Imitating Human Search Strategies for Assembly

Dennis Ehlers[1], Markku Suomalainen[1,2], Jens Lundell[1] and Ville Kyrki[1]

***Abstract*— We present a Learning from Demonstration method for teaching robots to perform search strategies imitated from humans in scenarios where alignment tasks fail due to position uncertainty. The method utilizes human demonstrations to learn both a state invariant dynamics model and an exploration distribution that captures the search area covered by the demonstrator. We present two alternative algorithms for computing a search trajectory from the exploration distribution, one based on sampling and another based on deterministic ergodic control. We augment the search trajectory with forces learnt through the dynamics model to enable searching both in force and position domains. An impedance controller with superposed forces is used for reproducing the learnt strategy. We experimentally evaluate the method on a KUKA LWR4+ performing a 2D peg-in-hole and a 3D electricity socket task. Results show that the proposed method can, with only few human demonstrations, learn to complete the search task.**

I. INTRODUCTION

To succeed in peg-in-hole type assembly tasks with small clearance between workpieces, a robot's internal model of the world must match with the environment to high precision, as even small errors in the model can cause the alignment to fail. It is possible to rely on handcrafted exception strategies to complete the task, but the process of manually defining the strategies is tedious and often not generalizeable. Even though recent research has shown significant improvement in planning of contact motions [1], they are ineffective when facing modeling errors. Our key insight is to make the robot behave similar to a human when faced with such uncertainties about the environment—probing the environment, like fitting a key into a lock in darkness or inserting a plug into an electric socket, as illustrated in Fig. 1. Thus we present a Learning from Demonstration (LfD) method for teaching robots to perform human-inspired search strategies.

In LfD, a human teacher performs one or more demonstrations that the robot then learns to mimic [2]. Methods to encode and parametrize the demonstrated skill are usually based on attractors along a trajectory, such as Dynamic Movement Primitives (DMP) [3] and Gaussian Mixture Model (GMM) with Gaussian Mixture Regression (GMR) [4]. However, these methods assume that the goal pose is either known or can be learned. In contrast, we target scenarios where the goal pose can vary between reproductions of the same task and cannot be visually identified with sufficient accuracy to complete the task, thus rendering the problem closer to area coverage than a trajectory following problem.

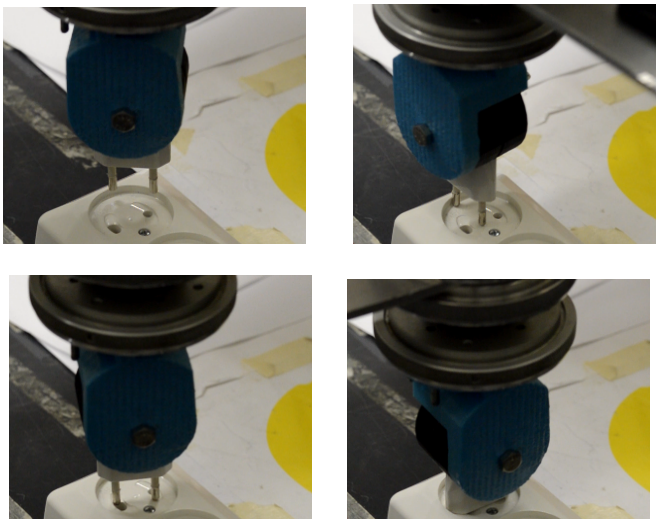


Fig. 1: An example sequence of a robot inserting a plug into a socket without vision sensing or knowledge about the location of the socket's holes.

We propose a method to learn search strategies by imitating human behavior in scenarios where the transformation between the start and end poses is not known but the human follows a deliberative strategy to cover an area that contains the end pose. The method consists of four steps: 1) gathering one or more human demonstrations from which forces and positions are recorded; 2) modeling the covered search area—the *exploration distribution*—and state-invariant dynamics of the task at hand probabilistically; 3) planning a search trajectory over the exploration distribution; and 4) executing the search trajectory with an impedance controller while superposing forces learned from dynamics. For planning the search trajectory we compare two algorithms: an ergodic control approach proposed by Miller et al. [5] and a new sampling-based algorithm Trajectory Sampling from Human-Inspired Exploration (TSHIX). We evaluate our method in both 2D (round peg-in-hole) and 3D (electric plug insertion) search tasks.

The main contributions in this work include 1) a novel 4-step framework for learning in-contact search strategies from human demonstrations, 2) a sampling based method, TSHIX, for creating trajectories from an exploration distribution and 3) showing through experiments that our learned search strategies are suitable to work as exception strategies in a 3D assembly task.

This work was supported by Academy of Finland, decision 286580 and 314180.

[1]D. Ehlers, J. Lundell and V. Kyrki are with School of Electrical Engineering, Aalto University, Finland firstname.surname@aalto.fi

[2]M. Suomalainen is currently with University of Oulu, Finland markku.suomalainen@oulu.fi

II. RELATED WORK

Existing work on search motions mainly focus on exception strategies on a 2D plane. Abu-Dakka et al. [6] presented an exception strategy based on stochastic search for failure case in a peg-in-hole task. However, their method did not use any task-specific information, which could improve the accuracy. Jasim et al. [7] used an Archimedean spiral to cover the 2D search area. However, the approach is not applicable in higher dimensional search tasks such as inserting a plug into a socket. Baum et al. [8] compared different physical exploration strategies in a 3D scenario. However, they did not learn the search strategies from human demonstrations.

A similar idea of using human demonstrations to learn search strategies was previously explored by De Chambrier in [9]. To learn to insert a plug into a socket, the method used 150 demonstrations to train a Partially Observable Markov Decision Process (POMDP) which learned a strategy that first localized itself in the environment with the help of physical barriers, and then followed a fixed trajectory to the socket. In comparison, our method focuses on cases with known approximate location of the goal, while the position or orientation of the goal can change from one demo to another in an unknown manner, requiring physical exploration for insertion. Moreover, our method does not require such an extensive amount of demonstrations.

In De Chambrier's work, the actual insertion was performed according to the work of Kronander [10, Chapter 5]. He also had an underlying hand-crafted strategy based on contact transients, and in case of jamming the human teacher intervened. In contrast, our method can learn the whole search process from humans in cases where the approximate location and orientation of the hole is known. In addition, our search strategy considers both position and force distributions, thus allowing also force-based search. Furthermore, the presented method does not require hand-crafting transient behaviors.

III. METHOD

Our proposed method for learning search strategies from human demonstrations consist of 4 steps: 1) learning the exploration distribution Section III-A, 2) creating the search trajectory for the manipulator Section III-B, 3) learning the task dynamics Section III-C, and 4) performing the search trajectory with an impedance controller Section III-D.

A. Learning the Exploration Distribution

We model the exploration distribution with a Gaussian[1] distribution over the task space,

$$p_e(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \ , \tag{1}$$

where $\boldsymbol{s} \in \mathbb{R}^D$, D is the dimensionality of the task (e.g. 3 if only translations are considered), and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance estimated by maximum likelihood.

To learn expressive exploration distributions, the training data must represent the entire search domain. In this work the humans demonstrated the movement with closed eyes to mimic the sensing modalities of the robot. To start a demonstration the end-effector was first placed in the proximity of the goal that resembles possible failure scenarios and then the human performed the demonstrations with his eyes closed. Demonstrations that ended either immediately at the goal or in a configuration from which success without vision was improbable were discarded.

The demonstrations were aligned with respect to their starting positions, termed the *search frame*. With this choice we can learn the *search* strategy of the teacher in situations where localizing the tool w.r.t. the world frame is impossible. Thus any trajectory generated will also start at the origin of the search frame, making the learned search strategies independent of their location in the world frame.

B. Creating the Search Trajectory

We propose two methods for generating a search trajectory that covers the modeled exploration distribution: 1) TSHIX, a stochastic sampling-based method, and 2) ergodic control [5], a deterministic method.

1) Trajectory Sampling from Human-Inspired eXploration: TSHIX samples points from the exploration distribution created in III-A, finds an approximation of the shortest itinerary through the sampled points and smooths it.

We draw samples from the exploration distribution using Marsaglia's ziggurat method [11]. Visualizations of sampling in 2D are presented in Fig. 4. Finding the shortest itinerary through the sampled points is a Travelling Salesman Problem (TSP), which we solved using an existing genetic algorithm based optimizer for simplicity of implementation[2]. The start of the itinerary is always set at the origin of the search frame in an effort to imitate the human search. The itinerary is then smoothed with a Savitzky-Golay Filter [12]. Fig. 5a illustrates the stages of creating a trajectory for the tool: A series of sample points are generated, through which an itinerary is calculated and smoothed to produce the final trajectory.

2) Ergodic Control: Ergodic control is a method for creating a trajectory such that at every step the time-averaged behavior of the system matches a known desired probability distribution. This is realized by minimizing a cost function that matches spatial statistics of the trajectory to those of the desired distribution. We quantify this as matching their Fourier coefficients [13]

$$C\big(x(t)\big) = \sum_{k=1}^{K} \Lambda_k \Big(c_k\big(x(t)\big) - \phi_k\Big)^2, \tag{2}$$

where Λ_k are frequency dependent weights, and ϕ_k are the Fourier coefficients of the exploration distribution learnt from the human demonstration. To minimize (2), we use the optimal control formulation of Miller and Murphey [5].

[1]For more complex exploration patterns, the single Gaussian can be replaced with a GMM.

[2]https://se.mathworks.com/matlabcentral/fileexchange/21198-fixed-start-open-travelling-salesman-problem-genetic-algorithm

C. Learning the Task Dynamics

When humans perform search tasks they do not simply follow a trajectory in space, but maintain contact between the workpieces and deduce vicinity of alignment based on contact forces. Learning the dynamics is useful to efficiently overcome friction as well as to reach threshold forces for phenomena such as "snapping" [14]. Therefore, learning search strategies similar to humans requires simultaneous learning of positions and forces.

For this reason, to record contact forces during a demonstration a force-torque sensor (F/T sensor) was attached to the wrist of the manipulator to record the interaction force (normal plus friction force). Because the pose of the goal is not known, we learn location-invariant dynamics that we assume to be applicable for the entire search.

We model the dynamics as a joint Gaussian distribution of velocities and wrenches

$$p_d(\Delta \boldsymbol{s}, \boldsymbol{a}) = \mathcal{N}(\Delta \boldsymbol{s}, \boldsymbol{a} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \ , \tag{3}$$

where $\Delta \boldsymbol{s}_t = \boldsymbol{s}_{t+1} - \boldsymbol{s}_t$ is the instantaneous motion and $\boldsymbol{a}_t = [-\boldsymbol{f}_t^T \ \ -\boldsymbol{t}_t^T]^T$ is the action wrench (inverse of the opposing measured interaction force). The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are fitted to demonstrations using maximum likelihood.

We can then predict the required action wrench for a desired state change by conditioning the Gaussian distribution (3) on the state change. Partitioning $\boldsymbol{\mu} = [\boldsymbol{\mu_a}^T \ \ \boldsymbol{\mu_s}^T]^T$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{aa}} & \boldsymbol{\Sigma_{as}} \\ \boldsymbol{\Sigma_{sa}} & \boldsymbol{\Sigma_{ss}} \end{bmatrix} \ , \tag{4}$$

the predicted action wrench for a desired state change $\Delta \boldsymbol{s}^*$ is

$$(\boldsymbol{a}^* \mid \Delta \boldsymbol{s}^*) \sim \mathcal{N}(\boldsymbol{\mu^*}, \boldsymbol{\Sigma}^*) \ , \tag{5}$$

where

$$\boldsymbol{\mu}^* = \boldsymbol{\mu_a} + \boldsymbol{\Sigma_{as}} \boldsymbol{\Sigma_{ss}^{-1}} (\Delta \boldsymbol{s}^* - \boldsymbol{\mu_s}) \ , \tag{6}$$

and

$$\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma_{aa}} - \boldsymbol{\Sigma_{as}} \boldsymbol{\Sigma_{ss}^{-1}} \boldsymbol{\Sigma_{sa}} \ . \tag{7}$$

D. Performing the Search

To reproduce the search motions, we use an impedance controller with superposed forces and torques, illustrated in Fig. [reffig:impedanceControl. We use an impedance controller to enable the robot to follow a trajectory in free space as well as in contact while being compliant to the environment. Using the superposed feed-forward forces enables force based searching, provided the teacher demonstrates such behavior.

To determine the position set-points, the continuous search trajectory (Sec. III-C) is first sampled in time. For each sample, the desired action wrench $\boldsymbol{a}^* \equiv [\boldsymbol{f}_f^T \ \boldsymbol{t}_f^T]^T$ is calculated as the mean $\boldsymbol{\mu}^*$ of the conditional distribution given in (6). The resulting impedance control law for translations is then

$$\boldsymbol{f}_c = K_p \boldsymbol{e}_p + B_p \dot{\boldsymbol{e}}_p + \boldsymbol{f}_f \ , \tag{8}$$

where $\boldsymbol{f}_c$ is the Cartesian force commanded to the robot, $\boldsymbol{e}_p$ is the positional error between current and desired position, K_p the stiffness matrix, $\dot{\boldsymbol{e}}_p$ the derivative of the positional error, B_p is the positional damping coefficient and $\boldsymbol{f}_f$ the superposed feed-forward force.

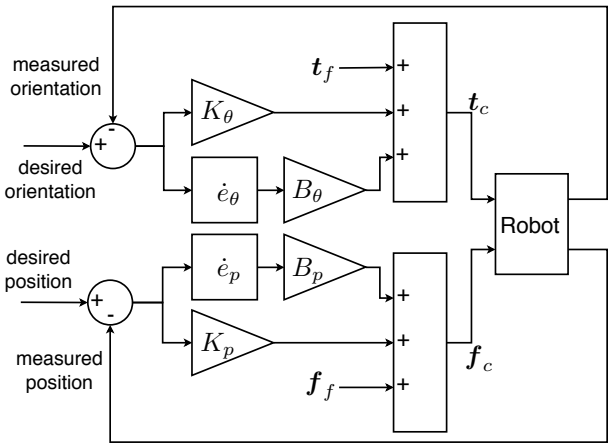


Fig. 2: An impedance controller with F/T feed-forward is used to reproduce the search motions.

For the orientation, the impedance control law can be written

$$\boldsymbol{t}_c = K_\theta \boldsymbol{e}_\theta + B_\theta \dot{\boldsymbol{e}}_\theta + \boldsymbol{t}_f \ , \tag{9}$$

where $\boldsymbol{t}_c$ is the Cartesian torque command, $\boldsymbol{e}_\theta$ is the angular error between current and desired orientation, K_θ is the angular stiffness matrix, $\dot{\boldsymbol{e}}_\theta$ is the derivative of the angular error, B_θ is the angular damping matrix and $\boldsymbol{t}_f$ the superposed feed-forward torques.

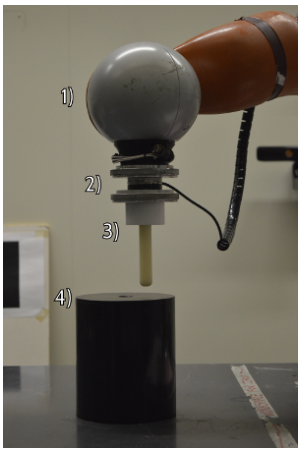
IV. EXPERIMENTS AND RESULTS

We used a KUKA LWR4+ lightweight arm to experimentally validate our method. Demonstrations were recorded in gravity compensation mode, where the robot's internal sensors recorded the pose of the robot and an ATI mini45 F/T sensor at the wrist of the robot recorded the wrench. We implemented the controller through the Fast Research Interface (FRI) [15] similarly as in [16], [17] such that KUKA's internal controller handles the feed-forward dynamics on top of the impedance controller defined in (8-9).

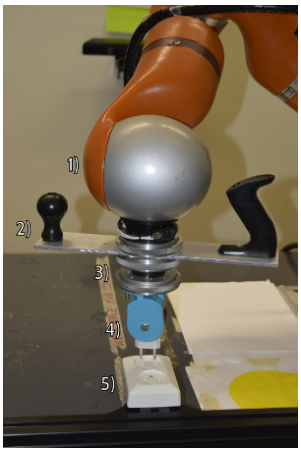
We evaluated the method in two scenarios: a peg-in-hole task (Fig. 3a) and an electrical plug and socket task (Fig. 3b). With the peg-in-hole setup, we first compared the two presented trajectory creation algorithms, TSHIX and ergodic control (Section IV-A), then examined how the number of itinerary points for TSHIX affected success rate (Section IV-B), and finally analyzed the trajectory following performance of the controller (Section IV-C) and whether involving the dynamics in a 2D task improves the performance (Section III-C). With the electrical plug task (Section IV-D) we study the performance of the method in a more complex 3D search task.

A. Comparison Between Ergodic Control and TSHIX

We used the 2D peg-in-hole task (Fig. 3a) to compare the sampling based TSHIX and ergodic control approaches for



(a) Peg-in-Hole setup: 1) KUKA LWR4+, 2) F/T sensor, 3) Peg tool, 4) Cylinder with hole



(b) The plug and socket setup: 1) KUKA LWR4+, 2) Grab-handle for users to manipulate the robot, 3) F/T sensor, 4) Plug tool, 5) Socket

Fig. 3: The setups used for the experiments.

trajectory creation[3]. We first learned the exploratory distribution and the dynamics from a single demonstration. Since the peg-in-hole task is conducted in the 2D plane on top of a cylinder, the orientation of the peg tool was fixed during both demonstration and reproduction. With TSHIX we generated 30 different trajectories in the x-y-plane, each consisting of 300 points sampled from the learned exploration distribution. With the ergodic control we generated a trajectory that was approximately the same length as the mean of the TSHIX trajectories. Finally, for the trajectories provided by both algorithms we used the learned dynamics model to compute the superposed forces.

We performed the comparison by randomly choosing 30 starting positions from within a circle of ca. 3 cm in diameter, located roughly halfway between the originally demonstrated starting position and the hole of the cylinder. The reason for moving the starting positions closer to the hole is that the demonstration acts as a worst case scenario, *i.e.* the one end of the exploration distribution will be approximately at the starting position. The same exploratory distribution and dynamics were used for both algorithms. For TSHIX we used a different randomly chosen trajectory with each starting position to average the inherent randomness, whereas for the ergodic control we used the same trajectory with each position, due to its deterministic nature. As this task does not involve rotations, the control was performed by feeding the trajectory and superposed forces to (8).

The results of the experiment are presented in Table I. They show that TSHIX performed slightly better than the ergodic control. In the five trials where TSHIX failed, the ergodic control failed as well. These failures seemed to occur because the starting positions were too far away from the goal, as neither trajectory lead close enough to the hole. In the five trials where TSHIX succeeded and ergodic control failed, the ergodic trajectory would usually go within the vicinity of the hole, but not cover it. This implies that the difference between the algorithms could stem from density of coverage.

[3]To generate ergodic control trajectories we used the following open source implementation `https://github.com/dressel/ErgodicControl.jl`.

Algorithm	Ergodic	TSHIX
Success rate	20/30	25/30
Success percentage	67%	83%

TABLE I: Success rates from peg-in-hole experiments from the two trajectory creation methods.

One advantage of using the sampling-based TSHIX is the flexibility in generating the trajectory. If a generated trajectory was unsuccessful or undesirable for other reasons, a new trajectory can easily be generated by changing the random seed for the sampling. Moreover, additional points could even be sampled during reproduction should the task fail. This is not possible for the ergodic trajectory, as they are per definition deterministic and thus when the chosen convergence criteria are met, the trajectory is finished. However, the ergodic trajectory can be considered optimal for covering a distribution [18] and the deterministic nature increases the predictability and repeatability of the algorithm.

B. Effect of Sample Size

To investigate the effect the sample size had on the success rate of TSHIX, one successful starting position from the previous section was selected for further analysis. We generated 20 trajectories for sample sizes of 100, 200, 300, 400 and 500 each. More samples will lead to a higher coverage (see Fig. 4) and the connection between coverage and success rate is investigated in this section. The trajectories and their corresponding predicted superposed forces were executed with the impedance controller from (8). The results, presented in Table II, show that initially the success of the search increases with higher sample sizes but so does the computational time. However, somewhat counterintuitively, after 300 samples the success rate consistently stays at 85%, a finding further explored in the next section.

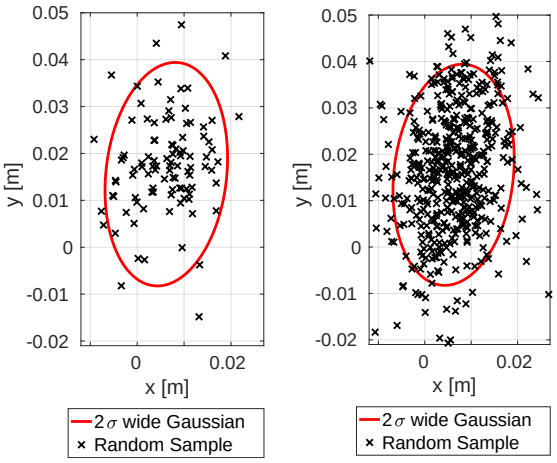
Sample size	100	200	300	400	500
Success rate	6/20	12/20	17/20	17/20	17/20
Success percentage	30%	60%	85%	85%	85%
Mean CPU time (s)[4] (±1 sd) (s)	13.8 (±0.3)	34.1 (±1.1)	61.9 (±4.4)	94.3 (±5.4)	135.6 (±10.4)

TABLE II: Success rates and computational times from varying the number of sampled points in TSHIX on the 2D peg-in-hole task

C. Controller Analysis

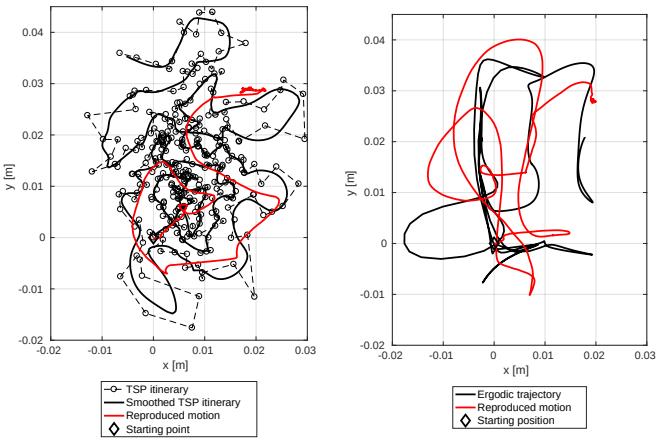
For in-contact tasks it is a challenge to choose the right stiffness value such that the trajectory is followed while allowing compliance to avoid excessive contact forces and facilitate easier completion of the task. For peg-in-hole tasks, lower stiffness perpendicular to the motion allows the peg to

[4]In this work all computations were performed using an Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66GHz.



(a) 100 samples (b) 500 samples

Fig. 4: Different number of sampled points covering the learned exploration distribution.



(a) (b)

Fig. 5: Trajectory following performance of the controller on trajectories created by TSHIX (a) and ergodic control (b).

be inserted without moving over the exact center of the hole as it exploits the mechanical gradients of the task.

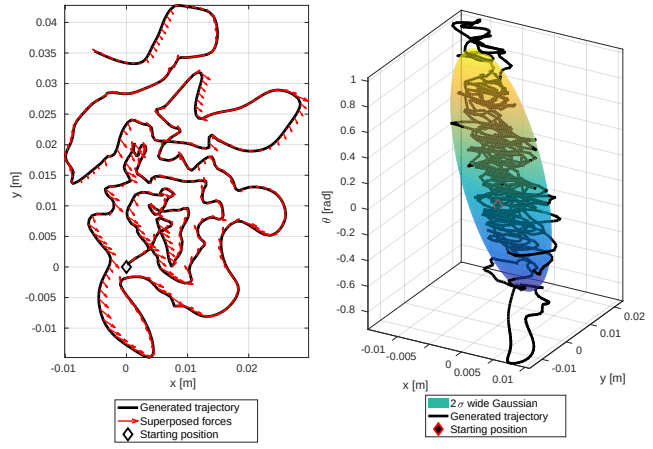
We chose to keep the stiffness low and instead provide the robot with the dynamics of the task. By knowing the dynamics instead of increasing the stiffness, we can predict the forces necessary to counter the environmental forces, such as friction, along the trajectory. The stiffness is therefore increased only in the direction of motion while leaving the tool compliant perpendicular to the direction of motion. We empirically observed that the perpendicular compliance indeed seemed to play a key role in the final stage of the task where the peg has to slide into the hole; without the compliance the peg was more likely to slide by close to the hole but not sink in.

To experimentally examine how much influence the usage of the superposed forces has on the outcome of the task, an additional experiment series was set up. 20 trajectories from 100, 200 and 400 sampled points were created. Each trajectory was then executed with the impedance controller in Section IV-B without the addition of the superposed predicted forces. For the 100 sample point trajectories, only 2/20 trials succeeded, 4 less then with force-feed-forward. Similar observations were made for 200 and 400 sample point trajectories (7/20 and 13/20 respectively). This consistently inferior performance is indicative of the importance of the superposed forces even on a task performed on a 2D plane. As the forces are expected to be countering the environmental forces, which in this task consist of mainly friction, the forces point in the direction of the state change. This is observable in Fig. 6a where the superposed forces are plotted over a trajectory sliding along a 2D plane. We note that in this example the forces pointing "north-west" have smaller magnitude and are more difficult to observe.

However, as seen in Fig. 5 even with the superposed forces the tool is not following the desired trajectory accurately. These trajectories were completed with stiffness values that were experimentally found to provide a good balance between trajectory following and compliance that facilitates the final insertion of the peg-in-hole task. The reason for this divergence lies presumably in the unknown friction forces of the robot's joints. When performing the demonstrations, the teachers also noticed that the joint configuration had a significant effect on the ease of moving the robot. In addition, feeding a Cartesian wrench profile recorded from a demonstration to the robot in pure force control did not produce any meaningful results. We assume this is due to static friction in the joints that prevents initiation of small motions. The same reason is why, in Table II, the success percentage of TSHIX did not reach 100% despite increased sample size: in all sample sizes between 300 and 500, there was one error where the peg slid past the hole even if the planned trajectory covered it, and two errors where the peg entered the hole but did not slide all the way in, most likely due to insufficient compliance. We hypothesize that better compensation of internal friction would allow trajectory following even with low stiffness and thus better facilitate the final insertion part.

D. Learning a 3D Search Task

To investigate the scalability of the proposed method to higher dimensions, we conducted a three-dimensional search experiment. The task was to insert an electrical plug (Europlug) into a socket. For the socket, a 12V socket usually found in caravans and sheds was used because of its design: one hole is slightly larger, and there are valley shaped indentations around the holes, allowing an easier insertion when motion is compliant. To make the demonstrations and the learning process easier, the rotation around the x- and y-axes of the tool are restricted, so that the search was only performed in the positional space of the x-y-plane, as well as in rotation around the tool's z-axis. The demonstrations were conducted with closed eyes, and the starting pose of the plug was inside the socket close to a side wall. The orientation around the z-axis was chosen randomly, but still reasonable enough to allow the teacher to complete the task like an everyday situation. Because of the higher complexity of this 3D task, we compared learning the exploration



(a) (b)

Fig. 6: (a) The superposed forces calculated from learned dynamics plotted along a generated search trajectory (b) Illustration of 3D trajectory created from 600 sample points.

distribution from either one or two demonstrations. The two demonstrations were performed starting from opposite ends of the upper half of the socket. From both a single demonstration and a combination of two demonstrations, the 3D exploration distribution (position in x and y, rotation around z) and the 6D task dynamics (position in x and y, rotation around z, forces in x and y, torques around z) were learned similarly to the peg-in-hole task, by fitting a single Gaussian distribution over the data. A trajectory was created by sampling 600 points, which was found to be a reasonable number for covering the 3D exploration distribution (see Fig. 6b). Subsequently the predicted interaction values from the environment (both forces and torques) for the given trajectory were calculated.

For the experiments, 15 starting positions were randomly chosen in the northern hemisphere of the socket, and for each starting position a different trajectory was generated from both one and two demonstrations. It was impossible to create an ergodic trajectory with Projection-based Trajectory Optimization in 3D, since the algorithm gets easily trapped in a local minimum which restricts the length of the created trajectory. In order to provide a baseline for comparison, random walk trajectories were created using the method proposed by Abu-Dakka et al. [6]. The results are presented in Table III. We can observe that for the 3D task having two demonstrations significantly increases the success rate and even for the much harder socket task the method achieves good results.

Algorithm	Random Walk	1-demo-TSHIX	2-demo-TSHIX
Success rate	0/15	3/15	10/15
Success percentage	0%	20%	67%

TABLE III: Success rates from plug-in-socket experiments from random walk and TSHIX with either one or two demonstrations.

E. Discussion

The results show that the proposed method performs well if the demonstrations' starting positions cover the variability of the scenarios. However, some real world scenarios such as plug-and-socket for a European 230V AC socket are surprisingly difficult, even human teachers found it very hard to solve the task blindly. This indicates that some search problems may be infeasible in higher dimensions. On the other hand, if the demonstrated search is performed in a lower-dimensional subspace of a high-dimensional state space, the proposed method will automatically perform the search in the lower-dimensional space by virtue of employing the Gaussian linear model.

We showed that while it is possible to use only one demonstration to learn a search if the task is sufficiently easy, for more complex tasks with higher dimensionality it is necessary to have more demonstrations in order to achieve a reasonable convergence region. Moreover, for multimodal or more complex tasks, the exploration distribution can be modeled as a GMM. We originally used GMMs, but initial experiments showed that only one Gaussian component was sufficient to model our experimental problems and more components lead to overfitting. Furthermore, a requirement for the demonstrations is that they need to be performed in worst case scenarios, so that they cover the variability of both exploration region and dynamics encountered in the task.

V. CONCLUSIONS AND FUTURE WORK

We presented a simple yet data efficient LfD method teaching robots to imitate human search strategies for assembly tasks. The key idea is to first learn from human demonstrations both an exploration distribution and a dynamics model and then use these to generate candidate search trajectories covering said exploration distribution with superposed forces extracted from the dynamics model. We evaluated two different trajectory generating methods: TSHIX, our own sampling based method, and ergodic control, a deterministic optimal path planner. The experimental results demonstrate that our method is data efficient as it learns to complete 2D and 3D search tasks from only one, respectively, two human demonstrations, a major improvement over current exception strategies that have been shown to only work in 2D.

The method suffers from the same issue most LfD methods do, mainly that it can only learn what it has seen. One remedy to this can be found in the research regarding Human-Robot Interaction, where a recent study investigated how to instruct human teachers to give informative demonstrations [19].

There are several interesting directions for future work. First of all, as we observed from demonstrations that humans tend to naturally perform their search in distinct phases and in no more than three dimensions, it stems to reason that it is more efficient to model high dimensional searches as a sequence of lower-dimensional ones with, *e.g.*, the method presented in [20]. Another interesting direction is to analyze whether the temporal correlations of the human search can be included in the exploration distribution. Furthermore, as experienced during the initial trials the quality of the human demonstrations varied substantially, raising an interesting question on how to develop teaching environment for both human and robot in order to get high quality demonstrations.

REFERENCES

[1] C. Guan, W. Vega-Brown, and N. Roy, "Efficient planning for near-optimal compliant manipulation leveraging environmental contact," in *Robotics and Automation, 2018. (ICRA). IEEE International Conference on*, pp. 215–222, IEEE, 2018.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[3] 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.

[4] 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.

[5] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *American Control Conference (ACC), 2013*, pp. 4196–4201, IEEE, 2013.

[6] 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.

[7] 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.

[8] M. Baum, M. Bernstein, R. Martin-Martin, S. Höfer, J. Kulick, M. Toussaint, A. Kacelnik, and O. Brock, "Opening a lockbox through physical exploration," in *Humanoid Robotics (Humanoids), IEEE-RAS 17th International Conference on*, pp. 461–467, IEEE, 2017.

[9] G. P. L. De Chambrier, "Learning search strategies from human demonstrations," tech. rep., EPFL, 2016.

[10] K. J. A. Kronander, "Control and learning of compliant manipulation skills," tech. rep., EPFL, 2015.

[11] G. Marsaglia and W. W. Tsang, "A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions," *SIAM Journal on scientific and statistical computing*, vol. 5, no. 2, pp. 349–359, 1984.

[12] Press, Flannery, Teukolsky, and Vetterling, *Numercial Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1989.

[13] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4-5, pp. 432–442, 2011.

[14] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, "Force controlled assembly of emergency stop button," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3751–3756, IEEE, 2011.

[15] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the KUKA lightweight robot," in *Proc. of the IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications – How to Modify and Enhance Commercial Controllers (ICRA 2010)*, IEEE, 2010.

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

[17] 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, IEEE, 2017.

[18] L. Dressel and M. J. Kochenderfer, "On the optimality of ergodic trajectories for information gathering tasks," in *2018 Annual American Control Conference (ACC)*, pp. 1855–1861, IEEE, 2018.

[19] A. Sena, Y. Zhao, and M. J. Howard, "Teaching human teachers to teach robot learners," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, IEEE, 2018.

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