# Using Learning from Demonstration to Generate Real-time Guidance for Haptic Shared Control

C. J. Pérez-del-Pulgar[1,2], Jan Smisek[1,3], V. F. Muñoz[2] and André Schiele[1,4]
[1]Telerobotics and Haptics Laboratory, ESTEC, European Space Agency, Noordwijk, the Netherlands
[2]System Engineering and Automation Department, Universidad de Málaga, Andalucia Tech, Malaga, Spain
[3]Faculty of Aerospace Engineering, Delft University of Technology, Delft, the Netherlands
[4]Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, the Netherlands
E-mail: {carlosperez, vfmm}@uma.es, {j.smisek, a.schiele}@tudelft.nl

***Abstract*—This paper introduces a new Learning from Demonstration (LfD)-based method that makes usage of robot effector forces and torques recorded during expert demonstrations, to generate force-based haptic guidance reference trajectories online, that are intended to be used during haptic shared control for additional operator 'guidance'. Derived haptic guidance trajectories are superimposed to master-device inputs and feedback forces within a bilateral control experiment, to assist an operator by the guidance during peg-in-hole insertion. We show that 96 peg-in-hole expert demonstrations were sufficient to obtain a good model of the task, which was used on-line to generate haptic guidance trajectories in real-time with a 1kHz sampling rate.**

## I. Introduction

During robot teleoperation, the inclusion of haptic feedback alone, already increases the awareness of an operator (master) of the situation between robot (slave) and its environment site [1]. Despite much progress on improving fidelity of bilateral control, still, teleoperation itself is limited by human limitations (e.g., judgment errors, loss of attention, fatigue). Autonomy, in contrast, does not provide the 'degree of human co-action' that is generally required to carry out varying and intuitive tasks in remote and unstructured environments. Haptic shared control, in which haptic teleoperation is merged with additional (synthetic) guidance force during task execution, is interesting for many application cases. Addition of this haptic guidance was shown to reduce operator workload and to improve teleoperated task performance [2], [3]. Conventionally, haptic guidance is implemented as a virtual spring whose output is summed to a haptic master device input to 'guide' an operator to follow a prescribed reference trajectory (virtual fixtures) [4]. This method has been implemented in different fields, such as remote assembly [3], tele-surgery [5] and vehicle control [6]. It is, in principle, position-input based, relying on a known and accurate reference position trajectory usually obtained from images or marker data [7]. However, there are tasks in which the position or trajectory can be affected by measurement errors [8] and by geometric and contact constraints. Moreover, it is often challenging to know an appropriate guidance trajectory a-priori. During teleoperated tasks, the guidance needs to be based on the momentary operator action. In particular for assembly tasks (e.g., peg-in-hole insertion), guidance must be based on contact geometry and forces. While a peg-in-hole insertion can still be modeled explicitly, more complex tasks, such as tying a knot, are unfeasible for explicit modeling. Learning from Demonstration (LfD) offers suitable tools to address these challenges. Previous similar works have been applied only to temporally continuous trajectories, which can be 'played back', not on force-based trajectories computed on-line. Temporal trajectories have been used to program robots to perform previously trained human tasks, such as pouring a glass of water [9], hitting a table tennis ball, feeding a robotic doll [10] or placing a ball in a hole inside a box [11]. However, none of these works addressed the use of LfD for guidance on-line and for trajectories that depend on contact forces instead of depending on time.

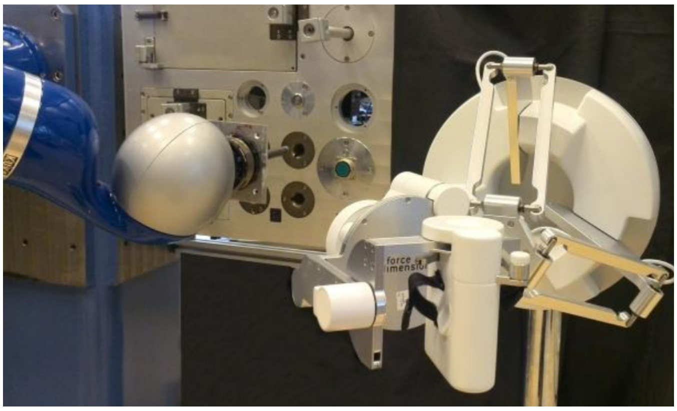


Fig. 1: Experimental setup with KUKA LWR4+ robot (slave) operated by a Sigma 7 haptic device (master). Learning from Demonstration was used to compute on-line haptic guidance trajectories to assist an operator with a 6 d.o.f. peg-in-hole insertion task.

The goal of this paper to develop and apply a Learning from Demonstration based algorithm to generate appropriate guidance trajectories on-line, for force-input-based haptic guidance during contact tasks. The method is experimentally verified with a teleoperation peg-in-hole experiment (the de facto benchmark test for robotic assembly [12]) implemented with the teleoperation setup shown in Fig. 1.

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

## II. Method

The proposed method uses interaction measurements from a slave robot and provides haptic guidance references to a haptic master device during the conducting of a task. For this purpose, a task needs to be divided into a set of steps $e_1, e_2, \ldots, e_p$, whose relationship between interaction measurements (e.g. forces, torques and positions) and haptic guidance references (forces) are different. For example, a task can consist of rotating a peg until it reaches a limit of a latching mechanism ($e_1$) and afterwards, inserting it ($e_2$). Therefore, in $e_1$ the haptic guidance references will be rotational forces until the peg reaches the limit, and longitudinal force references will be generated in $e_2$ until the peg is inserted.

During the demonstration-phase of each step (during overall training) a Gaussian Mixture Model (GMM) is encoded for each step, which is then stored in a library. During actual teleoperation with haptic guidance, this library is then used to generate haptic guidance references on the fly, in real-time, by means of Gaussian Mixture Regression (GMR).

### A. Training stage

In the training stage, a GMM is encoded for each step and stored in a steps library. The training procedure is illustrated in Fig. 2, which consists of 5 sequential processing stages.

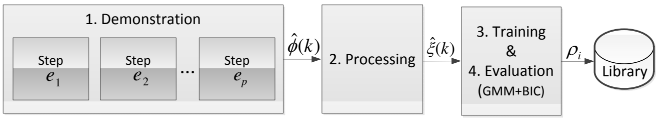


Fig. 2: Training stage data-flow diagram. Each "task-step" (Step $e_p$ ) is independently trained and stored into the library.

*1) Demonstration Stage:* Each task-step (from here-on called 'step') is demonstrated several times (e.g., by kinesthetic teaching) to obtain a sequence of $k$ elements of sensory information $\hat{\phi}(k) = \phi(1), \ldots, \phi(j), \phi(k)$, obtained at every time instant $j$. This sequence is composed of tuples defined as $\phi = (\vec{f}, \vec{\tau}, \vec{p})$, where $\vec{f} = (f_x, f_y, f_z)$ and $\vec{\tau} = (\tau_x, \tau_y, \tau_z)$ represent the manipulator end effector exerted forces and torques respectively (interaction measurements), which can be directly measured by a multiaxial Force/Torque sensor. $\vec{p} = (p_x, p_y, p_z)$ represents the Cartesian position of the manipulator end-effector.

*2) Processing Stage:* The ideal guidance references $\vec{h}(j)$ are obtained from $\vec{p}$ assuming the objective of each step is to reach a concrete position of the manipulator end effector, which depends on the interaction measurements. So, the difference between the final position $\vec{p}(k)$ and the obtained position at each instant in time $j$, $\vec{p}(j)$, is defined as $d(j)$. Using it, an ideal guidance reference $\vec{h} = (h_x, h_y, h_z)$ is calculated for every time instant $j$ as $\vec{h}(j)$, as:

$$\vec{h}(j) = K_H \vec{d}(j), \tag{1}$$

with a vector of spring constants $K_H$ and $\vec{d}(j) = \vec{p}(k) - \vec{p}(j)$. Using this information, a tuple that represents the relationship between interaction measurements and ideal guidance references can be defined as:

$$\xi = \left[\xi^i; \xi^o\right] = \left[\vec{f}; \vec{\tau}; \vec{h}\right] \tag{2}$$

where $\xi^i$ represents the input variables (interaction measurements), and $\xi^o$ represents the output variables (ideal guidance reference). Initially, the dimension of the tuple $\xi$ is $D = 9$ elements if all the interaction measurements and ideal guidance references are used. However, it could be reduced by removing elements with little added information. For example, if a step consists of rotating a peg when vertical forces $f_z$ are exerted, lateral forces $f_x$ and $f_y$ are not needed. Therefore, the tuple dimension could be reduced to $D = 7$ elements.

*3) Training Stage:* Having the tuple $\xi$, a training sequence can be defined as $\hat{\xi}(k) = \xi(1), \xi(2), \ldots, \xi(k)$, which will be used to encode the demonstrated step into a GMM as follows.

A GMM is defined as $\rho = \{\pi_n, \mu_n, \Sigma_n,\}_n^N = 1$, whose parameters are the number of Gaussians $N$, the prior probabilities $\pi_n$, the means $\mu_n$ and the covariance matrices $\Sigma_n$. Using a tuple $\xi$, the probability that this tuple belongs to an encoded GMM is:

$$P(\xi) = \sum_{n=1}^{N} P(n) P(\xi|n) \tag{3}$$

where $P(n)$ is the prior probability and $P(\xi|n)$ is a conditional probability density function, defined as:

$$P(n) = \pi_n, \tag{4}$$

$$P(\xi|n) = N(\xi, \mu_n, \Sigma_n,) \tag{5}$$

Combining Eqs. (3) to (5), the log-likelihood of a GMM for a training sequence is obtained by:

$$L(\hat{\xi}(k)) = \sum_{j=1}^{k} \log\left(P\left(\xi\left(j\right)\right)\right) \tag{6}$$

where $P\left(\xi\left(j\right)\right)$ is obtained by Eq. (3). Therefore, the objective is to find a set of parameters for each GMM that maximizes Eq. (6) for a training sequence $\hat{\xi}(k)$. In this paper, the Expectation-Maximization (EM) algorithm [13] has been used. The algorithm is an iterative method that estimates the parameters that maximize the likelihood of a statistical model based on a training sequence $\hat{\xi}(k)$ and a predefined number of Gaussians $N$. The later parameter is important: if it is too low, the GMM would not sufficiently capture the training sequence, and if it is too high, the computer processing time of the GMR function would be too long, violating the real-time constraints, as will be explained in Section II-B.

*4) Evaluation Stage:* To evaluate how well a GMM has been trained, the Bayesian Information Criterion (BIC) [14] has been used. This method provides a score for different estimated GMM that is calculated as:

$$\text{BIC} = -L(\hat{\xi}(k)) + \frac{n_p}{2} \log(k), \tag{7}$$

where $n_p = (N-1) + N(D + \frac{1}{2}D(D+1))$. The first part of this equation provides the log likelihood of the estimated

GMM for the training sequence, and the second part penalizes the score taking into account the dimension $D$ of the tuple $\xi$ and the number $N$ of Gaussians. In BIC, lower score signifies a better model. The criterion can be used for two purposes: 1) to choose the number of Gaussians $N$; and 2) to perform a comparison between different dimensions $D$ of the tuple $\xi$.

*5) Library Stage:* Finally, once trained and correctly evaluated, the encoded GMM $\rho_i$ for the $i$-th task-step is stored in the steps library that will be used to provide haptic guidance references during the real-time execution of the task.

### B. Generating the Haptic Guidance References

Having obtained the steps library for a specific task, it can be applied to derive the haptic guidance during teleoperation. Fig. 3 shows the data-flow of the proposed method to generate the haptic shared control references. The encoded task-step that is being executed is selected from the library. This selection can be performed in different ways, e.g., using predefined measurement thresholds that provide information of the step that is being carried out, or using machine learning algorithms for gesture recognition [15].

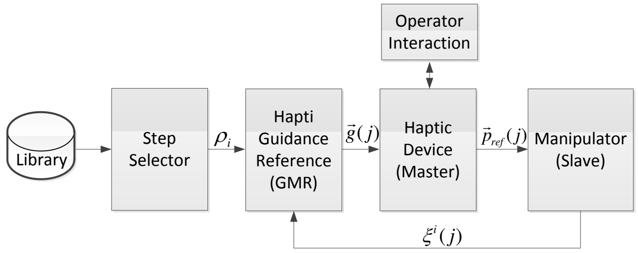


Fig. 3: Haptic guidance reference generation data-flow diagram for the processing of data during real-time teleoperation. The Step Selector selects the GMM of the task-step currently being identified.

Then, GMR is used to obtain the haptic guidance references $\vec{g} = (g_x, g_y, g_z)$ from the detected step $\rho_i$ using the interaction measurements $\xi^i$. For this purpose, the parameters of a GMM can be represented as:

$$\mu_n = \begin{bmatrix}\mu_n^i \\ \mu_n^o\end{bmatrix}, \quad \Sigma_{i,n} = \begin{bmatrix}\Sigma_n^i & \Sigma_n^{io} \\ \Sigma_n^{oi} & \Sigma_n^o\end{bmatrix}, \tag{8}$$

which are used to represent the GMR function in 9. In this equation, $\xi^i$ represents the input variables (interaction measurements), $\vec{g}$ is the estimated output vector (haptic guidance references) and $P(n|\xi^i)$ is the probability of an observed input belonging to each of the Gaussians $n$ as defined as:

$$\vec{g} = \sum_{n=1}^{N} P(n|\xi^i)\left[\mu_n^o + \frac{\Sigma_n^{io}}{\Sigma_n^i}(\xi^i + \mu_n^i)\right] \tag{9}$$

where

$$P(n|\xi^i) = \frac{P(n)P(\xi^i|n)}{\sum_{j=1}^{N} P(j)P(\xi^i|j)}. \tag{10}$$

Once obtained, the haptic guidance references $\vec{g}(j)$ at each instant in time $j$, are transmitted to the operator by the haptic master device. A new end-effector pose reference $p_{\text{ref}}(j)$ is generated as a result, in combination with the operator interaction, which is then sent to the slave robot manipulator.

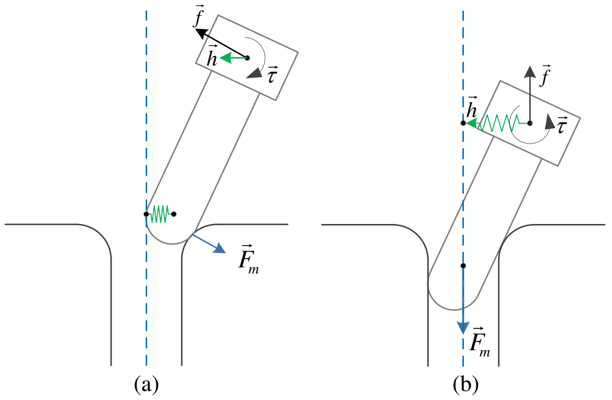
Equations (9) and (10) are used to estimate the haptic guidance references in real-time. This means the execution of these equations must be performed within a time-step of the real-time system. Looking at the scaling, there is one sum in Eq. (9) and another one in the divisor of Eq. (10). Although Eq. (10) is calculated inside the sum in Eq. (9), its divisor can be previously calculated because it is independent of the sum in Eq. (9). Therefore, the algorithm of the GMR scales as $O_{\text{GMR}}(N)$, i.e., the processing time increases linearly with the number of Gaussians.

## III. Experimental verification

### A. Peg-in-hole insertion task

The peg-in-hole insertion task, despite being trivial while manually performed, proved to be relatively challenging while carried out by a robot (both when teleoperated and when carried out by an automatic system) [16].

Firstly, the task is divided into two task-elements (steps), illustrated in Fig. 4, which depend on the geometric relation between peg and hole (one contact point vs. two contact points). In a first step, an operator places the peg at the of hole entrance (only lateral contacts with the hole) and then slides the tip towards the center of the hole (Fig. 4a). The hole surface is pushed with a normal force $\vec{F}_m$. With a rigid peg, a reaction force of the same magnitude is transmitted to the base of the peg as $\vec{f}$. Also, a small torque $\vec{\tau}$ is generated. The peg then continues to move, constrained by one contact point.



(a) Lateral movement (b) Push down movement

Fig. 4: Principal task-elements (steps) during a peg-in-hole insertion: (a) Lateral movement until contact. (b) Push-down movement with two contacts.

In the second task step, once two contacts have been established, but without a correct orientation (Fig. 4b), the operator has to align the peg with the hole by 'pushing' down with a force $\vec{F}_m$. Vertical and opposite lateral forces $\vec{f}$ arise because the peg is locked, and opposite direction torques $\vec{\tau}$

arise because of a leverage. Thus, a rotation of the peg may be performed in order to align it with the hole.

To summarize, forces and torques of different magnitudes and directions are caused, depending on the type of contact with the hole. This means, each configuration also requires a different 'type of movement' that facilitates the insertion. For this reason, the two identified task steps are: (a) surface contact and (b) lever effect.

To simplify the 'step selector' (Fig. 3), the vertical component of the forces exerted on the peg base $f_z$ has been used to define a threshold to switch between these two task-steps.(During (a), the force $f_z$ was observed to be much lower than during (b)).

To obtain the tuple $\xi$ , an analysis of both steps has been carried out in order to reduce the tuple dimension $D$. In the case of surface contact (Fig. 4a), only the force vector can be used because the torques do not provide any relevant information during the reproduction of this step. Moreover, $f_z$ has been removed because it was too low during the movement. Thus, $\xi$ can be defined for the surface contact step as:

$$\xi_{\text{sc}} = [\xi^i_{\text{sc}}; \xi^o_{\text{sc}}] = [f_x, f_y; h_x, h_y]. \tag{11}$$

In the case of the lever effect (Fig. 4b), only the torque vector is used, because forces do not provide any relevant information in this task-step. (Also, $\tau_z$ has be removed). Therefore, $\xi$ can be defined as:

$$\xi_{\text{le}} = [\xi^i_{\text{le}}; \xi^o_{\text{le}}] = [\tau_x, \tau_y; h_x, h_y]. \tag{12}$$

Once these tuples are defined, both training sequences can be obtained from demonstrations, for which the KUKA LWR4+ has been used in gravity assist teaching mode.

### B. Experimental Setup

A KUKA Lightweight Robot (LWR4+) [17] and a Sigma 7 haptic master device [18] have been used in a teleoperation work-cell, both of which were connected to a main computer. Fig. 5 shows the architecture of experimental setups with task-board.

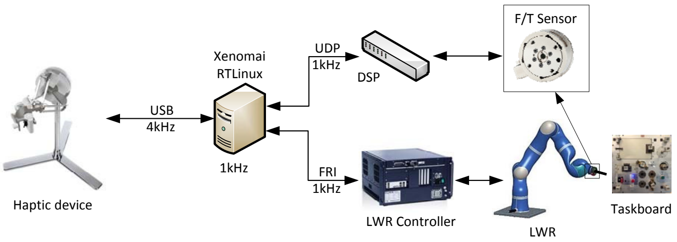
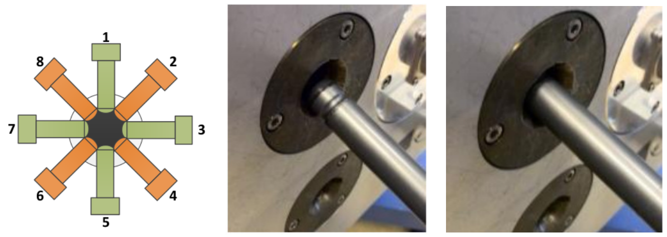


Fig. 5: SW/HW architecture of the teleoperation system.

The Kuka LWR 4+ was used in 'gravity compensated' mode for collecting the demonstration data. This allows the robot to be manually guided through the task by holding the end-effector (behind the Force-Torque (F/T) sensor) and manually perform the multiple peg-in-hole insertions from various initial angles. The Net F/T Gamma SI 65-5 from ATI Industrial Automation sensor measures forces $\vec{f} = (f_x, f_y, f_z)$ and torques $\vec{\tau} = (\tau_x, \tau_y, \tau_z)$, required as input to the proposed method. A 155 mm long titanium peg was rigidly mounted on the distal-side of the F/T sensor attached to the manipulator end-effector. During the 'reproduction experiment', the manipulator was teleoperated using a Sigma 7 haptic device. The main computer was a PC with a real-time Linux based operating system (Xenomai). This computer was used during the training and reproduction stage with 1 kHz sampling rate.

### C. Task learning by demonstration

The LWR was manually guided to demonstrate/teach both steps, which start from eight $45°$ displaced initial positions. Fig. 6a shows an illustrated front view on the hole along with the eight defined initial positions that covered a full circle. Fig. 6b and Fig. 6c show the starting position of the peg for the surface contact task-step and for the lever effect step respectively. Six training movements were carried out for each step and initial position, resulting in $6 \times 2 \times 8 = 96$ insertions being demonstrated.

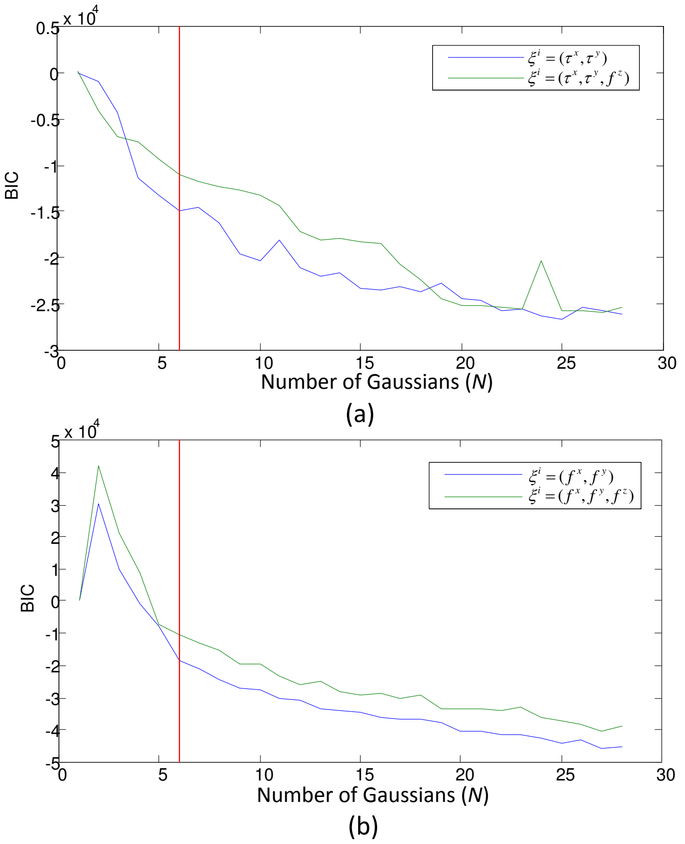


(a) Insertion directions (b) Surface contact (c) Lever effect

Fig. 6: Initial training positions: (a) Eight different insertion positions to cover a full circle over the hole. (b) 'Surface contact' initial position. (c) 'Lever effect' initial position.

Data of these movements was grouped to have only one sensory information sequence for each task-step. These sequences were processed to obtain the training sequences used to encode both GMMs with different number $N$ of Gaussians, and to evaluate them through the BIC as outlined above in Section II-A4. Fig. 7 shows a comparison of the BIC score for different number of Gaussians and also shows how the BIC score is improved when the dimension $D$ of $\xi$ is reduced. In this figure, the blue line represents a tuple $\xi^i$ of two dimensions and the green line represents three dimensions. As expected, a lower dimension of $\xi^i$ improves the score (lower). Specifically, a similar score is obtained using six Gaussians and two dimensions or eleven Gaussians and three dimensions as shown in Fig. 7a. It can be seen from both plots that the BIC score improves as more Gaussians are used. However, as stated in Section II-B, the GMR processing time increases linearly with the number of Gaussians. Due to the processing time constraint of the real-time system (1 ms), several tests were carried out, and the maximum number of Gaussians that the main computer was able to execute in real-time was $N = 6$ (red line in Fig. 7).

In order to verify the newly obtained models, the best encoded GMMs were selected and compared offline with newly recorded validation data. For this purpose, $8 \times 2 = 16$ new movements were performed, one for each initial position



(a)

(b)

Fig. 7: BIC scores for different gestures using two different training tuples. Lower values mark better model fit.

and task-step. These new movements were used to compare the difference between the processed ideal guidance references and the obtained haptic guidance references through GMR, calculated by the Root-Mean-Square (RMS) as:

$$E_{\text{RMS}} = \sqrt{\frac{\sum_{i=1}^{M} \sum_{j=1}^{k} \left( \vec{g}_i(j) - \vec{h}_i(j) \right)^2}{\sum_{i=1}^{M} k_i}}, \tag{13}$$

where $\vec{g}_i(j)$ and $\vec{h}_i(j)$ represent the haptic and ideal guidance references respectively, calculated for the insertion $i$ and $k_i$ number of elements within the validation sequence, over all time instants $j$.

Fig. 8 shows the RMS error for each axis of the guidance references. The RMS error is lower than 0.28N during the lever effect task-step and lowest with 0.2N during surface contact, which appears to be showing a good fit.

### D. Task reproduction for haptic shared control

To validate the guidance trajectory reproduction, the Kuka LWR slave manipulator was configured in compliance mode with a Cartesian stiffness of $k_{\text{trans}} = 500\,\text{N/m}$ and $k_{\text{rot}} = 50\,\text{Nm/rad}$, for translations and rotations, respectively. These parameters allowed the operator to teleoperate the robot without exerting excessive forces on the taskboard. Moreover, the vertical forces threshold, used to choose the task-step that is being performed, was fixed to 5 N.

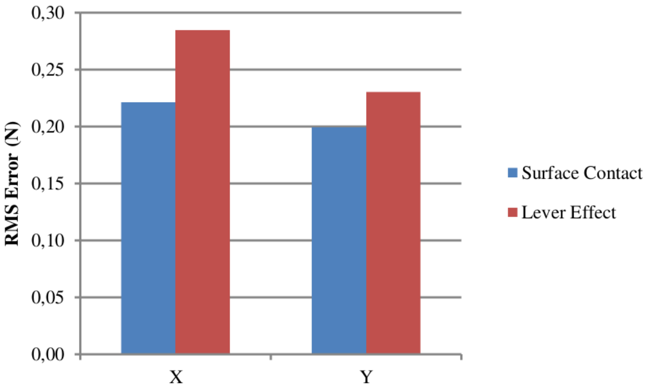


Fig. 8: RMS Error between the ideal and haptic guidance references for the performed insertions, evaluated in a validation dataset with 16 insertions.

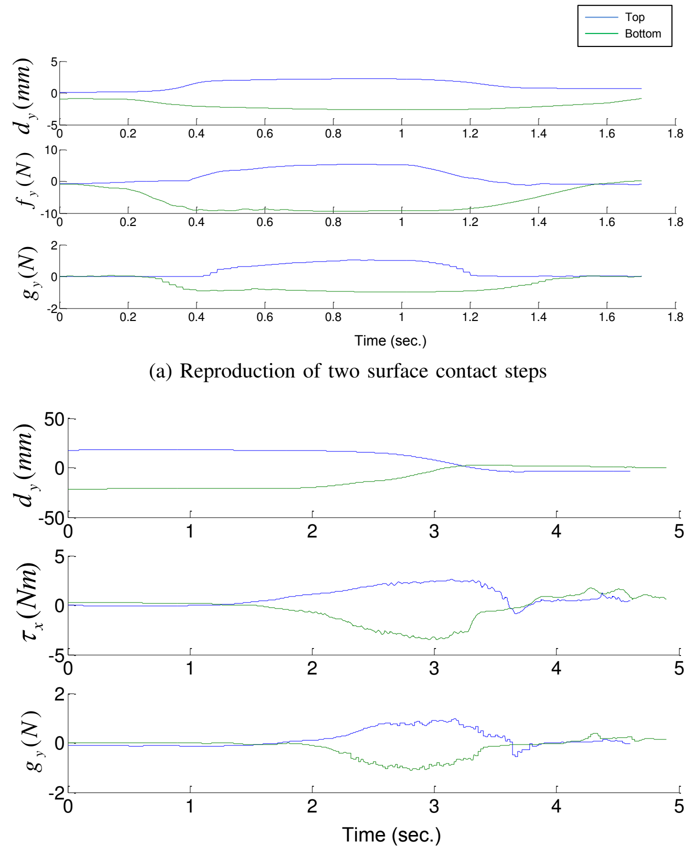
The operator used the Sigma 7 haptic master device to insert the peg into the hole on the taskboard, receiving haptic guidance forces and controlling full six degrees of freedom. To keep the focus mainly on the actions of the guidance system, the measured force feedback loop was disconnected, and the operator had to rely only on the aid of the GMR obtained haptic guidance references. Using this configuration, each task-step was teleoperated, starting from positions 1 and 5 in Fig. 6a, i.e. vertical movements to insert the peg. Fig. 9 shows the measured information where only the applicable axes are represented. The surface contact task-step results are shown in Fig. 9a. $d_y$ indicates the difference between the current position of the manipulators end-effector and the final one that correctly places the peg tip in the hole, $f_y$ represents the vertical interaction forces, measured by the F/T sensor, which is used to provide haptic guidance references $g_y$ that reduces the error during the teleoperation. During the guidance display in the lever task-step, illustrated in Fig. 9b, the situation is similar. Here, $d_y$ represents the difference between the current position of the end effector and the final one that aligns the peg with the hole. In this step, the torque $\tau_x$ is the interaction measurement that is used to generate the vertical haptic guidance references $g_y$ that align the peg with the hole, reducing the mentioned error as shown in the figure.

Finally, in the attached video[1], the operator was blindfolded, and his task was to provide the forward motion (i.e., into the hole) and follow the guidance force references. The peg-in-hole insertion was carried out twice by the operator to compare the use of the proposed approach with direct haptic teleoperation.

## IV. Conclusions

This paper demonstrates the feasibility of a new method for generating force-based haptic guidance trajectories that uses interaction information (i.e., forces and torques) to generate haptic guidance references that assist an operator to solve a task through teleoperation. The fundamental concept underlying the proposed method is the use of Learning from Demonstration through multiple demonstrations, which can

[1] Available online: http://tiny.cc/LfDHapticGuidanceSMC2016



(a) Reproduction of two surface contact steps

(b) Reproduction of two lever effect steps

Fig. 9: Example of peg-in-hole insertions from positions 1 and 5 (in Fig. 6a). The first one was carried out starting at the top of the hole (green line) and the second one at the bottom of the hole (blue line).

generate haptic guidance references in real-time and on-line during telemanipulation.

The proposed method has been verified with an peg-in-hole insertion experiment. It is shown that 96 demonstrated insertions are sufficient to obtain a good model of the task (RMS error of predicted guidance force on a validation dataset bellow 0.28 N) that allows to generate guidance independently from a time-base. The task was broken down into two task-steps that were modeled using GMMs with 6 Gaussians each, which allows generating haptic guidance references in real-time with a sampling rate of 1 kHz.

However, the method to decide the task-step that is being performed on-line had been simplified in this experiment. A more generic detector, e.g., based on gesture recognition, would be needed for more complex tasks. So far, the proposed approach is only tested on one task (peg-in-hole insertion). Therefore, the next step will be to confirm how this approach applies to different tasks and to perform user studies.

## References

[1] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control.* MIT Press, Aug. 1992.

[2] M. K. O'Malley, A. Gupta, M. Gen, and Y. Li, "Shared Control in Haptic Systems for Performance Enhancement and Training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, p. 75, 2006.

[3] H. Boessenkool, D. Abbink, C. Heemskerk, F. Van der Helm, and J. Wildenbeest, "A Task-Specific Analysis of the Benefit of Haptic Shared Control During Tele-Manipulation," *IEEE Transactions on Haptics*, vol. PP, no. 99, p. 1, 2012.

[4] E. Nuno and L. Basanez, "Haptic guidance with force feedback to assist teleoperation systems via high speed networks," *VDI Berichte*, vol. 1956, p. 213, 2006.

[5] A. Chowriappa, R. Wirz, A. R. Ashammagari, and Y. W. Seo, "Prediction from expert demonstrations for safe tele-surgery," *International Journal of Automation and Computing*, vol. 10, no. 6, pp. 487–497, Dec. 2013.

[6] P. G. Griffiths and R. B. Gillespie, "Sharing Control Between Humans and Automation Using Haptic Interface: primary and secondary task performance benefits," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 47, no. 3, pp. 574–590, 2005.

[7] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research.* Springer, 2007, pp. 49–64.

[8] J. Smisek, M. M. van Paassen, and A. Schiele, "Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy," in *World Haptics Conference (WHC), 2015 IEEE.* IEEE, 2015, pp. 500–505.

[9] R. Jäkel, S. R. Schmidt-Rohr, M. Lösch, and R. Dillmann, "Representation and constrained planning of manipulation strategies in the context of Programming by Demonstration," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 162–169.

[10] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and Reproduction of Gestures by Imitation," *IEEE Robotics Automation Magazine*, vol. 17, no. 2, pp. 44–54, Jun. 2010.

[11] L. Rozo, P. Jiménez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intelligent Service Robotics*, vol. 6, no. 1, pp. 33–51, Jan. 2013.

[12] B. J. Unger, A. Nicolaidis, P. J. Berkelman, A. Thompson, R. L. Klatzky, and R. L. Hollis, "Comparison of 3-D haptic peg-in-hole tasks in real and virtual environments," in *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings*, vol. 3, 2001, pp. 1751–1756 vol.3.

[13] J. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," Tech. Rep., 1998.

[14] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, 1998*, vol. 2, May 1998, pp. 645–648 vol.2.

[15] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.

[16] S. R. Chhatpar and M. S. Branicky, "Search strategies for peg-in-hole assemblies with position uncertainty," in *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings*, vol. 3, 2001, pp. 1465–1470 vol.3.

[17] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing," *German Conference on Robotics*, pp. 1 –8, 2010.

[18] A. Tobergte, P. Helmer, U. Hagn, P. Rouiller, S. Thielmann, S. Grange, A. Albu-Schäffer, F. Conti, and G. Hirzinger, "The sigma.7 haptic interface for MiroSurge: A new bi-manual surgical console," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 3023–3030.