the human guidance during the task. Recorded force and motion patterns are then used to learn a Hidden Markov Model (HMM) which can predict the human's next action, and over time the robot learns to take over a more active role in the interaction. Kulvicius et al. [KBA+13] also address a transportation task where the two agents are modeled as two point particles coupled by a spring. The forces applied by the other agent tell the robot how to adapt its own trajectory.

Although graphical models and HMMs have a history of successful applications for action and intention recognition in a discretized symbolic level, the generation

of the commands that determine the interaction dynamics at the low level is usually addressed by a different representation, e.g. a lower-level HMM [LON10] or DMPs [GNIU14]. This needs is clear in the work of [LBA] where Petri nets are used to control the state of the actions in a higher level representations while the actions itself are predicted by partial least-square regression. As shown in [MEL+14], the principal distinction of our method in relation to the vast bodies of work of discrete graphical models and continuous interaction dynamics is that it provides solutions to both problems with a single representation that fully correlates human-robot interaction that can be easy to acquire by imitation learning.

References

[BAVE+13] Heni Ben Amor, David Vogt, Marco Ewerton, Erik Berger, Bernhard Jung, and Jan Peters. Learning responsive robot behavior by imitation. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3257–3264, 2013.

[BKP14] S. Brandl, O. Kroemer, and J. Peters. Generalizing manipulations between objects using warped parameters. In *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2014.

[BML+15] Andrea Baisero, Yoan Mollard, Manuel Lopes, Marc Toussaint, and Ingo Lutkebohle. Temporal segmentation of pair-wise interaction phases in sequential manipulation demonstrations. In *Submitted to: Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[ENL+15] 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 *Submitted to: Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[GNIU14] A Gams, B. Nemec, AJ. Ijspeert, and A Ude. Coupling movement primitives: Interaction with the environment and bimanual tasks. *Robotics, IEEE Transactions on*, 30(4):816–830, Aug 2014.

[KBA+13] Tomas Kulvicius, Martin Biehl, Mohamad Javad Aein, Minija Tamosiunaite, and Florentin Wörgötter. Interaction learning for dynamic movement primitives used in cooperative robotic tasks. *Robotics and Autonomous Systems*, 61(12):1450–1459, 2013.

[KS13] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. In *Robotics: Science and Systems*, 2013.

[LBA] Baldin Llorens-Bonilla and H Harry Asada. A robot on the shoulder: Coordinated human-wearable robot control using coloured petri nets and partial least squares predictions.

[LMLH12] M. Lawitzky, J.R. Medina, Dongheui Lee, and S. Hirche. Feedback motion planning and learning from demonstration in physical robotic assistance: differences and synergies. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3646–3652, Oct 2012.

[LMPN15] R. Lioutikov, G. Maeda, J. Peters, and G. Neumann. A parser for constructing movement primitive libraries. In *In preparation*, 2015.

[LON10] Dongheui Lee, Christian Ott, and Yoshihiko Nakamura. Mimetic communication model with compliant physical contact in humanhumanoid interaction. *The International Journal of Robotics Research*, 29(13):1684–1704, 2010.

[MEL+14] G. Maeda, M. Ewerton, R. Lioutikov, H. Ben Amor, J. Peters, and G. Neumann. Learning interaction for collaborative tasks with probabilistic movement primitives. In *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2014.

[MMRL15] Yoan Mollard, Thibaut Munzer, Pierre Rouanet, and Manuel Lopes. Learning and representing object assembly tasks. In *under preparation*, 2015.

[MPG+15] Thibaut Munzer, Bilal Piot, Mathieu Geist, Olivier Pietquin, and Manuel Lopes. Inverse reinforcement learning in relational domains. In *submitted to AAAI'15*, 2015.

[ORP00] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, 2000.

[RCC+13] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras. Learning collaborative impedance-based robot behaviors. In *AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA, 2013.

[TKSK12] Yasufumi Tanaka, Jun Kinugawa, Yusuke Sugahara, and Kazuhiro Kosuge. Motion planning with worker's trajectory prediction for assembly task partner robot. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1525–1532. IEEE, 2012.

[WMD+13] Z. Wang, K. Muelling, M. P. Deisenroth, H. Ben Amor, D. Vogt, B. Schoelkopf, and J. Peters. Probabilistic movement modeling for intention inference in human-robot interaction. (7):841–858, 2013.

2 Annexes

- Learning Interaction for Collaborative Tasks with Probabilistic Movement Primitives.
  Maeda, G.; Ewerton, M.; Lioutikov, R.; Amor, H.; Peters, J. Neumann, G.
  Accepted for publication: Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), 2014

  This paper introduces the use of Probabilistic Movement Primitives (ProMPs) for interaction. This paper uses the TUDa-based experimental platform as a third hand to perform the assembly of a toolbox.

  Abstract:
  This paper proposes the formulation of a probabilistic interaction model based on movement primitives for robots that work in collaboration with

a human partner. Since the human partner can execute a variety of unforeseen tasks a requirement of our system is that the assistant robot must be able to adapt and learn new skills on-demand, without the need of an expert programmer. Thus, this paper leverages on the framework of imitation learning and also on its application to human-robot interaction using the recently introduced concept of Interaction Primitives (IPs). We present the use of Probabilistic Movement Primitives (ProMPs) as an interaction framework that both recognizes the action of a human and generates the appropriate movement primitive of the assistant robot. We evaluate our method on experiments using a lightweight arm interacting with a human partner and also using motion capture trajectories of two humans assembling a box. The advantages of ProMPs in relation to the original formulation for interaction are exposed.

- Learning Multiple Collaborative Tasks with a Mixture of Interaction Primitives
  Ewerton, M.; Neumann, G.; Lioutikov, R.; Amor, H.; Peters, J.; Maeda, G. Submitted to: 2015 IEEE International Conference on Robotics and Automation (ICRA)

  This paper proposes a mixture of interaction primitives to learn different collaborative tasks from unlabeled training data. This paper uses the TUDa based experimental platform as a third hand to perform the assembly of a toolbox.

  Abstract:
  Learning human-robot interaction from demonstration presents a number of open issues. This paper addresses the problem of learning interaction from unlabeled demonstrations. Our proposed method builds upon the framework of *Interaction Primitives* to encode the parameters that describe human-robot interaction in a lower dimensional space. Due to the fact that one single Interaction Primitive does not apply across different interaction scenarios, it is usually assumed that the training data is labeled and one model is learned for each label. Our contribution is to propose unsupervised learning of models of different human-robot interactions. This work endows Interaction Primitives with the capability of performing clustering and conditioning on non-linear distributions using *Gaussian Mixture Models* (GMMs). We validate our algorithm by assembling a box with the help of a KUKA lightweight arm. We record the movements of a human using a motion capture system and the movements of the robot, while being driven by kinesthetic teaching. Given a number of unlabeled demonstrations of different interactions, our algorithm learns a number of probabilistic interaction models. In the test phase, we apply conditioning to infer the reaction of the robot given the movement of the human.

- Generalizing Manipulations Between Objects using Warped Parameters
  Brandl, S.; Kroemer, O.; Peters, J. Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), 2014

  This paper proposes grasping generalization by extracting parameters of point-cloud data in the form of warping functions. The method is evaluated in with a pouring task using the 3rdHand experimental platform at TU-Darmstadt.

Abstract:
One of the key challenges for learning manipulation skills is generalizing between different objects. The robot should adapt both its actions and the task constraints to the geometry of the object being manipulated. In this paper, we propose computing geometric parameters of novel objects by warping known objects to match their shape. We refer to the parameters computed in this manner as warped parameters, as they are defined as functions of the warped objects point cloud. The warped parameters form the basis of the features for the motor skill learning process, and they are used to generalize between different objects. The proposed method was successfully evaluated on a pouring task both in simulation and on a real robot.

Learning Interaction for Collaborative Tasks with Probabilistic Movement Primitives

Guilherme Maeda[1], Marco Ewerton[1], Rudolf Lioutikov[1], Heni Ben Amor[2], Jan Peters[1,3], Gerhard Neumann[1]

***Abstract*— This paper proposes a probabilistic framework based on movement primitives for robots that work in collaboration with a human coworker. Since the human coworker can execute a variety of unforeseen tasks a requirement of our system is that the robot assistant must be able to adapt and learn new skills on-demand, without the need of an expert programmer. Thus, this paper leverages on the framework of imitation learning and its application to human-robot interaction using the concept of Interaction Primitives (IPs). We introduce the use of Probabilistic Movement Primitives (ProMPs) to devise an interaction method that both recognizes the action of a human and generates the appropriate movement primitive of the robot assistant. We evaluate our method on experiments using a lightweight arm interacting with a human partner and also using motion capture trajectories of two humans assembling a box. The advantages of ProMPs in relation to the original formulation for interaction are exposed and compared.**

Fig. 1. Illustration of two collaborative tasks where a semi-autonomous robot helps a worker assembling a box. The robot must predict what is the action to execute, to hand over the screw driver or to hold the box. Its movement must also be coordinated relative to the location at which the human worker executes the task.

I. Introduction

While the traditional use of robots is to *replace* humans in dangerous and repetitive tasks we motivate this paper by semi-autonomous robots that *assist* humans. Semi-autonomous robots have the ability to physically interact with the human in order to achieve a task in a collaborative manner. The assembly of products in factories, the aiding of the elderly at home, the control of actuated prosthetics, and the shared control in tele-operated repetitive processes are just a few examples of application.

Only recently, physical human-robot interaction became possible due advances in robot design and safe, compliant control. As a consequence, algorithms for collaborative robots are still in the early stages of development. Assistance poses a variety of challenges related to the human presence. For example, Fig. 1 illustrates a robot assistant that helps a human to assemble a box. The robot must not only predict what is the most probable action to be executed based on the observations of the worker (to hand over a screw driver or to hold the box) but also the robot movement must be coordinated with the worker movement. Pre-programming a robot for all possible tasks that a worker may eventually need assistance with is unfeasible. A robot assistant must be able to learn the interaction and to adapt to a variety of unforeseen tasks without the need of an expert programmer.

Motivated by the described scenario, this work proposes the use of imitation learning [1] in the context of collaboration. Imitation learning has been widely used as a method to overcome the expensive programming of autonomous robots. Only recently, however, its application for physical interaction has been introduced under the concept of Interaction Primitives (IP) by Lee at al. in [2], defined as skills that allow robots to engage in collaborative activities with a human partner by Ben Amor et al. in [3].

Leveraging on the framework of [3], our approach is based on probabilistically modeling the interaction using a distribution of observed trajectories. We propose using Probabilistic Movement Primitives (ProMPs) [4] for modeling such a distribution. In a manufacturing scenario such a distribution of trajectories can be obtained by observing how two co-workers assemble a product, several times throughout the day, providing a rich data set for imitation learning. Such a collection of trajectories is used to create a prior model of the interaction in a lower dimensional weight space. The model is then used to recognize the intention of the observed agent and to generate the movement primitive of the unobserved agent given the same observations. The movement primitive of the unobserved agent can then be used to control a robot assistant.

The main contribution of this paper is the introduction of the Probabilistic Movement Primitives [4] in the context of imitation learning for human-robot interaction and action

[1]Intelligent Autonomous Systems Lab, Technische Universitaet Darmstadt, 64289 Darmstadt Germany. Correspondence should be addressed to maeda@ias.tu-darmstadt.de

[2]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA.

[3]Max Planck Institute for Intelligent Systems Spemannstr. 38, 72076 Tuebingen, Germany

recognition. We will show how Interaction ProMPs can be applied to address the three main problems previously illustrated in Fig. 1, that is: (a) learning a collaborative model by imitation learning and thus avoiding expert programming, (b) the ability to recognize a task by observing the worker, and (c) the coordination of the assistant movement in relation to the worker movement. We also show the advantages of ProMPs over the original DMP-based framework [3], and present an algorithm for aligning data using local optimization in order to avoid the issue of slope constraints typical of dynamic time warping.

II. Related Work

The data-driven analysis and classification of interactions between multiple persons has long been addressed within the computer vision community. In particular visual surveillance tasks, e.g., tracking of multiple pedestrians, require methods for identifying the occurrence and type of person-to-person interactions. In a seminal paper, Oliver et al. [5] show that hidden Markov models (HMMs), and more generally graphical models, are suited for representing the mutual dependencies of the behaviors between interacting agents. Graphical models have gained popularity in the field of human-robot interaction as they naturally include temporal information into the inference process and the Bayesian semantics provides a simple way to encode prior knowledge. In [6], Lee et al. use a hierarchical HMM to learn and represent responsive robot behaviors. In their approach, a high-level HMM identifies the current state of the interaction and triggers low-level HMMs which correspond to the robot's motor primitives. In order to ensure that the robot adapts to the movement of the human partner, virtual springs are attached between markers on the human body and corresponding positions on the robot. In a similar vein, Ben Amor et al. [7] use a path-map HMM to model interactions in cooperative tasks. In their approach, a backbone of shared hidden states correlates the actions of the interacting agents.

Tanaka et al. [8] use a Markov model to predict the positions of a worker in an assembly line. The space in which the worker moves is discretized into different regions. A Gaussian mixture model relates positions to procedures. Using this information a robot, then, delivers tools and parts to a human worker along the assembly line. Besides HMMs, other probabilistic graphical models have also been used to address interaction tasks. Koppula et al. [9] use a conditional random field with sub-activities, human poses, object affordances and object locations over time. Inference on the graphical model, allows a robot to anticipate human activity and choose a corresponding, preprogrammed robot response. Wang et al. [10] propose the intention-driven dynamics model, which models human intentions as latent states in graphical model. Intentions can be modeled as discrete variables, e.g., action labels, or continuous variables, e.g., an object's final position. The transitions between latent states and the mapping from latent states to observations are modeled via Gaussian Processes. As evidenced by these works, graphical models can be very powerful tools in classifying interactions. However, this often requires a substantial set of training data. In particular for humanoid motion generation with many degrees-of-freedom, it is often challenging to acquire sufficiently large and general data sets.

For more efficient learning and generalization, various authors investigated the projection of the original trajectories into a new, low-dimensional space where correlations between the agents are easier to unravel. Llorens et al. [11] show how such a low-dimensional interaction space can be used to implement an assistive robot arm. Similarly in [7], probabilistic principal component analysis is used to find a shared latent space. Dynamic Movement Primitives (DMPs) [12] allows for a low-dimensional, adaptive representation of a trajectory. The general idea is to encode a recorded trajectory as dynamical systems, which can be used to generate different variations of the original movement. In the context of interaction, Prada et al. [13] present a modified version of DMPs, that adapts the trajectory of one agent to a time-varying goal. By setting the goal to the wrist of another agent, the method can be used to generate handover motions.

Although graphical models and HMMs have been successfully used for action and intention recognition in a discretized symbolic level, the generation of trajectories for the continuous dynamic control of the robot is usually addressed by a different level of representation (e.g. a lower-level HMM [6] or DMPs). In relation to the previously cited works, here, we propose a framework based solely on a continuous movement representation that is used to both recognize actions and generate trajectories in the form of movement primitives; mainly leveraging on DMP-based Interaction Primitives [3] and Probabilistic Movement Primitives (ProMPs) [4]. By using ProMPs rather than DMPs our prosed method naturally correlates different agents directly in the same space in which observations are made, since observations of a task are usually given by their trajectories. This is an advantage in relation to the original framework of [3] since the representation of collaboration in the space of accelerations/forces due to the use of DMPs obfuscates the algorithm and increases its sensitivity to noise in the observations.

III. Proposed Method

This section briefly introduces Probabilistic Movement Primitives for a single degree of freedom as presented in [4] and proposes its extension for interaction and collaboration. Although not covered in this work, in its original proposition, the design of a feedback controller that tracks the distribution of trajectories is also part of ProMPs and the interested reader is referred to [4] for details; here we assume the existence of a human-safe standard feedback controller such as a low-gain PD controller. This section also exposes the main characteristics of the interaction framework based on DMPs in [3] and its relation to the approach of this paper. Finally, a simple local optimization algorithm is proposed for aligning several demonstrations provided by a human.

A. ProMPs for a Single DOF

For the purposes of the following derivations we generically refer to each joint or Cartesian coordinates of a human or robot simply as a degree of freedom (DOF) with position q and velocity $\dot{q}$. Starting with the case of a single DOF, we denote $\boldsymbol{y}(t) = [q(t)\ \dot{q}(t)]^T$ and a trajectory as a sequence $\boldsymbol{\tau} = \{\boldsymbol{y}(t)\}_{t=0,...T}$. We adopt linear regression with n Gaussian basis functions $\boldsymbol{\psi}$. The state vector $\boldsymbol{y}(t)$ can then be represented by a n-dimensional column vector of weights $\boldsymbol{w}$ as

$$\boldsymbol{y}(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}(t) \\ \dot{\boldsymbol{\psi}}(t) \end{bmatrix} \boldsymbol{w} + \boldsymbol{\epsilon}_y, \tag{1}$$

where $\boldsymbol{\Psi}_t = [\boldsymbol{\psi}(t),\ \dot{\boldsymbol{\psi}}(t)]^T$ is a 2×n dimensional time-dependent basis matrix and $\boldsymbol{\epsilon}_y \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_y)$ is zero-mean i.i.d. Gaussian noise. The probability of observing the whole trajectory is then

$$p(\boldsymbol{\tau}|\boldsymbol{w}) = \prod_0^T \mathcal{N}(\boldsymbol{y}(t)|\boldsymbol{\Psi}_t \boldsymbol{w}, \boldsymbol{\Sigma}_y). \tag{2}$$

Similar to DMPs the speed of the execution of the movement is decoupled from the speed of the original trajectory by using a phase variable $z(t)$. The phase variable replaces the time in order to control the location of the basis functions with $\boldsymbol{\psi}(z(t))$. For simplicity we will use $z(t) = t$ such that $\boldsymbol{\psi}(t) = \boldsymbol{\psi}(z(t))$ while remembering that any monotonically increasing function can be used [4].

Each trajectory is now represented by a low-dimensional vector $\boldsymbol{w}$ since the number of basis is usually much smaller than the number of time steps. Trajectory variations obtained by different demonstrations are captured by defining the distribution over the weights $p(\boldsymbol{w}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the learning parameter. The probability of the trajectory becomes

$$p(\boldsymbol{\tau}|\boldsymbol{\theta}) = \int p(\boldsymbol{\tau}|\boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{\theta}) d\boldsymbol{w}. \tag{3}$$

So far $\boldsymbol{\theta}$ captures the correlation among the weights within the trajectory and between demonstrations of the same DOF.

B. ProMPs for Collaboration

The key aspect for the realization of the interaction primitives is the introduction of a parameter $\boldsymbol{\theta}$ that captures the correlation of all DOFs of multiple agents. Assuming that the distribution of trajectories of different agents is normal, then $p(\boldsymbol{w}; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$. Under this assumption we redefine the vector of weights $\boldsymbol{w}$ to account for all degrees of freedom of multiple-agents. Following the definitions in [3] we will refer to the assisted human as the *observed agent*, and assume that he/she provides the observed DOFs of the model (e.g by motion capture). The robot will be referred to as the *controlled agent*.

For an observed agent with P DOFs and a controlled agent with Q DOFs, we construct a row weight vector by concatenating the trajectory weights

$$\bar{\boldsymbol{w}}_d = \{[\boldsymbol{w}_1^T, ...\boldsymbol{w}_p^T, ..., \boldsymbol{w}_P^T]^o,\ \ [\boldsymbol{w}_1^T, ...\boldsymbol{w}_q^T, ..., \boldsymbol{w}_Q^T]^c\} \tag{4}$$

where $\bar{\boldsymbol{w}}_d$ is the augmented weight vector corresponding to the d-th demonstration, $\boldsymbol{w}_p$ is the n-dimensional column vector of weights of the p-th DOF of the observed agent, and $\boldsymbol{w}_q$ is the vector of weights of the q-th DOF of the controlled agent. The mean and covariance are then computed by stacking all demonstration weights

$$\begin{aligned} \boldsymbol{\mu}_w &= \text{mean}([\bar{\boldsymbol{w}}_1, ..., \bar{\boldsymbol{w}}_d, , ..., \bar{\boldsymbol{w}}_D]^T), \\ \boldsymbol{\Sigma}_w &= \text{Cov}([\bar{\boldsymbol{w}}_1, ..., \bar{\boldsymbol{w}}_d, , ..., \bar{\boldsymbol{w}}_D]^T), \end{aligned} \tag{5}$$

where D is the number of demonstrations.

Gaussian conditioning can then be applied on-line as each new observation is made using recursive updates in the form

$$\begin{aligned} \boldsymbol{\mu}_w^+ &= \boldsymbol{\mu}_w^- + \boldsymbol{K}(\boldsymbol{y}^*(t) - \boldsymbol{H}_t^T \boldsymbol{\mu}_w^-) \\ \boldsymbol{\Sigma}_w^+ &= \boldsymbol{\Sigma}_w^- - \boldsymbol{K}(\boldsymbol{H}_t^T \boldsymbol{\Sigma}_w^-) \\ \boldsymbol{K} &= \boldsymbol{\Sigma}_w^- \boldsymbol{H}_t^T (\boldsymbol{\Sigma}_y^* + \boldsymbol{H}_t^T \boldsymbol{\Sigma}_w^+ \boldsymbol{H}_t)^{-1}, \end{aligned} \tag{6}$$

where $\boldsymbol{K}$ is the Kalman gain matrix, $\boldsymbol{y}^*(t)$ is the observed value at time t, $\boldsymbol{\Sigma}_y^*$ is the measurement noise, and the upper-scripts $-$ and $+$ the values before and after the update. The observation matrix $\boldsymbol{H}_t$ is block diagonal and each diagonal entry contains the 2×n basis $[\boldsymbol{\psi}(t),\ \dot{\boldsymbol{\psi}}(t)]^T$ for each observed joint

$$\boldsymbol{H}_t = \begin{bmatrix} \boldsymbol{\Psi}_t & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{\Psi}_t \end{bmatrix} \tag{7}$$

In the collaboration case only measurements of the observed agent are provided. By maintaining consistency with definition (4) where the entries of the observed agent come before the controlled agent, the mean is then $\boldsymbol{\mu}_w = [\boldsymbol{\mu}_w^o \quad \boldsymbol{\mu}_w^c]^T$ and the observation matrix $\boldsymbol{H}_t$ is partitioned as

$$\boldsymbol{H}_t = \left[\begin{array}{cc:cc} (\boldsymbol{\Psi}_t^o)_{(1,1)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{\Psi}_t^o)_{(P,P)} & \boldsymbol{0} & \boldsymbol{0} \\ \hdashline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}^c_{(1,1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}^c_{(Q,Q)} \end{array}\right] \tag{8}$$

where each zero entry is of 2×n dimension. Note that if only positions of the observed agent are provided $(\boldsymbol{\Psi}_t^o)_{(p,p)} = [\boldsymbol{\psi}(t),\ \boldsymbol{0}(t)]^T$.

In general, since (6) is a full state linear estimator, any partial combination of observations (for example when $\boldsymbol{y}^*$ only contains positions, or velocities, or a mixture of both) provides the optimal estimate of states $\boldsymbol{\mu}_w^+$ and their uncertainty $\boldsymbol{\Sigma}_w^+$.

C. Action Recognition for Primitive Activation

Here we use the ProMP framework in a multi-task scenario where each task is one encoded by one interaction primitive.

Consider a specific task $s \in \{1, .., K\}$ and assume that for each task an Interaction ProMP has been generated as it was proposed in section III-B. Using the recursive notation of (6), the upper script $(\cdot)^-$ refers to the parameters of the Interaction ProMP updated up to the previous observation, that is $\boldsymbol{\theta}_s^- = \{\boldsymbol{\mu}_w^-, \boldsymbol{\Sigma}_w^-\}_s$. The probability of the observation

at a subsequent time t given the parameters $\boldsymbol{\theta}_s^-$ of one of the tasks is

$$p(\boldsymbol{y}^*(t);\boldsymbol{\theta}_s^-) = \int p(\boldsymbol{y}^*(t)|\boldsymbol{\Psi}_t\boldsymbol{w},\boldsymbol{\Sigma}_y^*)p(\boldsymbol{w}|\boldsymbol{\theta}_s^-)d\boldsymbol{w} \quad (9)$$

$$= \mathcal{N}(\boldsymbol{y}^*(t)|\boldsymbol{\Psi}_t\boldsymbol{\mu}_w^-, \boldsymbol{\Psi}_t\boldsymbol{\Sigma}_w^-\boldsymbol{\Psi}_t + \boldsymbol{\Sigma}_y^*). \quad (10)$$

The task s can now be recognized by applying Bayes rule

$$p(s|\boldsymbol{y}^*(t)) = \frac{p(\boldsymbol{y}^*(t)|\boldsymbol{\theta}_s^-)p(s)}{\sum_{k=1}^K p(\boldsymbol{y}^*(t)|\boldsymbol{\theta}_k^-)p(k)}, \quad (11)$$

where $p(s)$ is the initial probability of the task (e.g. $p(s) = 1/K$ for uniform distribution). We will evaluate Eqs. (9)-(11) using real collaboration data in the experimental section of this paper.

D. Relation to Interaction DMPs

It is now straightforward to relate our proposed method with the previous interaction primitives based on DMPs [3]. The principal difference is that in the framework of interaction DMPs the weights are mapped from the forcing function $f(t)$ as opposed to the positions $q(t)$. Using the linear basis-function model

$$f(t) = \boldsymbol{\psi}(t)^T\boldsymbol{w}, \quad (12)$$

where $\boldsymbol{\psi}(t)$ are the normalized Gaussian basis functions. Similarly to the ProMP case a distribution of weights $p(\boldsymbol{w})$ is learned based on several demonstrations of a task.

For each DOF, the forcing function adds a nonlinear behavior on the movement which complements a linear and stable spring-damper system

$$\ddot{q} = [\alpha_y(\beta_y(g-q) - \dot{q}/\tau) + f(t)]\tau^2, \quad (13)$$

where g is the goal attractor, α_y, β_y are user-defined parameters that characterize the spring-damper behavior and τ controls the speed of execution. For details on DMPs please refer to [12] and references therein.

When using imitation learning a demonstration is executed and measurements are usually given in the form of positions, which must be differentiated twice such that the forcing function can be computed

$$f(t) = \ddot{q}/\tau^2 - \alpha_y(\beta_y(g-q) - \dot{q}/\tau). \quad (14)$$

Referring back to (6) the Gaussian conditioning is now based on the observation of forces or accelerations, that is $y^*(t) = f(\ddot{q},(\cdot),t)^*$. As our evaluations will show, the fact that forces are usually computed using second derivatives of the position can be restrictive for applications with asynchronous or sparse measurements as the observed accelerations needed for conditioning are hard to obtain in this case. In contrast, in the ProMP framework, it is possible to directly condition on the observed quantities, i.e., the position of the agent.

E. Time Warping with Local Optimization

One issue of imitation learning for trajectories is that multiple demonstrations provided by humans are usually, sometimes severely, warped in time. Demonstrations must be unwarped or time-aligned before the distribution of the weights can be computed. Here we propose aligning trajectories by taking one of the demonstrations as a reference $\boldsymbol{y}_r$ and using local optimization of the time warping function with

$$\boldsymbol{t}_w^{j+1} = v_0^j + \boldsymbol{g}(\boldsymbol{v}^j)\boldsymbol{t}_w^j, \quad (15)$$

where $\boldsymbol{t}_w^j$ represents a vector containing the warped time of demonstration $\boldsymbol{y}_w$ at the j-th iteration of the optimization.

We propose $\boldsymbol{g}$ as a smooth, linear Gaussian-basis-model with P weights $\boldsymbol{v}^j = [v_1^j, ..., v_P^j]$ as the parameters to be optimized. The extra parameter v_0^j is used to shift the time which is useful when the reference and warped trajectories are, in fact, identical but start at different instants. The optimization is initialized with $v_0^j = 0$ and $\boldsymbol{t}_w^j = \boldsymbol{t}_r$ for $j=1$. The parameters $\boldsymbol{v}^j$ are optimized with gradient descent to decrease the absolute cumulative distance between the reference and warped trajectories

$$\boldsymbol{v} = \arg\min_v \sum_{k=0}^{K} |\boldsymbol{y}_r(\boldsymbol{t}_r(k)) - \boldsymbol{y}_w(v_0^j + \boldsymbol{g}(\boldsymbol{v}^j)\boldsymbol{t}_w^j)|. \quad (16)$$

While Dynamic Time Warping (DTW) [14] is widely used for such problems, our local method forces alignment without "jumping" the indexes of the warped time vector which is an usual outcome of DTW and renders unrealistic and non-smooth trajectories. While this problem is usually minimized by imposing a slope constraint [14], the use of a smooth function $\boldsymbol{g}$ not only avoids the tunning of this parameter but also preserves the overall shape of the trajectory.

IV. Experiments

This section presents results on a simple simulated scenario to compare the differences between the original work of Interaction DMPs with Interaction ProMPs. Next, we evaluate the accuracy of Interaction ProMPs for generating reference trajectories for an anthropomorphic robot arm conditioned on the movement of a human. Finally, we will show experimental results with Interaction ProMPs used in a collaborative scenario of a box assembly to both recognize and predict the action of two collaborators.

A. Comparison with Interaction DMPs

In a typical interaction task the observations of a coworker might arrive asynchronously, at irregular periods of time, for example, when the measurement signal is prone to interruption (a typical case is occlusion in motion capture systems). Fig. 2 (a) illustrates a simple case where both observed and controlled agents have a single joint each. The training data was created by sketching two sets of trajectories on a PC screen using a computer mouse. We than use these two sets as proxies of the observed and controlled agents resulting on the initial distribution of trajectories (in blue). The upper

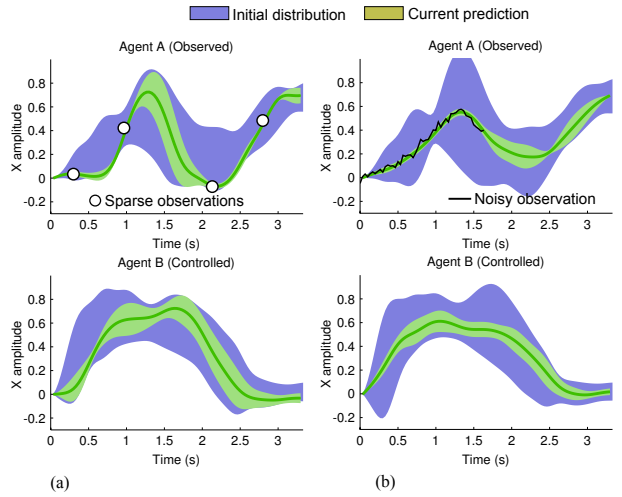


Fig. 2. Two scenarios where the Interaction ProMPs are advantageous over Interaction DMPs. (a) Sparse and asynchronous observations. (b) Noisy stream of observed data ($\sigma^2 = 0.04$). The patches represent the $\pm 2\sigma$ deviations from the mean.

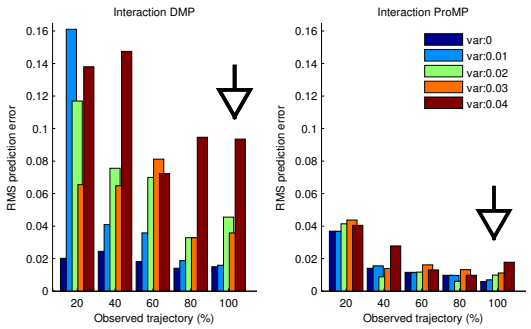


Fig. 3. Root-mean-square prediction error of the movement of collaborator B as a function of the number of observed samples of the trajectory of collaborator A. The different bars indicate the amount of noise added to the observation of the position of collaborator A.

plot shows the prediction (in green) of the observed agent after observing four measurements. Note that following (4) the predicted mean $\boldsymbol{\mu}_w^+$ has the dimension of the augmented weight vector, that is, if each single-DOF agent trajectory is encoded by n basis functions $\boldsymbol{\mu}_w^+$ is a column vector of size $2n$. The bottom figure depicts the predicted distribution of the controlled agent. Note that the same experiment can not be reproduced with Interaction DMPs as the second derivative on such sparse measurement is hard to compute and introduce innacuracies on the representation of the true force.

In Fig. 2 (b) the ProMP is being conditioned on a constant synchronous stream of noisy position measurements. The plot shows the case where the true trajectory is corrupted with a Gaussian noise with variance $\sigma^2 = 0.04$. Interaction DMPs suffer from noisy position measurements as the observation must be differentiated twice to compute the forcing function. While low-pass filters alleviate this problem, the introduction of phase lag is an issue that can be potentially

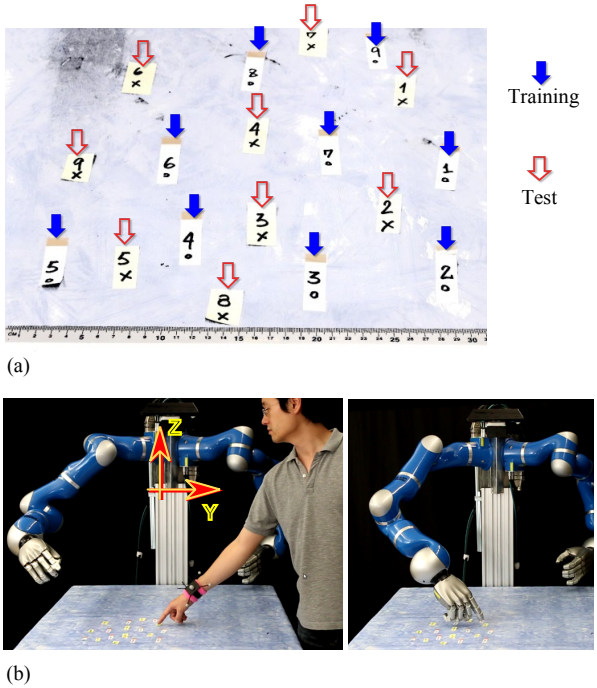


Fig. 4. An interactive task where the robot has to point at the same position previously pointed by the human. The robot, however, has no exteroceptive sensors and its predicted trajectory is based solely on the correlation with the observed human movement. (a) The nine positions used to create the Interaction ProMP (dot markers) and the extra nine positions used to test the method (cross markers). (b) An example where the human points at the test position #1 and the robot points to the same position.

avoided with ProMPs.

Fig. 3 compares the prediction error over the whole trajectory of Interaction DMPs and ProMPs given the same noisy observed data. With DMPs the error is greatly influenced by the amount of noise while ProMPs show much less sensitivity. For the case where the full trajectory of collaborator A is observed (indicated by the arrow) the prediction error increased by a factor of five times using the Interaction DMPs when noise ranged from a clean signal to a signal of noise variance 0.04. In contrast, the error deteriorates by a factor of two with Interaction ProMPs.

B. Robot Control with Interaction ProMPs

We evaluated the ability of Interaction ProMPs in generating the appropriate movement primitive for controlling a robot based on observations of a human partner. The experiment consisted in measuring the $[x, y, z]$ trajectory coordinates of the wrist of an observed agent via motion capture[1] while pointing at a certain position on a table placed in front of our robot (a 7-DOF anthropomorphic arm with a 5-finger hand). Then, the robot was moved in gravity compensation mode to point with its index finger at the same position on the table while its joint positions were being recorded (kinesthetic teaching). This pair of

[1]All human positions were measured in relation to the world reference frame located at the torso of the robot (as shown in Fig. 4(b)).

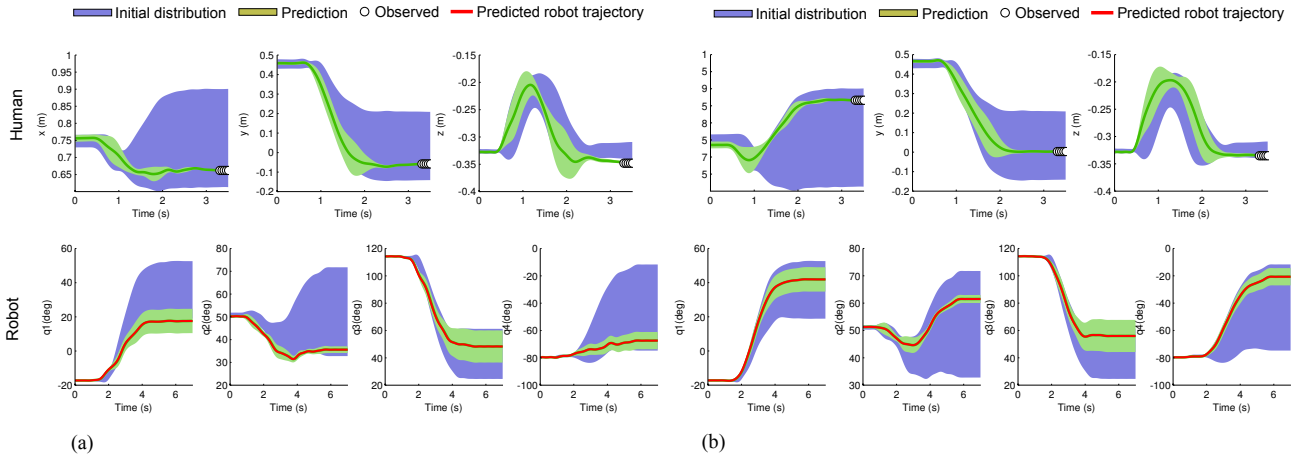


Fig. 5. The upper row shows the human Cartesian coordinates of the wrist. The bottom row shows the first four joints of the 7-DOF robotic arm. (a) Conditioned results of the test position #6. (b) Conditioned results of the test position #8.

trajectories formed by the Cartesian positions of the wrist and the joint positions of the arm where then mapped into the weight space and concatenated as in Eq. (4). In total, nine different positions were pointed to collect training data, sparsely covering an approximate circular area of diameter 30 cm. The pointed positions are shown in Fig. 4(a) by the dots.

After creating the Interaction ProMPs, as described in Section III-B, we defined extra nine marked positions shown in Fig. 4(a) by the crosses. The human then pointed at one of the crosses while motion capture was used to measure the trajectory of the wrist. These observations were then used to condition the Interaction ProMP to predict trajectories for each joint of the robot, whose mean values where used as reference for a standard trajectory tracking inverse dynamics feedback controller with low gains.

Fig. 4(b) shows one example of the interactive task where the human pointed to the position marked by the cross #1, which was not part of the training; the robot was capable of pointing to the same position. Note that the robot was not provided with any exteroceptive feedback, such as cameras, to reveal the location of the pointed position. Although the robot was not directly "aware" of the position of the pointed cross, the interaction primitive provides the robot the capability to predict what movement to make based solely on the observed trajectories of the partner.

Figure 5 shows two examples on the conditioned interaction primitives when the human pointed at positions #6 in (a) and #8 in (b) (refer back to Fig. 4(a) for the physical position of the crosses). The first row in each subplot shows the $[x, y, z]$ coordinates of the wrist. The second row shows the first four joints of the robot, starting from the shoulder joint. Since we are only interested in the final pointing position, the interaction primitive was conditioned on the final measurements of the wrist position. As positions #6 and #8 were physically distant from each other, the difference between their predicted trajectories were quite large in relation to each other, roughly covering the whole span of the prior distribution (in blue) for some certain DOFs of the arm.

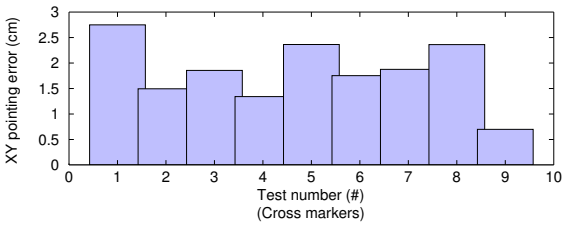


Fig. 6. Accuracy of the pointed positions by the robot when using the test positions given by the cross markers. The error was computed by taking the actual pointed position and the true position of the markers on the table.

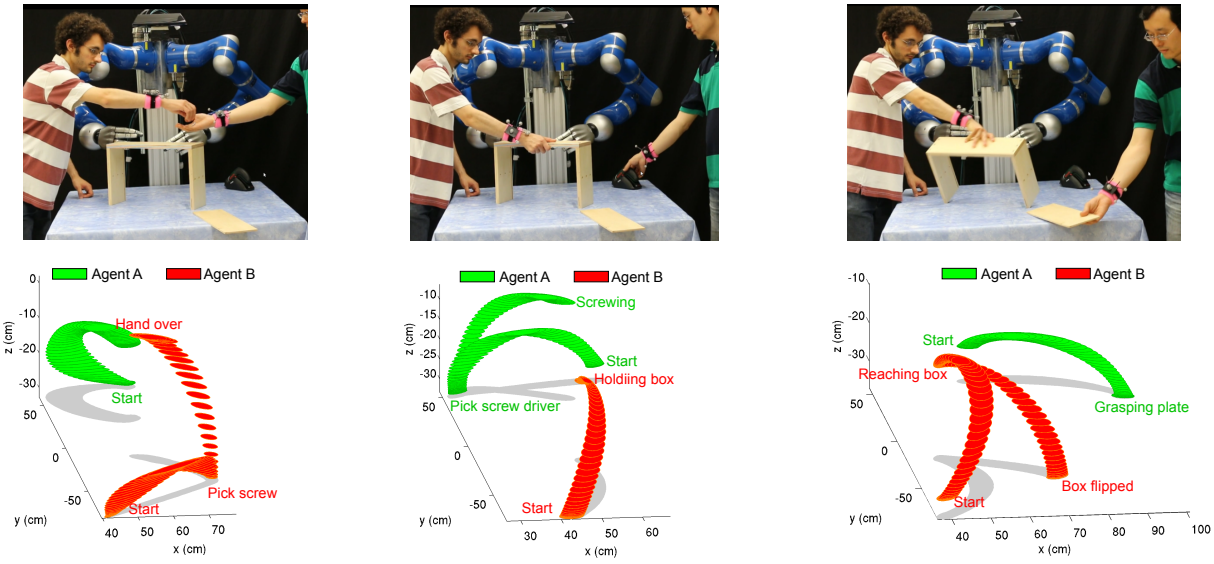
Figure 6 shows the distance error on the plane of the table between the position pointed by the robot and its true position. The robot was able to reasonably point even at locations at the limits of the training data such as position #1, #7, and #8 (see Fig. 4). The maximum error was of 3 cm, or 10% in relation to the total area covered by the training points (approximately a circle of diameter 30 cm). The experiments show that the physical movement of the robot is clearly conditioned by the position indicated by the human (see the accompanying video[2]).

Note that this not a precision positioning experiment; the markers on the wrist were not fixed in a rigid, repeatable manner, neither the finger of the robot could be positioned with milimetric precision during the kinesthetic teaching phase. The framework of Interaction ProMPs allows, however, to seamlessly integrate additional sensing to increase accuracy in precision tasks. This is naturally achieved adjusting the observation vector $\boldsymbol{y}^*$ in (6) and the zero entries in (8) to include new sensory information such as the reference position of a hole in which the robot must insert a peg.

C. Action Recognition for Primitive Activation

While in the previous experiments interaction primitives were evaluated for the case of a single task, here we show

[2]also available from http://youtu.be/2Ok6KQQQDNQ



(a) Hand over (b) Fastening (c) Plate insertion

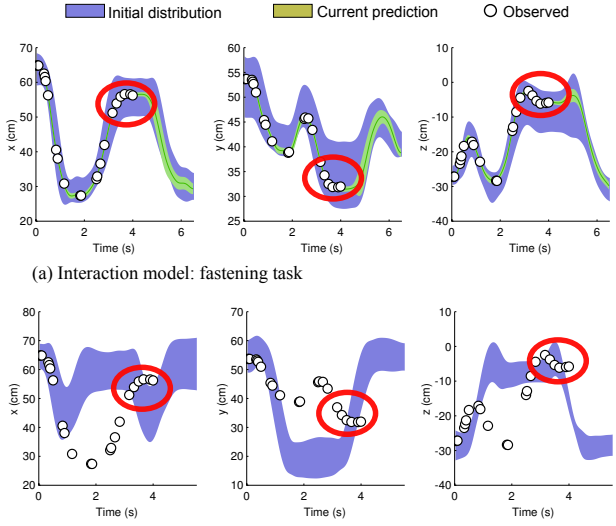
Fig. 7. Three collaborative tasks involved when assembling a box by two co-workers. From left to right, the photos show the hand over of a screw, the fastening of the screw where one agent grasps the screw driver while the other holds the box steadily, and the insertion of a plate, which requires one agent to flip the box such that the slot becomes accessible to the other agent. The distribution of aligned demonstrations for each task are shown under their respective photos. The plot shows the covariance in the x-y plane at each corresponding z height.

how Interaction ProMPs can be used for recognizing the action of the observed agent and to select the appropriate desired movement primitive of the controlled agent. This capability allows the robot to maintain a library of several tasks encoded as Interaction ProMPs and to activate the appropriate primitive based on the observation of the current task.

As shown in the photos of Fig. 7, we collected collaborative data in the form of the Cartesian coordinate positions of the wrists of two humans assembling a box. As in the previous experiment of section IV-B, all measurements were taken in relation to the torso of the robot. The collaborator on the right plays the role of the observed agent while the collaborator at the left plays the role of the controlled agent. The controlled agent will be referred to as the *predicted agent* since he/she can not be controlled.

In the ”hand-over” task shown in Fig. 7(a), the observed agent stretches his hand as a gesture to request a screw. The predicted agent then grasps a screw sitting on the table and hand it over to the collaborator. In the ”fastening” task shown in Fig. 7(b), the observed agent grasps an electrical screwdriver. The predicted agent reacts by holding the box firmly while the observed agent fastens the screw. In the ”plate insertion” task shown in Fig. 7(c), the observed agent grasps the bottom plate of the box. The predicted agent then flips the box such that the slots to which the plate slides in are directed towards the observed agent.

Each task is repeated 40 times. All trajectories are aligned using the method described in Section III-E. The aligned trajectories are shown in Fig. 7 under their respective photos as three-dimensional plots for each of the tasks where the covariance in x-y directions are shown at the corresponding



(a) Interaction model: fastening task

(b) Interaction model: hand over task

Fig. 8. Action recognition based on conditioning the movement primitives of the observed agent. In this example the observations of the fastening task also overlaps with the primitives of the hand over task.

heights (Z direction) of the movement. Interaction ProMPs are created for each task using the distribution of aligned trajectories.

We evaluated action recognition using Eqs. (9)-(11) on the three presented tasks for box assembly. Fig. 8 shows one evaluation as an example. Note from the figure that the majority of observations indicate that the fastening task is taking place. The last five observations (surrounded by the ellipse), however, fits both tasks and could be a potential source of ambiguity in task recognition. Even in those

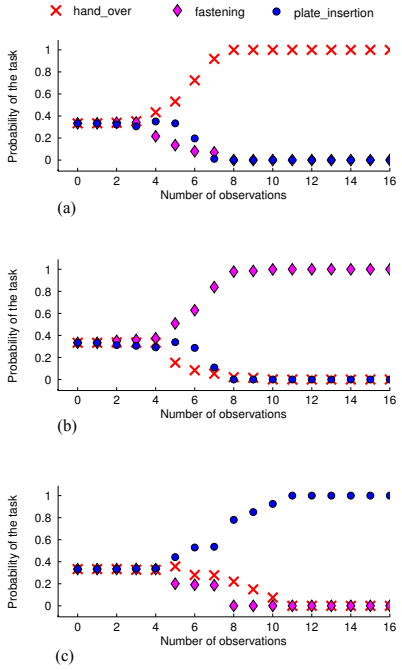


Fig. 9. Action recognition given three different Interaction ProMPs, one for each task involved in assembling the box. The three Interaction ProMPs are conditioned on the same observations of the observed agent. Probabilities of tasks are shown as a function of the number of observations along the trajectory of the observed agent. (a) Recognition of the hand over task. (b) Recognition of the fastening task. (c) Recognition of the plate insertion task.

cases, ProMPs can clearly distinguish among tasks as shown by the plots in Fig. 9 where the probabilities of the task are given as a function of the number of observations. Subplots (a), (b) and (c) show the task recognition for the fastening, hand over and plant insertion tasks, respectively. In general, we observed that 3-5 observations are required to achieve a 100% certainty for each task. (The last part of the accompanying video shows our method controlling the robot assistant to assembly a box with recognition of two different handover tasks).

V. Conclusion

This paper introduced a method for collaboration suited for new applications using semi-autonomous robots whose movements must be coordinated with the movements of a human partner. By leveraging on the original framework of Interaction Primitives [3] we proposed the use of ProMPs for the realization of primitives that capture the correlation between trajectories of multiple agents. This work compared the main differences between DMPs and ProMPs for interaction and advocates the later for applications where measurements are noisy and/or prone to interruption. Using a 7-DOF lightweight arm we evaluated the capability of Interaction ProMPs in generating the appropriate primitive for controlling the robot in an interactive task involving a human partner. We also proposed a method for task recognition that naturally fits the ProMP framework.

Our current work addresses the use of mixture-models to automatically generate different Interaction ProMPs without a priori hand labeling of multiple tasks. We are also investigating tasks in which some of the involved DOFs do not correlate linearly and also when certain tasks do not induce correlation. The later is especially true for tasks where the movement of the agents are not related by causality.

VI. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programmes (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and (FP7-ICT-2009-6) under grant agreement 270327 (ComPLACS). The authors would like to acknowledge Filipe Veiga, Tucker Hermans and Serena Ivaldi for their assistance during the preparation of this manuscript.

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in humanhumanoid interaction," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1684–1704, 2010.

[3] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

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

[5] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, Aug 2000.

[6] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human-humanoid interaction," *Int. Journal of Robotics Research.*, vol. 29, no. 13, pp. 1684–1704, Nov. 2010.

[7] H. Ben Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung, and J. Peters, "Learning responsive robot behavior by imitation," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 3257–3264.

[8] Y. Tanaka, J. Kinugawa, Y. Sugahara, and K. Kosuge, "Motion planning with worker's trajectory prediction for assembly task partner robot," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 1525–1532.

[9] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response." in *Robotics: Science and Systems*, 2013.

[10] Z. Wang, K. Mülling, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013.

[11] B. Llorens-Bonilla and H. H. Asada, "A robot on the shoulder: Coordinated human-wearable robot control using coloured petri nets and partial least squares predictions," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*, 2014.

[12] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[13] M. Prada, A. Remazeilles, A. Koene, and S. Endo, "Dynamic movement primitives for human-robot interaction: comparison with human behavioral observation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1168–1175.

[14] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978.

Learning Multiple Collaborative Tasks with a Mixture of Interaction Primitives

Marco Ewerton[1], Gerhard Neumann[1], Rudolf Lioutikov[1], Heni Ben Amor[2], Jan Peters[1,3] and Guilherme Maeda[1]

***Abstract*— Robots that interact with humans must learn to not only adapt to different human partners but also to new interactions. Such a form of learning can be achieved by demonstrations and imitation. A recently introduced method to learn interactions from demonstrations is the framework of *Interaction Primitives*. While this framework is limited to represent and generalize a single interaction pattern, in practice, interactions between a human and a robot can consist of many different patterns. To overcome this limitation this paper proposes a *Mixture of Interaction Primitives* to learn multiple interaction patterns from unlabeled demonstrations. Specifically the proposed method uses Gaussian Mixture Models of Interaction Primitives to model non-linear correlations between the movements of the different agents. We validate our algorithm with two experiments involving interactive tasks between a human and a lightweight robotic arm. In the first we compare our proposed method with conventional Interaction Primitives in a toy problem scenario where the robot and the human are not linearly correlated. In the second we present a proof-of-concept experiment where the robot assists a human in assembling a box.**

I. Introduction

Robots that can assist us in the industry, in the household, in hospitals, etc. can be of great benefit to the society. The variety of tasks in which a human may need assistance is, however, practically unlimited. Thus, it is very hard (if not impossible) to program a robot in the traditional way to assist humans in scenarios that have not been exactly prespecified.

Learning from demonstrations is therefore a promising idea. Based on this idea, Interaction Primitive (IP) is a framework that has been recently proposed to alleviate the problem of programming a robot for physical collaboration and assistive tasks [1], [2]. At the core, IPs are primitives that capture the correlation between the movements of two agents—usually a human and a robot. Then, by observing one of the agents, say the human, it is possible to infer the controls for the robot such that collaboration can be achieved.

A main limitation of IPs is the assumption that the movements of the human and the movements of the robot assistant are linearly correlated. This assumption is reflected in the underlying Gaussian distribution that is used to model the demonstrations. While this assumption holds for tasks that cover a small region of the workspace (a high-five task in [1] or handover of objects in [2]), it limits the use of IPs in two aspects. First, as illustrated in Fig. 1, a task such as the assembly of a toolbox consists of several interaction patterns that differ significantly from each other and therefore can not be captured by a single Gaussian. Moreover, even within a single interaction pattern, the correlation between the two agents may not be linear, for example, if the movements of the human are measured in the Cartesian space, while the movements of the robot are measured in joint space.

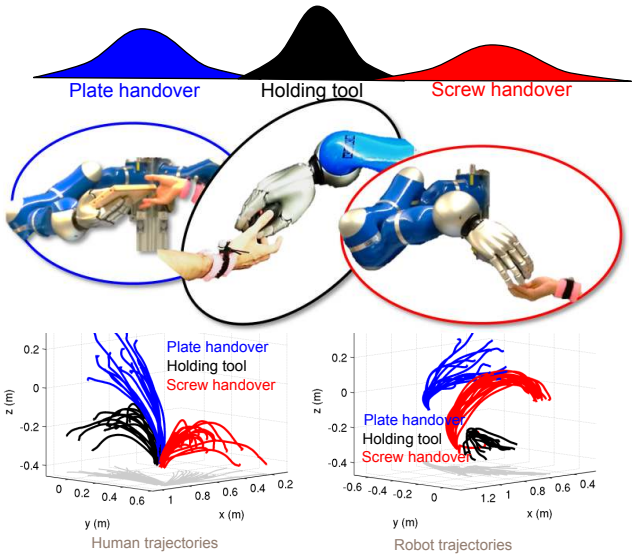


Fig. 1. Illustration of a task consisting of multiple interaction patterns, where each can be represented as an Interaction Primitive. In this work, we want to learn multiple interaction patterns from an unlabeled data set of interaction trajectories.

Manually labeling each subtask (e.g. “plate handover", “screw handover", “holding screw driver") is a way to model interactions with multiple subtasks. Ideally, however, robots should be able to identify different subtasks by themselves. Moreover, it may not be clear to a human how to separate a number of demonstrated interactions in different, linearly correlated groups. Thus, a method to learn multiple interaction patterns from unlabeled demonstrations is necessary. The main contribution of this paper is the development of such a method. In particular, this work uses Gaussian Mixture Models (GMMs) to create a Mixture of Interaction

[1]Intelligent Autonomous Systems Lab, Department of Computer Science, Technical University Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany {ewerton, neumann, lioutikov, peters, maeda}@ias.tu-darmstadt.de

[2]Institute of Robotics and Intelligent Machines, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA hbenamor@cc.gatech.edu

[3]Max Planck Institute for Intelligent Systems, Spemannstr. 38, 72076 Tuebingen, Germany jan.peters@tuebingen.mpg.de

Probabilistic Movement Primitives [2].

The remainder of this paper is organized as follows. Section II presents related work. In Section III, Probabilistic Movement Primitives (ProMPs) and Interaction ProMPs are briefly introduced, followed by the proposition of the main contribution of this paper: a Mixture of Interaction ProMPs based on Gaussian Mixture Models (GMMs). Section IV evaluates the proposed method, first on a toy problem that is useful to clarify the characteristics of the method, and then on a practical application of a collaborative toolbox assembly. Section V presents conclusions and ideas for future work.

II. Related Work

Physical human-robot interaction poses the problem of both action recognition and movement control. Interaction dynamics need to be specified in a way that allows for robust reproduction of the collaborative task under different external disturbances, and a common approach is based on direct force sensing or emulation. Rozo et al. [3] proposed a framework for haptic collaboration between a human and a robot manipulator. Given a set of kinesthetic demonstrations, their method learns a mapping between measured forces and the impedance parameters used for actuating the robot, e.g., the stiffness of virtual springs governing the collaborative task. In another force-based approach, Lawitzky et al. [4] proposed learning physical assistance in a collaborative transportation task. In the early learning phase, the robot uses the measured force values to follow the human guidance during the task. Recorded force and motion patterns are then used to learn a Hidden Markov Model (HMM) which can predict the human's next action, and over time the robot learns to take over a more active role in the interaction. Kulvicius et al. [5] also address a transportation task where the two agents are modeled as two point particles coupled by a spring. The forces applied by the other agent tell the robot how to adapt its own trajectory.

Our work differs significantly from the cited works in the sense that our method does not use nor emulate force signals, but instead learns the correlation between the trajectories of two agents. Correlating trajectories not only simplifies the problem in terms of hardware and planning/control but also allows us to correlate multi-agent movements that do not generate force during the interaction, for example, the simple gesture of asking and receiving an object.

Graphical models have also been used to describe interaction dynamics. In the computer vision community, HMMs have been widely adopted to model interaction dynamics from input video streams [6], [7]. As a result, graphical models have also gained considerable attention in the field of human-robot interaction. In [8], Hawkins and colleagues use a Bayes network to improve the fluency in a joint assembly task. The Bayes network learns to infer the current state of the interaction, as well as task constraints and the anticipated timing of human actions. Tanaka et al. [9] use a Markov model to predict the positions of a worker in an assembly line. Wang et al. [10] propose the Intention-Driven Dynamics Model (IDDM) as a probabilistic graphical model with observations, latent states and intentions where the transitions between latent states and the mapping from latent states to observations are modeled as Gaussian Processes. Koppula et al. [11] use a conditional random field with sub-activities, human poses, object affordances and object locations over time. Inference on the graphical model, allows a robot to anticipate human activity and choose a corresponding, preprogrammed robot response. Lee et al. [12] learn a hierarchical HMM which triggers action primitives in response to observed behaviors of a human partner.

While very successful for classifying actions, graphical models, however, may not be the best option when it comes to generating motions. In [13], for example, the use of a HMM with discrete states, although very successful in action classification, introduces artifacts into the motion generation part that hinders motion generalization. Therefore, a clear problem in physical human-robot interaction is that while graphical models may be suitable in the action recognition domain, motion generation at the continuous level must also be taken into account. Llorens et al. [14] present hybrid design for a robot to be used on the shoulder. In their work, Petri Nets accounts for discrete control transitions while at the motion level, Partial Least Squares Regression has been used to find the best action of the robot at future time steps.

Perhaps the principal distinction of our method is the use of Interaction Primitives (IPs), introduced by Ben Amor et al. [1] and initially based on dynamical movement primitives [15] and later extended to Probabilistic Movement Primitives [16] with action recognition in the work of Maeda et al. [2]. As shown in [2], Interaction Primitives can be used to not only recognize the action of an agent, but also to coordinate the actions of a collaborator at the movement level; thus overcoming in a single framework both layers of discrete action recognition and continuous movement control. Differently from [2], where different interaction patterns must be hand-labeled, our contribution is the unsupervised learning of a Mixture of Interaction Primitives.

III. Mixture of Interaction Primitives

In this section, we will briefly discuss the Interaction Primitive framework based on Probabilistic Movement Primitives [2], [16], followed by the presentation of the proposed method, based on Gaussian Mixture Models.

A. Probabilistic Movement Primitives

A Probabilistic Movement Primitive (ProMP) [16] is a representation of movement based on a distribution over trajectories. The probabilistic formulation of a movement primitive allows operations from probability theory to seamlessly combine primitives, specify via points, and correlate joints via conditioning. Given a number of demonstrations, ProMPs are designed to capture the variance of the positions q and velocities $\dot{q}$ as well as the covariance between different joints.

For simplicity, let us first consider only the positions q for one degree of freedom (DOF). The position q_t at time

step t can be approximated by a linear combination of basis functions,

$$q_t = \boldsymbol{\psi}_t^T \boldsymbol{w} + \epsilon, \tag{1}$$

where ϵ is Gaussian noise. The vector $\boldsymbol{\psi}_t$ contains the N basis functions ψ_i, $i \in \{1,\ 2,\ 3, ...,\ N\}$, evaluated at time step t where we will use the standard normalized Gaussian basis functions.

The weight vector $\boldsymbol{w}$ is a compact representation of a trajectory[1]. Having recorded a number of trajectories of q, we can infer a probability distribution over the weights $\boldsymbol{w}$. Typically, a single Gaussian distibution is used to represent $p(\boldsymbol{w})$. While a single $\boldsymbol{w}$ represents a single trajectory we can obtain a distribution $p(q_{1:T})$ over trajectories $q_{1:T}$ by integrating $\boldsymbol{w}$ out,

$$p(q_{1:T}) = \int p(q_{1:T}|\boldsymbol{w})p(\boldsymbol{w})\mathrm{d}\boldsymbol{w}. \tag{2}$$

If $p(\boldsymbol{w})$ is a Gaussian, $p(q_{1:T})$ is also Gaussian. The distribution $p(q_{1:T})$ is called a Probabilistic Movement Primitive (ProMP).

B. Interaction ProMP

An Interaction ProMP builds upon the ProMP formulation, with the fundamental difference that we will use a distribution over the trajectories of all agents involved in the interaction. Hence, $\boldsymbol{q}$ is multi-dimensional and contains the positions in joint angles or Cartesian coordinates of all agents. In this paper, we are interested in the interaction between two agents, here defined as the observed agent (human) and the controlled agent (robot). Thus, the vector $\boldsymbol{q}$ is now given as $\boldsymbol{q} = [\boldsymbol{q}^o,\ \boldsymbol{q}^c]^T$, where $(\cdot)^o$ and $(\cdot)^c$ refer to the observed and controlled agent, respectively.

Let us suppose we have observed a sequence of positions $\boldsymbol{q}_t^o$ at m specific time steps t, $m \leq T$. We will denote this sequence by D. Given those observations, we want to infer the most likely remaining trajectory of both the human and the robot.

Defining $\bar{\boldsymbol{w}} = [\boldsymbol{w}_o^T,\ \boldsymbol{w}_c^T]^T$ as an augmented vector that contains the weights of the human and of the robot for one demonstration, we write the conditional probability over trajectories $\boldsymbol{q}_{1:T}$ given the observations D of the human as

$$p(\boldsymbol{q}_{1:T}|D) = \int p(\boldsymbol{q}_{1:T}|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|D)\mathrm{d}\bar{\boldsymbol{w}}. \tag{3}$$

We compute a normal distribution of n demonstrations by stacking several weight vectors $[\bar{\boldsymbol{w}}_1,\ ...,\ \bar{\boldsymbol{w}}_n]^T$, one for each demonstration, such that $\bar{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$. A posterior distribution can be obtained after observing D with

$$\begin{aligned} \boldsymbol{\mu}_w^{new} &= \boldsymbol{\mu}_w + \boldsymbol{K}(D - \boldsymbol{H}_t^T\boldsymbol{\mu}_w) \\ \boldsymbol{\Sigma}_w^{new} &= \boldsymbol{\Sigma}_w - \boldsymbol{K}(\boldsymbol{H}_t^T\boldsymbol{\Sigma}_w) \end{aligned} \tag{4}$$

[1]In order to cope with the different speeds of execution during demonstration, the trajectories must be time-aligned before parameterization. The interested reader is referred to [2] for details.

where $\boldsymbol{K} = \boldsymbol{\Sigma}_w\boldsymbol{H}_t^T(\boldsymbol{\Sigma}_D + \boldsymbol{H}_t^T\boldsymbol{\Sigma}_w\boldsymbol{H}_t)^{-1}$, $\boldsymbol{\Sigma}_D$ is the observation noise, and

$$\boldsymbol{H}_t = \left[\begin{array}{cc:cc} (\boldsymbol{\psi}_t^o)_{(1,1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{\psi}_t^o)_{(P,P)} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0}^c_{(1,1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}^c_{(Q,Q)} \end{array}\right] \tag{5}$$

is the observation matrix where the unobserved states of the robot are filled with zero bases. Here, the human and the robot are assumed to have P and Q DOFs, respectively.

Now, by combining (1), (3) and (4), we can compute the probability distribution over the trajectories $\boldsymbol{q}_{1:T}$ given the observation D. For a detailed implementation the interested reader is referred to [2].

C. Mixture of Interaction ProMPs

The goal of our method is to learn several interaction patterns given the weight vectors that parameterize our unlabeled training trajectories. For this purpose, we learn a GMM in the weight space, using the Expectation-Maximization algorithm (EM) [17].

Assume a training set with n vectors $\bar{\boldsymbol{w}}$ representing the concatenated vectors of human-robot weights as defined in section III-B. In order to implement EM for a GMM with a number K of Gaussian mixture components, we need to implement the Expectation step and the Maximization step and iterate over those steps until convergence of the probability distribution over the weights, $p(\bar{\boldsymbol{w}}|\alpha_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K})$, where $\alpha_{1:K} = \{\alpha_1, \alpha_2, \cdots, \alpha_K\}$, $\boldsymbol{\mu}_{1:K} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_K\}$ and $\boldsymbol{\Sigma}_{1:K} = \{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \cdots, \boldsymbol{\Sigma}_K\}$. Here, $\alpha_k = p(k)$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the prior probability, the mean and the covariance matrix of mixture component k respectively. We initialize the parameters $\alpha_{1:K}$, $\boldsymbol{\mu}_{1:K}$ and $\boldsymbol{\Sigma}_{1:K}$ using k-means clustering before starting the Expectation-Maximization loop. The number K of Gaussian mixture components is found by leave-one-out cross-validation.

The mixture model can be formalized as

$$p(\bar{\boldsymbol{w}}) = \sum_{k=1}^{K} p(k)p(\bar{\boldsymbol{w}}|k) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\bar{\boldsymbol{w}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{6}$$

Expectation step: Compute the *responsibilities* r_{ik}, where r_{ik} is the probability of cluster k given weight vector $\bar{\boldsymbol{w}}_i$.

$$r_{ik} = p(k|\bar{\boldsymbol{w}}_i) = \frac{\mathcal{N}(\bar{\boldsymbol{w}}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\alpha_k}{\sum_{l=1}^{K} \alpha_l \mathcal{N}(\bar{\boldsymbol{w}}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \tag{7}$$

Maximization step: Update the parameters α_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of each cluster k, using

$$n_k = \sum_{i=1}^{n} r_{ik},\ \alpha_k = \frac{n_k}{n}, \tag{8}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} r_{ik}\bar{\boldsymbol{w}}_i}{n_k}, \tag{9}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k}\left(\sum_{i=1}^{n} r_{ik}(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)^T\right). \tag{10}$$

Finally, we want to use our model to infer the trajectories of the controlled agent, given observations from the observed agents. We need to find the posterior probability distribution over trajectories $\mathbf{q}_{1:T}$ given the observations D, as in Section III-B.

In order to compute this posterior using our GMM prior, first we find the most probable cluster k^* given the observation D, using the Bayes' theorem. The posterior over the clusters k given the observation D is given by

$$p(k|D) \propto p(D|k)p(k), \tag{11}$$

where

$$p(D|k) = \int p(D|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|k)\mathrm{d}\bar{\boldsymbol{w}}$$

and

$$p(\bar{\boldsymbol{w}}|k) = p(\bar{\boldsymbol{w}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Thus the most probable cluster k^* given the observation D is

$$k^* = \arg\max_k p(k|D). \tag{12}$$

The output of our algorithm is the posterior probability distribution over trajectories $\mathbf{q}_{1:T}$, using cluster k^* to model our prior in the weight space,

$$p(\mathbf{q}_{1:T}|k^*) = \int p(\mathbf{q}_{1:T}|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|k^*)\mathrm{d}\bar{\boldsymbol{w}}, \tag{13}$$

where

$$p(\bar{\boldsymbol{w}}|k^*) = p(\bar{\boldsymbol{w}}|\boldsymbol{\mu}_{k^*}, \boldsymbol{\Sigma}_{k^*}).$$

Algorithms 1 and 2 provide a compact description of the methods we propose for training and inference respectively.

The mixture model, given observation D, can be written as

$$p(\bar{\boldsymbol{w}}|D) = \sum_{k=1}^{K} p(k|D)p(\bar{\boldsymbol{w}}|k, D), \tag{14}$$

where $p(\bar{\boldsymbol{w}}|k, D)$ is the conditioned mixture component k. In the case of mixture models, we have a combination of linear models which is gated by $p(k|D)$ and hence we can model non-linear correlations.

IV. Experiments

This section presents experimental results in two different scenarios using a 7-DOF KUKA lightweight arm with a 5-finger hand[2].

The goal of the first scenario is to expose the issue of the original Interaction Primitives [1], [2] when dealing with trajectories that have a clear multi-modal distribution. In the second scenario we propose a real application of our method where the robot assistant acts as a third hand of a worker assembling a toolbox (please, refer to the accompanying video[3]).

[2]Regarding the control of the robot, the design of a stochastic controller capable of reproducing the distribution of trajectories is also part of ProMPs and the interested reader is referred to [16] for details. Here we use a compliant, human-safe standard inverse-dynamics based feedback controller.

[3]Also available from http://youtu.be/9XwqW_V0bDw

Algorithm 1 Training

1) Parameterize demonstrated trajectories:
Find vector of weights $\bar{\boldsymbol{w}}$ for each trajectory, such that $\mathbf{q}_t \approx \boldsymbol{\psi}_t^T \bar{\boldsymbol{w}}$.

2) Find GMM in parameter space, using EM:
Initialize GMM parameters $\alpha_{1:K}$, $\boldsymbol{\mu}_{1:K}$ and $\boldsymbol{\Sigma}_{1:K}$ with k-means clustering.

repeat

E step

$$r_{ik} = p(k|\bar{\boldsymbol{w}}_i) = \frac{\mathcal{N}(\bar{\boldsymbol{w}}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\alpha_k}{\sum_{l=1}^{K} \alpha_l \mathcal{N}(\bar{\boldsymbol{w}}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

M step

$$n_k = \sum_{i=1}^{n} r_{ik}, \;\; \alpha_k = \frac{n_k}{n}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} r_{ik}\bar{\boldsymbol{w}}_i}{n_k}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k}\left(\sum_{i=1}^{n} r_{ik}(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)(\bar{\boldsymbol{w}}_i - \boldsymbol{\mu}_k)^T\right)$$

until $p(\bar{\boldsymbol{w}}|\alpha_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K})$ converges

Algorithm 2 Inference

1) Find most probable cluster given observation:

$$p(k|D) \propto p(D|k)p(k)$$

$$k^* = \arg\max_k p(k|D)$$

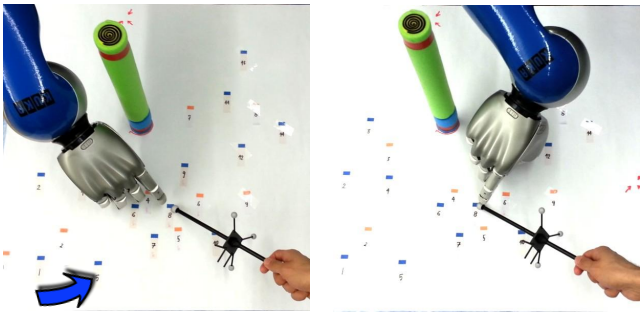
2) Condition on observation, using cluster k^* as prior:

$$p(\mathbf{q}_{1:T}|k^*) = \int p(\mathbf{q}_{1:T}|\bar{\boldsymbol{w}})p(\bar{\boldsymbol{w}}|k^*)\mathrm{d}\bar{\boldsymbol{w}}$$

A. Non-Linear Correlations between the Human and the Robot on a Single Task

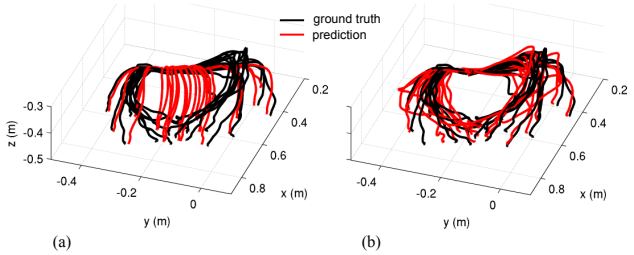
To expose the capability of our method of dealing with multi-modal distributions, we propose a toy problem where a human specifies a position on a table and the robot must point at the same position. The robot is not provided any form of exteroceptive sensors; the only way it is capable to generate the appropriate pointing trajectory is by correlating its movement with the trajectories of the human. As shown in Fig. 2, however, we placed a pole in front of the robot such that the robot can only achieve the position specified by the human by moving either to the right or to the left of the pole. This scenario forces the robot to assume quite different configurations, depending on which side of the pole its arm is moving around.

During demonstrations the robot was moved by kinesthetic teaching to point at the same positions indicated by the human (tracked by motion capture) without touching the



(a) (b)

Fig. 2. Experimental setup of a toy problem used to illustrate the properties of the Mixture of Interaction Primitives. The robot is driven by kinesthetic teaching to point at the positions specified by the human (pointed with the wand). Certain pointed positions can be achieved by either moving the arm to the right (a) or to left (b) of the pole placed on the table. Other positions, such as the one indicated by the arrow, can only be achieved by one interaction pattern.



(a) (b)

Fig. 3. Results of the predictions of the robot trajectories in Cartesian space. Both subplots show the same ground truth trajectories generated by driving the robot in kinesthetic teaching. The predictions are generated by leave-one-out cross-validation on the whole data set comprised of 28 demonstrations. (a) Prediction using the conventional Interaction ProMPs with a single Gaussian. (b) Prediction using the proposed method with a mixture of Gaussians.

pole. For certain positions, as the one indicated by the arrow in Fig. 2(a), only one demonstration was possible. For other positions, both right and left demonstrations could be provided as shown in Fig. 2(a) and 2(b). The demonstrations, totaling 28 pairs of human-robot trajectories, resulted in a multi-modal distribution of right and left trajectory patterns moving around the pole.

In this scenario, modeling the whole distribution over the parameters of the trajectories with one single Gaussian (as in the original Interaction Primitive formulation) is not capable of generalizing the movements of the robot to other positions in a way that resembles the training, as the original framework is limited by assuming a single pattern. This limitation is clearly shown in Fig. 3(a) where several trajectories generated by a single cluster GMM (as in the original Interaction Primitive) cross over the middle of the demonstrated trajectories, which, in fact, represents the mean of the single Gaussian distribution.

Fig. 3(b) shows the predictions using the proposed method with a mixture of Gaussians. By modeling the distribution over the parameters of the trajectories using GMMs as described in section III-C, a much better performance could be achieved. The GMM assumption that the parameters are

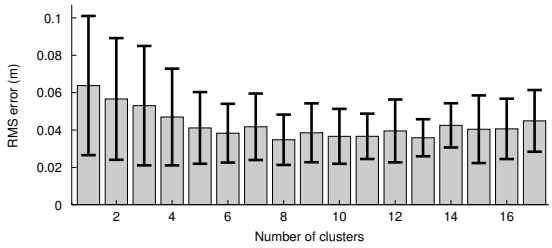


Fig. 4. Root Mean Square Error with models using up to 17 Gaussians.

only locally linear correlated seemed to represent the data much more accurately. As shown in Fig. 4, this improvement is quantified in terms of the Root Mean Square (RMS) Error of the prediction of the trajectory in relation to the ground truth using leave-one-out cross-validation over the whole data set. The same figure also shows that there is a sharp decrease in the RMS error up to six clusters, especially when taking into account the variance among the 28 tests. Beyond seven clusters it is observed that the prediction error fluctuates around 4 cm. The experiments previously shown in Fig. 3(b) were done with eight clusters.

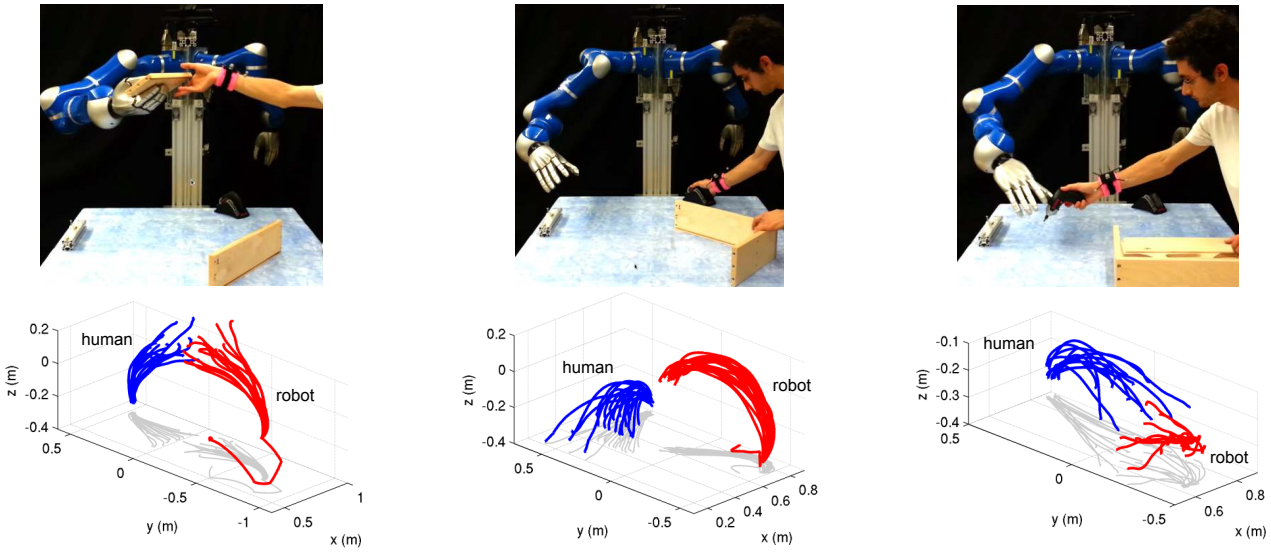
B. Assembling a Box with a Robot Assistant

In this experiment, we recorded a number of demonstrations of different interaction patterns between a human and the robot cooperating to assemble a box. We used the same robot described in the previous experiment. During demonstrations the human wore a bracelet with markers whose trajectories in Cartesian coordinates were recorded by motion capture. Similar to the first scenario, the robot was moved in gravity compensation mode by another human during the training phase and the trajectories of the robot in joint space were recorded.

There are three interaction patterns. Each interaction pattern was demonstrated several times to reveal the variance of the movements. In one of them, the human extends his/her hand to receive a plate. The robot fetches a plate from a stand and gives it to the human. In a second interaction the human fetches the screwdriver and the robot grasps and gives a screw to the human as a pre-emptive collaborator would do. The third type of interaction consists of giving/receiving a screwdriver. Each interaction of plate handover, screw handover and holding the screwdriver was demonstrated 15, 20, and 13 times, respectively. The pairs of trajectories of each interaction are shown in Fig. 5[4].

As described in section III, all training data is fed to the algorithm resulting in 48 human-robot pairs of unlabeled demonstrations as shown in the upper row of Fig. 7. The presented method parameterizes the trajectories and performs

[4]Due to the experimental setup, for the sub-tasks of plate and screw handover we added an initial hand-coded trajectory that runs before the kinesthetic teaching effectively starts. These trajectories are used to make the robot grasp and remove the plate or screw from their respective stands. This is reflected in the figure as the deterministic part at the beginning of the trajectory of the robot. This initial trajectory, however, has no effect on the proposed method itself.



(a) Handing over a plate (b) Handing over a screw (c) Holding the screw driver

Fig. 5. Demonstrations of the three different interactions and their respective trajectories. For the case of plate and screw handover the beginning of the robot trajectory shows a deterministic part that accounts for the fact that the robot has to remove objects from their respective stands, which is not part of the kinesthetic teaching, and which does not affect the algorithm in any sense.

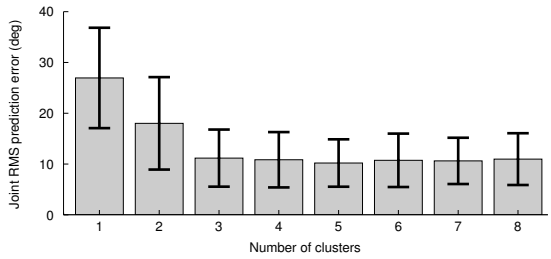


Fig. 6. Root-mean-square error of the joint trajectories (averaged over all tests) using a leave-one-out cross-validation as a function of the number of clusters (mixture components). The plateau after three clusters seem to be consistent with the training data since it consists of three distinct interaction patterns.

clustering in the parameter space in order to encode the mixture of primitives. In the figure, the human is represented by the (x, y, z) Cartesian coordinates while the robot is represented by the seven joints of the arm. Due to lack of space, the figure only shows up to the first four joints (starting from the base).

Figure 6 shows the RMS prediction error averaged over all tests as the number of mixture components increase. The prediction is obtained by leave-one-out cross-validation over the whole set of 48 demonstrations. As one would expect, since the unlabeled data contains three distinct interaction patterns, the improvement is clearly visible up to three mixture components. No significant improvement is obtained afterwards, thus the GMM with three mixture components was selected for experiments.

In the inference/execution phase, the algorithm first computes the most probable Interaction Primitive mixture component based on the observation of the position of the wrist of the human with (12). Using the same observation, we then condition the most probable Interaction Primitive, which allows computing a posterior distribution over trajectories for all seven joints of the robot arm as in (13). Finally, the mean of each joint posterior distribution is fed to a standard inverse dynamics feedback tracking controller.

The lower row of Fig. 7 depicts the posterior distribution for one test example where a three-cluster GMM was trained with the other 47 trajectories. The GMM prior is shown in gray where the patches of different clusters overlap. The observation consists only of the final position of the wrist, shown as asterisks in the figure. The black lines are the ground truth trajectories of each degree of freedom. The posterior, in red, is represented by its mean and by the region inside $\pm$ two standard deviations. The mean of this posterior is the most probable trajectory for each degree of freedom given the observed end position of the wrist of the human.

We assembled the toolbox, consisting of seven parts and 12 screws, two times. The experiments demanded more than 40 executions of the Interaction Primitives. The selection of the right mixture component was 100% correct. (Please refer to the accompanying video).

We evaluated the precision of the interactions by computing the final position of the hand of the robot with forward kinematics. The forward kinematics was fed with the conditioned robot trajectories predicted by leave-one-out cross validation. The interactions of plate handover and holding screwdriver resulted in mean error with two standard deviations (mean error $\pm 2\sigma$) of 3.2 ± 2.6 cm and 2.1 ± 2.3 cm, respectively. We did not evaluate the precision of the handover of the screw, as the position at which

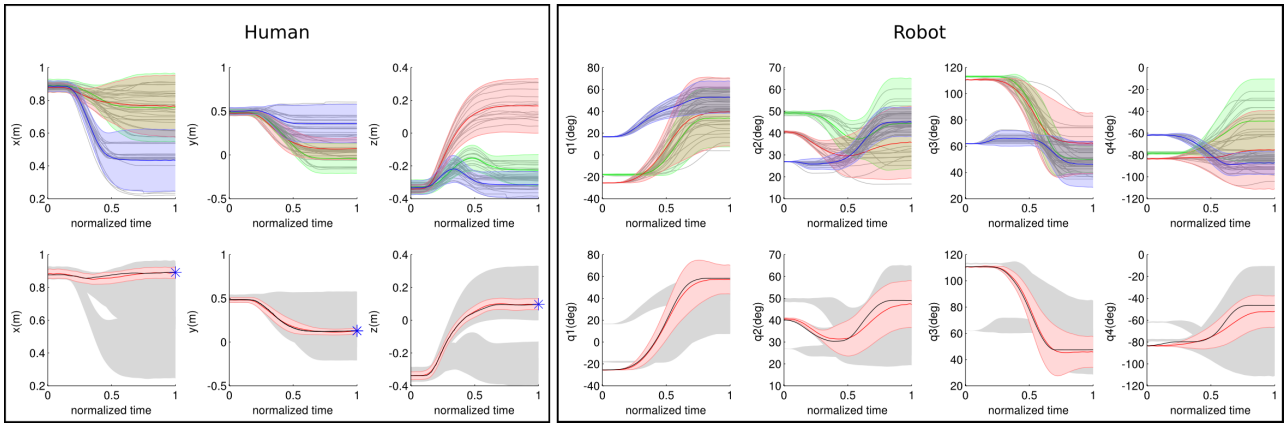


Fig. 7. **Upper row**: Mixture components represented by their mean trajectories and the region inside two standard deviations ($\mu \pm 2\sigma$). Obs.: The plots show only the part of the trajectories generated by kinesthetic teaching. **Lower row**: Posterior probability distribution given observation depicted by the blue asterisks.

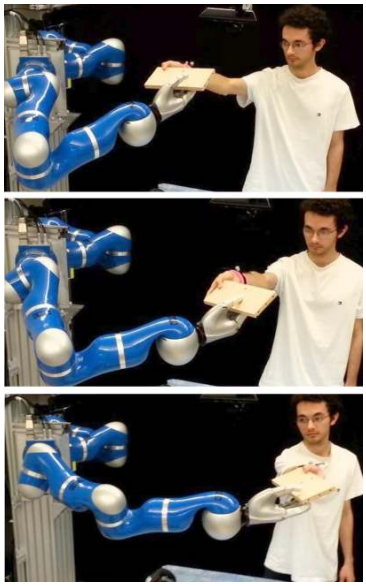


Fig. 8. Handover of a plate. Conditioning on three different positions of the wrist (using motion capture) of a human coworker.

the robot hands the screw is not correlated to the human (please refer to the accompanying video). As an example, Fig. 8 shows the robot executing the plate handover at three different positions based on the location of the wrist marker. Note that the postures of the arm are very different, although they are all captured by the same Interaction Primitive.

V. Conclusions

In this paper we presented a Mixture of Interaction Primitives where Gaussian Mixture Models are used to model multiple interaction patterns from unlabeled data. The multimodal prior probability distribution is obtained over parameterized demonstration trajectories of two agents working in collaboration. During the execution, the algorithm selects the mixture component with the highest probability given the observation of the human, which is then conditioned to infer the appropriate robot reaction. The proposed method is able to learn and recognize multiple human-robot collaboration tasks from an arbitrary number of demonstrations consisting of unlabeled interaction patterns, what is not possible with the current Interaction Primitive framework.

In the context of human-robot interaction we are currently addressing the estimation of the phase of the execution of the primitive for switching tasks in real time. Also, we are addressing the use of the stochastic feedback controller provided by the original ProMP work in [16]. Although this work focused on human-robot trajectories, we are currently considering extensions of our work where the human is replaced by other variables of interest. For example, the same framework can be used to correlate joint and end-effector trajectories of the same robot to learn nonlinear forward/inverse kinematic models. Similarly the Mixture of Interaction Primitives can be used to correlate the interaction between motor commands and joint trajectories to learn inverse dynamic models.

VI. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programmes (FP7-ICT-2013-10) under grant agreement 610878 (3rdHand) and (FP7-ICT-2009-6) under grant agreement 270327 (ComPLACS).

References

[1] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[2] G. Maeda, M. Ewerton, R. Lioutikov, H. Ben Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2014.

[3] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA, 2013.

[4] M. Lawitzky, J. Medina, D. Lee, and S. Hirche, "Feedback motion planning and learning from demonstration in physical robotic assistance: differences and synergies," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 3646–3652.

[5] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1450–1459, 2013.

[6] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, ser. CVPR '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 994–.

[7] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, Aug 2000.

[8] K. Hawkins, N. Vo, S. Bansal, and A. F. Bobic, "Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration," in *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2013.

[9] Y. Tanaka, J. Kinugawa, Y. Sugahara, and K. Kosuge, "Motion planning with worker's trajectory prediction for assembly task partner robot," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 1525–1532.

[10] Z. Wang, K. Mülling, M. P. Deisenroth, H. Ben Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013.

[11] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response." in *Robotics: Science and Systems*, 2013.

[12] D. Lee, C. Ott, Y. Nakamura, and G. Hirzinger, "Physical human robot interaction in imitation learning," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3439–3440.

[13] H. Ben Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung, and J. Peters, "Learning responsive robot behavior by imitation," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 3257–3264.

[14] B. Llorens-Bonilla and H. H. Asada, "A robot on the shoulder: Coordinated human-wearable robot control using coloured petri nets and partial least squares predictions," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation*, 2014.

[15] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

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

[17] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

Generalizing Pouring Actions Between Objects using Warped Parameters

Sascha Brandl, Oliver Kroemer, and Jan Peters

***Abstract*— One of the key challenges for learning manipulation skills is generalizing between different objects. The robot should adapt both its actions and the task constraints to the geometry of the object being manipulated. In this paper, we propose computing geometric parameters of novel objects by warping known objects to match their shape. We refer to the parameters computed in this manner as *warped parameters*, as they are defined as functions of the warped object's point cloud. The warped parameters form the basis of the features for the motor skill learning process, and they are used to generalize between different objects. The proposed method was successfully evaluated on a pouring task both in simulation and on a real robot.**

I. Introduction

In order to perform tasks in everyday environments, robots will need to be capable of manipulating a wide range of different objects. As objects of the same type may have different shapes and sizes, the robot will have to adapt its actions to the geometry of the specific object that it is manipulating. The shape of objects is particularly important when manipulating liquids, e.g., pouring a glass of water, as liquids conform to the shape of their container. The robot must therefore take into consideration a container's geometry when using it in a pouring task.

Although containers come in a wide variety of shapes and sizes, the important differences can usually be defined by a few geometric parameters [1], [2]. For example, the volume of a container indicates how much fluid it can hold, regardless of whether it has a spherical, or cylindrical shape. A robot can generalize pouring actions between different containers by using these geometric parameters. However, the robot will not be provided with the geometric parameters for most of the novel objects that it encounters. While a human may annotate the geometric information for a couple of objects, the robot will usually need to compute these parameters on its own.

In this paper, we investigate using warped parameters to generalize pouring skills between different objects. A warped parameter is defined as a function on the points of a known object's point cloud. For example, a warped parameter may compute the volume of a set of points' convex hull. When the robot encounters a novel object, it warps the point cloud of the known object to the new object's shape. As a result of the warping, the value of the warped parameter changes to match the geometry of the new object. Once the geometric parameters have been computed, the robot can use them to generalize actions and task constraints between different objects.

All authors are members of the IAS group at the TU Darmstadt, Germany. Jan Peters is also a member of the MPI for Intelligent Systems, Germany. {brandl, kroemer, peters}@ias.tu-darmstadt.de

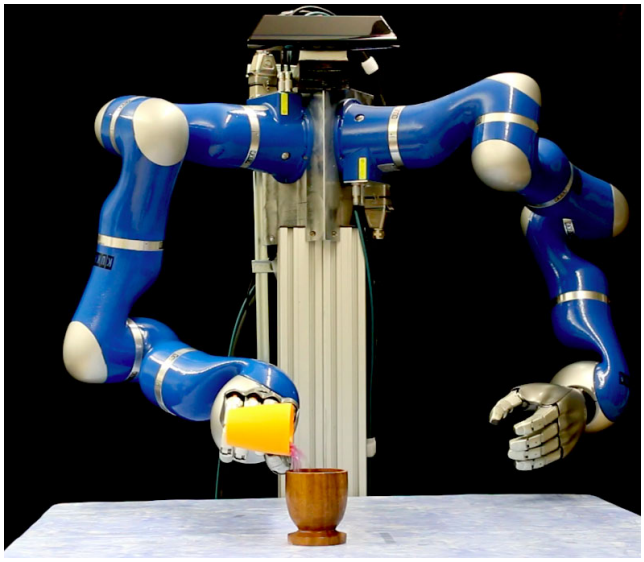


Fig. 1. The robot performs a pouring task with two previously unknown objects. The pouring action was learned from human demonstrations using a taller cup and a wider container to pour into.

In Section II, we explain the process of computing the warped parameters. In Section III, we describe how the robot can learn pouring actions and task constraints that generalize between objects using the warped parameters. The proposed method was successfully evaluated both in simulation and on the robot shown in Fig. 1. The results of the experiments are detailed in Section IV.

Related Work

Several previous works have used warping to generalize manipulations between objects. Hillenbrand et al. [3], [4] used warping to map contact points onto novel objects, in order to transfer grasps between objects. A similar approach was used by Rainer et. al [5], [6] for transferring coordinate frames of task constraints between objects. However, the size and shape of the constraint regions were not adapted to the new object's geometry. Rather than warping only the points on the object, Schulman et al. [7] computed warping functions for the entire scene. The warping was then applied to the demonstrated trajectory of the source scene in order to obtain a trajectory for the current scene. These approaches focus on mapping specific points from the source scene to the target scene, and are therefore especially well-suited for contact-based manipulations. Warped parameters can be used

to model more general features of the objects, such as areas and volumes.

Several methods have also been proposed for learning to perform pouring tasks. Pastor et al. [8] learned dynamic motor primitives (DMPs) for pouring from human demonstrations, and used these to generalize to different cup placements. Similarly, Muehlig et al. [9] encoded demonstrated bimanual pouring trajectories using Gaussian mixture models. Rozo et al. [10] proposed learning a controller for pouring tasks based on the observed forces. The work on learning pouring from demonstration has mainly focused on learning with the same set of objects. In comparison, we propose learning in a feature space defined by the warped parameters, in order to automatically generalize between objects.

Some work has also been done on generalizing pouring actions between different objects using reinforcement learning. Kroemer et al. [11] learned a pouring DMP from human demonstrations, and then used a trial-and-error approach to learn the location of a novel container's opening. The opening was detected using a shape-similarity kernel. Tamosiunaite et al. [12] used reinforcement learning to learn the shape of the pouring DMP, as well as the goal point. Reinforcement learning was also used to adapt the learned motion to novel objects, without explicitly considering the differences in geometry.

II. Generalization with Warped Parameters

In this section, we describe how a robot can compute geometric parameters of an object by warping a known object to match its shape. The object models and the warping process used in this paper are described in Sections II-A to II-C. The computation of the warped parameters for pouring tasks is described in Section II-D.

A. Geometric Object Models

In order to generalize manipulations to a novel object, the robot first computes correspondences between a known source object O_s and the unknown target object O_t. An object O_i is modeled as a set of c_i points located at positions $\boldsymbol{p}_{ij} \in \mathbb{R}^3$ with corresponding normals $\boldsymbol{n}_{ij} \in \mathbb{R}^3$, where $j \in \{1, ..., c_i\}$.

Objects often consist of multiple parts, and a manipulation may only depend on the shape of a part of an object. Hence, geometric parameters often describe the shape of a part rather than the whole object. We therefore also assign each point $\boldsymbol{p}_{ij}$ a vector $\boldsymbol{l}_{ij}$ of length ρ with binary labels, which indicate which of the ρ object parts the point corresponds to. The labels of the target object O_t are initially unknown, but can be computed using the warping process.

An example of an annotated cup can be seen in Fig. 2. The first part is the CONTAINER, which holds the liquids. The second part is the RIM around the opening. We also label the HANDLE as a *dummy* part. As not all containers have handles, it is not used to define any warped parameters for the pouring task, and is only included to help align objects during the warping process.

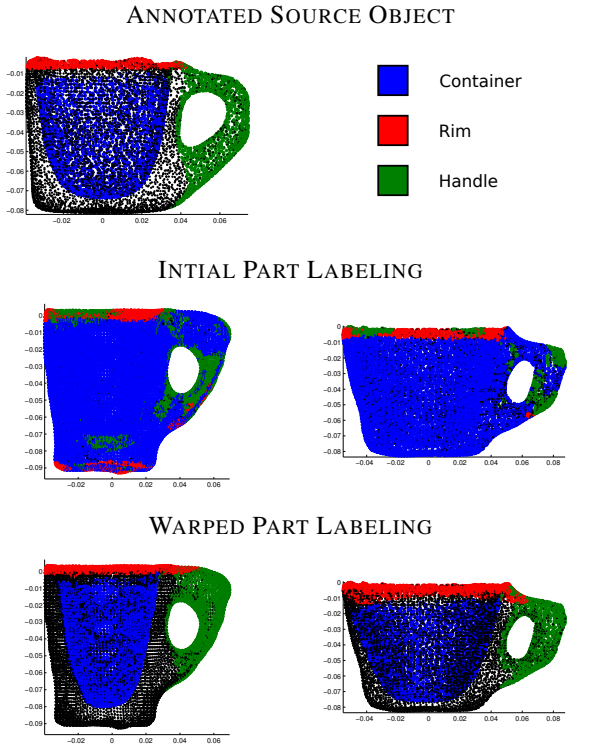


Fig. 2. The *top* row shows the point cloud of the source object, annotated by a human user. The *middle* row shows the point clouds of two target objects. The points were labelled using a classifier based on local features. This intial estimate is only used to compute a coarse alignment with the source object. The point clouds were pre-aligned for this figure to show more clearly how the labels change during the warping process. The *bottom* row shows the final results of the label mapping approach.

B. Warping

Given a source object and a target object, the robot can compute correspondences between the two objects. These correspondences are determined by warping the shape of the source object onto that of the target object. There are various methods for computing 3D warpings between object [13], [14], and the proposed approach does not depend on a specific warping algorithm. We therefore employ a basic warping algorithm for finding correspondences between the containers. The warping process consists of two stages: 1) object alignment, and 2) point mapping

In the first stage, the source object is coarsely aligned with the target object, such that their corresponding object parts are close together. This alignment is accomplished by computing a coordinate system based on the objects' parts. The origin of the coordinate frame is the mean of the container points. The first axis is given by the direction to the mean of the rim points, and the second axis is the orthogonal direction to the mean of the handle points. The third axis is computed by the cross product of the first two axes. As the part labels of the target object $\boldsymbol{l}_t$ are unknown, an initial estimate of the labels is computed using logistic regression. One classifier is trained for each of the three object parts. Each point $\boldsymbol{p}_{ti}$ is classified based on the local distribution

of points in its neighborhood. The features used to describe the local distribution of points include the eigenvalues of the covariance matrix, and the distance from the point $\boldsymbol{p}_{ti}$ to the mean of the neighborhood points. The classifiers were trained on the labelled points of the source object. An example of the initial labeling can be seen in Fig. 2. The coordinate frame of the object is estimated using this initial labeling of points. Once the two objects are aligned, the source object was scaled in each direction such that the variances of its container part matched those of the target object. We denote the aligned source objects and target objects by $\tilde{O}_s$ and $\tilde{O}_t$ respectively.

In the second stage of the warping algorithm, the points from the source object $\tilde{O}_s$ are mapped onto the target object $\tilde{O}_t$. This step is similar to the approach proposed by Hillenbrand [15]. Each point of the aligned source object is mapped to the mean of the k nearest neighbors in the aligned target object. In our experiments, we set $k = 1$. Hence, the warped source point $\boldsymbol{p}_{wi}$, with corresponding normal $\boldsymbol{n}_{wi}$ and labels $\boldsymbol{l}_{wi}$, is given by

$$\boldsymbol{p}_{wi} = \boldsymbol{p}_{tj} \text{ , } \boldsymbol{n}_{wi} = \boldsymbol{n}_{tj} \text{ , and } \boldsymbol{l}_{wi} = \boldsymbol{l}_{si},$$
$$\text{s.t. } j = \arg\min \left\| \tilde{\boldsymbol{p}}_{si} - \tilde{\boldsymbol{p}}_{tj} \right\| \text{ and } \tilde{\boldsymbol{n}}_{si}^T \tilde{\boldsymbol{n}}_{tj} > 0.$$

Thus, each source point is mapped to the closest target point with a normal pointing in the same direction. The warped object and its point cloud are denoted by O_w.

C. Point Mapping vs. Label Mapping

The warping process defines a new position and normal for each of the c_s point of the source object O_s. The location of these new points can be used to define warped parameters, as detailed in the next section. We refer to this approach as *point mapping*, as the points of the source object are mapped onto the target object.

However, if the source object has considerably fewer points than the target object, then some details of the target object may not be captured by the warped object. This issue can be addressed by warping the target object to match the source object. The alignment and scaling of the objects is performed as before. However, the points of the target object are mapped onto the source object. The label of each of the target points is then determined using a k-nearest neighbors classifier. In our experiments, we again used $k = 1$, such that

$$\boldsymbol{p}_{wi} = \boldsymbol{p}_{ti} \text{ , } \boldsymbol{n}_{wi} = \boldsymbol{n}_{ti} \text{ , and } \boldsymbol{l}_{wi} = \boldsymbol{l}_{sj},$$
$$\text{s.t. } j = \arg\min \left\| \tilde{\boldsymbol{p}}_{si} - \tilde{\boldsymbol{p}}_{tj} \right\| \text{ and } \tilde{\boldsymbol{n}}_{si}^T \tilde{\boldsymbol{n}}_{tj} > 0.$$

We refer to this approach as label mapping, as the labels of the source object are mapped onto the target object. When using multiple neighbors $k > 1$, the point is assigned to a part if the majority of its k neighbors belong to that part.

The benefit of using the label mapping approach is that it guarantees that all of the points of the target object are used for computing the warped parameters. However, when using label mapping, points can only be referred to by their label and not as individual points. In comparison, when using point mapping, one can refer to individual points, e.g., $\boldsymbol{p}_{w72}$, which correspond to specific points on the source object. The bottom row of Fig. 2 shows an example of using label mapping.

D. Warped Parameters

Having computed the correspondences between the known source object and the novel target object, the robot can compute the warped parameters for the target object. A warped parameter is defined as a function on the warped point cloud $f(O_w)$. Warped parameters can be used to define geometric reference parameters, such as lengths, areas, and volumes, of an object's part. Warped parameters can also be used to define task frames.

For pouring, the task frame is defined by the lip point of the first container, and the center of the second container's opening. The center of the opening is defined as the mean of the rim points. The lip point is defined as the rim point that is the closest to the other container. A pouring motion is defined by the trajectory of the held container's lip point relative to the center of the second container's opening. The trajectory includes the relative 3D position and the tilt of the first container about its lip point. The other two rotation dimensions are usually assumed to be zero. If there is no second container, the lowest rim point is defined as the lip point.

The geometric reference parameters for pouring include the radius of the opening, the volume of the container, the height of the container, and a reference angle for tilting the cup. The radius of the opening is given by the mean distance between the rim points and the center of the opening. The volume of the container is given by the volume of the container points' convex hull. The height of the container is given by the range of all of the points along the first dimension. A tilt reference angle is defined by the amount that the cup must be rotated about the lip point, such that half of the container's volume is above the lip point. As the warping process reshapes the points of the source object, the estimates of the reference parameters will change accordingly. In this manner, the warped parameter function defines how the parameter's value is grounded in the object's geometry.

As the above examples show, warped parameters can be used to define various object properties, and can even build on each other. These parameters can then be automatically computed for new objects using the warping process.

III. Learning with Warped Parameters

In this section, we describe how a robot can learn pouring actions and task constraints that generalize to new objects using the warped parameters.

A. Learning Task Constraints

When performing a pouring task, the liquid should remain in the cup while it is being transported, and it should only be poured out if it will be transferred to another container. These task constraints correspond to phase transitions [16] and can

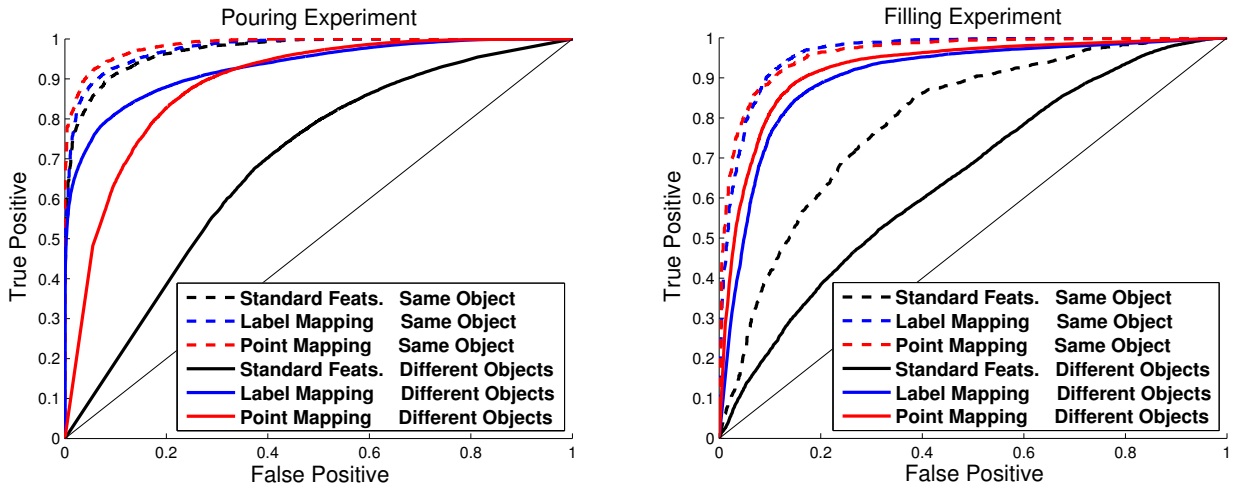


Fig. 3. The figure shows the ROC curves for the learned classifiers for both the pouring experiment and the filling experiment. The *dashed* lines indicate the performance when the classifier is applied to data from the same object that was used for training the classifier. The *solid* lines indicate the performance when the classifiers are applied to novel objects, for which they had no training data. A classifier is generally considered to perform better if it gets closer to the top left corner. Classifiers were trained using features based on the warped parameters computed using both the label mappings and point mappings approaches. The standard features approach did not use the reference values given by the warped parameters.

be fulfilled by learning to predict when the held container will start to pour and when the poured liquid will fill the second container. The conditions for pouring and filling are learned by training a classifier for each condition. The classification is performed using logistic regression, which is a form of probabilistic classifier. The probability of pouring $y_p = 1$ from the first container is given by

$$p(y_p = 1|\boldsymbol{x}_u) = (1 + \exp(-\boldsymbol{\omega}^T \boldsymbol{\varphi}(\boldsymbol{x})))^{-1}$$

where $\boldsymbol{\varphi}(\boldsymbol{x})$ is a vector of features describing the state of the container $\boldsymbol{x}$, and the weight vector $\boldsymbol{\omega}$ is computed from training data using iterative reweighted least squares. The features $\boldsymbol{\varphi}(\boldsymbol{x})$ are of the form α/α_r, where α is a variable and α_r is a reference value defined by a warped parameter. For predicting pouring, the features include the tilt angle of the cup divided by the tilt reference angle, and the fluid volume divided by the volume of the container. The resulting features are dimensionless quantities that automatically adapt to the geometry of the container.

For predicting when the poured liquid increases the fluid volume in the second container $y_f = 1$, we expand the set of features to include both objects and their relative positions. The vertical distance between the containers is divided by the height of the first container. The horizontal distances between the containers are divided by the radius of the second container. These features allow the robot to learn when the poured liquid will miss the second container, as well as predict when the container will overflow.

B. Learning Motor Primitives in Warped Spaces

The proposed warping approach can also be used to learn motor primitives that adapt to the shape of the objects being manipulated. Motor primitives are often used to define desired trajectories that can be easily adapted to different situations. In order to model distributions of trajectories, we use the probabilistic motor primitives (ProMPs) [17]. These motor primitives encode correlations between the different dimensions of the trajectory, and can be conditioned on the initial state of the objects.

The learned motor primitive defines a desired trajectory in the task space described in Section II-D. Similar to the features used to generalize task constraints, the trajectories are defined as dimensionless quantities. The vertical distance between the objects is divided by the height of the held container, and the tilt angle is divided by the reference tilt angle. The horizontal distances are divided by the radius of the second container.

The motor primitives are learned by scaling the demonstrated trajectories according to the warped parameters of the objects used in the demonstrations. In order to execute a pouring action, the robot samples a trajectory from the ProMP, and rescales it according to the current objects' warped parameters.

IV. Experiments

The proposed method was implemented and evaluated both in simulation and on a real robot. The robot, shown in Fig. 1, consists of two Kuka light weight robot arms, each equipped with a five-fingered DLR hand [18]. The robot observes the table-top scene from above using a Microsoft Kinect camera. Ten different cups and bowls were scanned from multiple views. 3D mesh models were generated using an implicit surface representation and marching cubes [19].

A. Simulated Pouring and Filling Experiments

In the first experiment, we evaluated how well task constraints generalize between objects when using warped parameters. The objects were simulated using the Bullet physics engine [20] together with Fluids 2 for incorporating smoothed particle hydrodynamics [21].

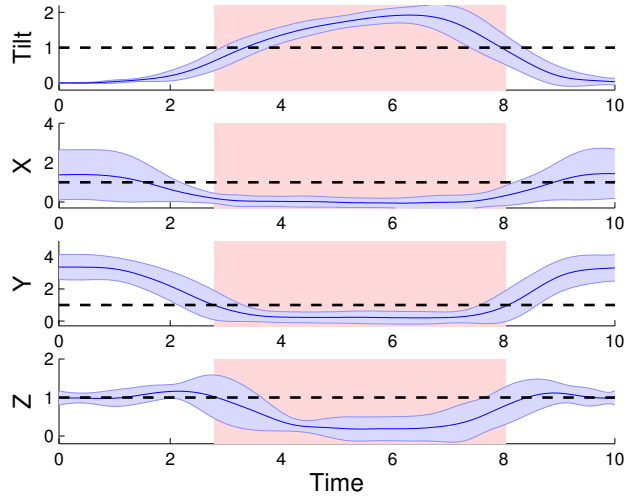


Fig. 4. The plots show the distribution over trajectories learned by the ProMPs in the generalized space. The blue line indicates the mean trajectory, and the shaded regions correspond to +/- two standard deviations. The black horizontal lines indicate when the value is one. The tilt is one when the cup is tilted such that half of the container's volume is above the lip points. The X and Y values are one when the lip point is one radius away from the second container's center. The Z value is one when the vertical distance between the cup and the container is the same as the height of the cup. The red region indicates when the X-Y position of the cup's lip point is within one radius of the container's center.

Each object was filled 1000 times with a random amount of liquid, and tilted by a random angle around the lip point. If the volume of the fluid in the cup decreased, the trial was labelled as pouring $y_p = 1$. Otherwise it was labelled as not pouring $y_p = 0$. The classifiers were trained on sets of 50 samples. The classifiers were tested on two test sets: the 950 other samples from the same object, and the 9000 samples from the other objects. The latter dataset is used to test how well the classifiers generalize between different objects.

A similar procedure was used for the filling experiment. However, the cup used for pouring always contained 10 particles at the start of the trial, and the second container was filled by a random amount. The cup was always tilted by $120°$. The relative positions of the cups were varied between trials. A trial was considered as successful $y_f = 1$ iff none of the particles ended up outside of the second container.

For each training set, three classifiers were computed. The first two classifiers were trained using the warped parameters from the point mapping and the label mapping approaches respectively. The features used for training the classifiers were described in Section III-A. As a benchmark, we also evaluated the classifiers without using the warped parameters. In this case, all of the reference values α_r were set to one, regardless of the objects being manipulated, and the relative positions of the objects were defined by their centers.

The results of the pouring and filling experiments can be seen in Fig. 3. As one would expect, the classifiers generally achieved similar levels of performance when evaluated on the training object. The standard features performed considerably worse in the filling experiment, as different cups were used for pouring even though the second container remained the same. The ROC curves show that the performance of all three classifiers decreases when generalizing to novel objects. However, the drop in performance is considerably less when using the warped parameters. The features based on the warped parameters are therefore better at separating the positive and negative examples across different objects. While the two warping methods performed similarly well on the filling experiment, the label mapping approach performed better in the pouring experiment, detecting more than 50% of the true positives with almost no false positives. The results show that the warping parameters can be used to reliably generalize the constraints of the pouring task between different containers.

B. Robot Pouring Experiment

In the second experiment, the robot used warped parameters to generalize pouring actions between different objects. The robot was provided with ten demonstrations of a pouring task using kinaesthetic teaching. All of the demonstrations were performed with the same two objects shown in the left picture of Fig. 5. For safety reasons, the task was performed with gel balls rather than an actual liquid. The cup was half full at the start of each trial. Using the ten demonstrations, the robot learned a ProMP for pouring, as described in Section III-B. The learned distribution over trajectories is shown in Fig. 4. The robot was then given the task of pouring with different objects. The robot successfully learned to pour from a shorter cup into a bigger bowl, a smaller cup, and a square bowl, as shown in Fig. 5. Only a couple of gel balls were spilled during the experiments. A video of the robot performing the pouring task is also provided in the supplementary material.

As the cups were half-full, pouring usually commenced when the tilt value went above one. Fig. 4 shows that the distribution over trajectories remains safely below this value until the lip point is above the opening. When moving the cup back, most of the liquid has been poured out, and hence the cup can be tilted more. The pictures in Fig. 5 show that the cup was often placed close to the rim of the second container, which indicates that the robot was able to adapt the learned trajectory to the geometry of the object being manipulated.

C. Future Work

The autonomy of the robot can be increased by learning warped parameters. The points could be labelled by using an unsupervised approach to segmenting the objects into parts. A set of generic geometric functions could then be applied to each part in order to generate a library of warped parameters. Feature selection methods could then be applied to select a suitable set of warped parameters for the given task.

The focus in this paper was on learning pouring skills from a single object. The generalization between objects therefore relies on using the warped parameters to construct dimensionless features for the robot. However, given data

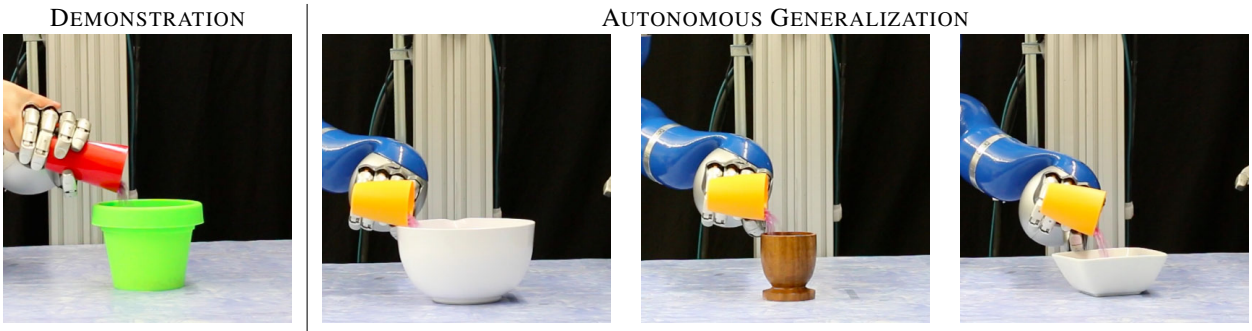


Fig. 5. The pictures show the key results of the real robot experiment. The robot was provided with multiple demonstrations of the pouring task using kinaesthetic teaching, as shown on the left. Using the warped parameters approach, the robot successfully generalized the demonstrated actions to novel objects with different shapes and sizes, as shown on the right.

from multiple objects, the robot can also learn how to generalize between objects. In this case, the warped parameters could be treated as separate features that describe the object. For example, ProMPs can be used to learn the correlations between the trajectory parameters and the warped object parameters. Object-specific trajectories can be obtained by conditioning on the current object parameters. This approach would even allow the robot to learn that only some segments of the trajectory depend on the object parameters. However, learning ProMPs in this manner would require additional training trajectories with different objects. These trajectories could be obtained from human demonstrations, or by adapting trajectories using reinforcement learning [12].

V. CONCLUSION

We proposed using warped parameters to generalize pouring skills between different objects. Warped parameters are functions defined on the point cloud of a known object. The parameter can be computed for a novel object by warping the known object's point cloud to match the geometry of the novel object.

The proposed method was successfully evaluated both in simulation and on a real robot pouring task. The experiments showed that the warped parameters can be used to generalize task constraints and motor primitives between containers of different shapes and sizes

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreements 248273 (GeRT) and 610878 (3rdHand). The authors would also like to thank Ullrich Hillenbrand for the insightful discussions.

REFERENCES

[1] G. Bartels, I. Kresse, and M. Beetz, "Constraint-based movement representation grounded in geometric features," in *International Conference on Humanoid Robots*, (Atlanta, Georgia, USA), 2013.

[2] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz, "Decomposing cad models of objects of daily use and reasoning about their functional parts," in *International Conference on Intelligent Robots and Systems*, pp. 5943–5949, 2013.

[3] U. Hillenbrand and M. Roa, "Transferring functional grasps through contact warping and local replanning," in *International Conference on Intelligent Robots and Systems*, pp. 2963–2970, Oct 2012.

[4] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, "Generalization of human grasping for multi-fingered robot hands," in *International Conference on Intelligent Robots and Systems*, 2012.

[5] R. Jäkel, *Learning of Generalized Manipulation Strategies in Service Robotics*. PhD thesis, Institut für Anthropomatik, Karlsruhe, 2013.

[6] R. Jäkel, S. Schmidt-Rohr, S. Rühl, A. Kasper, Z. Xue, and R. Dillmann, "Learning of planning models for dexterous manipulation based on human demonstrations," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 437–448, 2012.

[7] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *International Symposium on Robotics Research*, 2013.

[8] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *International Conference on Robotics and Automation*, 2009.

[9] M. Muehlig, M. Gienger, S. Hellbach, J. J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *International Conference on Robotics and Automation*, (Kobe, Japan), pp. 1177–1184, IEEE, 05/2009 2009.

[10] L. Rozo, P. Jimenez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in *Robot Motion and Control*, pp. 227–232, July 2013.

[11] O. Kroemer, E. Ugur, E. Oztop, and J. Peters, "A kernel-based approach to direct action perception," in *International Conference on Robotics and Automation*, 2012.

[12] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter, "Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives," *Robotics and Autonomous Systems*, 2011.

[13] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder, "Multiresolution mesh morphing," in *Annual Conference on Computer Graphics and Interactive Techniques*, (New York, NY, USA), pp. 343–350, 1999.

[14] F. Steinke, B. Schölkopf, and V. Blanz, "Learning dense 3d correspondence.," in *Advances in Neural Information Processing Systems*, pp. 1313–1320, MIT Press, 2006.

[15] U. Hillenbrand, "Non-parametric 3d shape warping," in *International Conference on Pattern Recognition*, pp. 2656–2659, IEEE, 2010.

[16] O. Kroemer, H. van Hoof, G. Neumann, and J. Peters, "Learning to predict phases of manipulation tasks as hidden states," in *International Conference on Robotics and Automation*, 2014.

[17] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, 2013.

[18] Z. Chen, N. Y. Lii, T. Wimboeck, S. Fan, M. Jin, C. Borst, and H. Liu, "Experimental study on impedance control for the five-finger dexterous robot hand dlr-hit ii.," in *International Conference on Intelligent Robots and Systems*, pp. 5867–5874, IEEE, 2010.

[19] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *International Conference on Robotics and Automation*, 2011.

[20] "Bullet physics library," October 2013.

[21] "Bullet-fluids project," October 2013.