Proceedings
ISBN 978–952-5183-54-2

Learning compliant assembly skills from human demonstration

Markku Suomalainen and Ville Kyrki

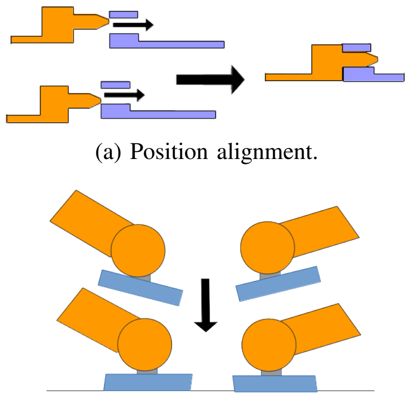
***Abstract*—Robotic assembly is mainly used inside factories where both the environment and the task for each robot stays constant and the batch sizes are large, with car factories presenting a prime example. However, in manufacturing Small and Medium-sized Enterprises (SMEs) or construction yards the level of automation is very low, mainly due to the changing environment causing two major problems for robots: firstly, the programming of robots is often difficult and thus it can take too long to make the same robot perform multiple tasks interchangeably. Secondly, the use of robots with traditional control methods requires an accurate model of the environment, which can be either costly to acquire and prone to accidental changes in the real environments (SMEs) or simply infeasible (construction). To enable the use of robots in new environments, robots must be easy to teach and able to adapt to small changes in the environment. In this paper we propose methods to use Learning from Demonstration (LfD) with compliant motions to facilitate the usage of robots in new environments.**

I. INTRODUCTION

The strenuousness of programming a robot to perform different tasks is a major reason holding back the widespread use of robots in industry and at people's homes. Industrial robots are mainly used only when the same product is manufactured for long periods of times. One of the next places where the usage of robots can really increase is enterprises where production batches can be small. But to enable this step, domain experts must be able to teach the robots the required task, such that a robotics expert is not required at the stage every time the robot needs to learn a new task or fails at completing a taught task.

Learning from demonstration (LfD) is an established paradigm in robotics, where the goal is easily programmable robots. In short, the idea is to show the robot an example of a skill, which the robot learns to reproduce and generalize into other locations and similar situations. Methods to show an example include *e.g.* kinesthetic teaching (holding a gravity-compensated robot and leading it through the motions) and teleoperation. However, traditional LfD techniques struggle with compliant motions, which are required in many industrial assembly tasks.

In this paper we propose to use LfD with compliant motions to overcome the aforementioned problems. In LfD the user can show the robot how to perform a required task, using either teleoperation or kinesthetic teaching where the teacher physically holds a gravity-compensated robot and leads it through the desired task. We developed methods to ease the use of compliance on three different levels in programming a robot: on the control level, on the primitive level and on the motion sequencing level. On the control level, we propose using impedance control for cases where both the manipulator and object are ground based. On the primitive level we present a new impedance control– based motion primitive which can be used to learn and encode motions that use the environment to mitigate pose uncertainties– humans naturally have the skill to exploit contact forces in insertion tasks, and we want to convey the skill from human to robot in an efficient way. On the motion sequencing level we first show how a complex human demonstration can be segmented into phases, each of which can be modelled with the primitive. Then we present how the primitives can be sequenced online to successfully reproduce the task. Additionally, we show that the presented motion primitive can also be applied effectively for bimanual assembly tasks. Finally, we present how to learn from human teachers search motions similarly as a human inserting a plug into a socket in darkness, which can be used as efficient exception strategies in assembly. To conclude, this paper presents a framework that can accelerate the degree of automation in tasks where currently the use of robots is infeasible.



(a) Position alignment.

(b) Orientation alignment

Fig. 1: Compliant motions can be used for aligning both position and orientation of a workpiece [1]

This paper is an overview of seven publications by the authors following the aforementioned paradigms, and the mathematical details of the methods can be found from those publications. In [2], we learned desired direction and axes of compliance for a motion by assuming we can directly measure the direction of the force which the human teacher applies to the robot in kinesthetic teaching. However, we noticed that this assumption only holds true for certain force/torque sensor configurations, and hence we wanted to solve also the more general problem, where we can only measure the force between the end-effector and the environment. We solved this problem in [3], with the observation that in a compliant sliding motion there is always a certain sector of directions from which the robot can apply force to perform the observed motion. We managed to take the intersection of these sectors in a 3-D motion, over one or more demonstrations, and thus learn the parameters for a dynamically linear compliant motion. In [1] we generalized the task to work with rotational motions as well. Furthermore, in [4] we learned how to sequence these motions to perform a full task, such as pipeline assembly. To make the robots more independent even in case of changes in the environment, in [5] we looked into whether a robot could learn to search using contact forces, similarly as a human tries to fit a key into the keyhole in darkness. Finally, in [6] we showed that our method can be applied to dual-arm tasks and examined the role of compliance in dual-arm assembly with a little more detail. Additionally, to show that the method from [3] is robust enough to work with systems where errors in measurements can be higher, we combined the method with a stability-guaranteed Virtual Decomposition Control- based impedance controller for a heavy-duty hydraulic manipulator with a 475kg payload [7].

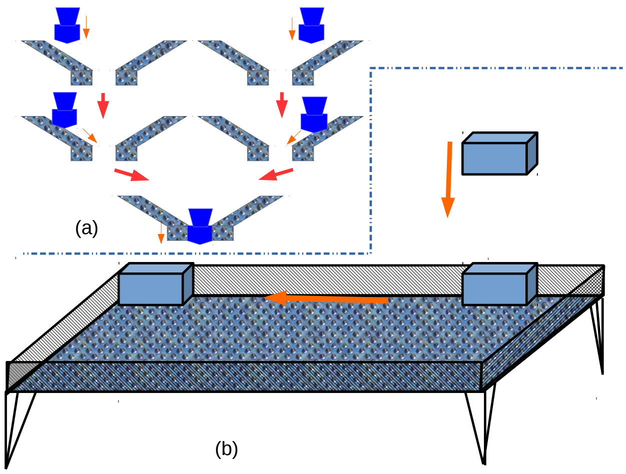


Fig. 2: Compliant motion policy used (a) to align workpieces and (b) to place a box at the corner of the table [4].

II. METHOD

To make a robot execute a task, the task must be represented in a manner that is understandable for the robot, often called a *policy* consisting of *primitives* each of which models a *phase* of the task, such as shown in Fig. 2b where moving the block to touch the table is one phase and moving it into the corner is another. In this chapter we consider modelling and learning from a human demonstration a single phase of such a policy. This is the *continuous* level of a hierarchical policy, and the simplest examples of this sort of behaviour for modelling a trajectory are splines [8] or Bezier curves [9]. If the task requires contact with the environment, the simplest approach is to augment the trajectory with a *force profile* consisting of forces the robot should apply to the environment at each position. The obvious downside of this kind of representation is that even small changes in the environment or in the robot's coordinate system can easily cause the task to fail. Thus simply recording the trajectory and forces from a human demonstration and replaying them is not a valid LfD strategy. There exist more general and popular primitives than the simple trajectory encoders mentioned that are used nowadays to represent tasks learned from human demonstrations. However, especially when trying to learn how to take advantage of the environment with compliant motions, there are certain downsides in the currently popular motion primitives.

Perhaps the most recognized primitives used currently in LfD are Dynamic Movement Primitives (DMP) [10] and Gaussian Mixture Model (GMM) with either Gaussian Mixture Regression (GMR) [11] or Stable Estimator of Dynamical Systems (SEDS) [12]. Strengths of the DMPs include the ability to be learned online and that they can be coupled with wrench or impedance profiles. These attributes, along with the simplicity, make DMPs a popular choice for learning and encoding complicated trajectories. Additionally, with correctly chosen gains DMPs can be shown to be stable, and DMPs have been shown to be generalizable through *task-parametrization* to new situations [13], [14] by simply modifying a parameter relevant to the current task. A downside of DMPs is that to learn from multiple demonstrations, tools such as Dynamic Time Warping (DTW) [15] need to be used to temporally align the demonstrations. Especially with more than two demonstrations this becomes a tedious task.

Similarly to DMPs, GMMs can be task-parametrized and augmented with a wrench profile. SEDS uses GMMs as well to model the task, but due to the use of a dynamical system, the stability can be guaranteed with correctly chosen parameters, unlike when using the statistical methods in GMR. SEDS has also been used to produce impedance to allow compliant motions [16]. The main downside of all these methods when used for compliant motions is the tight coupling of the trajectory and force profile, which makes the methods susceptible to pose uncertainties and makes taking full advantage of compliant difficult. Thus we propose new methods to efficiently learn robust compliant skills from human demonstrations.

A. Linear Motion with Compliance

In this section we present the approach called Linear Motion with Compliance (LMC), which was gradually developed in [1]–[3] and used successfully as a component in [4], [6], [7]. The key idea is that we model a task as a sequence of linear motions with compliance such that we take advantage of the environment to guide and align the tool, as shown in Figs. 2 and 1. We assume that many workpieces have a mechanical gradient such as depicted in Fig. 3, which can be used to guide workpieces into alignment. The problem we address is the following: how to learn from human demonstration a task such that the *convergence region*, *i.e.* the set of starting poses from which alignment with a same set of parameters is successful, is maximized. Thus the uncertainties related to the relative pose between the workpieces to be aligned can be efficiently mitigated. Such uncertainties can rise from, for example, small modifications to the environment or simply the uncertainty of grasping an object. To clarify this even further, let us consider the situation in Fig. 3 and assume that the goal is to slide the

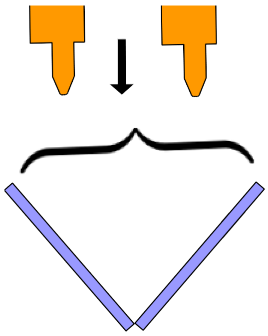


Fig. 3: Illustration of the theoretical convergence region (black brace) of the algorithm in a pure translational case [1].

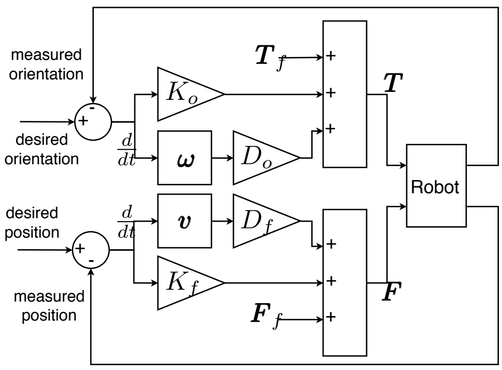


Fig. 4: An impedance controller with F/T feed-forward is used to reproduce the search motions [5].

tool to the bottom of the valley. The error in translation is the horizontal difference between the orange tools tip and the center bottom point of the valley. Now, if the tool is sliding directly downwards from anywhere within the convergence region and the tool is compliant perpendicular to this motion, it will slide along the surface all the way to the bottom.

We use impedance control to reproduce the motions. Impedance control is defined as

$$\begin{aligned} \boldsymbol{F} &= K_f(\boldsymbol{x}_d - \boldsymbol{x}) + D_f\boldsymbol{v} + \boldsymbol{F}_f \\ \boldsymbol{T} &= K_o(\boldsymbol{\beta}_d - \boldsymbol{\beta}) + D_o\boldsymbol{\omega} + \boldsymbol{T}_f \end{aligned} \tag{1}$$

where $\boldsymbol{F},\boldsymbol{T}$ are the force and torque used to control the robot, $\boldsymbol{x}_d$ the desired position, $\boldsymbol{x}$ the current position, $\boldsymbol{\beta}_d$ the desired orientation, $\boldsymbol{\beta}$ the current orientation, K_f and K_o stiffness matrices and $D_f\boldsymbol{v}$ and $D_o\boldsymbol{\omega}$ linear damping terms. Parameters $\boldsymbol{F}_f$ and $\boldsymbol{T}_f$ are the superposed feed-forward force and torque, which can be used if additional implicit force on top of the standard impedance controller is required. The block diagram of the controller is shown in Fig. 4.

Thus, now if we analyse (1), we see two parameters on both translations and rotations that must be adjusted to provide this sort of behaviour: the stiffness matrices K_f, K_o and the desired position and orientation $\boldsymbol{x}_d, \boldsymbol{\beta}_d$. As observed, with efficient exploitation of compliance linear motions suffice to achieve assembly goals in scenarios such as in Fig. 3: thus, we can write the the desired position $\boldsymbol{x}_d$ and orientation $\boldsymbol{\beta}_d$ in a feed-forward manner as

$$\begin{aligned} \boldsymbol{x}_{d,t} &= \boldsymbol{x}_{d,t-1} + \nu\Delta t\hat{\boldsymbol{v}}^*_{\boldsymbol{d}} \\ \boldsymbol{\beta}_d, t &= \boldsymbol{\beta}_{d,t-1} + \lambda\Delta t\hat{\boldsymbol{\omega}}^*_{\boldsymbol{d}} \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{v}}^*_{\boldsymbol{d}}$ and $\hat{\boldsymbol{\omega}}^*_{\boldsymbol{d}}$ are the desired directions describing the human teachers intended motion in translation and rotation, Δt the sample time of the control loop and ν and λ the translational and rotational speeds.

B. Learning the LMC primitive

In this section we explain the process of learning the parameters $\hat{\boldsymbol{v}}^*_{\boldsymbol{d}}$,$\hat{\boldsymbol{\omega}}^*_{\boldsymbol{d}}$, K_f and K_o such that the conditions described in the previous section are met. The data required for learning are the measured Cartesian poses of the end-effector and the corresponding wrenches measured by the Force-Torque sensor (F/T sensor) according to Fig. 5 from a human demonstration.

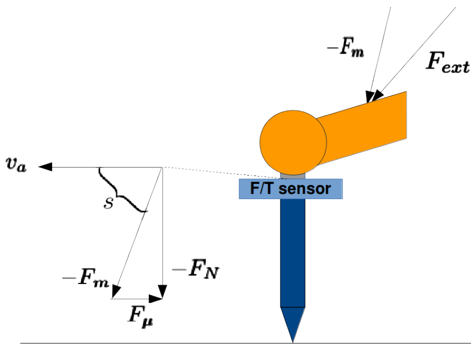


Fig. 5: Forces recorded by an F/T sensor when sliding along a surface. v_a is the actual velocity, F_{ext} the force applied by the teacher and s the sector of desired directions [1].

A general flow of the learning process is shown in Fig. 6: the pitch, *i.e.* the relation between mean translational speed of the demonstration ν and rotational speed λ, must be computed first from the raw data. After this, the same steps are taken for both translational and rotational motions: the first thing is to check if the teacher keeps either translations or rotations 3-Degree of Freedom (DoF) compliant, *i.e.* either position or orientation can change freely without affecting the execution of the task. Whenever this is not the case, the next thing is to check whether the teacher intentionally translated or rotated the tool, *i.e.* check the existence of $\hat{\boldsymbol{v}}^*_{\boldsymbol{d}}$ and $\hat{\boldsymbol{\omega}}^*_{\boldsymbol{d}}$. If either or both exist, the compliant axes must be deduced such that they are perpendicular to the desired direction, otherwise the compliant axes can be directly deduced from the data. Finally, if both $\hat{\boldsymbol{v}}^*_{\boldsymbol{d}}$ and $\hat{\boldsymbol{\omega}}^*_{\boldsymbol{d}}$ exist, ν and λ must be set according to the pitch, finalizing the LMC primitive.

The intuition into 3-DoF compliance can be observed from Fig. 7: the teacher tries to only rotate the tool, but due to contact forces, rotational force causes also translation in the wrist. In this case observed translation is caused completely by the environment and the corresponding degrees of freedom need to be set compliant (i.e. 3-DOF compliance). To numerically

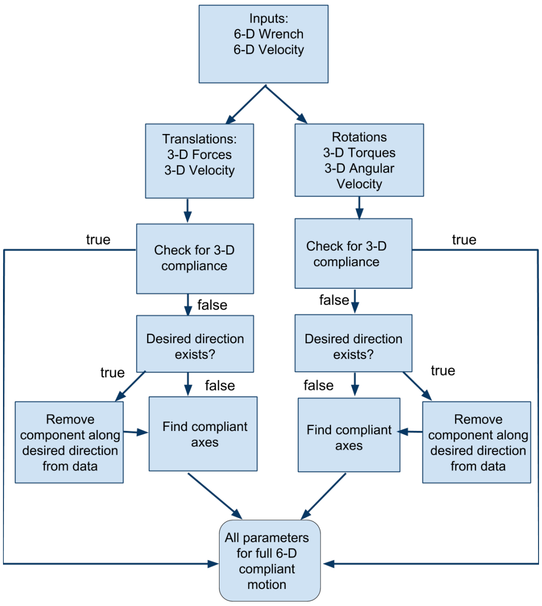


Fig. 6: A flowchart describing the whole process of finding the 6-D compliant primitive to reproduce a demonstrated motion [1].

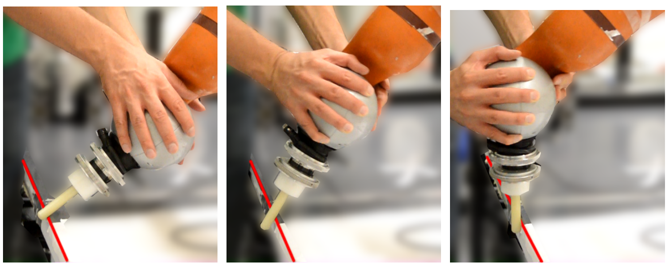


Fig. 7: A demonstration of rotating the peg around the edge of the table, where the teacher only rotates the tool and the translational motion in the wrist is caused by the contact forces. The edge of the table is highlighted in red [1]].

detect this, we looked whether more *work* was done by the environment or the teacher, where work in physics is defined as

$$\begin{aligned} W_x &= \boldsymbol{F}_m \cdot \Delta \boldsymbol{x} \\ W_\beta &= \boldsymbol{T}_m \cdot \Delta \beta \end{aligned} \tag{3}$$

where W is the work, $\Delta \boldsymbol{x}$ the change in translation, $\Delta\beta$ the change in angle and $\boldsymbol{F}_m$ and $\boldsymbol{T}_m$ the force and torque measured by the F/T sensor. The idea is that when the work measured at an interval is positive, the environment has done the work since the forces and torques are caused by the environment. By comparing the amount of positive and negative work measured we can deduce whether the teacher or the environment did more work to move the tool.

The idea for learning the desired direction stems from Fig. 5, where the forces acting on a tool sliding along a surface are shown. The key is to observe sector s, which marks the 2-D sector between the actual direction of motion $\boldsymbol{v_a}$ and the

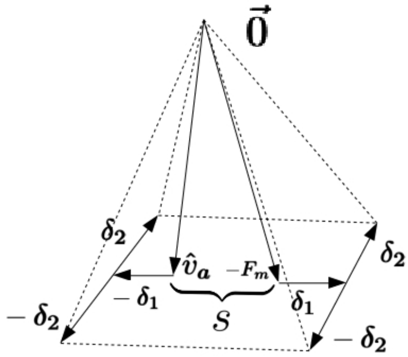


Fig. 8: Illustration of expanding 2-D sector s from Fig. 5 for translations into 3-D set of directions. Continuous lines represent the vectors and dotted lines highlight the pyramid shape [1].

negative of the force measured by the F/T sensor, $-\boldsymbol{F_m}$. Firstly, we observe that the width of this sector depends on the friction force $\boldsymbol{F_\mu}$ such that with higher friction force the sector s becomes more narrow. Secondly, if the external force, $\boldsymbol{F_{ext}}$ is applied from anywhere within sector s, the observed motion will be exactly the same, along the direction of $\boldsymbol{v_a}$. This gives us, at every time instant, a range of directions from which the robot can apply force to achieve a certain motion. We hypothesize that by computing an intersection of these sectors s during a demonstration consisting of sliding motions, we can find the desired direction $\hat{\boldsymbol{v}}_{\boldsymbol{d}}^{*}$ where the human intends to push the tool, and the same logic can be applied to rotations to find $\hat{\boldsymbol{\omega}}_{\boldsymbol{d}}^{*}$.

To transfer this intuition into 3-D, several steps are required. Firstly, there is always uncertainty in a humans demonstration, even if the teacher tries to draw a straight line. Thus, there is a risk that taking an intersection results in an empty set. In Fig. 8 is shown how we propose extending the sector s from Fig. 5 into a pyramid shape in 3-D. These pyramid shapes are then projected into 2-D and an intersection is calculated with suitable outlier rejection. If more than one demonstrations are supplied, the intersection is simply computed from a concatenation of the demonstrations.

The next step in Fig. 6 is finding the compliant axes. The first question in this process is, separately for translations and rotations, whether the desired direction was observed. If it was, this already reduces the dimensionality of the possible compliant axes by one, since the compliant axes must be orthogonal to desired direction as per Condition **??**. The key idea for finding the compliant axes for reproducing the observed motion from the remaining DoFs is the following: if motion along other directions besides the desired direction was observed, it must have been caused by the environment, and thus compliance is required to replicate the motion. Without a desired direction, we assume that all motion is caused by the environment. Thus, when the desired direction exists, we first subtract it from the raw motion data before advancing, and then the compliant axes are computed similarly for the cases where desired direction exists and where it does not. To compute comparable values, we take inspiration from the Bayesian Information Criterion (BIC) [17] to choose the correct number of compliant axes.

C. *Learning search motions*

In Section II-A we made the assumption that there is a physical gradient, such as a chamfer, that can guide the robot's tool into the goal pose. However, it is often the case that this sort of guidance is not available and the environment surrounding the goal pose cannot be exploited to mitigate uncertainties. Even in such a case a human still can have task-dependent intuition of how to efficiently locate the goal– a human might, for example, use a different strategy for fitting a key into a lock than inserting an electric plug into a socket. It would be highly useful if this sort of task-dependent information could be directly conveyed to the robot from a human demonstration– in industry this sort of search could be used as an *exception strategy* in case of a failed task due to error in the environment model.

The existing work on exception strategies for assembly tasks is very limited. Abu-Dakka *et al.* [18] used random walk in case an assembly task failed and searching had to be done. Jasim *et al.* [19] used an Archimedean spiral, which is guaranteed to find the goal with the correct resolution and starting position. However, the spiral is limited to 2-D case and randomness is something to be used as a baseline for better methods. Kronander [20]Chapter 5 used incremental learning, where the human assists the robot during insertion if the robot gets stuck. However, none of these methods took advantage of any intuition a human may have on the tasks at hand.

The approach in this paper is to not assume any contact that can help guide the search, either for guiding or localization purposes, meaning that no earlier experience with this particular plug pose is expected and no visual or auditory sensor input is allowed. Gathering such demonstrations from a human is non-trivial, and we managed this by blindfolding the teacher and varying the relative pose between the start of the demonstrations and the goal. Essentially, there are two things we can learn from the human from such a demonstration without further assumptions: the area in which to search, and the dynamics of the in-contact motions. We call the area to search the *exploration distribution*, which we learn by fitting a Gaussian to the teacher's search trajectory. As the proposed method is for use in environments where the location information can be erroneous, we regard the information conveyed by the exploration distribution and recorded forces as more important than the starting location of the search. Thus we set each search into a common coordinate frame, which in this thesis is called the *search frame*. Furthermore we choose to align the demonstrations based on their origin, not the goal, even though in the world coordinate system they share the goal but not the origin. With this choice we can better learn the *search* strategy of the teacher in situations where localizing the tool w.r.t the world frame is impossible.

D. *Learning Sequence of Motions*

In this chapter we address segmenting a demonstrations into phases, each of which is used to learn a single LMC primitive, and then during execution a correct primitive must be chosen at a correct moment– a single primitive is often insufficient to encode a whole task, and thus the primitives must be *sequenced* to be useful in real-life tasks. We show that LMC primitives can be successfully combined to perform assembly skills, such as attaching hose couplers together. As LMC depends neither on time nor pose, the length of a single primitive does not need to be known beforehand, which brings more flexibility and error tolerance– however, the price of that flexibility is that an additional algorithm is required for learning to detect these changes.

Intuitive approaches for segmenting human demonstrations into phases are often simple heuristics, such as Zero-Velocity Crossing (ZVC) (*i.e.* a change of direction in velocity) or threshold values in contact force. However, this sort of simple heuristics are often not error-tolerant and have to be manually designed for each task. At the other end of spectrum in complexity are usage of multimodal inputs including vision, which can then yield impressive results in, for example, success detection in screwing [21]. However, we use strictly pose and wrench signals acquired from a demonstration since we do not want to depend on vision due to challenges in occlusion and accurate detection of contact.

In this paper the goal is to segment synchronized pose and wrench recordings from human demonstrating an assembly task into segments, each consisting of a single LMC primitive presented in Section II-A. Equations (2) and (1) defining LMC inspired us to model the state dynamics of a single phase by a linear Gaussian model, in which the next state depends on current state $\mathbf{s}$ (pose or a subset), measured interaction (wrench or a subset) $\mathbf{a}$ concatenated with value 1, and current phase $\boldsymbol{\rho}$, where each phase consists of a single LMC– an example of what a phase can look like is seen in Fig. 2. Thus, we write $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t, \boldsymbol{\rho}_t)$. The distribution of the next state is then

$$\mathbf{s}_{t+1} \sim \mathcal{N}\left(A_{\boldsymbol{\rho}_t}\mathbf{s}_t + B_{\boldsymbol{\rho}_t}\mathbf{a}_t, \Sigma_{\boldsymbol{\rho}_t}\right) \tag{4}$$

where $A_{\boldsymbol{\rho}_t} \in \mathbb{R}^{m\times m}$ represents the uncontrolled dynamics, $B_{\boldsymbol{\rho}_t} \in \mathbb{R}^{m\times d}$ models compliance through interaction forces and constant offset velocity through the concatenated 1, and $\Sigma_{\boldsymbol{\rho}_t} \in \mathbb{R}^{m\times m}$ is the covariance matrix corresponding to phase $\boldsymbol{\rho}_t$. Thus, the model assumes linear system dynamics and $B_{\boldsymbol{\rho}_t}$ is used to model contact interaction effects and constant desired direction of motion. This model is used for detecting the correct phase, but the actual control is performed with LMC.

An inspiration for the segmenting approach of this work was the work of Kroemer *et al.* [22], who use an autoregressive state space model together with an Hidden Markov Model (HMM) to segment a demonstration: essentially, their idea is that each phase depends on the previous phase and previous state, *i.e.* $p(\boldsymbol{\rho}_t|\mathbf{s}_t, \boldsymbol{\rho}_{t-1})$. However, as we are modeling compliant in-contact motions while assuming pose uncertainties, depending on the pose for phase changes is not desirable. Thus, we use the graphical model depicted in Fig. 9 with $p(\boldsymbol{\rho}_t|\mathbf{a}_t, \boldsymbol{\rho}_{t-1})$ such that each phase depends on the previous phase and previous action to better take into account the compliant nature of the task being modelled. For learning the model, we adapt the Expectation-Maximization (EM) algorithm from [22] to learn from multiple demonstrations the model parameters $\theta = (w, A, B, \Sigma)$. During reproduction, we take advantage of the joint probabilities to choose the correct primitive.

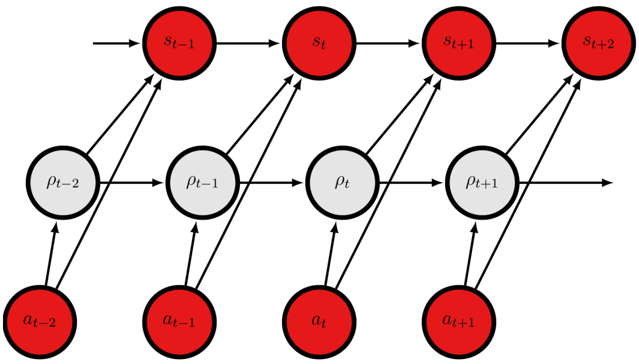


Fig. 9: The graphical model we used for segmenting the demonstration [4].

III. EXPERIMENTS AND RESULTS

We tested our approaches on various tasks and setups. On the hose-coupler setup in Fig. 12 we performed experiments on the LMC primitive and combining primitives. On the peg-in-hole setup shown in Fig. 15 we tried the LMC primitive both on single and dual arms and also the search. Additionally, we performed search experiments on a plug-and-socket setup shown in Fig. 10 and used LMC with a heavy-duty hydraulic manipulator shown in Fig. 11.

In the hose-coupler setup we defined both the Tool Center Point (TCP) and the Center of Compliance (CoC) in the flange of the robot to achieve rotational compliance around the flange and to observe the translations occurring at the flange when the orientation of the tool changes. In this task there is a high likelihood of orientation error when commencing the task due to difficulties in pose estimation, with examples shown in Fig. 12. We showed that with one desired direction and a correctly identified stiffness matrix the hose couplers can be aligned with the same set of parameters starting from both Figs. 12a and b and ending up in Fig. 12c after two demonstrations from different starting positions.

In the hose-couple alignment task, for translations a desired direction is found, but for rotations it is not– visualization of the rectangles representing the limits of desired direction at each time interval are shown in Fig. 13, where the red rectangles are from a demonstration starting from the pose of Fig. 12a and the blue ones from Fig. 12b. It can be observed that for translations the rectangles from both demonstrations are aligned, but for rotations the two demonstrations are clearly

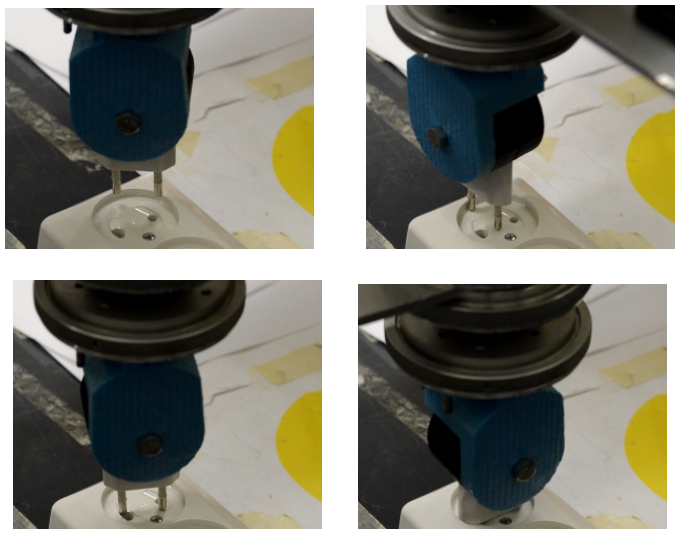


Fig. 10: An example sequence of a robot inserting a plug into a socket without vision sensing [5].

separate, leading to the conclusion that a desired direction for translations is required but for rotations there is not a desired direction.

Finding the number of compliant axes is visualized in Fig. 14, where each blue cross represents the mean directions of motion of a demonstration and the red axes are the axes of Principal Component Analysis (PCA) performed on all mean directions of a demonstration. Since for translations there exists a desired direction, it is plotted in cyan (overlapping the first PCA axis, as expected) and subtracted from the mean direction of motion data, *i.e.* the blue crosses are projected into the plane of the other principal axes, resulting in the green crosses. Now it can be observed that one of the principal components connects the green crosses, thus explaining the observations and resulting in choosing one compliant axis along that component. As the TCP was set in the flange, translation is required to perform the alignment. For rotations, the analysis is done directly on the PCA data, as there is no desired direction. It can be observed that the rotations are close

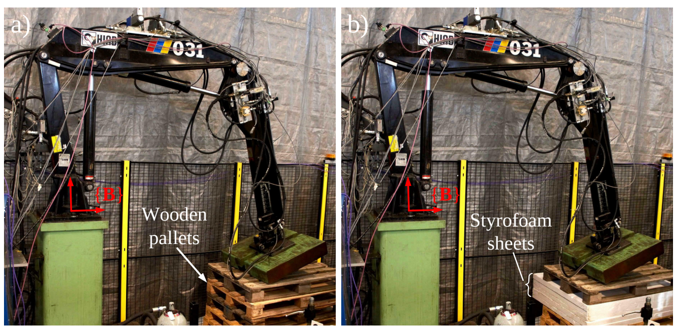
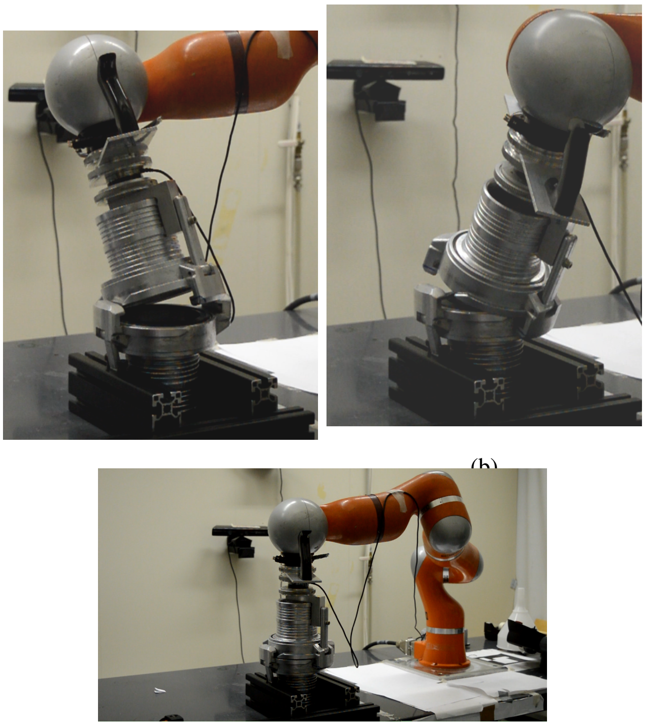


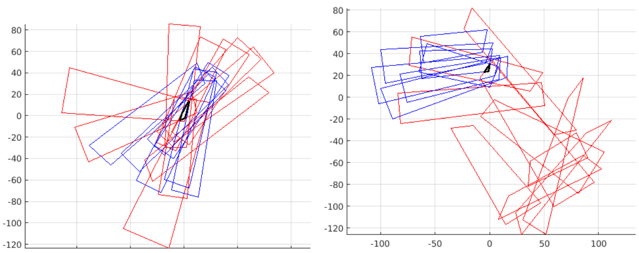
Fig. 11: a) Experiment setup with wooden pallets. b) Experiment setup with styrofoam sheets and wooden pallets. The manipulator's position in the figures show the starting point of the test trajectories (same starting position in the both cases) [7].



(b)

(c)

Fig. 12: Two possible starting poses and the final pose of the hose-coupler alignment task [1].

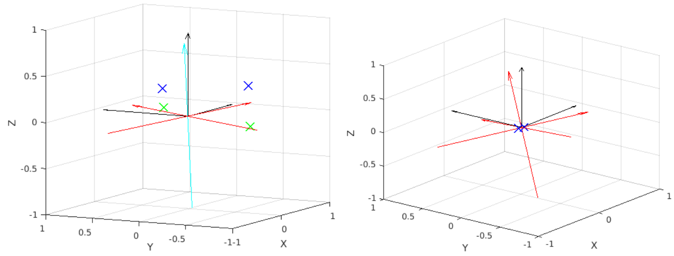


(a) Translations (b) Rotations

Fig. 13: Visualization of finding the desired direction, shown for translations and rotations of the hose-coupler alignment task. The red and blue colors indicate the two separate demonstrations of the task and the black rectangle is the intersection, the set of all desired directions in the projection coordinate

to the origin, but still far enough that one compliant axis was detected, as required to align the tools.

We experimentally verified that we can successfully reproduce the alignment motions. Additionally, we showed successful learning and reproduction of a peg-in-hole task with a varying starting orientation error. Screenshots from a reproduction are shown in Fig. 15. Moreover, we also showed that this primitive can be successfully used with teleoperated demonstrations, which are shown to be noisier than by kinesthetic teaching [23] with a heavy-duty hydraulic



(a) Translations (b) Rotations

Fig. 14: Illustrations of choosing the directions of compliant axes on the hose-coupler alignment experiment. The black arrows are coordinate axes, the red ones the eigenvectors U, the blue crosses the average motions of each demonstration and the green crosses their projections to the first principal component. In (a) the desired direction is plotted in cyan (overlapping the third eigenvector as expected). In both (a) and (b) 1 compliant axis is chosen [1].

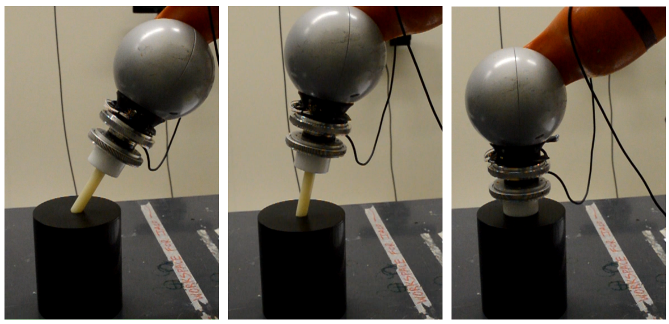
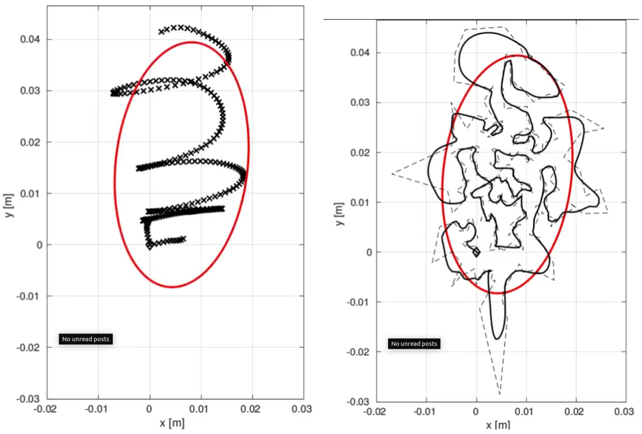


Fig. 15: Screenshots from a reproduction video of the P-I-H motion. The motion starts from the leftmost picture, and the peg is rotated and pushed to the bottom. The peg has radius 16.5 mm, length 80 mm and a rounded tip, and the hole's radius is 0.25 mm more than the peg's [1].

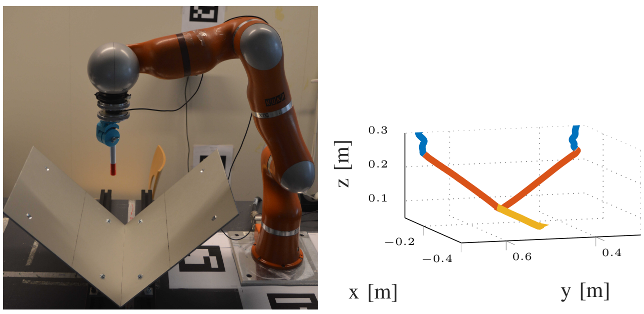
We performed the search motions on the peg-in-hole setup with 85% accuracy and the plug-in-socket task with 67% accuracy, which we consider good considering the difficulty of the tasks (essentially a near-blind search in 2-D or 3-D). In Fig. 16a is shown how the exploration distribution is learned from human demonstration, and in Fig.16b how a search trajectory is created from the exploration distribution by sampling.

We tested segmenting and sequencing of motions on both the hose-coupler setup (Fig. 12) and valley setup seen in Fig. 17a. In the hose-coupler setup, the algorithm correctly identified lowering the coupler as one LMC phase and interlocking the couplers as another and reproduction was successful. In the valley setup, the algorithm correctly identified that sliding down either side is the same phase, as seen in Fig. 17b, thus showing that the robot learned to take advantage of the guidance of either chamfer.



(a) Demonstration and exploration distributions.
(b) Exploration distribution and search trajectory.

Fig. 16: Visualizations of creating a search trajectory from one or more human demonstrations.



(a)
(b)

Fig. 17: The physical valley setup (a) and the phases learned from a demonstration of sliding to the bottom of the valley and then towards the camera (b).

IV. Conclusions

We successfully showed that we can learn from human demonstrations various tasks requiring compliance. The results from [2]- [7] can be used to greatly advance the usage of robots in SMEs by three very important factors: firstly, the usage of LfD makes teaching the robot new tasks easy and efficient, thus allowing the robot to perform varying tasks when production batch sizes are small. Secondly, by the use of compliance, small changes in the workplace due to *e.g.* vibrations may not cause the task to fail. Thirdly, even if the task fails, if a proper exception strategy is learned with the search, the robot can recover even from errors by itself and carry on it's task without need of an employee to re-teach everything. We believe that these results have the potential to significantly boost the usage of robots in Finland.

References

[1] M. Suomalainen and V. Kyrki, "Learning 6-d compliant motion primitives from demonstration," *Autonomous Robots*, 2019, submitted. arXiv:1809.01561.

[2] M. Suomalainen and V. Kyrki, "Learning compliant assembly motions from demonstration," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 871–876, IEEE, 2016.

[3] M. Suomalainen and V. Kyrki, "A geometric approach for learning compliant motions from demonstration," in *Humanoid Robots (Humanoids), 2017 IEEE-RAS 17th International Conference on*, pp. 783–790, 2017.

[4] T. Hagos, M. Suomalainen, and V. Kyrki, "Segmenting and sequencing of compliant motions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6057–6064, IEEE, 2018.

[5] D. Ehlers, M. Suomalainen, J. Lundell, and V. Kyrki, "Imitating human search strategies for assembly," *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, Accepted for Publication. arXiv:1809.04860.

[6] M. Suomalainen, S. Calinon, E. Pignat, and V. Kyrki, "Improving dual-arm assembly by master-slave compliance," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, Accepted for Publication.

[7] M. Suomalainen, J. Koivumäki, S. Lampinen, J. Mattila, and V. Kyrki, "Learning from demonstration for hydraulic manipulators," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3579–3586, IEEE, 2018.

[8] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths," *Robotics and Autonomous Systems*, vol. 11, no. 2, pp. 113–127, 1993.

[9] J.-H. Hwang, R. C. Arkin, and D.-S. Kwon, "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1444–1449, IEEE, 2003.

[10] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*, pp. 261–280, Springer, 2006.

[11] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[12] S. M. 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.

[13] J. Lundell, M. Hazara, and V. Kyrki, "Generalizing movement primitives to new situations," in *Conference Towards Autonomous Robotic Systems*, pp. 16–31, Springer, 2017.

[14] M. Hazara and V. Kyrki, "Model selection for incremental learning of generalizable movement primitives," in *Advanced Robotics (ICAR), 2017 18th International Conference on*, pp. 359–366, IEEE, 2017.

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

[16] N. Figueroa and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87 of *Proceedings of Machine Learning Research*, pp. 927–946, PMLR, 2018.

[17] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[18] F. J. Abu-Dakka, B. Nemec, A. Kramberger, A. G. Buch, N. Krüger, and A. Ude, "Solving peg-in-hole tasks by human demonstration and exception strategies," *Industrial Robot: An International Journal*, vol. 41, no. 6, pp. 575–584, 2014.

[19] I. F. Jasim, P. W. Plapper, and H. Voos, "Position identification in force-guided robotic peg-in-hole assembly tasks," *Procedia Cirp*, vol. 23, pp. 217–222, 2014.

[20] K. J. A. Kronander, *Control and Learning of Compliant Manipulation Skills*. PhD thesis, EPFL, 2015.

[21] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Learning manipulation graphs from demonstrations using multimodal sensory signals," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2758–2765, IEEE, 2018.

[22] O. Kroemer, H. Van Hoof, G. Neumann, and J. Peters, "Learning to predict phases of manipulation tasks as hidden states," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4009–4014, IEEE, 2014.

[23] K. Fischer, F. Kirstein, L. C. Jensen, N. Krüger, K. Kukliński, T. R. Savarimuthu, *et al.*, "A comparison of types of robot control for programming by demonstration," in *Proceedings of the 2016 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 213–220, 2016.