2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Augmenting Action Model Learning by Non-Geometric Features

Iman Nematollahi Daniel Kuhner Tim Welschehold Wolfram Burgard

Abstract— **Learning from demonstration is a powerful tool for teaching manipulation actions to a robot. It is, however, an unsolved problem how to consider knowledge about the world and action-induced reactions such as forces imposed onto the gripper or measured liquid levels during pouring without explicit and case dependent programming. In this paper, we present a novel approach to include such knowledge directly in form of measured features. To this end, we use action demonstrations together with external features to learn a motion encoded by a dynamic system in a Gaussian Mixture Model (GMM) representation. Accordingly, during action imitation, the system is able to couple the geometric trajectory of the motion to measured features in the scene. We demonstrate the feasibility of our approach with a broad range of external features in real-world robot experiments including a drinking, a handover and a pouring task.**

I. Introduction

In the past, service robotics has grown in importance and received rising interest. Several service robotic systems are commercially available as robots that mow the lawn or vacuum the floor. However, applications involving manipulation tasks as organizing shelves or interactions with the environment rely on feedback as handing over objects to a human are still tough to accomplish. Mostly such systems depend on predefined motion trajectories and require expert knowledge. On the other hand, learning from demonstration alleviates the robot teaching process allowing even non-expert users to extend the robot's usability by teaching new skills. Common techniques rely on learning a geometric representation of the action which determines a trajectory. They are able to adapt to other geometric setups and dynamically react to external perturbations on the trajectory course. Many real-life tasks however are not only described by geometric constraints but also depend on additional properties as contact forces or other measurable parameters like the liquid height in a container while pouring. So far such reacting behavior is either triggered externally or regulated by time.

In this paper, we present a novel approach that learns actions from demonstration considering both the geometric information about the motion as well as its interdependence to other, non-geometric features in the scene. Those can be direct results of the motion, like when pouring a liquid, or caused by independent sources as measured contact forces. Both cases exert influences on the motion. Thus, we aim at coupling the knowledge about these influences directly into the action model by including the measured feature as an additional dimension.

All authors are with the Department of Computer Science, University of Freiburg, Germany. {nematoli, kuhnerd, twelsche, burgard}@cs.uni-freiburg.de

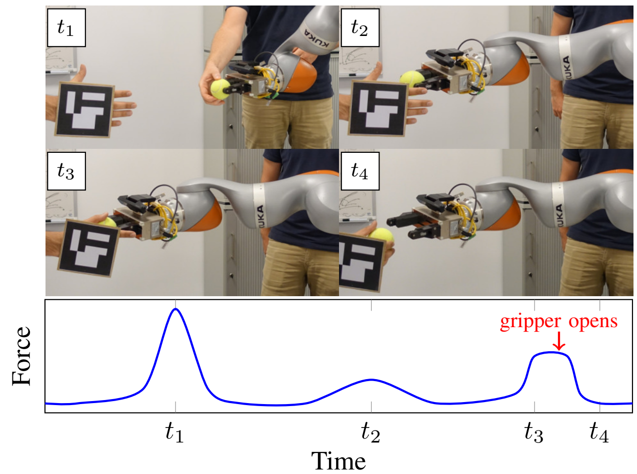

Fig. 1: Our approach uses Gaussian Mixture Models to create a motion model for a manipulator. Besides the common geometric description of demonstrations, the model takes external parameters as contact forces or liquid levels into account. The pictures depict how force as an external parameter can influence the motion generation. Our model is able to deliver the object properly, by learning the correlation between force measurements and geometric course of the demonstrations.

Fig. 1 shows an example application. Handing over an object requires to react when the collaborator successfully grasped the object. For such manipulation tasks, many approaches create a motion model that only considers the trajectory of the object at hand. While these models can create trajectories that closely imitate the training data, the safe and smooth interaction with the human can constitute a problem. Obviously, the opening of the robot gripper should not only depend on its relative position to the human hand but rather be influenced by the measured contact force. Moreover, this force should only cause the gripper to be opened when the object is also close to the human hand. Therefore, considering a threshold on the measured forces is not sufficient. By incorporating external parameters into the model one can overcome these issues. As can be seen in Fig. 1, our model is able to cope with undesired forces acting on the robot (t_1) and forces caused by the sudden accelerations (t_2). After reaching the human hand the robot does not open its gripper before the human has actually grasped the object (t_3). Other examples include drinking and pouring tasks. In the first one, the robot should not start tilting the cup as long as no contact between the human lips and the rim of the cup is established. In pouring tasks the robot has to consider the liquid height in the cup.

In this paper the following contributions are made: we couple the geometric course of a motion described in a dynamical system with other quantifiable entities like measured forces or the height of a liquid in a cup. This correlation is learned through a small number of action demonstrations and can in principle be applied to any action that should react to arbitrary, measurable external features. We demonstrate the feasibility of our approach for different applications in real-world robot experiments including a drinking task, a handover task and a pouring task.

II. RELATED WORK

In the context of learning from demonstrations the two major question addressed in the research community over the last years focus on *how* and *what* to imitate. The question of what to imitate refers to learning relevant context or frames of reference for generalizing demonstrated actions. This paper addresses the *how* to imitate question and extends it into the direction of *why* and *when*. We explore how externally perceived information that should influence the trajectory's course can be incorporated into the modeling of actions. Recent work dealing with imitation learning on trajectory level describes the motion using dynamic systems [1]. The underlaying representations vary from movement primitives as described in [2] to probabilistic approaches as presented by Calinon *et al.* [3], [4]. In [3] the authors propose a formulation regarding the motion mechanism as driven by a virtual spring-damper system. The dynamical system's parameters are then learned as a mixture of Gaussians. This approach can be implemented as an autonomous system, i. e., independent of time, or as directly time dependent. Khansari-Zadeh *et al.* [5] further extend the probabilistic representation with a mixture of Gaussians by proposing a learning approach that ensures asymptotic stability of the generated motions. Recent work by Welschehold *et al.* [6], [7], [8] use similar dynamical systems to learn a variety of mobile manipulation actions directly from human demonstrations. Mühlig *et al.* [9] presented a system that learns manipulation actions using an articulated model of the human body. Although [6], [9] interact with the world, the actions they imitate are not critically dependent on reaction of the environment as it is for instance in a pouring task. All of these approaches focus on imitating and generalizing seen demonstrations from a purely geometric viewpoint. In our work we also rely on autonomous dynamical system using a mixture of Gaussians to learn the demonstrated behavior. In contrast to the described approach above however, we include additional non-geometric features that are not controlled by the system.

In [10] Kim *et al.* present an approach that allows catching of object in flight using the dynamical system representation from [5]. The approach reacts online to pose measurements of the flying object. In contrast to our work the measurement of the object motion is not directly part of the learned model but triggers an adaption of the system to the best predicted catching pose. In [11] Do *et al.* address the task of pouring water into a cup, which is also one of our example applications. While we use their pipeline to measure the liquid height, their implementation of the pouring action relies on a manually designed PID controller. They recently extended their work by learning the pouring trajectory from simulation using an approach based on deep reinforcement learning and applied it in real-world experiments [12]. In the context of our drinking experiment, Burget *et al.* have proposed a framework in which an autonomous robotic drinking assistant detects and localizes the user's mouth and then generates a motion to bring a cup to the mouth [13]. In contrast to our approach, their system cannot handle dynamic movements of the user's head.

III. APPROACH

For acquisition of new robot skills, our approach learns a dynamic action model which adopts the geometric representation of the intended motion in response to external features from human demonstrations. Hence, our proposed framework consists of three steps: (1) data collection, which gathers demonstration data as trajectories recorded as motion of humans, robots or objects as well as their corresponding effects on the scene or other parameters that exert influence on the motion, (2) motion model learning, which allows to learn robot actions from this set of demonstrations constrained on the perceived parameters, and (3) an online execution system which ensures reactive behavior of the system in presence of disturbances. In general, the demonstrations can contain parameters obtained by arbitrary sensors as liquid heights determined by a vision-based system or end-effector contact forces measured by the robot.

A. Problem Statement

Given N demonstrations of an action we want to learn action models that capture their geometric route and link it to other non-geometric features. A demonstration $\boldsymbol{X} = \langle(\boldsymbol{\xi}_1, \boldsymbol{\rho}_1), \ldots, (\boldsymbol{\xi}_n, \boldsymbol{\rho}_n)\rangle$ consists of n vectors $(\boldsymbol{\xi}_i, \boldsymbol{\rho}_i) \in \mathbb{R}^d$, where $\boldsymbol{\xi}_i$ is a geometric pose and $\boldsymbol{\rho}_i$ a feature describing additional information. Thus, a typical task is described by a 6-DOF (Degree-of-Freedom) trajectory and associated features as contact forces or liquid level heights. The features can not be controlled directly by the robot but are a reaction of the environment to the geometric trajectory or a result of an external source. When imitating the action, the system has to adopt the geometric execution in response to the perceived external features.

B. Learning a Dynamical System with Gaussian Mixture Models

Khansari-Zadeh *et al.* [5] use autonomous dynamical systems to explain encoding of point-to-point motions. An autonomous dynamical system employs a first order ordinary differential equation to map the state of the system to its first derivative. In other words, if $\boldsymbol{\xi}$ defines the state of the system, then

$$\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}) + \boldsymbol{\epsilon}, \tag{1}$$

where f is a nonlinear steady and continuously differentiable function and $\boldsymbol{\epsilon}$ is a zero-mean additive Gaussian noise. Note that Eq. (1) is autonomous since the derivative only depends on the state $\boldsymbol{\xi}$ and not on time explicitly. The noise-free estimate of f can be learned by a mixture of Gaussians, i. e., as a linear superposition of multivariate Gaussian distributions which aims to capture a model for the density of the nonlinear data at hand. This provides an estimate of the variation and correlation across all variables of the data. A GMM with K Gaussian components represents the joint probability of the pose $\boldsymbol{\xi}$ and the corresponding velocity $\dot{\boldsymbol{\xi}}$ by

$$\mathcal{P}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \sum_{k=1}^{K} \mathcal{P}(k)\mathcal{P}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}} \mid k). \tag{2}$$

$\mathcal{P}(k)$ is a prior and $\mathcal{P}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}} | k)$ is the joint probability density corresponding to the k-th Gaussian. Hence, the estimate of f is parametrized by $\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, where π_k is the prior, $\boldsymbol{\mu}_k$ the mean and $\boldsymbol{\Sigma}_k$ the covariance of the k-th Gaussian function. The means and covariances are defined by:

$$\boldsymbol{\mu}^k = \begin{pmatrix} \boldsymbol{\mu}^k_{\boldsymbol{\xi}} \\ \boldsymbol{\mu}^k_{\dot{\boldsymbol{\xi}}} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}^k = \begin{pmatrix} \boldsymbol{\Sigma}^k_{\boldsymbol{\xi}} & \boldsymbol{\Sigma}^k_{\boldsymbol{\xi}\dot{\boldsymbol{\xi}}} \\ \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}} & \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}} \end{pmatrix}. \tag{3}$$

These parameters can be estimated iteratively with different optimization techniques such as Expectation-Maximization (EM) algorithms.

By using this estimated joint probability density function we can retrieve $\dot{\boldsymbol{\xi}}$ given $\boldsymbol{\xi}$ as the conditional distribution $\mathcal{P}(\dot{\boldsymbol{\xi}} \mid \boldsymbol{\xi})$ through Gaussian mixture regression [14]:

$$\hat{\boldsymbol{\mu}}_{\dot{\boldsymbol{\xi}}} = \sum_{k=1}^{K} h_k(\boldsymbol{\xi}) \left(\boldsymbol{\mu}^k_{\dot{\boldsymbol{\xi}}} + \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}} \left(\boldsymbol{\Sigma}^k_{\boldsymbol{\xi}} \right)^{-1} (\boldsymbol{\xi} - \boldsymbol{\mu}^k_{\boldsymbol{\xi}}) \right) \tag{4}$$

$$\hat{\boldsymbol{\Sigma}}_{\dot{\boldsymbol{\xi}}} = \sum_{k=1}^{K} h_k^2(\boldsymbol{\xi}) \left(\boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}} - \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}} \left(\boldsymbol{\Sigma}^k_{\boldsymbol{\xi}} \right)^{-1} \boldsymbol{\Sigma}^k_{\boldsymbol{\xi}\dot{\boldsymbol{\xi}}} \right), \tag{5}$$

where the weights $h_k(\boldsymbol{\xi})$ correspond to

$$h_k(\boldsymbol{\xi}) = \frac{\mathcal{P}(k)\mathcal{P}(\boldsymbol{\xi} \mid k)}{\sum_{k=1}^{K} \mathcal{P}(k)\mathcal{P}(\boldsymbol{\xi} \mid k)} \tag{6}$$

and $\hat{\boldsymbol{\mu}}_{\dot{\boldsymbol{\xi}}}$ and $\hat{\boldsymbol{\Sigma}}_{\dot{\boldsymbol{\xi}}}$ are the estimated parameters of $\mathcal{P}(\dot{\boldsymbol{\xi}} \mid \boldsymbol{\xi})$. The autonomous dynamical system can now reproduce the learned behavior by estimating the next velocity at the given current pose. Since $\hat{f}(\boldsymbol{\xi}) = \hat{\boldsymbol{\mu}}$ we get:

$$\dot{\boldsymbol{\xi}} = \sum_{k=1}^{K} h_k(\boldsymbol{\xi}) \left(\boldsymbol{\mu}^k_{\dot{\boldsymbol{\xi}}} + \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}} \left(\boldsymbol{\Sigma}^k_{\boldsymbol{\xi}} \right)^{-1} (\boldsymbol{\xi} - \boldsymbol{\mu}^k_{\boldsymbol{\xi}}) \right). \tag{7}$$

By updating the pose $\boldsymbol{\xi}$ with the generated velocity $\dot{\boldsymbol{\xi}}$ and proceeding iteratively we can generate a robot trajectory.

C. Incorporating External Features into Action Models

The system described in Sec. III-B is able to reproduce trajectories in a geometrically suitable way. There are however countless cases in which just reproducing a trajectory will not yield the desired outcome since the system should also dynamically react to other features of the scene. In this section we describe our contribution to treat external features of the system as additional dimensions in the action model learning approach.

As we have changed the definition of a trajectory $\boldsymbol{X}$ to include feature values $\boldsymbol{\rho}$ at each of the n poses (see Sec. III-A), accordingly the representation of the Gaussian function parameters θ change to

$$\boldsymbol{\mu}^k = \begin{pmatrix} \boldsymbol{\mu}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})} \\ \boldsymbol{\mu}_{\dot{\boldsymbol{\xi}}} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}^k = \begin{pmatrix} \boldsymbol{\Sigma}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})} & \boldsymbol{\Sigma}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})\dot{\boldsymbol{\xi}}} \\ \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}(\boldsymbol{\xi},\boldsymbol{\rho})} & \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}} \end{pmatrix} \tag{8}$$

With Gaussian mixture regression we can again estimate the conditional probability $\mathcal{P}(\dot{\xi}|(\xi, \boldsymbol{\rho}))$ as a new Gaussian distribution

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{\dot{\boldsymbol{\xi}}} = \sum_{k=1}^{K} h_k(\xi, \boldsymbol{\rho}) \Big(& \boldsymbol{\mu}^k_{\dot{\boldsymbol{\xi}}} \\ & + \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}(\boldsymbol{\xi},\boldsymbol{\rho})} \left(\boldsymbol{\Sigma}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})} \right)^{-1} \left((\boldsymbol{\xi}, \boldsymbol{\rho}) - \boldsymbol{\mu}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})} \right) \Big) \end{aligned} \tag{9}$$

$$\hat{\boldsymbol{\Sigma}}_{\dot{\boldsymbol{\xi}}} = \sum_{k=1}^{K} h_k^2(\xi, \boldsymbol{\rho}) \left(\boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}} - \boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}(\boldsymbol{\xi},\boldsymbol{\rho})} \left(\boldsymbol{\Sigma}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})} \right)^{-1} \boldsymbol{\Sigma}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})\dot{\boldsymbol{\xi}}} \right) \tag{10}$$

and retrieve velocity commands accordingly. Note that we are not interested in generating any prediction for $\dot{\boldsymbol{\rho}}$ as we assume that the external features cannot be controlled directly by the robot. $\boldsymbol{\rho}$ is only considered as an observation of the dynamical system to augment the action model learning by the further knowledge about the world in order to generate reactive motion commands. This way the correlation between the behavior of external features and velocity of geometric motion is captured in the action representation by the covariances $\boldsymbol{\Sigma}^k_{(\boldsymbol{\xi},\boldsymbol{\rho})\dot{\boldsymbol{\xi}}}$ and $\boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}(\boldsymbol{\xi},\boldsymbol{\rho})}$ of the GMM. Accordingly, Eq. (8) demonstrates that covariance matrices and means of all K Gaussian functions capture the relation between positions and external features observed during demonstrations and their corresponding velocities.

D. Online Execution

The model described in the previous two sections is used to generate velocities for our robot end-effector. In tasks which involve object manipulation or human-robot interactions it is often sufficient to train a model that generates the desired velocities for the end-effector and ignores the joints velocities. Thus, our approach relies on a Jacobian-based task space controller [15] that transforms the end-effector velocities of the dynamic system into joint velocities to be executed by the robot. Furthermore, to enable online adaption of the trajectories at every time step we update the dynamic model to fit the current scenario and send the resulting new commands to the motion controller. By using a task space controller, our approach can cope with both human demonstrations and kinesthetic teaching.

IV. Evaluation

Learning a skill from demonstrations can be accomplished through different approaches. In the following, we evaluate our approach in three scenarios and investigate different learning modalities, scene tracking approaches and external features.

A. Experimental Setup

In order to detect the demonstrated motions in the scene we employ a marker-less optical approach for motion capturing in some of the experiments. OpenPose [16], [17] can be used to efficiently detect keypoints of multiple people in real-time. However, for body parts like hands the estimation of orientations is not always robust. We thus also relied on a marker-based tracking system yielding smoother trajectory recordings. Furthermore, for detecting and tracking of relevant objects we used Simtrack [18], a model-based tracking system that retrieves the object's 6D poses. All experiments have been conducted with a KUKA iiwa manipulator in a real-world environment.

To train our GMMs we used the SEDS library presented in [5]. It estimates the parameters of the mixture model by solving a constraint optimization problem and ensures global asymptotic stability of the model. In doing so, SEDS guarantees that the system always converges to the target point. This approach is designed to estimate the joint probability of a set of variables and its corresponding velocities. It furthermore assumes that all parameters can be controlled by the model. In our case, we add feature values as additional input dimensions since we are interested in their influence on the geometric course of the model. The estimated rates of change for these parameters, i. e., their velocities, are not relevant in our approach as they are not directly controlled by the robot. This breaks the guaranteed asymptotic stability of the system. However, in practice this does not cause critical behavior of the system.

B. Drinking

One important area that can greatly benefit by autonomous robotic agents is assessment of people with disabilities. A patient who is paralyzed due to spinal cord injury or stroke might suffer from an inability to perform tasks like liquid intake. In this experiment, we aim to show that by performing only a few drinking demonstrations and treating the force interaction between cup and human lip as an external feature we can ensure a compliant and dynamic behavior of the robot. To measure contact forces between cup and human lips we mounted a force-sensitive resistor around the rim of a cup. With this setup, a human teacher performed three drinking demonstrations. This imitation learning approach is the most intuitive and convenient way of teaching skills to a robot. The cup trajectories are specified relative to the teacher's nose frame and contain 6-DOF poses as well as corresponding force sensor measurements. Using these demonstrations, we trained a GMM model with two Gaussian functions. Our intention behind this experiment is that the future service robot is able to learn how to perform everyday tasks by observing a human, non-expert demonstrator.

To evaluate the trained model, we compared it with another GMM model which is learned by taking only the geometric representation of the same demonstrations into account. Fig. 2 illustrates the relation between the tilting angle of the cup and the amount of force which is exerted on the rim for both models. The upper image shows the experiment which is performed by the model that relies only on the geometric description. The lower image illustrates the behavior of our model which additionally takes force interaction into consideration. Evidently, the geometric model has already started to tilt the cup while approaching the user's lips (t_1) and would spill liquid before reaching them (e. g., t_2). In contrast, our extended model keeps the cup upright until the lips get in contact with the rim (t_2) which finally triggers the tilting motion of the cup (t_3). Thus, it has learned the correlation of desired force and the relative position of the cup to the lips at each point of the trajectory. Note that by employing the extended model users additionally have the possibility to actively control the motion. As long as the user has no contact with cup it will stay upright in front of the lips. This also yields a convenient stop condition since reducing the pressure between lip and cup causes a return motion into the upright position.

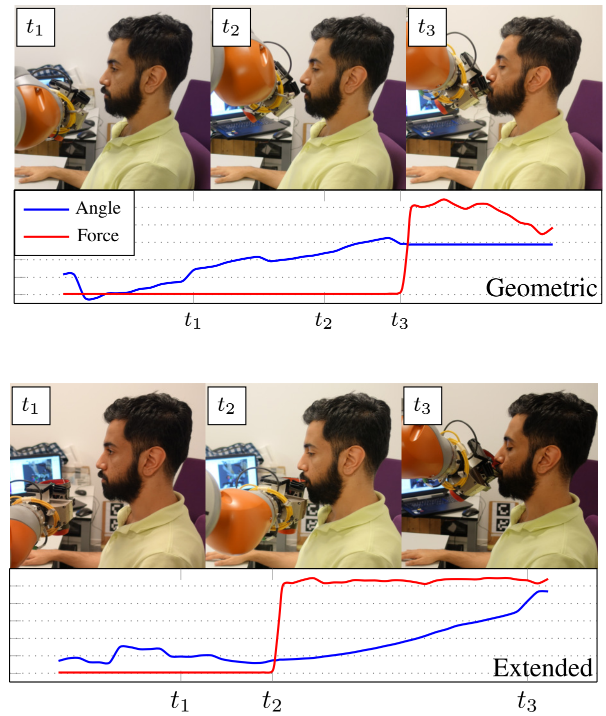

Fig. 2: The upper image shows the drinking task executed by our dynamic system that uses a geometric description of the trajectory only. At t_1 the cup already starts tilting without having contact to the lips. The contact is established in t_3 which results in potential spilling of liquid. The lower image shows our extended model which also considers the contact forces between lips and the cup ensures an upright and thus safe position of the cup as long as there is no contact. The two graphs visualize the corresponding tilting angles and contact forces. Compared to the extended model which yields the desired behavior, the geometric one does not know about the forces and starts tilting the cup too early.

C. Pouring Liquids into a Cup

Another essential skill for a domestic service robot which intends to serve beverages or cook meals is to pour liquids appropriately and accurately into containers. In the following, we demonstrate that our approach is able to adopt the liquid height in a cup as an external feature and learns an action model which pours an intended amount of liquid into the cup – regardless of whether the cup is empty or partially filled.

Do *et al.* have proposed a probabilistic framework for estimating the liquid level in a cup using low-cost and widely available RGB-D cameras [11]. Using this framework, the setup for this experiment consists of a RGB-D camera which is located above the cup to estimate its liquid height and another RGB-D camera to track the bottle of water. In the training phase, three demonstration were recorded by a human in which the cup was filled until reaching a liquid height of approximately 80%. The trajectory of the bottle has been defined as the bottle orientation in the frame of reference of the cup which we record alongside the liquid height. Using this information, we trained a GMM model with two Gaussian components that allows execution of pouring tasks and properly fills cups to a liquid height of 80%.

In order to assess the effectiveness of adding the liquid height as an external feature to the action model, we compared it with a model which was trained from the same demonstrations but without using the liquid height. Thus, it only relies on the geometrical description of the motion. The objective of this experiment was to pour water into the cup until 80% of the cup was filled. Table I illustrates the difference between both models. While the geometric model fails if there is already a specified amount of water in the cup, our model is able to cope with different initial liquid heights and still reaches the target height.

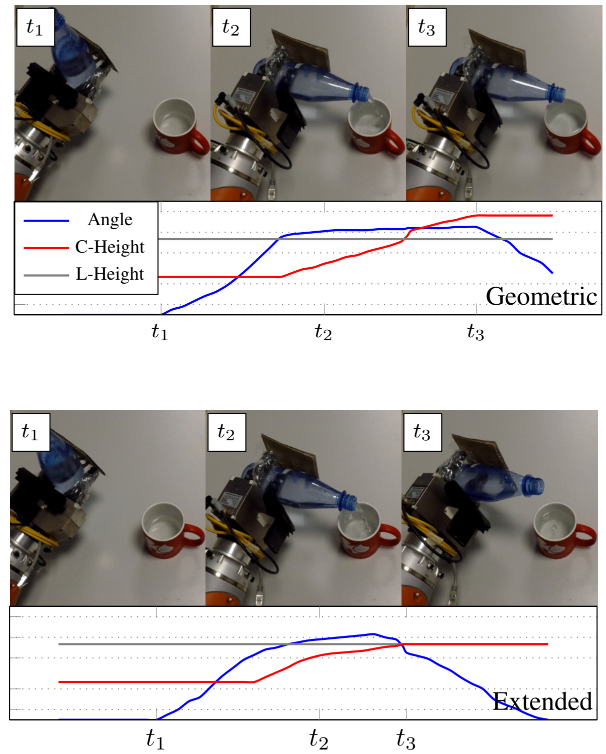

Fig. 3: The Geometric Model (upper image) fails to fill the learned height (L-Height) of the cup when the cup is already partially filled at the beginning, and the current height (C-Height) of liquid increases and results in an overflow of water. The Extended Model (lower image) manages to deal with the liquid height which is already in the cup and stops pouring immediately when the learned height of the cup is filled.

Goal Height	Initial Height	Final Height Extended Model	Final Height Geometric Model
80%	0%	81.02%	79.58%
	10%	82.84%	92.66%
	40%	78.30%	Overflow

TABLE I: The table illustrates the results of the pouring task. While the goal liquid height was always 80%, the initial heights varied from 0% to 40%. The two right columns compare the final liquid heights resulting from our extended model and a GMM model that only uses geometric parameters. It can be seen that independently of the initial height our model always reaches approximately 80%. The geometric model tries to fill additional 80% to the cup which finally results in an overflow.

D. Handing Over Objects to a Human

A fundamental yet complex skill for human-robot collaborations is a seamless handover of different objects to a human. For such an interaction, the robot has to resolve where to transfer the object at hand and more importantly when to exchange it in a dynamical setting to ensure that the human has control of the object immediately. With this experiment, we show that our GMM model which is based on a small set of handover demonstrations performed by kinesthetic teaching is not only able to capture the physical process of approaching an object to the human, but also does not release the object before the human has grasped it. To this end, our approach also takes into account the interacting forces. A widely used approach to transfer skills to robots is kinesthetic teaching. Although hand guiding a robot arm with many degrees of freedom could be challenging for non-expert users, it opens up the possibility of utilizing integrated sensors of the robot while performing demonstrations. In this experiment, we used kinesthetic teaching to collect five handover demonstrations with three different objects. During collection of training data, the hand was always at the same position and the starting positions of the robot's end-effector was moderately similar. For each demonstration we recorded (a) the 6-DOF poses of the human hand and the robot's end-effector, (b) the opening width of the gripper and (c) the Euclidean norm of the Cartesian force measurement of the robot at its end-effector. As in our other experiments, the trajectory of the end-effector was defined relative to the target frame, i. e., the human hand. To be invariant to specific objects and their weight we defined the extended features to be relative to the first measurement in each demonstration by computing their differences. Based on these demonstrations, we trained a GMM model with two Gaussian components which was able to successfully execute handover tasks with various objects.

To elaborate the characteristics of our trained model, we compared it with two baseline approaches. The first baseline method *GeoGMM* only considers the geometry of demonstrations to complete the handover task. Thus, it releases the object as soon as the robot reaches the human hand. Furthermore, *ThGMM* also uses the geometric parameters but additionally incorporates a static and user defined threshold to control opening of the robot's gripper based on the forces exerted to it. Finally, our approach – *ExtGMM* – combines the poses of the robot's end-effector and the forces exerted to it to train a model that resolves the shortages of the baseline methods. For each approach, we ran 25 experiments while using five different objects, two start positions of the robot end-effector and four hand positions.

Table II shows the results of this comparison in more detail. For the evaluation of the handover task we used different objects than what we used in the training. The limitation of the first baseline (GeoGMM) is that the model has no capability to infer if the human has already grasped the object or not. Thus, it typically opens the gripper too early resulting in a success rate of 48%. In contrast, the second baseline (ThGMM) is able to reach the goal pose correctly. However, the deficiency of this method is that it still opens the gripper before reaching the hand of the human in cases of external perturbations , e. g., by collisions with the environment or other persons interfering. Hence, the success rate of this approach was 56%. Our proposed approach (ExtGMM) is able to dynamically react to force changes since it implicitly couples the required contact forces that triggers an opening of the gripper with the distance of it to the human hand. Thus, the user can ask a robot to bring an object towards its hand and still controls when the grasp takes place, leading to the success rate of 88%. Our learned model only failed to deliver the object in scenarios where the pose of hand has been detected incorrectly by the perception system. However, since new trajectory waypoints are determined in an online manner the motion can mostly be recovered seamlessly.

Approach	# Runs	# Fails Geometry	# Fails Dropped	# Fails Forces	Success Rate
GeoGMM	25	2	11	0	48%
ThGMM	25	3	0	8	56%
ExtGMM	25	3	0	0	88%

TABLE II: Results of the handover in a human-robot interaction task. The table compares two baseline approaches with our model (*ExtGMM*). The *GeoGMM* only employs geometrical parameters to describe handovers resulting in a poor performance because of frequent unwanted drops of the object. *ThGMM* incorporates an additional threshold that monitors the forces exerted to the end-effector. While this model has no problems with dropping too early, it still fails if unexpected forces act on the robot. Finally, our model is able to reach the desired goal pose close to the user's hand and waits until the object has been grasped before opening the gripper.

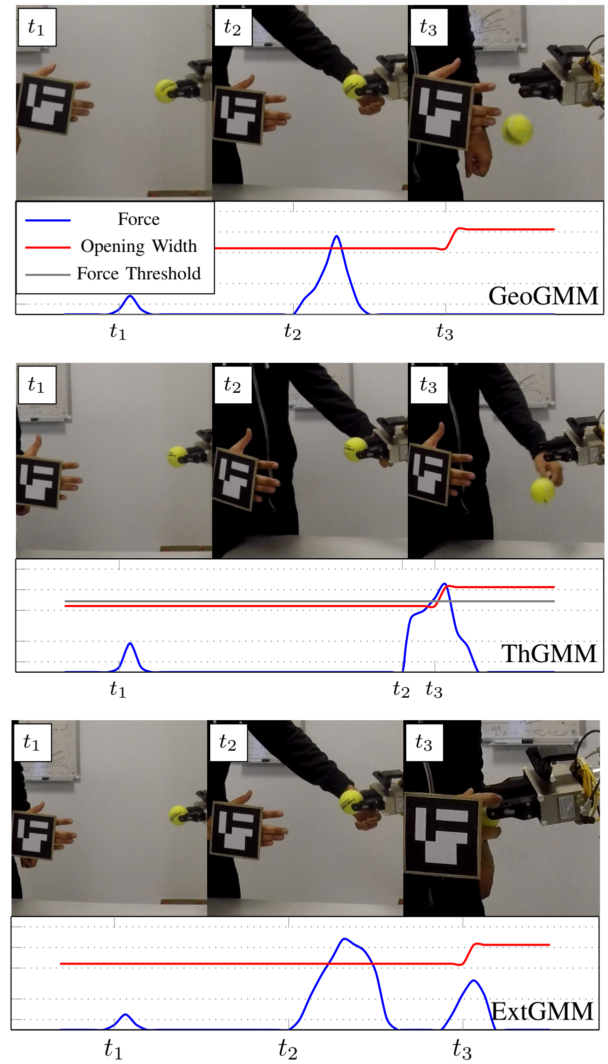

Fig. 4: GeoGMM (upper image) drops the object as soon as the robot end-effector reaches to the human hand. ThGMM (middle image) considers a static force threshold for opening the gripper but it also fails when unexpected forces work on the robot. ExtGMM (lower image) brings the object to the human hand and delivers it when the human has successfully grasped the object.

V. Conclusions

In this paper, we presented an approach to integrate non-geometric features into action model learning. We encode the motion in a dynamical system parametrized by a Gaussian Mixture Model. The correlation between the geometric course of the imitated motions and the additional features is captured in the covariances of the model. Our experiments show that we achieve a significant improvement over purely geometric approaches in the motion imitation. In all evaluated scenarios our approach is able to establish the desired geometric reaction to the perceived non-geometric features, thus yielding a convenient and safe way for robots to learn and execute new skills. In this fashion arbitrary reproducible signals could be integrated to influence the geometric course of a motion. A potential extension could build on trying to detect promising influential features automatically based on the available sensors.

References

[1] J. A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[3] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2012.

[4] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[5] S. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[6] T. Welschehold, C. Dornhege, and W. Burgard, "Learning mobile manipulation actions from human demonstrations," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[7] C. Zimmermann, T. Welschehold, C. Dornhege, T. Brox, and W. Burgard, "3d human pose estimation in rgbd images for robotic task learning," in *Int. Conf. on Robotics & Automation (ICRA)*, 2018.

[8] T. Welschehold, C. Dornhege, F. Paus, T. Asfour, and W. Burgard, "Coupling mobile base and end-effector motion in task space," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[9] M. Mühlig, M. Gienger, and J. Steil, "Interactive imitation learning of object movement skills," *Autonomous Robots*, vol. 32, no. 2, pp. 97–114, 2012.

[10] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1049–1065, Oct 2014.

[11] C. Do and W. Burgard, "Accurate pouring with an autonomous robot using an rgb-d camera," in *The 15th International Conference on Intelligent Autonomous Systems (IAS)*, Baden Baden, Germany, 2018.

[12] C. Do, C. Gordillo, and W. Burgard, "Learning to pour using deep deterministic policy gradients," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018.

[13] F. Burget, L. D. J. Fiederer, D. Kuhner, M. Voelker, J. Aldinger, R. T. Schirrmeister, C. Do, J. Boedecker, B. Nebel, T. Ball, and W. Burgard, "Acting thoughts: Towards a mobile robotic service assistant for users with limited communication skills," *CoRR*, vol. abs/1707.06633, 2017.

[14] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[15] B. Siciliano, L. Sciavicco, S. Chiaverini, P. Chiacchio, L. Villani, and F. Caccavale, "Jacobian-based algorithms: A bridge between kinematics and control," in *Proceedings of the Special Celebratory Symposium In the honor of Professor Bernie Roth's 70th Birthday*. Citeseer, 2003, pp. 4–35.

[16] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[17] M. Arduengo, S. Jorgensen, K. Hambuchen, L. Sentis, F. Moreno-Noguer, and G. Alenyà, "Ros wrapper for real-time multi-person pose estimation with a single camera," Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Tech. Rep., 2017.

[18] K. Pauwels and D. Kragic, "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 1300–1307.