2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Imitation Learning of Human Grasping Skills from Motion and Force Data

Alexander M. Schmidts, Dongheui Lee, Angelika Peer
Institute of Automatic Control Engineering
Technische Universität München
D-80290 München, Germany
{Alexander.Schmidts, dhlee, Angelika.Peer}@tum.de

Abstract— **Imitation learning, also known as Programming by Demonstration, allows a non-expert user to teach complex skills to a robot. While so far researchers focused on abstracting kinematic relations, only little attention has been paid to force information. In this work we study imitation learning of human grasping skills from motion and force data. For this purpose a teleoperation system is realized that allows a human to control a simulated robotic hand and to grasp objects in a virtual environment. Haptic rendering algorithms are implemented to calculate interaction forces that occur when touching the virtual object. While learning of fingertip interaction forces is shown to result in physical inconsistency compared to the demonstrations, we show that learning of internal tensions leads to stable reproductions of the demonstrated grasping skill. Obtained results further indicate an enlarged generalisation capability of grasping skills learnt on the basis of motion and force data compared to grasping skills that encode kinematic relations only.**

I. INTRODUCTION

While today's industrial robots are capable of solving specific tasks in well known environments, robots that share their working space with humans are confronted with unknown and timely varying environments. To increase capabilities of such robots, the possibility of teaching them complex tasks is of paramount importance.

Programming by Demonstration (PbD), also known as Imitation Learning [1], is an often adopted approach in this context. Recorded data is used in an abstraction step to extract most general information (encoding) which is then utilized in a reproduction step (decoding) to synthesize the learned task. The approach can be adopted on a high level with complex task knowledge or on a low level with elementary skills. As encoding models Neural Networks [2], [3], Hidden Markov Models (HMMs) [4] or Gaussian Mixture Models (GMMs) [5] are commonly applied.

Grasping represents a very complex and frequently used task, which requires simultaneous coordination of motion and force. In Ekvall and Kragić HMMs were applied for human grasp recognition [6]. After mapping recognised grasps to a robot manipulator they executed different grasp types with the help of hand shape primitives. Hsiao and Lozano-Pérez used a teleoperation system to learn whole-body grasps from demonstrations on a simulated humanoid robot [7]. The demonstrations are sequences of keyframes in which representative interaction points are recorded. Gräve et al. built a system which learnt parametric motion primitives from human demonstrations and improved them by reinforcement learning [8]. Bohg and Kragić also investigated grasping points, but with a non-linear classification algorithm [9]. Rao et al. used a supervised localisation method to identify graspable segments [10].

The mentioned approaches have in common that kinematic relations of a grasping skill are learnt, while neglecting force information. However, force information is an important component for the succesful execution of a grasping task. Therefore we explicitly study imitation learning of human grasping skills from motion and force data with multiple interaction points in this paper. In contrary to recent approaches which used force information for single point interactions [11], [12], the additional complexity of multiple interaction points in the grasping task leads to new challenges. While direct learning of fingertip interaction forces will be shown to result into inconsistent physical behavior compared to the demonstrations, we will demonstrate that learning of internal tensions leads to stable reproductions of the demonstrated grasping skill. We will further demonstrate an enlarged generalisation capability of grasping skills learnt on the basis of motion and force data compared to grasping skills that encode kinematic relations only.

The paper is structured as follows. Section II provides a general overview of our approach, while section III details the implemented imitation learning algorithms. Section IV discusses the necessary physical consistency of forces when grasping skills are learnt from motion and force data. Details of the experimental implementation are provided in section V. Finally, section VI reports on experimental results that demonstrate the benefit of including force information when learning human grasping skills.

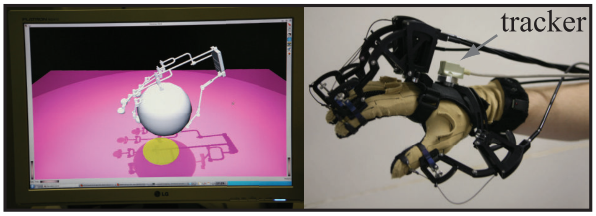

Fig. 1. Teleoperation system used for demonstration. Using the data glove and motion tracker attached to the users hand (right) the simulated Elumotion Hand is controlled (left).

978-1-61284-455-8/11/$26.00 ©2011 IEEE

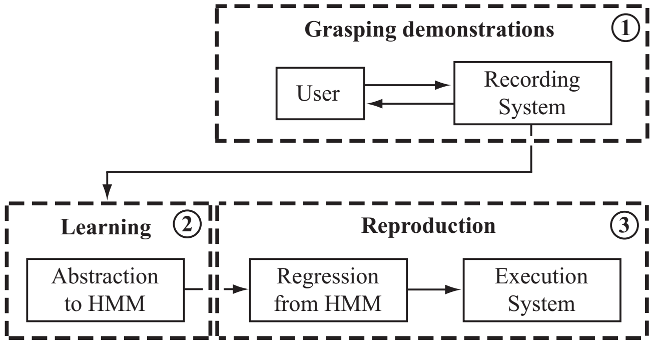

Fig. 2. (1) During demonstrations force and motion data is recorded. (2) This information is abstracted by a learning algorithm. (3) Regression is performed to extract generalised time-series of data, which are used for reproduction of the task.

II. OVERVIEW

In order to study imitation learning of human grasping skills we adopt the setup shown in Fig. 1. Demonstrations of grasping skills are realized by means of a teleoperation system. On operator site human finger motions are measured using a dataglove and mapped to a robotic hand, which is simulated by means of a dynamical simulation. Haptic rendering is used to simulate interaction forces between the robotic hand and the object. Force information is fed back to the human operator and displayed by means of an exoskeleton. Fig. 2 provides the overview of our implemented learning approach.

III. IMITATION LEARNING

The abstraction and regression processes represent the core of an imitation learning framework. While Gaussian Mixture Regression (GMR) [13] was found to have the better regression quality, HMMs have the advantage of containing time as an implicit variable, which results into an enhanced recognition quality. Thus, herein an HMM is used for abstraction, while GMR for regression. For a general explanation of HMMs please refer to Rabiner [14].

A. Abstraction to HMM

An HMM is defined by the initial state probability, the probability transition matrix, the covariance matrix $\boldsymbol{\Sigma}_{ik}$ and the center $\boldsymbol{\mu}_{ik}$ of Gaussian k in state i:

$$\boldsymbol{\Sigma}_{ik} = \begin{bmatrix} {}^{tt}\Sigma_{ik} & {}^{ts}\boldsymbol{\Sigma}_{ik} \\ {}^{st}\boldsymbol{\Sigma}_{ik} & {}^{ss}\boldsymbol{\Sigma}_{ik} \end{bmatrix} \qquad \boldsymbol{\mu}_{ik} = \begin{bmatrix} {}^{t}\mu_{ik} \\ {}^{s}\boldsymbol{\mu}_{ik} \end{bmatrix}$$

Temporal values are marked with an index t and spatial values with an index s. The e-th demonstration is referred to with $\boldsymbol{o}^e(t)$.

At the current implementation, the topology of the HMM (i.e., the number of states) are determined heuristically (cf. subsection III-C). So far, existing methods (e.g., Bayesian Information Criterion [15]) for topology estimation are often based on trade-off between the classification performance and model complexity. However, reproductions quality, which is crucial for grasping tasks, is not taken into account. In the future, it will be interesting to investigate on topology selection for dynamical tasks.

After topology determination the HMM parameters are estimated by a following two step procedure. First, parameters of the HMM are initialised. Due to the high-dimensional and non-linear problem, an arbitrary initialisation may result into poor local maxima and thus, the k-means algorithm is typically adopted in this context. Second, the estimation is done based on the EM-algorithm. The resulting HMM parameters do not contain explicit time information which is necessary for GMR. Thus, the probability that an observation belongs to each state and each Gaussian is calculated after the convergence of the EM-algorithm. Given this probability, the means and covariances of temporal data can be determined. An improvement of these values can be achieved by taking into account that parts of different demonstrations have been executed with differing time ranges. Therefore the correlation of the temporal information in the demonstrations is enhanced as detailed in the following paragraphs.

1) Transformations of demonstrations:

a) Time-scaling: All demonstrations are scaled in time to the mean time T which is calculated over all demonstrations. This results into demonstrations of an equal time length.

b) Dynamic Time Warping (DTW) [15]: Classical DTW de-skews temporally warped one-dimensional value sequences by building a warping path which results from the smallest cumulative error. However, in PbD multiple N-dimensional value sequences are given. We propose an approach addressing the following two points: (*i*) a reference demonstration is chosen, (*ii*) one or multiple 1D warping paths are built for a demonstration consisting of N spatial dimensions.

To solve the first problem, we selected the reference by considering the information content of a demonstration. In general a varying trajectory contains more information than a static one. Thus, we choose the trajectory with the maximum difference between maximum and minimum value as reference.

The second problem is solved by projecting the N dimensions to one representative dimension. The main issue is to maintain the type-dependent influence of the dimensions (e.g. joint angles or endeffector positions). Therefore, the method described in (1) is chosen

$$\begin{aligned} z(t_i) &= \sum_g \left[\frac{k_g}{N_g} \sum_j |o^e_{g,j}(t_i)| \right] \\ k_g^{-1} &= \max(o^e_{g,j}(t_1), o^e_{g,j}(t_2), ...o^e_{g,j}(T)) \qquad \forall j \qquad (1) \\ \boldsymbol{z}^e &= [z(t_1), z(t_2), ...z(T)] \end{aligned}$$

where a weighted sum is built depending on the type g by normalisation with k_g and the dimensionality N_g. j denotes the different dimensions of a type. In $\boldsymbol{z}^e$ the representative one dimensional value sequence for a demonstration is contained. From this procedure follows that every weighted summand equals at maximum 1. Consequently, significant values have a high influence, while non-significant values have little influence on $\boldsymbol{z}^e$. Raw demonstrations and the

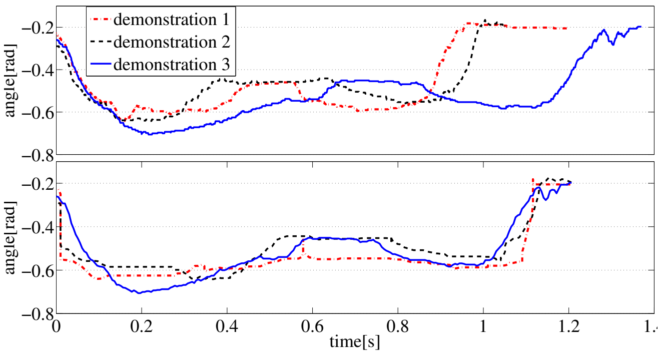

Fig. 3. Preprocessing of observations. Top: Raw demonstrations. Below: Results after adaptation and DTW

obtained results after time-scaling and DTW are shown in Fig. 3.

2) Initialisation: For initialisation the following two-step procedure is adopted:

a) Scaling: The time range is scaled to be one order of magnitude larger than the magnitude range. This enhances the quality of the k-means algorithm [5].

b) k-means algorithm: The algorithm returns centers which can be distributed among the Gaussians after a rescaling to fit the original data as well as the indices of the points belonging to these centers. Thus, the covariance matrices can be initialised as follows:

$$\boldsymbol{\Sigma}_{ik} = \frac{\sum (\boldsymbol{o}_{ik} - \boldsymbol{\mu}_{ik})(\boldsymbol{o}_{ik} - \boldsymbol{\mu}_{ik})^T}{N}$$

3) EM algorithm: After the initialisation steps, the Baum-Welch algorithm is executed for parameter estimation [16]. Beside the probability transition matrix and the initial state probability the algorithm only returns the spatial centers and covariance matrices. However, the auxiliary variables $\gamma^e_{ik}(t)$ describing the probabilities that an observation belongs to a certain Gaussian k and state i is also estimated. Thus, these probabilities are used to calculate ${}^t\mu_{ik}$, ${}^{tt}\Sigma_{ik}$ and ${}^{ts}\boldsymbol{\Sigma}_{ik}$. Interested readers please refer to [4] for details.

B. Regression

Regression is performed in two steps. First, linear functions are expressed by the parameters of the Gaussians

$$\boldsymbol{x}_{ik} = {}^s\boldsymbol{\mu}_{ik} + {}^{st}\boldsymbol{\Sigma}_{ik}({}^{tt}\Sigma_{ik})^{-1}(t - {}^t\mu_{ik}).$$

Second, smoothing between these linear functions is done by a weighted sum. The weight h_{ik} is based on the probability that a Gaussian is responsible for a given time point

$$h_{ik} = \frac{\mathcal{N}(t; {}^t\mu_{ik}, \epsilon^{tt}\Sigma_{ik})}{\sum_{k=1}^{K}\sum_{i=1}^{I}\mathcal{N}(t; {}^t\mu_{ik}, \epsilon^{tt}\Sigma_{ik})}$$

where $\mathcal{N}(t;^t\mu_{ik}, \epsilon\ ^{tt}\Sigma_{ik})$ is an univariate Gaussian distribution with the time as input variable and a smoothing factor ϵ. The denominator is summed over all I states and K Gaussians. This factor is an extension to the common regression and used as a tuning parameter to keep the maximum velocities in a certain range (see subsection III-C for details). The final regression formula is given by

$$\boldsymbol{x} = \sum_{k=1}^{K}\sum_{i=1}^{I} h_{ik}\boldsymbol{x}_{ik}.$$

C. Smoothing factor

For the evaluation of the smoothing factor three artificial demonstrations g_i of similar nature are given that span over a time range of $t \in [0;4]$s:

$$\begin{aligned} g_1 &= \cos(\omega t) \\ g_2 &= 1.01\cos(1.05\omega(t + 0.05)) + 0.05 \\ g_3 &= 0.97\cos(0.97\omega(t - 0.03)) - 0.03 \end{aligned} \tag{2}$$

where $\omega = 0.6 \cdot 2\pi$ s^{-1} is the angular frequency.

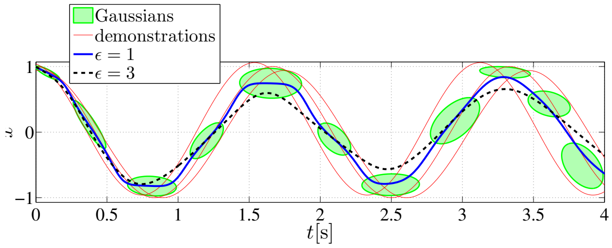

Fig. 4. Generalised trajectories for different smoothing factors ϵ.

From these three demonstrations an HMM is estimated assuming a left-right topology, one Gaussian per state and a total number of 11 states. This number follows from the fact that every Gaussian should describe a certain extremum or linear function contained in the observations. Then, the generalised trajectories are retrieved with different smoothing factors. Some specific values are calculated and shown in Table I. The result of the regression is shown in Fig. 4. Increasing the smoothing factor leads to a larger deviation from the demonstrations (see RMS error) but also to decreasing amplitudes of the derivatives. This characteristic can be used to restrict the generalised trajectories to realizable velocities and accelerations. Thus, the smoothing factor can be used to define a trade-off between kinematic constraints of the manipulator and task constraints which are given by the desired precision of the execution.

TABLE I

CHARACTERISTICS OF GENERALISED TRAJECTORIES FOR DIFFERENT VALUES OF THE SMOOTHING FACTOR

smoothing factor	RMS error	max($\lvert\dot{x}\rvert$)	max($\lvert\ddot{x}\rvert$)
0.5	0.50	6.32	128.43
1.0	0.50	3.73	44.78
2.0	0.56	2.67	19.91
3.0	0.62	2.28	15.52

IV. LEARNING OF GRASPING SKILLS FROM MOTION AND FORCE DATA

If a grasping skill is learnt based on motion data only, kinematic task constraints can be extracted from a series of demonstrations that allow its reproduction on similar objects. However, when applying the learnt skill to objects of a different size or stiffness, high interaction forces might result and this can damage the robotic hand or object. Such damages can be avoided by considering also force information in the learning process.

Including force information, however, requires special considerations to be taken into account. Due to the closed kinematic chains formed between fingers and object, forces applied to the object have to balance each other in steady state. Simple reproduction of forces applied during multiple demonstrations at the single interaction points, however, can lead to physically inconsistent forces and consequently unstable grasping behaviors. Given for example demonstrations of a two-finger precision grasp which is abstracted and reproduced with the interaction forces that have been applied at the two contact points. When learning and reproducing these interaction forces, they do not necessarily balance each other. As a consequence, the object will deviate from its desired path and in the worst case, the grasp will become unstable.

In order to overcome these problems when reproducing grasping skills from motion and force data, we split forces acting on the object into external and internal forces (cf. Fig. 5). For rigid bodies they are defined as follows: External forces mean resulting forces from contact interactions with the environment which are acting in sum on the center of mass of the object. The static part compensates volumetric forces like gravitation or magnetism. The dynamic part leads to an acceleration or deceleration of the object. Internal forces are forces that act on the cross section area of the object and lead to tension in the object. The static part means forces that squeeze or stretch the object, while the dynamic part describes forces that act on the object in consequence of the inertia of the object.

Static internal forces as defined above, correlate with internal tensions. The latter have the advantage that they are decoupled of the actual kinematic constellation of the hand and the location of the contact points, however, they contain the same information as the static internal forces. Thus, learning internal tensions instead of the interaction forces guarantees physically consistent behavior and consequently the reproduction of stable grasps. For two interaction points between manipulator and object, the external $\boldsymbol{f}_{ext}$, the static internal forces $\boldsymbol{f}_{int,i}$ for each finger i and the internal tension σ can be calculated based on the scheme of Yoshikawa and Nagai [17] that models a virtual linkage between the two interaction points:

$$\begin{aligned}\boldsymbol{f}_{ext} &= \sum_i \boldsymbol{f}_i \\ \sigma &= \min(||\boldsymbol{f}_1^T \boldsymbol{e}_{12}||, ||\boldsymbol{f}_2^T \boldsymbol{e}_{12}||) \\ \boldsymbol{f}_{int,i} &= \pm\sigma \boldsymbol{e}_{12}\end{aligned}$$

where $\boldsymbol{f}_i$ denotes the interaction forces at finger i and $\boldsymbol{e}_{12}$ the unit vector from interaction point 1 to 2. If more than two interaction points exist, internal forces and tensions $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, ..., \sigma_L]^T$ corresponding to the L virtual linkages can be calculated with the grasp description matrix $\boldsymbol{E}$ and its pseudoinverse $\boldsymbol{E}^\dagger$ [18].

$$\begin{aligned}\boldsymbol{\sigma} &= \boldsymbol{E}^\dagger [f_1, f_2, ..., f_n]^T \\ \boldsymbol{f}_{int,i} &= \boldsymbol{E}\boldsymbol{\sigma}\end{aligned}$$

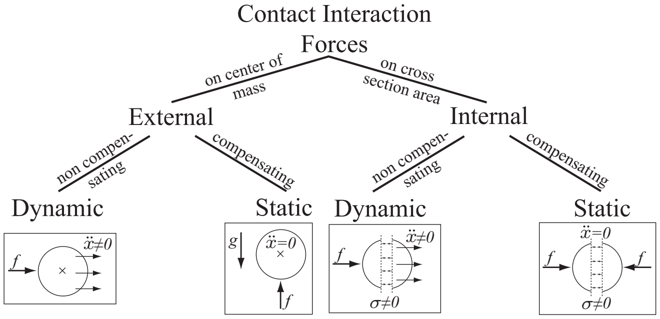

Fig. 5. Decoupling of interaction forces into internal and external force components: In the boxes examples are given with f forces, $\ddot{x}$ acceleration, g gravitation and σ internal tensions

V. EXPERIMENTAL SETUP

In this paper we analyse a two finger precision grasp that consists of the following four phases: reaching, grasping, lifting and holding of the object. A 13-dimensional vector containing the joint angles $\boldsymbol{\varphi}$, the tension σ as well as the position and orientation of the robot palm were recorded during multiple demonstrations. Subsequently a left-right HMM is trained with these demonstrations. Every state contains one Gaussian and reproduction is done using GMR.

The following subsections provide details about the used hardware setup as well as the implemented controllers and haptic rendering.

A. Hardware setup

The experimental setup consists of the following components (cf. Fig. 1).

- **Cyberglove**: a strain-gauge-based finger angle measurement unit; human finger motions are mapped to robot motions using a deterministic mapping algorithm
- **Cybergrasp**: a hand exoskeleton to apply forces to the fingertips and to provide a realistic grasping sensation
- **Flock of Birds**: a magnetic tracking system to measure palm position and orientation which is mapped to the robot hand
- **Elumotion Hand**: simulation of an anthropomorphic hand with nine actuated degrees of freedom and 20 joints

B. Dynamic simulation of robotic hand

For the simulation of the Elumotion Hand the human palm position and orientation in space is directly mapped to the robot palm. Each robot finger is simulated by means of a dynamical equation

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T \boldsymbol{f} \qquad (3)$$

that models inertial, gravitational and interaction forces acting on the fingertip and by taking into account kinematic constraints resulting from couplings between the joints.

C. Robot hand controller

In order to realise grasping skills on the robotic hand, two kinds of controllers are implemented. A PD position controller with diagonal gain matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$

$$\boldsymbol{\tau}_p^* = \boldsymbol{K}_p(\boldsymbol{\varphi_d} - \boldsymbol{\varphi_a}) + \boldsymbol{K}_d(\dot{\boldsymbol{\varphi}}_d - \dot{\boldsymbol{\varphi}}_a) \tag{4}$$

and a PI force controller

$$\boldsymbol{\tau}_f^* = \boldsymbol{J}^T \left[\boldsymbol{\sigma}_d \boldsymbol{e}_{12,a} - \boldsymbol{f}_{int,a} + k_p \int \boldsymbol{\sigma}_d \boldsymbol{e}_{12,a} - \boldsymbol{f}_{int,a} \mathrm{d}t \right] \tag{5}$$

where $\boldsymbol{J}$ denotes the Jacobian. The variable $\boldsymbol{\sigma}_d$ denotes the desired internal tension which results from the regression and $\boldsymbol{f}_{int,a}$ the actual static internal force applied at the interaction points, while $\boldsymbol{e}_{12,a}$ denotes the actual unit vector from one finger to the other. The integral part of the force controller is activated when the first finger touches the object. Equations (4) and (5) define 20 joint torques. Since only 9 of them are actuated, the motor torques $\boldsymbol{\tau}_f = [\tau_{f,1}, ..., \tau_{f,9}]^T$ and $\boldsymbol{\tau}_p = [\tau_{p,1}, ..., \tau_{p,9}]^T$ are calculated by summing up related torques of $\boldsymbol{\tau}_f^*$ and $\boldsymbol{\tau}_p^*$. The final commanded torque is given by

$$\boldsymbol{u} = \boldsymbol{\tau}_p + \boldsymbol{\tau}_f. \tag{6}$$

Note that during the demonstration phase only position control (4) is applied, while both position and force control is used during execution of the learned grasping skill.

D. Haptic rendering

Haptic rendering for realistic grasping of a virtual sphere is implemented. We modelled a single point contact between the sphere and the ground by means of a spring-damper system that simulates the compliance of the sphere and allows to calculate interaction forces due to contact with the floor, see Fig. 6. For the finger-sphere interaction a god-object/proxy based approach was chosen that assumes a spring to be positioned between fingertip and respective proxy position [19]. This approach is extended according to [20] to include static and sliding friction. For damping the relative finger-sphere movement, a damper is included between a fingertip and the sphere and not between a fingertip and its proxy to prevent oscillations originating from numerical integration. A schematic of the implemented approach is shown in Fig. 6.

VI. EXPERIMENTAL RESULTS

A. Learning of interaction forces versus internal tensions

For the experimental evaluation four demonstrations of the two finger precision grasp were recorded. After the abstraction step the generalised trajectories were reproduced using three states/Gaussians. In order to study the influence of the learnt force on reproduction, learning of finger interaction forces was compared to learning of the internal tensions. As can be seen in Fig. 7, learning of interaction forces led to an unstable grasp. This is the case, because the demonstrated and learnt forces are not physically consistent anymore, i.e. demonstrated compensating forces do not compensate each other in steady state. The index finger has a higher desired internal force than the thumb. Thus, the index finger pushes and tries to increase the force while the thumb is retreating and trying to reduce the force. This leads to the observed unstable grasp. In contrary learning and reproduction of internal tensions avoids these drawbacks and leads to a reproduction of the grasp, which is close to human demonstrations.

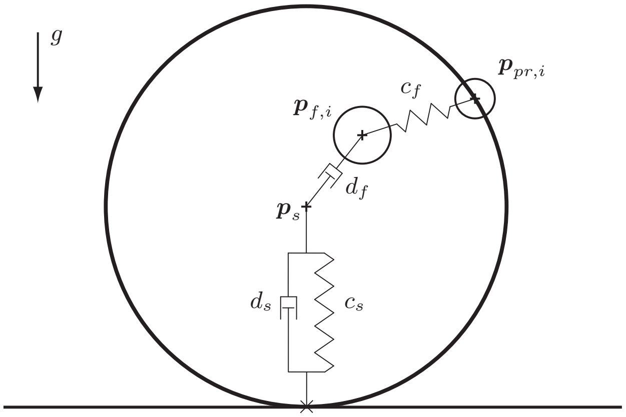

Fig. 6. Sphere-finger and sphere-ground contacts simulated by spring-damper systems. Sliding and static friction is added to make grasping of the object possible. $\boldsymbol{p}_{pr,i}$, $\boldsymbol{p}_{f,i}$ and $\boldsymbol{p}_s$ are the positions of the finger proxies, the fingertips and the center of the sphere. c_f and c_s denote the spring constants and d_f and d_s denote the damping constants. g indicates the direction of gravity.

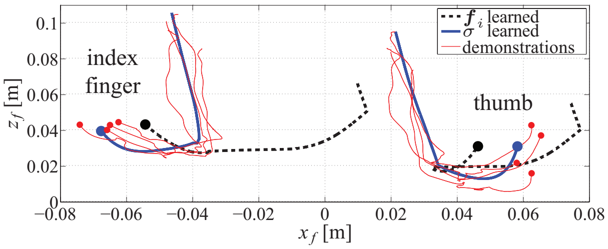

Fig. 7. Grasp attempts after having learnt interaction forces $\boldsymbol{f}_i$ of each finger i or internal tension σ. The former attempt fails. Zero time is indicated by the filled circles.

B. Benefits of learnt force information for generalisation

Additionally, we studied the generalisation capability for similar objects and compared learning of the grasping skill from motion data only with learning from motion and force data. A series of reproduction attempts with a varying mass m_s and radius r of the object were carried out and the maximum and mean of the applied internal tensions ($\overline{\sigma}$ denotes the mean internal tension which acts in steady-state) were recorded. As can be seen from Table II, learning of internal tensions and control of these tensions prevents the object and the robot hand to be damaged, i.e. there are no excessive or too small forces. Another positive effect of force control is that contact with the sphere is established with less time delay compared to approaches based on position control. The time difference ΔT observed between the two fingertips coming into contact with the sphere is used as measure for the synchronisation capability. This capability is important, since forces applied by one finger only, result into motion of the object, which can prevent successful grasps (compare e.g. lowest entry in Table II for varying radii). Force control also assures that contact with the object is established in case of too small objects, which otherwise would result into unsuccessful grasping attempts (compare e.g. first entry in Table II for varying radii). Summarising, we found that learning of force in addition to motion data leads to an enlarged generalisation capability of the learnt grasping skill compared to learning based on motion data only. Excessive forces are omitted when grasping larger objects or objects with greater stiffness. Grasping of smaller objects or more compliant objects becomes possible and fingertip synchrony is enhanced which all increases the success rate of grasping attempts.

TABLE II

INTERNAL TENSIONS AND SYNCHRONIZATION CAPABILITY FOR REPRODUCTION ATTEMPTS WITH AND WITHOUT FORCE CONTROL FOR VARYING OBJECT MASS AND RADII - THE HIGHLIGHTED PARAMETERS ARE USED FOR LEARNING

Varying object mass						
m_s[kg]	$\max(\sigma)$[N]		$\overline{\sigma}$[N]		ΔT[ms]	
0.24	3.43	5.34	3.20	4.36	1	163
0.32	3.21	5.44	3.20	4.47	11	200
0.40	**3.21**	**5.41**	**3.20**	**5.10**	**11**	**209**
0.48	3.21	5.35	3.20	5.06	11	212
0.56	3.21	5.30	3.20	5.01	11	215
Force control	ON	OFF	ON	OFF	ON	OFF

Varying object radii						
r[cm]	$\max(\sigma)$[N]		$\overline{\sigma}$[N]		ΔT[ms]	
3.6	3.21	-*	3.20	-*	28	-*
4.0	**3.21**	**5.41**	**3.20**	**5.10**	**11**	**209**
4.8	3.21	7.12	3.20	7.04	39	371
5.6	3.21	12.92	3.20	12.84	88	531
6.0	3.21	-*	3.20	-*	106	-*
Force control	ON	OFF	ON	OFF	ON	OFF

* unsuccessful grasping attempt

VII. SUMMARY AND CONCLUSION

In this work an approach for learning grasping skills in an imitation learning framework is presented. While in the state of the art grasping skills are mainly learnt based on motion data only, force information is considered to enhance the quality of the reproduction attempt. While learning of fingertip interaction forces is shown to result in physical inconsistency compared to the demonstrations, learning of internal tensions leads to stable reproductions of the demonstrated grasping skill. We also found an enlarged generalisation capability of a grasping skill learnt on the basis of motion and force information compared to skills learnt on the basis of motion data only. Excessive forces were avoided when grasping larger objects or objects with greater stiffness while grasping of smaller objects or more compliant objects became possible. Additionally fingertip synchrony was enhanced. As a consequence this led to an increased success rate of the performed grasping attempts and moderate force levels compared to reproduction attempts that neglected force information.

Future research will be twofolded. First, experiments will be done with a real manipulator to underline the theoretic results. Second, an analysis of learning power grasps will be carried out. In contrast to precision grasps where only a few interaction points are considered, a power grasp requires to handle a big amount of interaction points or even areas. This leads to new theoretic and numerical challenges.

REFERENCES

[1] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zöllner, and M. Bordegoni, “Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm,” in *9th International Symposium of Robotics Research*, 1999, pp. 229–238.

[2] K. MacDorman, R. Chalodhorn, and M. Asada, “Periodic nonlinear principal component neural networks for humanoid motion segmentation, generalization, and generation,” in *17th International Conference on Pattern Recognition*, vol. 4, 2004, pp. 537 – 540 Vol.4.

[3] R. Chalodhorn, D. Grimes, G. Maganis, R. Rao, and M. Asada, “Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional spaces,” in *IEEE International Conference on Robotics and Automation*, May 2006, pp. 3693 –3698.

[4] D. Lee and C. Ott, “Incremental motion primitive learning by physical coaching using impedance control,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4133 –4140.

[5] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

[6] S. Ekvall and D. Kragić, “Learning and evaluation of the approach vector for automatic grasp generation and planning,” in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4715 –4720.

[7] K. Hsiao and T. Lozano-Perez, “Imitation learning of whole-body grasps,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5657 –5662.

[8] K. Gräve, J. Stüandckler, and S. Behnke, “Improving imitated grasping motions through interactive expected deviation learning,” in *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 397 –404.

[9] J. Bohg and D. Kragic, “Grasping familiar objects using shape context,” in *International Conference on Advanced Robotics*, 2009, pp. 1 –6.

[10] D. Rao, Q. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Ng, “Grasping novel objects with depth segmentation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2578 –2585.

[11] C. H. Park, J. H. Kyung, D. I. Park, K. T. Park, D. H. Kim, and D. G. Gweon, “Direct teaching algorithm for a manipulator in a constraint condition using the teaching force shaping method,” *Advanced Robotics*, vol. 24, pp. 1365–1384, 2010.

[12] P. Kormushev, S. Calinon, and D. G. Caldwell, “Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input,” *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2010.

[13] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[14] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, 1989.

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

[16] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[17] T. Yoshikawa and K. Nagai, “Manipulating and grasping forces in manipulation by multifingered robot hands,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 1, pp. 67 –77, Feb. 1991.

[18] D. Williams and O. Khatib, “The virtual linkage: a model for internal forces in multi-grasp manipulation,” in *IEEE International Conference on Robotics and Automation*, 1993, pp. 1025 –1030.

[19] C. Zilles and J. Salisbury, “A constraint-based god-object method for haptic display,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems 95. ’Human Robot Interaction and Cooperative Robots’*, vol. 3, Aug. 1995, pp. 146 –151 vol.3.

[20] D. Ruspini, K. Kolarov, and O. Khatib, “Haptic interaction in virtual environments,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1997, pp. 128 –133.