Fitted Policy Iteration for a POMDP Peg-In-Hole search task.

Guillaume de Chambrier, Aude Billard

Abstract

Acting optimally given environmental uncertainty is still currently a major concern for autonomous robotic systems. Uncertainty arises from the absence of informative prior knowledge, insufficient sensor capabilities and imprecise motion. Such uncertainty, if not considered appropriately by a policy or planner can lead to either sub-optimal task execution or failure. We consider a plug power-socket search and connection task also known as Peg-in-Hole (PiH) in which both a human and robot must be able to find a socket and connect a power plug to it without using any vision, making the state space partially observable, whilst solely relaying on haptic information.

A policy can be obtained by encoding the tasks as a partially observable markov decision process (POMDP) and then solve it by dynamic programming. This quickly become infeasible for relatively high-dimensional continuous tasks. To address this problem we demonstrate how human intuition can be leveraged to learn a belief space policy. A set of human teachers demonstrate the search PiH task during which the position uncertainty, represented by a Point Mass Filter (PMF), is recorded and compressed to the most likely state and entropy. A policy parametrised by a Gaussian Mixture Model (GMM) is learned and refined in an Actor-Critic Fitted reinforcement learning framework. We evaluate Actor-Critic policy, called Q-EM, against a Greedy and non-optimised GMM policy with respect to the distance taken to localise the socket and the distance taken to establish a connection between the plug and power socket. We test the ability of the learned models to generalise to different socket types and locations. We found that the Actor-Critic policy is always better in terms of distance travelled to localise the socket. We tested on a KUKA LWR robot, across three different power sockets, the ability of the Q-EM, GMM and Greedy policies to find and connect a plug to the power socket. We found when the socket has no distinctive features both data learned policy are better but when features are present the Greedy policy does just as good, or better.

Keywords: Programming by Demonstration, Actor-Critic, Belief Space, POMDP, Fitted Reinforcement Learning

1. Introduction

The ability to act optimally given uncertainty is paramount for robotic systems to be successful in environments which are not fully observable. Depending on the task and structure of the uncertainty, if not taken into consideration by the control policy, can lead to waistfull usage of resources and even fail to accomplish the task. Given the potential adverse (disastrous if considering a search and rescue task) consequences, it is important to design uncertainty robust policies and planners. An approach which gives optimality guarantees when dealing with uncertainty is to formulate the task in the framework of a partially observable markov decision process (POMDP) which is subsequently solved by dynamic programming. It is well known that solving a POMDP directly is infeasible even for the simplest problems. In this work we consider a plug power-socket search and connection task given no visual information; the second component is also known as Peg-in-Hole (PiH). In our task a robot must first successfully localise a power socket and then establish a connection. Our approach is based on an Actor-Critc Reinforcement Learning (RL) and Programming by Demonstrations (PbD) framework in belief state space, which we name RL-PbD-POMDP. No vision system will be used in this task and we will solely relay on haptic information, provided via force-torque sensor mounted on the end-effector of the robot. We chose to not use vision for two reasons. The first is that we want to know the humans capabilities of accomplishing the task under these conditions and whether we can learn a POMDP policy from their demonstrations. The second is that PiH is a very important component in manufacturing processes and we seek to demonstrate that we accomplish this without the need of vision system which would add additional costs to a manufacturing plant.

1.1. Peg-in-hole

The Peg-in-Hole (PiH) task is one of the most widespread step in industrial assembly and manipulations processes, with examples including the assembly of vehicular transmission components [4] and valves [3]. To be successful, the estimated position of the robot's end-effector and workpiece must be precise. Typically, the clearance between peg and plug is very small leaving little room for error. As a result, variations in the assembly's components in combination with position uncertainty can result in either jamming during the insertion process or in failure for the plug finding the hole. This created a need for adaptive search and insertion policies for PiH, which has been driving research in this area.

From the literature, we identified the different components in PiH solutions.

All approaches use to some extent a vision system to estimate the position of the workpiece. For instance in [9] a PR2 is equipped with a checkerboard to facilitate pose estimation of the plug with respect to a power outlet whose position is extracted through a vision processing pipeline. An initial connection is attempted by visual servoing which is successful 10% of the time. Given an estimate of the workpiece's position, a common approach is to follow either a blind increasing spiral Cartesian trajectory or parametrised policies which guarantee that all positions on the workpiece have been visited. In [9], if the PR2 initially fails to connect the plug to the socket a spiralling outward motion is carried out with 2mm increments which obtains an overall success rate of 95%. For this approach to be applicable to a generic robot, it would require the addition of an external camera and checkerboard to the robot in question which might be cumbersome. In our work we consider a vision free system.

Another approach (which has been confined to academic circles) follows the data driven Programming by Demonstration (PbD) framework. Teleoperated or kinesthetic demonstrations by a human teacher are recorded and a policy is learned and fine-tuned so as to reproduce the same (F)orce/(T)orque profile as that demonstrated by the human teacher.

In [13] the authors learn a PiH policy for the Cranfield benchmark object. A vision system obtains the pose parameters of the object whilst a human teacher demonstrates trajectories, through teleoperation, in the frame of reference of the object. A time-dependent policy represented with Dynamic Movement Primitives (DMP) [12] encodes the recorded Cartesian end-effector pose. In [10], a F/T profile is encoded separately by a regressor parameterised by radial basis functions. Successive refinements of the DMP policy are achieved through using force feedback to adapt the parameters of an admittance controller. This results in the policy having similar force profiles to the human teachers. Further applications based on this method have been performed [1] with the incorporation of a disturbance rejection policy. Reproducing exactly the same force torque profile for the full trajectory which is encoded in a time dependent dynamical system might be unnecessary as the force torque profile is predominantly useful during the final stage of the PiH task, where the insertion can cause jamming. The force torque information can be used to rectify this problem [**?** , Chap. 5]. A hybrid control paradigm [5] can also be used to control the sensed force feedback with the environment. We make use of the hybrid control paradigm in this work in combination with a time-independent dynamical system.

Reinforcement learning has also been used in combination with DMP to learn PiH policies. In [7] an DMP policy is initialised with kinesthetic demonstrations of opening a door and picking up a pen. The recorded Cartesian trajectories are encoded in a parameterised DMP policy and augmented with a F/T regressor profile. A reward function is designed, encoding desirable properties of the F/T profile such as smoothness and continuity. After 110 trials the policy was found to be a 100% successful. In [6] a 18 dimensional input (sensed position, previous position and force) and a 6 dimensional output (linear and angular velocity) neural network is learned by associative reinforcement learning. During the learning process the plug is randomly positioned within the vicinity of the hole. After a 100 executions and updates, the policy was shown to be successful and was able to generalise across different geometries and clearances. Our work is similar in its approach, however we will not be considering autonomous rollouts common in RL, but will rely solely on the initial data provided by human teachers.

All the above policies were learned from human demonstrations and encoded by a regressor function and optimised to reproduce a desired F/T profile. Other approaches to the PiH problem are predominantly based on heuristic search mechanisms and compliant controllers.

In [4] different blind search policies are analysed for the insertion of a spline toothed hub into a forward clutch. The state space is discretised into points so that the distance between two neighbours is smaller than the clearance of the hole, which is known as a spray point coverage. Different search strategies are evaluated which ensure that all the points are visited. It is found that paths following concentric circles gradu-

ally spiralling inwards are the most effective method for finding the hole. This concentric circle search strategy has been applied in many PiH tasks. For instance in [2], a PiH heuristic policy was developed to connect a 5-pin waterproof industrial charger to an electric socket. The authors estimated the pose of the socket through a vision system and used a force controller in combination with a blind spiral search policy to achieve a connection and demonstrated their approach to be reliable. These blind search strategies do not consider actual state uncertainty and only work well when the plug or peg is within the vicinity of the socket. In our work we consider no visual information which leads to high state uncertainty making the direct application of such blind search methods ill-suited.

In [11] the authors observe that humans lack the precision and sensing accuracy of robotic systems, but nevertheless, are more proficient than robots at PiH. The authors state that when humans try to connect a square plug to a socket, they rub the plug against the socket's outlet without looking. It is thought that the inherent compliance in humans' motor control is the key to our success at PiH tasks [8]. The authors introduce an Intuitive Assembly Strategy (IAS) inspired by the above observation which does not require the hole to be precisely localised. The IAS search strategy is based on compliant spiral motion and the execution of the search trajectory is performed with a hybrid force/position controller. We also have observed that humans are good at accomplishing such tasks and we exploit this in our own PiH policy. We further consider different types of geometric objects whilst only considering haptic information.

The spiral strategy is widely used in industrial applications due to its simplicity, however, it is a blind search method. Another approach when dealing with the assembly process consists of fine-tuning parameters of predefined policies. In [3] the authors develop an online Gaussian Process policy optimisation of an assembly task. They demonstrate that by learning the dynamical model of the task during execution, it is faster than offline methods, such as Design of Experiment (DOE) or Genetic Algorithms.

1.2. Actor-Critic & Fitted Reinforcement Learning

The Peg-in-Hole (PiH) task is one of the most widespread steps in industrial assembly and manipulations processes, with examples including the assembly of vehicular transmission components [4] and valves [3]. To be successful, the estimated position of the robot's end-effector and workpiece must be precise. Typically the clearance between and peg and the workpiece's hole is very small, leaving little room for error. As a result, variations in the assembly's components in combination with position uncertainty can result in either jamming during the insertion process or the peg is unable to find the hole. This created a need for adaptive search and insertion policies for PiH, which has been driving research in this area.

We identified, from the literature, the different components in PiH solutions. All approaches use to some extend a vision system to estimate the position of the workpiece. Given an estimate of the workpiece's position, a common approach is to either follow a blind increasing spiral Cartesian trajectory or parametrised policies which guarantee that all positions on the workpiece have been visited. To increase the chance of a connection these approaches use a compliant controller, which usually includes a hybrid force/position controller. The second predominant approach (which has been confined to academy) follows the data driven Programming by Demonstration (PbD) framework. Teleoperated or kinesthetic demonstrations of a human teacher are recorded and a policy is learned and fine tuned such to reproduce the same (F)orce/(T)orque profile as the ones demonstrated by the human teacher. The first approach does not consider reproducing the F/T profile but rather follows a position trajectory whilst being compliant.

In [9] a PR2 executes a parameterised policy designed to connect a plug to a power outlet in order for the PR2 to recharge itself. The plug was equipped with a checkerboard to facilitate pose estimation of the peg with respect to power outlet who's position was extracted through a vision processing pipeline. An initial connection was attempt by visual servoing which was successful 10% of the time. When unsuccessful a spiralling outward motion with 2mm increments was carried out. They achieved an overall success rate of 95%. The hybrid control paradigm [5] was used throughout the execution of the task.

In [13] the authors learned a PiH policy for the Cranfield benchmark object. A vision system obtained the pose parameters of the object whilst a human teacher demonstrated trajectories, through teleoperation, in the frame of reference of the object. A time dependent policy represented with Dynamic Movement Primitives (DMP) [12] encodes the recorded Cartesian end-effector pose. A F/T profile is encoded separately by a regressor parameterised by radial basis functions. Successive refinements of the DMP policy are achieved through using force feedback to adapt the parameters of an admittance controller. This resulted in the policy having a

similar force profiles as the human teachers. Such an approach was first proposed by [10] and further applications based on this method have been done [1] with the incorporation of a disturbance rejection policy. Reinforcement learning has also been used in combination with DMP to learn PiH policies. In [7] an DMP policy is initialised with kinesthetic demonstrations of a door opening and pen pick up task. The recorded Cartesian trajectory are encoded in parameterised DMP policy and augmented with a F/T regressor profile. A reward function is designed encoding desirable properties of the F/T profile such as smoothness and continuity and after a 110 trials a policy was found to be successful 100% of the time. In [6] a 18 dimensional (sensed position, previous position and force) input and 6 dimensional output (linear and angular velocity) neural network is learned by associative reinforcement learning. During the learning process the peg would be randomly positioned (both position and orientation) within the vicinity of the hole and after a 100 executions and updates, the policy was successful at the task and was able to generalise across different geometries and clearances.

The policy of the above methods were learned from human demonstrations and encoded by a regressor function and optimised to reproduce a desired F/T profile. The next approaches to the PiH problem are predominantly based on heuristic search mechanism and compliant controllers.

In [4] different blind search policies for the insertion of a spline toothed hub into a forward clutch are analysed. The state space was discretised into points such that the distance between two neighbours was smaller than the clearance of the hole, which is known as a spray point coverage. Different search strategies which ensure that all the points are visited were evaluated. It was found that paths following a concentric circles gradually spiralling inwards were the most effective in finding the hole. The concentric circle search strategy has been applied in many PiH tasks. For instance in [2], a PiH heuristic policy was developed to connect a 5-pin water proof industrial charger to an electric socket. The authors estimated the pose of the socket through a vision system and use a force controller in combination with a spiral search policy to achieve a connection and demonstrated their approach to be reliable.

In [11] the authors make the remake that humans do not have the precision and sensing accuracy of robotic systems, but nevertheless we are more proficient than robots at PiH. The authors make the observation that when humans try to connect a square peg to a socket, they rub the peg against the socket's outlet without looking. It is thought that the inherent compliance in humans motor control is key to our success at PiH tasks [8]. The authors introduce an Intuitive Assembly Strategy (IAS), inspired by the above observation, which does not require the hole to be precisely localised. The IAS search strategy is based on compliant spiral motion and the execution of the search trajectory is done with a hybrid force/position controller.

The spiral strategy is widely used in industrial applications due to its simplicity, however it is a blind search method. Another approach to the assembly process consists of fin tuning parameters of predefined policies. In [3] the author develops an online Gaussian Process parameter policy search of an assembly task. The authors demonstrate that by learning the dynamics of the task during execution the model it is much more rapid in fine tuning the parameters in contrast with offline methods such as Design of Experiment (DOE) or Genetic Algorithms.

2. Experiment methods

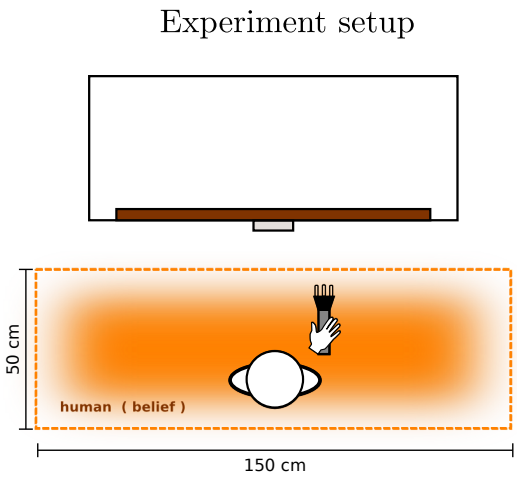
Figure 1 (*Top-left*), illustrates the PiH-search experiment setup. The orange area represents the teachers starting area and is assumed prior knowledge. The sockets are always positioned at the center of a fake wall (wooden plank) which is clamped to a table, see Figure 1 (*Top-right*) for an illustration.

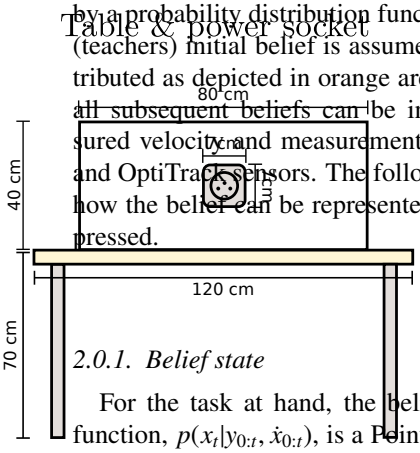
We consider one type of plug, Type J[1], and three different power sockets. Power *socket A*, has a ring around its holes, *socket B* has a funnel, which we hypothesize should make it easier to connect, and *socket C* has a flat elevated surface. See Figure 1 (*Bottom*) for an illustration.

The human teacher holds the plug which is attached to a cylindrical handle with an ATI 6 axis force torque sensor (Nano25 [2]) to provide **raw** wrench $\phi \in \mathbb{R}^6$ measurements. We define the **actual** measurement to be a function of the raw wrench, $\tilde{y}_t = h(\phi_t)$, which is a binary feature vector. The feature vector encodes whether a contact is present and the direction in which it occurs, which is discretized to the four cardinalities.

On top of the cylinder there is a set of markers used by a motion capture system OptiTrack[3] (which has millimeter tracking accuracy), see Figure 2, to measure both linear, $\dot{x} \in \mathbb{R}^3$, and angular velocity, $\omega \in \mathbb{R}^3$, at each time step which is recorded at a rate of 100 Hz. The force and torque information from the ATI sensor is recorded at the same rate.

[1] http://www.iec.ch/worldplugs/typeJ.htm
[2] http://www.ati-ia.com/products/ft/sensors.aspx
[3] http://www.optitrack.com/





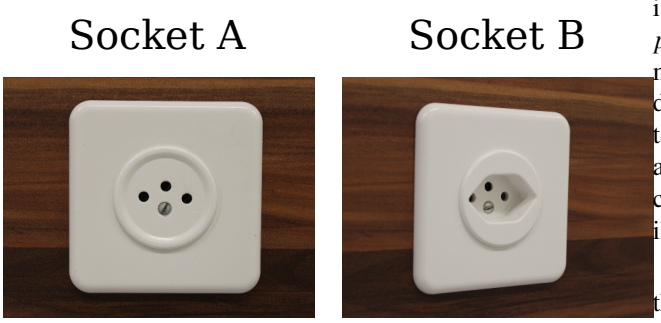


Figure 1: The experimental setup. *Top-left:* A participant (human teacher) is blindfolded and placed within the orange rectangular area always facing the wall. *Top-right:* Dimensions of the the wall and socket. *Bottom:* Three different power sockets, only socket A and B are used for data collection, socket C is purely used for evaluating the generalisation of the learned policy.

Figure 2: Human holding the cylinder plug holder, which is equipped with OptiTrack markers.

In this task, the human's location belief is represented by a probability distribution function. The participants' (teachers) initial belief is assumed to be uniformly distributed as depicted in orange area of Figure 1 and that all subsequent beliefs can be inferred from the measured velocity and measurements provided by the ATI and OptiTrack sensors. The following section describes how the belief can be represented, computed and compressed.

2.0.1. *Belief state*

For the task at hand, the belief probability density function, $p(x_t|y_{0:t}, \dot{x}_{0:t})$, is a Point Mass Filter (PMF) [**?** , p.87], which is a Bayesian filter. It is parametrised by a set of grid cells containing valid probabilities and is recursively updated by the application of a **motion**, $p(x_t|x_{t-1}, u_t)$ and **measurement**, $p(y_t|x_t)$ model. The motion model updates the position of the probability density function and subsequently increases the uncertainty of the position. The measurement model indicates areas of the state space from which a measurement $\tilde{y}_t$ could have originated. In Figure 3 (*Bottom-right*) we illustrate the likelihood when an edge is sensed.

A PMF is chosen to represent the believed location of the plug as the sensing the sensing likelihoods are non-gaussian and lead to multi-modal distributions. A PMF is able to capture such non-gaussianity whilst remaining fully deterministic (which is not the case for a particle filter).

The probability density function $p(x_t|y_{0:t}, \dot{x}_{0:t})$ is high dimensional and thus it is impractical to directly learn a statistical policy $\pi_\theta : p(x_t|y_{0:t}, \dot{x}_{0:t}) \rightarrow \dot{x}_t$ without some form of compression. One possibility would be E-PCA [**?**] which extracts a set of representative basis functions which are aslo probability distributions. Although elegant this method requires a discretisation of the belief space which is computationally expensive. Instead we choose to compress the pdf to a belief space vector composed of the maximum a posteriori, $\hat{x}_t^{\text{MAP}} = \arg\max_{x_t} p(x_t|y_{0:t}, \dot{x}_{0:t}) \in \mathbb{R}^3$, and the differentiation entropy, $U = H\{p(x_t|y_{0:t}, \dot{x}_{0:t})\} \in \mathbb{R}$. All pdfs in our recorded data set D are transformed to a belief space feature vector, $b_t = [\hat{x}_t^{\text{MAP}}, U]^{\text{T}}$.

Each participant's demonstration results in a dataset $D = \{\dot{x}_{1:T}^{[i]}, \omega_{1:T}^{[i]}, \phi_{1:T}^{[i]}, b_{1:T}^{[i]}\}$, where the upper index $[i]$ references the ith search trajectory (also one execution of the task or one episode) and subscript $1:T$ denotes the time steps during the trajectory from initialisation $t = 1$ until the end $t = T$.

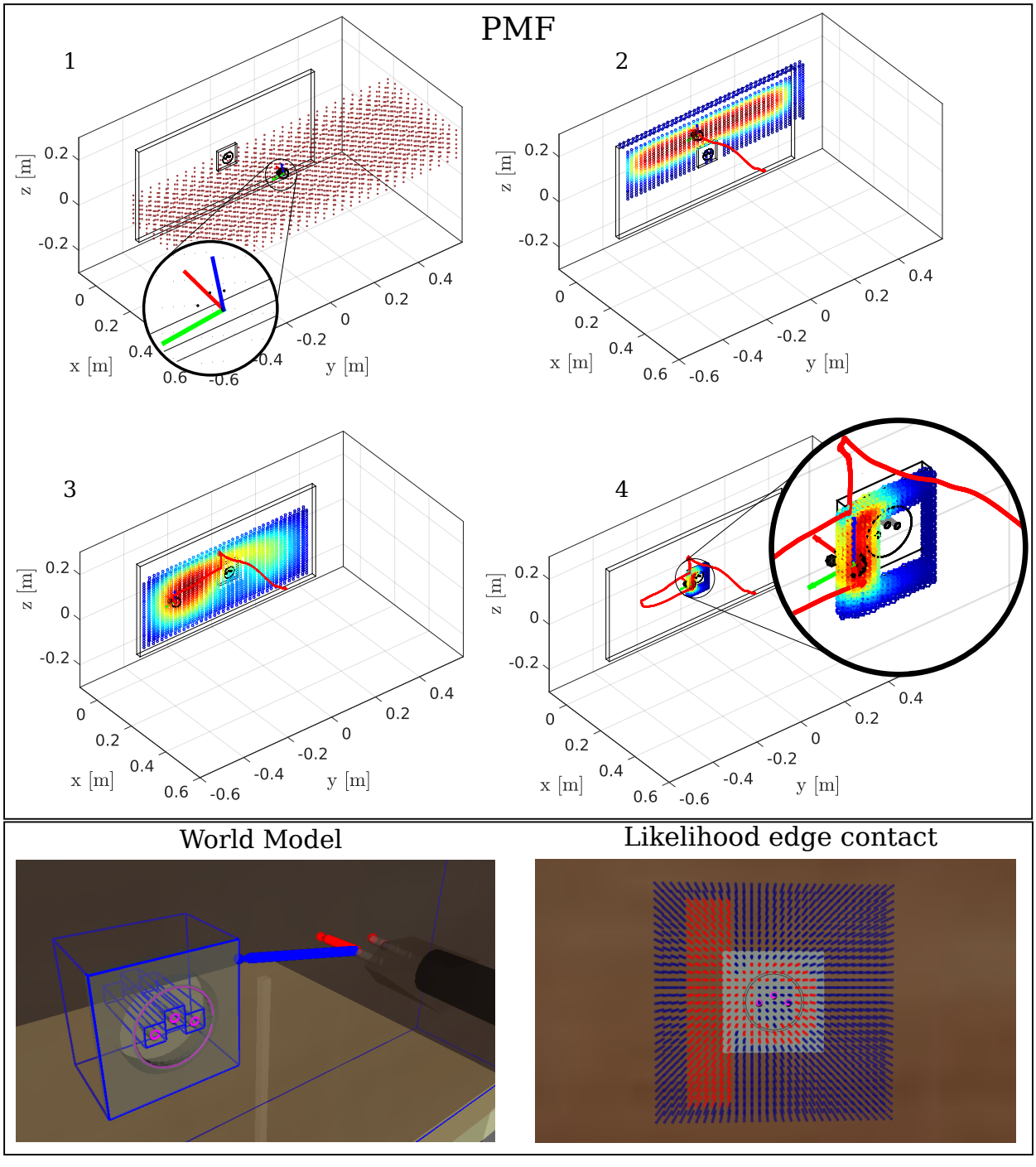


Figure 3: *Left:* Point Mass Filter (PMF) update of a particular human demonstration. (1) Initial uniform distribution spread over the starting region. Each grid cell represents a hypothetical position of the plug. The orientation is assumed to be known. (2) First contact, the distribution is spread across the surface of the wall. The red trace is the trajectory history. (3) motion noise increases the uncertainty. (4) The plug is in contact with a socket edge. *Right*: **World model**: The plug is modelled by its three plug tips and the wall and sockets are fitted with bounding boxes. **Likelihood**: The plug enters in contact with the left edge of the socket. As a result, the value of the likelihood in all the regions, x_t, close the left edge take a value of one (red points) whilst the others have a value zero (blue points) and areas around the socket's central ring have a value of one.

2.1. Participants and experiment protocol

To perform the PiH search tasks we recruited 10 student volunteers to be teachers (all male Master's and PhD students). The participants were aged between 24 and 30 with an average age of 26 years and a standard deviation of 2.4 years. Each participant carried out 30 demonstrations of the PiH search-task and each session lasted approximately 50 minutes and never exceeded one hour. The 10 participants were divided equally in two groups, A and B. Each member of group A began by performing 15 PiH searches with socket A, followed by a 10 minute break, finishing with an additional 15 searches with socket B. The members of group B performed the same protocol starting with socket B and ending with socket A. Figure 4 summarises a walk through of the experiment. The only exclusion criteria was the inability of the subject to accomplish the task. All participants gave written consent for taking part in this study.

The next section describes in detail the protocol for the search task:

1. Participant signs a form of consent before starting the experiment.
2. Each participant is given the opportunity to familiarise himself with the environment and become comfortable in wearing the sensor deprivation apparatus. During this time the participant is allowed to practice connecting the plug to the socket whilst standing within its vicinity.
3. Once the participant feels sufficiently ready to carry out the task to the best of his ability, the experimenter proceeds to disorient him through the usage of swivel chair. The disorientation process takes 30 seconds and includes both translation and rotation motions. After disorientation, the participant is signalled to stand up. The participant is reminded that he is facing the direction of the wall and that his starting location is within the orange rectangular area demarcated on the floor. He is then signalled by a light touch to the shoulder that he can start the task.
4. At task completion, the subject is once again disoriented and the process is repeated a total of 15 times. After 15 trials, the subject is given a 10 minute break whilst the experimenter changes the type of socket (A or B). A participant of group A will now continue with socket B. Similarly a participant of group B will continue after the break with socket A.

Each participant carried out a total of 30 PiH-search experiments, giving a total of 300 demonstrations.

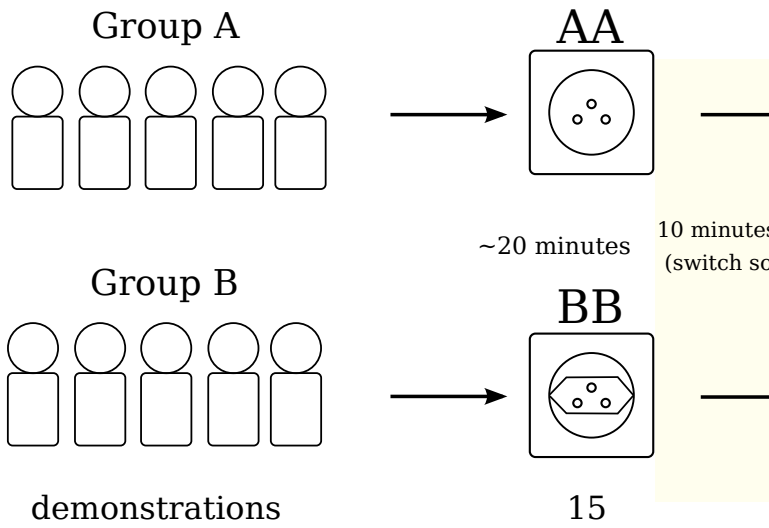


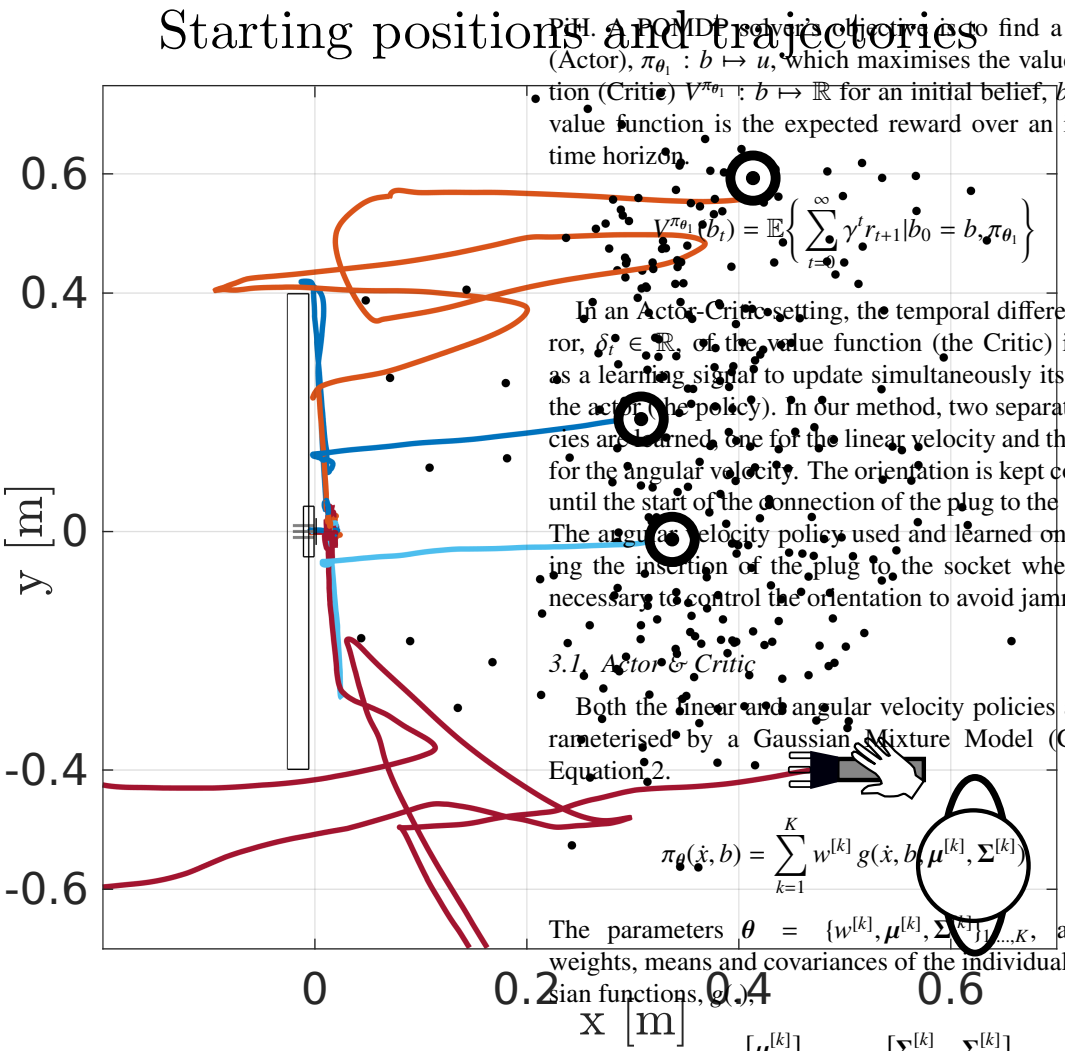
Figure 4: Experiment protocol. The participants are divided in two groups of 5, Group A begins with socket A and after a short break repeats the task with socket B. The same logic holds for Group B. For each socket 15 executions of the task are recorded.

Preliminary results. Both groups A and B took 9 ± 10s to find the socket's edge, regardless of the socket type. This is to be expected since the sockets are at the same location. It took a further 8 ± 7s on average for group B to connect socket B and 12 ± 10s on average for group A to connect socket A. As we can see this is not a straight forward task when considering the sensory deprivation. See Figure 5 (*Bottom*) for the time taken to connect the plug to the socket. In Appendix **??** we report the results of a non-parametric statistical analysis on the time taken to connect the sockets and we find that it takes 4 seconds more to connect socket A than socket B. This is somewhat expected as socket B has a funnel which can help to contain the subject to within the vicinity of the holes.

As connecting to socket A is more difficult we will **using only these demonstrations** as training data to learn a policy. Both socket B and C will be used solely to evaluate the generalisation of the policy.

3. Learning Actor and Critic

In our approach we learn two data driven policies. The first policy maps from belief space to linear velocity $\pi_{\theta_1} : b_t \mapsto \dot{x}_t$ and the second from angular sensed wrench to angular velocity, $\pi_{\theta_2} : \phi_t \mapsto \omega_t$. We chose to learn the belief policy π_{θ_1} in a Actor-Critic RL framework and the wrench policy π_{θ_2} directly from the demonstrated data as was done in [**?** , Chap. 5], which proved to be efficient in overcoming jamming during the



PiH. A POMDP solver's objective is to find a policy (Actor), $\pi_{\theta_1} : b \mapsto u$, which maximises the value function (Critic) $V^{\pi_{\theta_1}} : b \mapsto \mathbb{R}$ for an initial belief, b_0. The value function is the expected reward over an infinite time horizon.

$$V^{\pi_{\theta_1}}(b_t) = \mathbb{E}\left\{ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | b_0 = b, \pi_{\theta_1} \right\} \quad (1)$$

In an Actor-Critic setting, the temporal difference error, $\delta_t \in \mathbb{R}$, of the value function (the Critic) is used as a learning signal to update simultaneously itself and the actor (the policy). In our method, two separate policies are learned, one for the linear velocity and the other for the angular velocity. The orientation is kept constant until the start of the connection of the plug to the socket. The angular velocity policy used and learned only during the insertion of the plug to the socket where it is necessary to control the orientation to avoid jamming.

3.1. Actor & Critic

Both the linear and angular velocity policies are parameterised by a Gaussian Mixture Model (GMM), Equation 2.

$$\pi_{\theta}(\dot{x}, b) = \sum_{k=1}^{K} w^{[k]}\, g(\dot{x}, b, \boldsymbol{\mu}^{[k]}, \boldsymbol{\Sigma}^{[k]}) \quad (2)$$

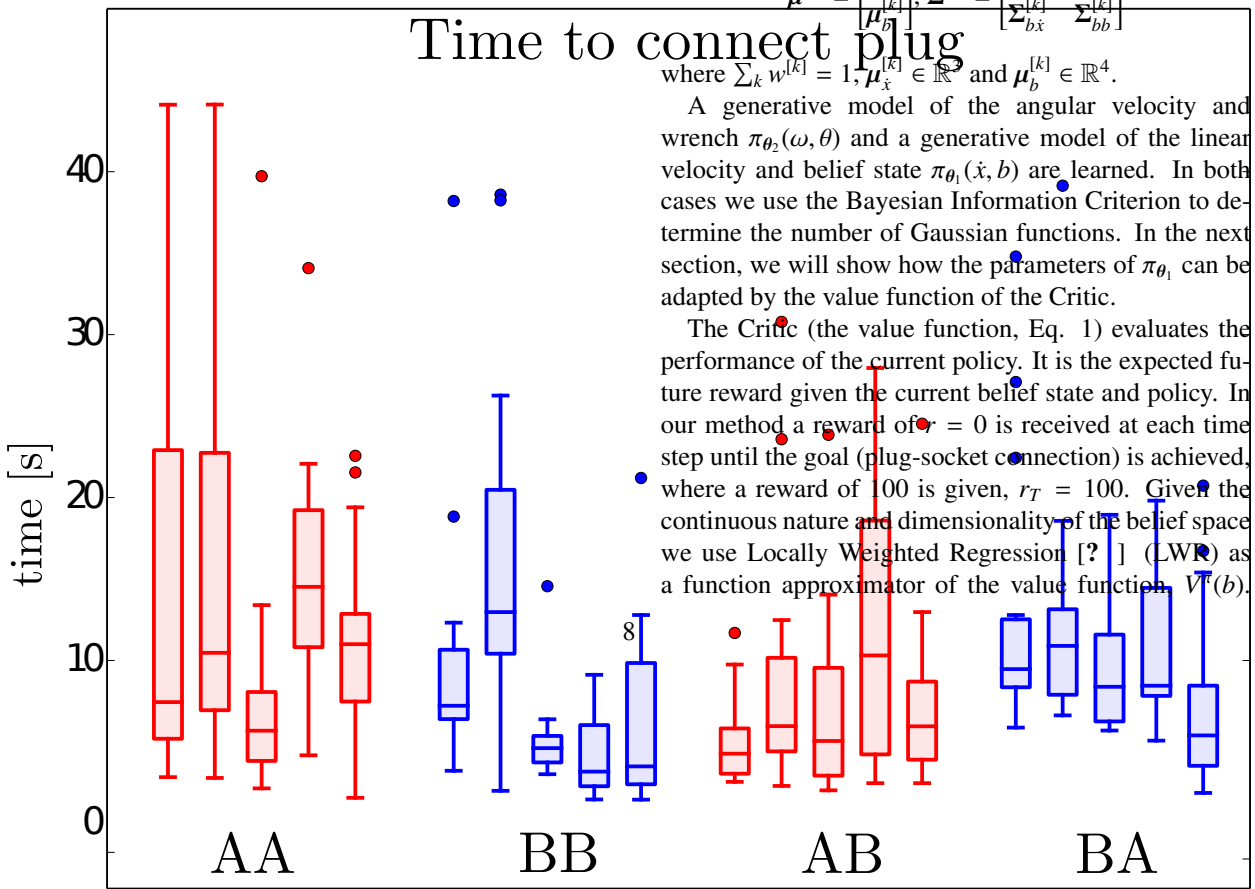
The parameters $\theta = \{w^{[k]}, \boldsymbol{\mu}^{[k]}, \boldsymbol{\Sigma}^{[k]}\}_{1,\ldots,K}$, are the weights, means and covariances of the individual Gaussian functions, $g(.)$,

$$\boldsymbol{\mu}^{[k]} = \begin{bmatrix} \boldsymbol{\mu}_{\dot{x}}^{[k]} \\ \boldsymbol{\mu}_{b}^{[k]} \end{bmatrix}, \boldsymbol{\Sigma}^{[k]} = \begin{bmatrix} \boldsymbol{\Sigma}_{\dot{x}\dot{x}}^{[k]} & \boldsymbol{\Sigma}_{\dot{x}b}^{[k]} \\ \boldsymbol{\Sigma}_{b\dot{x}}^{[k]} & \boldsymbol{\Sigma}_{bb}^{[k]} \end{bmatrix}$$

where $\sum_k w^{[k]} = 1$, $\boldsymbol{\mu}_{\dot{x}}^{[k]} \in \mathbb{R}^2$ and $\boldsymbol{\mu}_{b}^{[k]} \in \mathbb{R}^4$.

A generative model of the angular velocity and wrench $\pi_{\theta_2}(\omega, \theta)$ and a generative model of the linear velocity and belief state $\pi_{\theta_1}(\dot{x}, b)$ are learned. In both cases we use the Bayesian Information Criterion to determine the number of Gaussian functions. In the next section, we will show how the parameters of π_{θ_1} can be adapted by the value function of the Critic.

The Critic (the value function, Eq. 1) evaluates the performance of the current policy. It is the expected future reward given the current belief state and policy. In our method a reward of $r = 0$ is received at each time step until the goal (plug-socket connection) is achieved, where a reward of 100 is given, $r_T = 100$. Given the continuous nature and dimensionality of the belief space we use Locally Weighted Regression [?] (LWR) as a function approximator of the value function, $V^{\pi}(b)$.



LWR is a memory-based non-parametric function approximator. It keeps a set of input-target pairs $\{(b, r)\}$ as parameters. When a value, b, is queried, a set of p neighbouring points are chosen from the input space and are weighted according to a distance metric. The predicted output is given by a weighted least square of the p points. Equation 3 is the distance function used where D is a diagonal matrix.

$$W_{i,i} = \exp\left(-\frac{1}{2}(b - b_i)^{\mathrm{T}} D^{-1} (b - b_i)\right) \tag{3}$$

A new value is queried according to Equation 4,

$$V^{\pi}(b) = b\,(B^{\mathrm{T}} W B)^{-1} B^{\mathrm{T}} W \mathbf{r} \tag{4}$$

where $B = (b_1, \ldots, b_p)^{\mathrm{T}} \in \mathbb{R}^{(D \times p)}$, $W \in \mathbb{R}^{(p \times p)}$ is a diagonal matrix, $\mathbf{r} = (r_1, \cdots, r_p)^{\mathrm{T}} \in \mathbb{R}^{(p \times 1)}$

3.2. Fitted policy evaluation and improvement

Policy evaluation. To learn the value function we make use of batch reinforcement learning [**?**], also known as Experience replay. This is an offline method which applies multiple sweeps of the Bellman backup operator over a dataset of tuples $\{(b_t^{[i]}, \dot{x}_t^{[i]}, r_t^{[i]}, b_{t+1}^{[i]})\}_{i=1,\cdots,M}$ until the Bellman residual, $\|V_{k+1}^{\pi}(b) - V_k^{\pi}(b)\|$, converges.

Batch RL methods are used by a broad spectrum of research to learn policies. Most of them have focused on learning the Q-value function directly (Fitted Q-Iteration) [**?** **?** **?**]. Although this solves the control problem it requires discretisation of the action space or assumes quantifiable actions, as the Q-Bellman backups, $\hat{Q}(b_t, \dot{x}_t) \leftarrow \gamma \max_{\dot{x}_{t+1}} \hat{Q}(\dot{x}_{t+1}, b_{t+1})$, require an optimisation over the action space, $\dot{x}_{t+1}$, to find the best applicable action. Given the dimensionality and continuity of our problem we opt for an on-policy evaluation method which requires multiple *policy evaluation* and *policy improvement* iterations to achieve an optimal policy. In order for the RL-PbD-POMDP and PbD-POMDP to be comparable we will only be performing one iteration of policy evaluation and improvement, hence Algorithm **??** is applied only once to the dataset.

Policy improvement. The Temporal Difference (TD) error $\delta_t^{\pi} = r_{t+1} + \gamma V^{\pi}(b_{t+1}) - V^{\pi}(b_t)$ given by the critic is used to update the actor [**?** , Chap. 6]. In our offline approach the value function of the belief state, $\hat{V}^{\pi}(b)$, is estimated until convergence and then used to update the actor. This offline batch method has the advantage that no divergence can occur during the learning process.

We update the Actor policy given the Critic value function through a modification of the Maximisation step in Expectation-Maximisation (EM) for Gaussian Mixture Models. We refer to this modification as Q-EM which is strongly related to a Monte-Carlo EM-based policy search approach [**?** , p.50].

The reward of a demonstrated trajectory (one episode) is given by the discounted return, Equation 5,

$$R(b^{[i]}, \dot{x}^{[i]}) = \sum_{t=0}^{\mathrm{T}^{[i]}} \gamma^t\, r(b_t^{[i]}, \dot{x}_t^{[i]}) \tag{5}$$

where the index i stands for the ith episode. All policy gradient approaches seek to find a set of parameters, $\boldsymbol{\theta}$, of the Actor, which will maximise the expected reward, equivalent to maximising Equation 6,

$$\begin{aligned} J(\boldsymbol{\theta}) &= \mathbb{E}_{p_{\boldsymbol{\theta}}}\{R\} \\ &= \sum_{i=1}^{N} \underbrace{\left(\prod_{t=0}^{T^{[i]}} \pi_{\boldsymbol{\theta}}(\dot{x}_t^{[i]}, b_t^{[i]})\right)}_{p_{\boldsymbol{\theta}}(\tau_i)} R(\tau_i) \end{aligned} \tag{6}$$

where $\tau_i = \{(\dot{x}_0, b_0), \cdots, (\dot{x}_T^{[i]}, b_T^{[i]})\}$ are the state-action samples of the ith episode. To find the parameters which maximise the cost function, $\arg\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$, its derivative is set to zero. As this cannot be done directly, we maximise the logarithmic lower bound of the cost function which results in Equation 7, see Appendix **??** for the derivation.

$$\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=1}^{N} \sum_{t=0}^{T^{[i]}} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\dot{x}_t^{[i]}, b_t^{[i]})\, Q^{\pi_{\boldsymbol{\theta}'}}(\dot{x}_t^{[i]}, b_t^{[i]}) \tag{7}$$

Setting the derivative of Equation 7 to zero and solving for the parameters $\boldsymbol{\theta} = \{w, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ leads to a Maximisation update step of EM which is weighted by $Q^{\pi_{\boldsymbol{\theta}'}}$. We use the Critic's TD error as a substitute for Q^{π}. Assuming that our estimated value function, $\hat{V}^{\pi}$, is close to the true value function V^{π}, the TD error δ^{π} is an unbiased estimate of the advantage function, Equation 8 (see Appendix **??**).

$$A^{\pi}(\dot{x}_t, b_t) = Q^{\pi}(\dot{x}_t, b_t) - V^{\pi}(b_t) = \delta_t^{\pi} \tag{8}$$

Using the advantage function as means of policy search is popular with methods such as Natural Actor Critic (NAC) [**?**].

Each state-action sample j has an associated weight, $\delta_j \in \mathbb{R}$, where $\delta_j > 0$ means that the jth state action-pair lead to an increase in the value function and $\delta_j < 0$ lead to a decrease in the value function. The data log-likelihood is re-weighted accordingly, giving more importance to data points which lead to a gain. Since the

Q-EM update steps cannot allow negative weights, the TD error is rescaled to be between 0 and 1.

The reader is referred to Appendix **??** for the Maximisation update step of Q-EM for a GMM parameterization of the policy.

2D example fitted policy evaluation and improvement. To illustrate the mechanism of fitted policy evaluation and improvement, we give a 2D example of its application, see Figure 6. The *Top-left* subfigure depicts 10 trajectories demonstrated by two teachers going from start (white circle) to goal (orange star) state. The optimal path is a straight line passing in between two obstacles. Neither teacher demonstrated the optimal straight path.

In the *Bottom-left*, a GMM is fitted $\pi_{\theta}(\dot{x}, x)$ to the teachers' data, using the standard EM-algorithm. Taking the policy to be the output of Gaussian Mixture Regression (GMR) $\mathbb{E}\{\pi_{\theta}(\dot{x}|b)\}$ we obtain different behaviours than those demonstrated by the human teachers. The GMR averages the different modes encoded by the Gaussian functions which results in a mixing of the original demonstrated behaviours. No trajectories of the GMR policy truly replicate the demonstrated behaviour.

In the *Top-right* subfigure, we apply fitted policy evaluation to the original demonstrated data (discount factor $\gamma = 0.99$ and reward $r = 1$ when the goal is reached and zero otherwise) and compute the value function.

The *Bottom-right* subfigure illustrates the GMM policy learned with the Q-EM algorithm. As the advantage function $A^{\pi}(x, \dot{x})$ is highest along the start-goal axis, data points following this gradient will have a higher weight. This results in a policy with better rollouts (closer to the optimal path) than the trajectories generated by the policy learned via standard EM.

Belief state fitted policy evaluation. Returning to the PiH-search task with socket A, the Fitted Policy Evaluation (FPE) Algorithm **??** is applied to the demonstrations. In Figure 7 we illustrate the value function of the most likely state after the FPE algorithm converges. As expected, the value function is high closest to the socket and around the axis $z = 0$ and $y = 0$. When policy improvement via Q-EM is applied the Gaussian functions of the GMM will favour these locations.

In Figure 8 we illustrate the best and worst trajectories in terms of the accumulated value function. We can see that the five best trajectories (red) tend to be aligned with the socket (star position in front of socket), whilst the worst trajectories are towards the edges of the wall and tend to follow spiralling movements.

We learned two policies, one solely from the original human demonstrations which we call GMM and the sec-

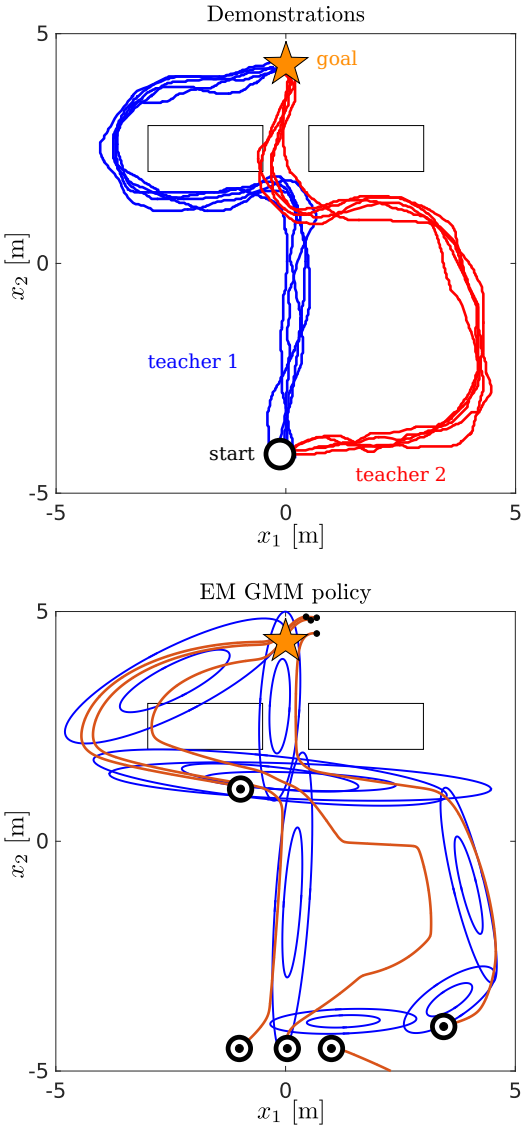


Figure 6: Fitted policy evaluation & improvement example. *Top-left:* The goal of the task is to reach the goal state. The first teacher (blue) demonstrates five trajectories which contours the obstacle in front of the goal. The second teacher (red) demonstrates 5 trajectories which initially deviate from the goal before passing between the two obstacles. *Bottom-left:* The EM algorithm is used to fit a GMM to the teachers' original data. The marginal $\pi_{\theta}(x)$ is plotted in blue and trajectories generated by the policy $\mathbb{E}\{\pi_{\theta}(\dot{x}|x)\}$ in orange. *Top-right Policy Evaluation:*. Value function after fitted policy evaluation terminated, the reward function is binary, $r = 1$ at the goal and zero otherwise, and a discount factor $\gamma = 0.99$ is used. *Bottom-right Policy Improvement:* the GMM is learned with the Q-EM algorithm in which each data point's weight proportional to the advantage function.

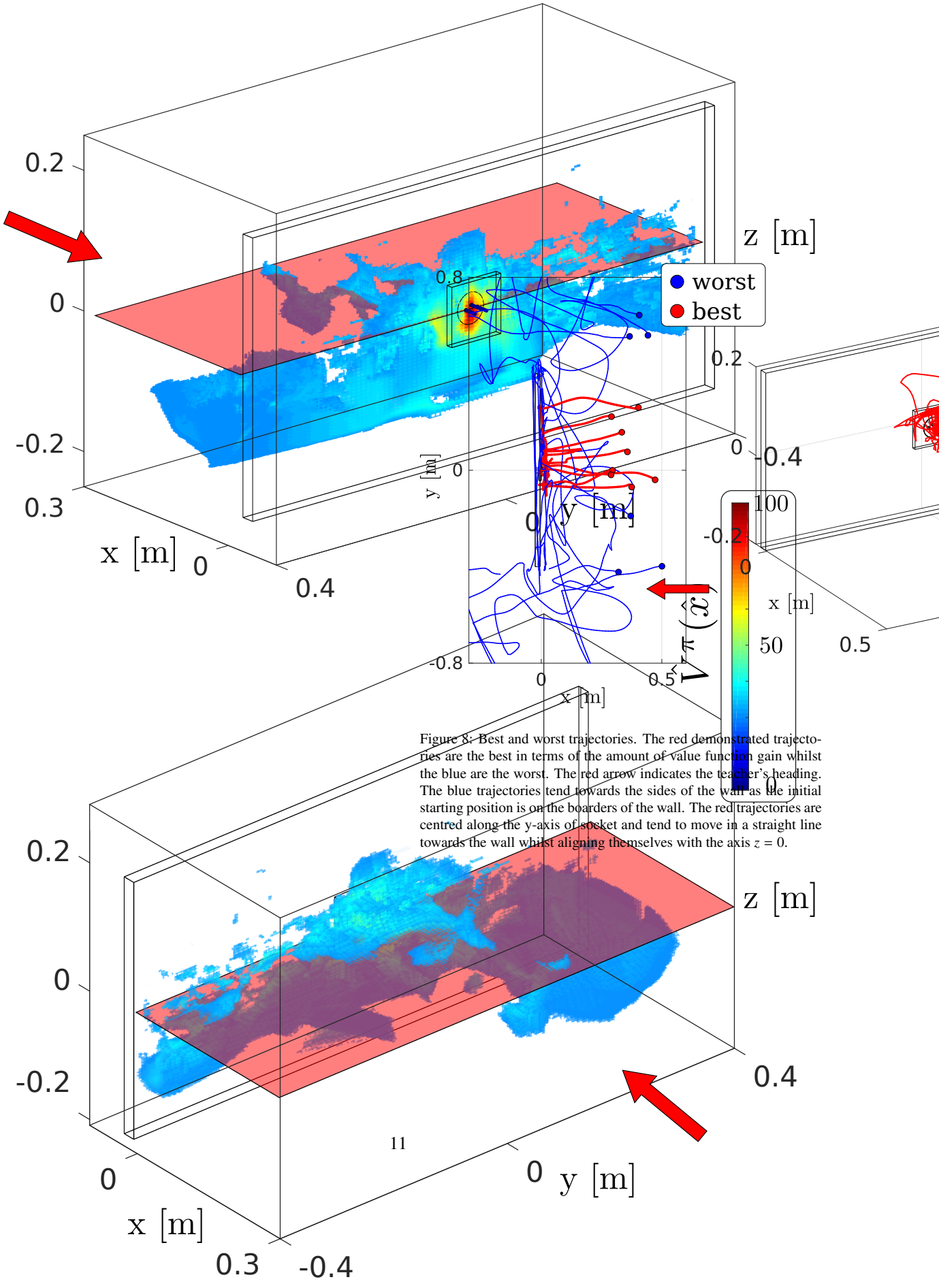


Figure 8: Best and worst trajectories. The red demonstrated trajectories are the best in terms of the amount of value function gain whilst the blue are the worst. The red arrow indicates the teacher's heading. The blue trajectories tend towards the sides of the wall as the initial starting position is on the boarders of the wall. The red trajectories are centred along the y-axis of socket and tend to move in a straight line towards the wall whilst aligning themselves with the axis $z = 0$.

Figure 7: LWR value function approximate $\hat{V}^{\pi}(\hat{x})$ for the most likely

ond which is the result of one iteration of fitted policy evaluation and improvement which we call Q-EM. In the section 5 we compare the GMM and Q-EM policies with the improvements which can be achieved within the RL-PbD-POMDP framework.

4. Control architecture

As detailed in section 3.2, a Gaussian Mixture Model was learned for both linear and angular velocity, although only the linear control policy is active until the plug is within the socket's hole, as the orientation is constant. The direction to search is given by the conditional, Equation 9,

$$\pi_{\boldsymbol{\theta}}(\dot{x}|b) = \sum_{k=1}^{K} w_{\dot{x}|b}^{[k]}\, g(\dot{x}; \boldsymbol{\mu}_{\dot{x}|b}^{[k]}, \boldsymbol{\Sigma}_{\dot{x}|b}^{[k]}) \tag{9}$$

which is a distribution over the possible normalised velocities. The function $g(\cdot)$ is a multivariate Gaussian function parameterised by mean $\boldsymbol{\mu}_{\dot{x}|b}^{[k]} \in \mathbb{R}^{(3\times1)}$ and Covariance $\boldsymbol{\Sigma}_{\dot{x}|b}^{[k]} \in \mathbb{R}^{(3\times3)}$. The subscript $\dot{x}|b$ indicates that the parameters are the result of the conditional. The reader is referred to [**?**], [**?**] for a detailed derivation of the conditional of a GMM. The learned model is multi-modal, as different search velocities are possible in the same belief state. Figure 9 illustrates the multi-modal vector fields of the conditional, Equation 9. In autonomous dynamical systems control, the velocity is obtained from the expectation of the conditional, Equation 9. However, the expectation which is a weighted linear combination of the modes, could result in unobserved behaviour or no movement if the velocities cancel out. As a result we use a modified version of the expectation operator which favours the current direction, Equation 10 - 11.

$$\alpha(\dot{x}) = w_{\dot{x}|b}^{[k]} \cdot \exp(-\cos^{-1}(< \dot{x}, \boldsymbol{\mu}_{\dot{x}|b}^{[k]} >)) \tag{10}$$

$$\dot{x} = \mathbb{E}_{\alpha}\{\pi_{\boldsymbol{\theta}}(\dot{x}|b)\} = \sum_{k=1}^{K} \alpha_k(\dot{x}) \cdot \boldsymbol{\mu}_{\dot{x}|b}^{[k]} \tag{11}$$

When the applied velocity mode is no longer present another direction is sampled. For example, when the robot enters in contact with a feature, greatly reducing the uncertainty, the current mode changes and a new search direction is computed. Figure 9 illustrates the policy vector field for GMM and Q-EM, both learned from teachers demonstrations.

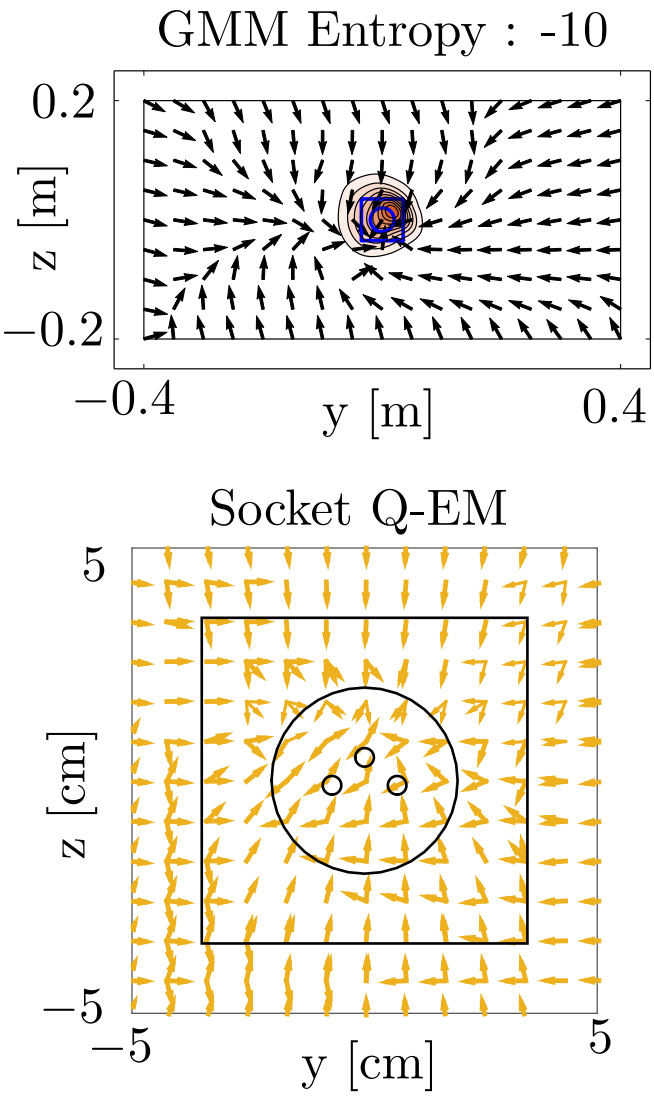


Figure 9: Q-EM and GMM policy vector fields. *Top*: The GMM policy is conditioned on an entropy of −10 and −5.2. For the lowest entropy level, most of the probability mass is close to the socket area since this level corresponds to very little uncertainty; we are already localised. We can see that the policy converges to the socket area regardless of the location of the believed state. For an entropy of −5.2 we can see that the likelihood of the policy is present across wall. The vector field directs the end-effector to go towards the left or right edge of the wall. *Bottom*: The entropy is marginalised out, the yellow vector field is of the Q-EM and orange of the GMM. The Q-EM vector field tends to be closer to a sink and there is less variation.

4.1. Robot Implementation

The GMM policy $\underline{\dot{x}} = \mathbb{E}_\alpha\{\pi_{\boldsymbol{\theta}}(\dot{x}|b)\}$ outputs a linear velocity which is normalised, $\underline{\dot{x}} \in \mathbb{R}^{(3\times1)}$. The amplitude of the velocity is computed separately and modulated according to sensed forces on the end-effector. This search task is haptic and the end-effector of the robot is always in contact with the environment. To make the robot compliant with the environment we use an impedance controller in combination with a hybrid position-force controller. A hybrid controller targets a sensed force F_x, in the x-axis, of 3N. The y and z velocity components of the direction vector are given by Equation 11. This is insufficient for the robot to reliably surmount the edges of the socket, hence the vector field of the GMM is modulated in y and z-axis, Equation 12.

$$\underline{\dot{x}} = R_y(c(F_z)\cdot\pi/2)\cdot R_z(c(F_y)\cdot\pi/2)\cdot\underline{\dot{x}} \qquad (12)$$

where R_y and R_z are (3×3) rotation matrices around the y and z-axis, and $c(F) \in [-1, 1]$ is a truncated scaling function of the sensed force. When a force F_z of 5N is sensed, a rotation of $R_y(\pi/2)$ is applied to the original direction resulting in the robot getting over the edge. The direction velocity is always normalised up to this point. The amplitude of the velocity is a proportional controller based on the believed distance to the goal,

$$\begin{aligned} v &= \max(\min(\beta_1, K_p(x_g - \hat{x}), \beta_2) \qquad (13)\\ \dot{x} &= v\underline{\dot{x}} \end{aligned}$$

where the lower and upper amplitude limits are given by β_1 and β_2, x_g is the position of the goal, and K_p the proportional gain which was tuned through trials.

The above procedure can control the general behaviour of the search but is insufficient for a successful implementation on a robotic system such as the 7 Degree of Freedom $q \in \mathbb{R}^7$ KUKA LWR, which we illustrate in Figure 10. The GMM policy $\dot{x} = \mathbb{E}_\alpha\{\pi_{\theta_1}(\dot{x}|b)\}$ outputs a linear velocity and the angular velocity is computed from a reference orientation which is constant. When the plug is to be connected to the socket, the angular velocity is the output of samples drawn from the conditional $\omega \sim \pi_{\theta_2}(\omega|\phi)$. From both linear and angular velocities a reference position $x^r \in \mathbb{R}^{(3\times1)}$ and orientation $R^r \in \mathbb{R}^{(3\times3)}$ are computed and used to define a linear and angular error $x_e = x^r - x$, $\psi_e = \text{angleaxis}(R^{\mathrm{T}}R^r)$ by using the the current position x and orientation R. Given the kinematic chain of the robot, the inverse of the Jacobian $J(q) \in \mathbb{R}^{6\times7}$ is used in an impedance control to transform the Cartesian error $c_e = [x_e, \psi_e]^{\mathrm{T}} \in \mathbb{R}^{6\times1}$ to torque commands $\tau_t \in \mathbb{R}^7$, Equation 14,

$$\tau_t = J^{\mathrm{T}}(q_t)\,(-Kc_e - D\dot{c}_e) + g(q_t) \qquad (14)$$

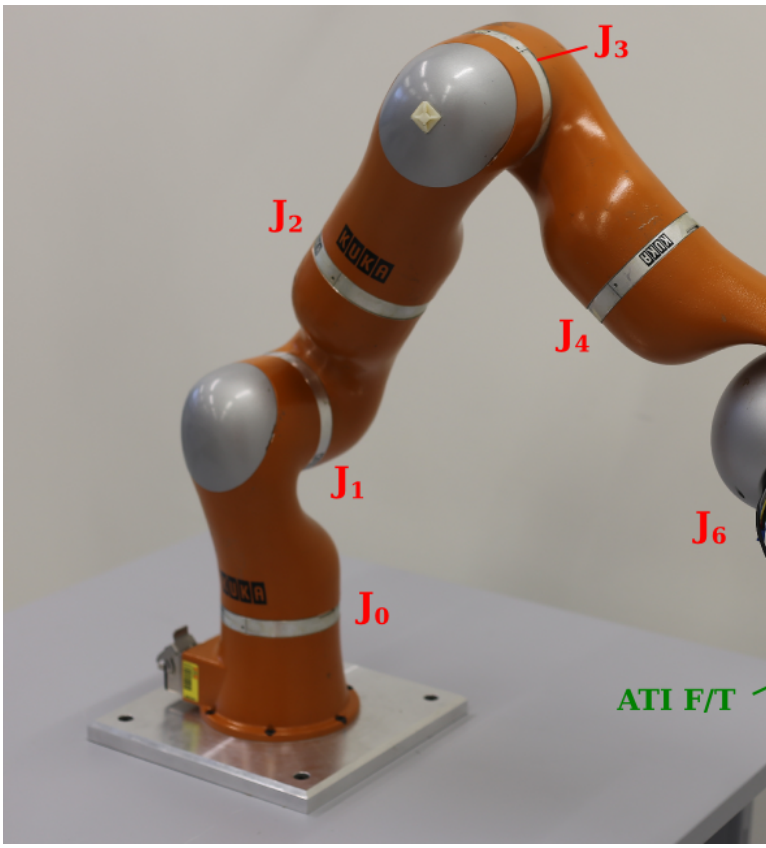


Figure 10: The KUKA LWR is a 7 Degree Of Freedom (DoF) robot, we illustrate in red each joint, which is controlled at a rate of 1kHz via an ethernet cable. The KUKA API provides a command interface to the stiffness, damping, position and torque variables of each joint.

where $K, D \in \mathbb{R}^{6\times6}$ are diagonal stiffness and damping matrices whose values were set experimentally and $g(q_t)$ compensates for gravity. Given an applied torque there is a resulting joint velocity $\dot{q}_t$ from which we can compute the measured Cartesian end-effector velocity used in the motion model of the PMF. Figure 11 illustrates the complete control flow.

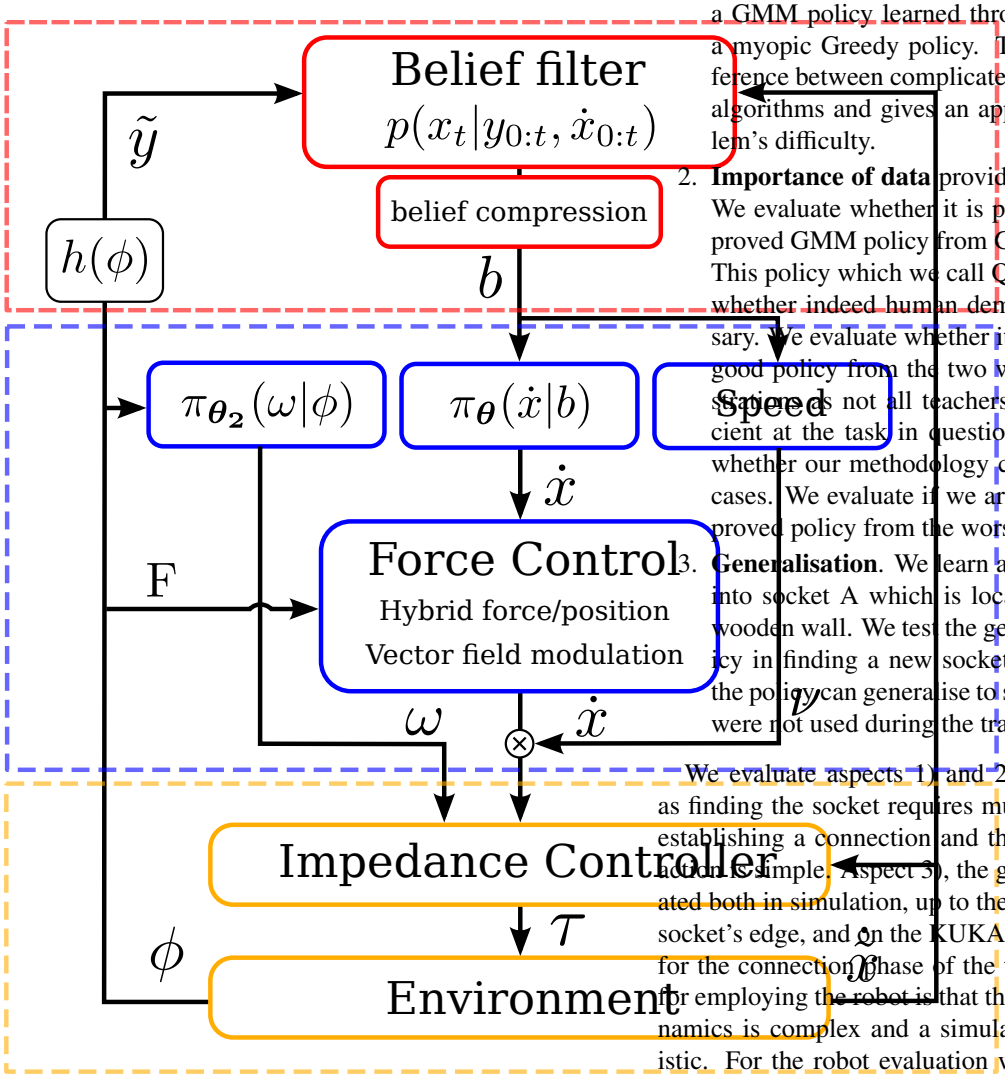


Figure 11: Control architecture. The PMF (belief) receives a measured velocity, $\dot{\tilde{x}}$, and a sensor measurement $\tilde{y}$ and is updated via Bayes rule. The belief is compressed and used by both the GMM policy and the proportional speed controller, Equation 13.

5. Results

We evaluate the following three aspects of the policy learned in our Actor-Critic framework:

1. **Distance taken to accomplish the goal** (connect plug to socket). We compare the Q-EM policy with a GMM policy learned through standard EM and a myopic Greedy policy. This highlights the difference between complicated and simplistic search algorithms and gives an appreciation of the problem's difficulty.
2. **Importance of data** provided by human teachers. We evaluate whether it is possible to learn an improved GMM policy from Greedy demonstrations. This policy which we call Q-Greedy is used to test whether indeed human demonstrations are necessary. We evaluate whether it is possible to obtain a good policy from the two worst teachers' demonstrations as not all teachers are necessarily proficient at the task in question and we want to test whether our methodology can be applied in these cases. We evaluate if we are able to obtain an improved policy from the worst two teachers.
3. **Generalisation**. We learn a policy to insert a plug into socket A which is located at the center of a wooden wall. We test the generalisation of the policy in finding a new socket location and whether the policy can generalise to sockets B and C, which were not used during the training phase.

We evaluate aspects 1) and 2) purely in simulation as finding the socket requires much less precision than establishing a connection and the physics of the interaction is simple. Aspect 3), the generalisation, is evaluated both in simulation, up to the point of localising the socket's edge, and on the KUKA LWR robotic platform for the connection phase of the task. The main reason for employing the robot is that the connection phase dynamics is complex and a simulation would be unrealistic. For the robot evaluation we consider the search starting already within the vicinity of the socket.

5.1. *Distance taken to reach the socket's edge (Qualitative)*

We consider three search experiments which we refer to as **Experiment 1, 2** and **3**, in order to evaluate the performance in terms of the distance travelled to reach the socket for the three search policies: GMM, Q-EM and Greedy. In these three experiments the task is considered accomplished when a search policy finds the socket's edge.

In **Experiment 1**, three starting locations are chosen: *Center*, *Left* and *Right*. See Figure 12, *Experiment 1*, for an illustration of the initial condition. This setup tests the effect of the starting positions. A total of 25 searches are carried out for each of the search policies.

In **Experiment 2**, two *Cases* are chosen in which the believed state (most likely state of the PMF) and the true position of the end-effector are relatively far apart. The location of the beliefs are chosen to be symmetric, see the Figure 12, *Experiment 2*. A total of 25 searches are carried for each of the two cases.

In **Experiment 3**, Figure 12, *Experiment 3*, the initial true starting positions of the end-effector are taken from a regular grid covering the whole start region, also used as the initial distribution for the human demonstrations. A total of a 150 searches are carried out for each of the three policies. This experiment compares the search policies with the human teachers' demonstrations.

We evaluate the performance of the three experiments in terms of the trajectories and their distribution in reaching the edge of the socket.

In **Experiment 1**, see Figure 12 *Experiment 1, second row* the results show a clear difference between the trajectories generated by the GMM and Q-EM policies. The orange GMM policy trajectories go straight towards the wall, whilst the yellow Q-EM policy trajectories drop in height making them closer to the socket. The same effect can be seen in Experiment 2 (*second row*). The Q-EM trajectories follow a downward trend towards the location of the socket. The gradient is less as the initial starting condition is lower than in Experiment 1.

In **Experiment 2**, see Figure 12, *Experiment 2, second row*, the trajectories of the Greedy policy depend on the chosen believed location (most likely state of the PMF). There is no variance in the Greedy's trajectories until it reaches the edge of the red square, where the branching occurs as the believed location is disqualified. This happens as no sensation has been registered at the point when the believed location reaches the wall. The true location is in fact situated further away from the wall than the believed location.

In **Experiment 3**, see Figure 12 *Experiment 3, second row*, Human and GMM show similar distributions of searched locations. They cover the upper region of the wall and top corners, to some extent. These distributions are not identical for two reasons. The first is that the learning of the GMM is a local optimisation which is dependent on initialisation and number of parameters. The second reason is that the synthesis of trajectories from the GMM is a stochastic process.

For the Q-EM policy, the distribution of the searched

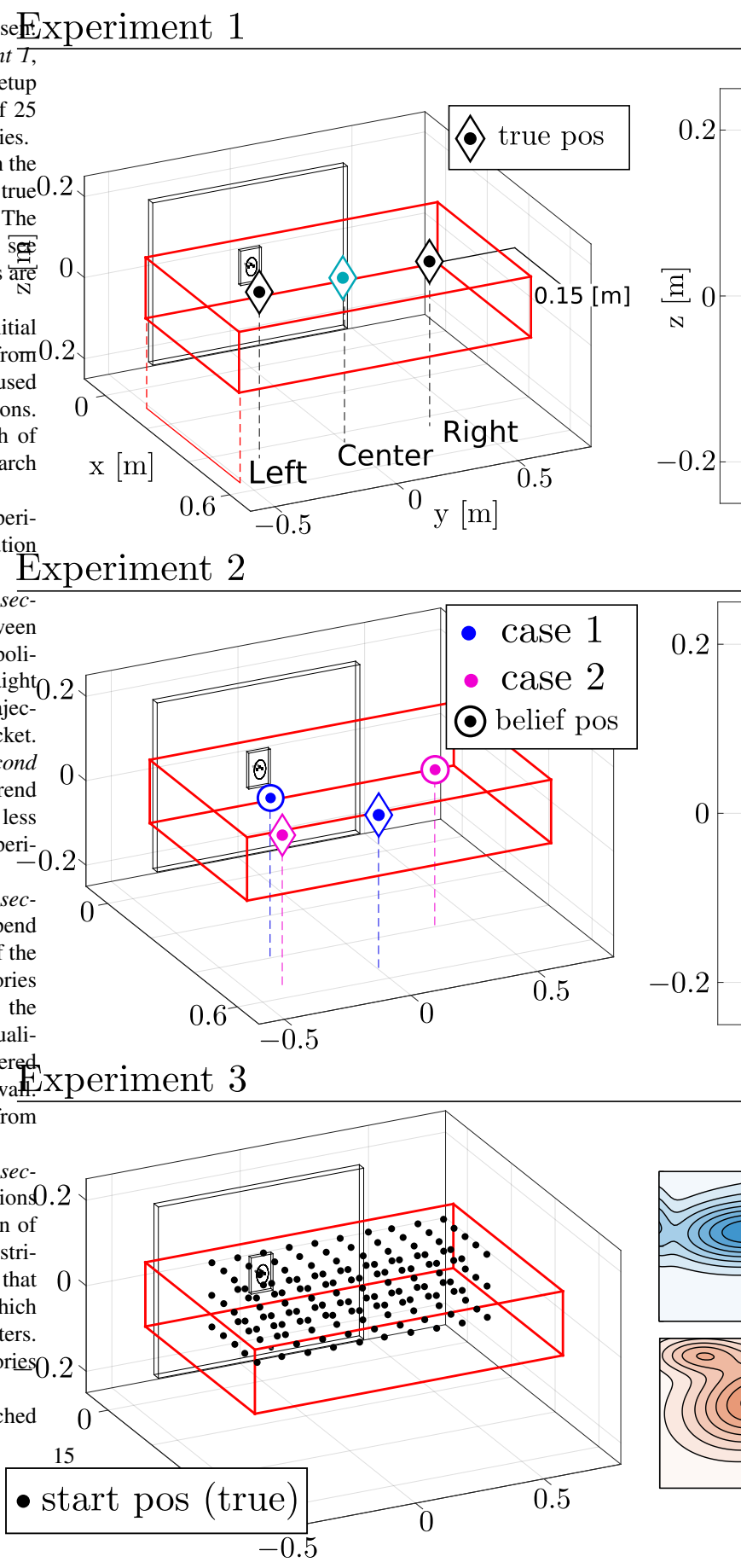


Figure 12: Three simulated search experiments. **Experiment 1:** Three start positions are considered: *Left*, *Center* and *Right* in which the triangles depict true position of the end-effector. The red cube illustrates the extent of the uncertainty. In the second row of Exper-

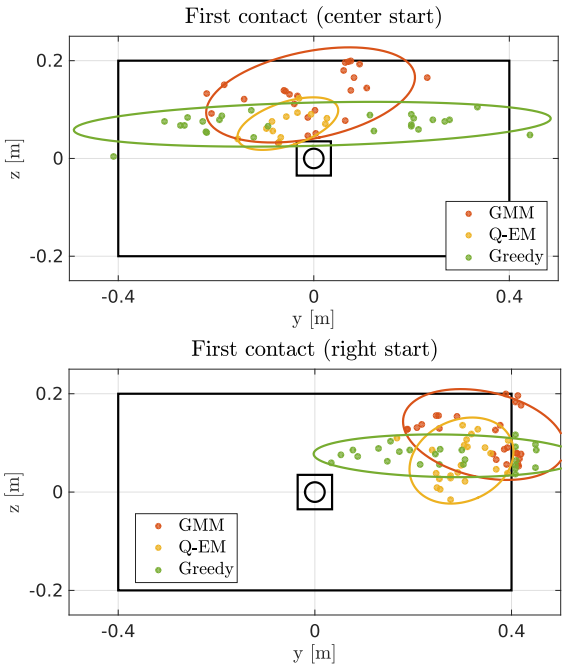


Figure 13: First contact with the wall, during experiment 1. (a) Contact distribution for initial condition "Center" . (b) Contact distribution for initial condition was "Right". The ellipses correspond to two standard deviations of a fitted Gaussian function.

locations is centred around the origin of the z-axis. The uncertainty is predominantly located in the x and y-axis. The Q-EM policy takes this uncertainty into consideration by restraining the search to the y-axis regardless of the starting position. The uncertainty is reduced when it is in the vicinity of the socket. The Greedy's policy search distribution is multi-modal and centred around the z-axis where the modes are above and below the socket. This shows that the Greedy policy acts according to the most likely state which changes from left to right of the socket, because of motion noise, resulting in left-right movements and little displacement. As a result the Greedy policy spends more time at these modes.

In Figure 13 (*Top-left*), we illustrate the distribution of the first contact with the wall during Experiment 1 for the *Center* initial conditions. The distribution of the first contact of the Greedy method is uniform across the entire y-axis of the wall. It does not take into account the variance of the uncertainty. In contrast, the GMM policy remains centred with respect to the starting position and the Q-EM is even closer to the socket and there is much less variance in the location of the first contact.

5.2. Distance taken to reach the socket's edge (Quantitative)

In Figure 14 we illustrate the quantitative results of the distance taken to reach the socket for all three experiments. In **Experiment 1**, for the *Center* initial condition, the Q-EM policy travels far less than the other search policies. Considering that the initial position of the search is 0.45 [m] away from the wall, the Q-EM policy finds the socket very quickly once contact has been established with the wall. For the *Right* and *Left* starting conditions both the GMM and Q-EM policies travel less distance to reach the socket, with a smaller variance when compared with the Greedy search policy.

In **Experiment 2**, Figure 14, the Q-EM search policy is the most efficient. For *Case 1* of Experiment 2, the initial most likely state is fixed to the left and the true position is facing the socket. As the belief is chosen to be to the left, upon contact with the wall the policy takes a left action since it is more likely to result in a localisation. On average this results in an exploration of the upper left area of the wall, which explains why *Case 1* of Experiment 2 performs worse than Experiment 1 for the *Center* initial condition. In *Case 2* however, where the true state is facing the left edge and the believed position is facing the right edge, less distance is taken to find the socket than for *Case 1*, Figure 14 (b). This improvement over *Case 1* is due to the true location of the end-effector being closer to an informative feature and results in a much faster localisation.

From **Experiment 3**, Figure 15, it is clear that all three search policies travel less to find the socket's edge compared with the teachers' demonstrations. All search policies are better than the human teachers with the exception of group B*, which is performing the task with socket A. The Q-EM policy remains the best.

We have shown that under three different experimental settings the Q-EM algorithm is predominantly the best in terms of distance taken to localise the socket. The GMM policy learned solely from the data provided by the human teachers also performs well in comparison to the human teachers and Greedy policy. We made, however a critical assumption in order to be able to use our (RL-)PbD-POMDP approach. This **assumption** is that a human teacher is proficient in accomplishing the task. If a teacher is not able to accomplish the task in a repetitive and consistent way so that a search patter can be encoded by the GMM, the learned policy will perform poorly. Next we evaluate the validity of this assumption and the importance of the training data provided by the human teachers.

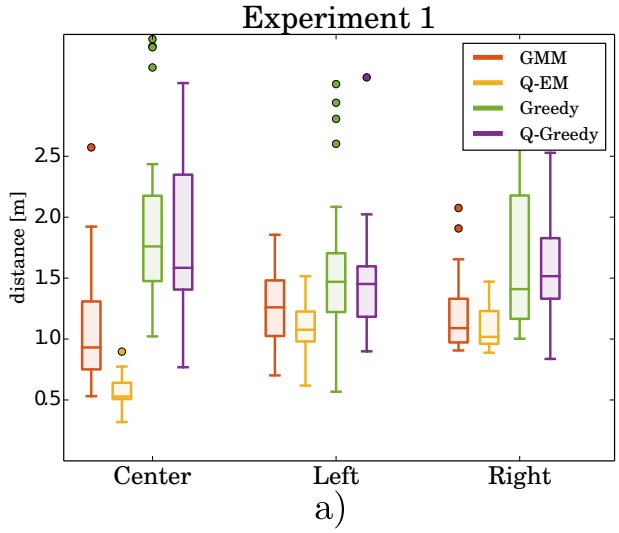


Figure 14: Distance travelled until the socket's edge is reached. a) Three groups correspond to the initial conditions: Center, Left and Right depicted in Figure 12, *top left*. The Q-EM method is always better than the other methods, in terms of distance. b) Results of the two initial conditions depicted in Figure 12, *top middle*, both the true position and most likely state are fixed. The Q-EM method always improves on the GMM.

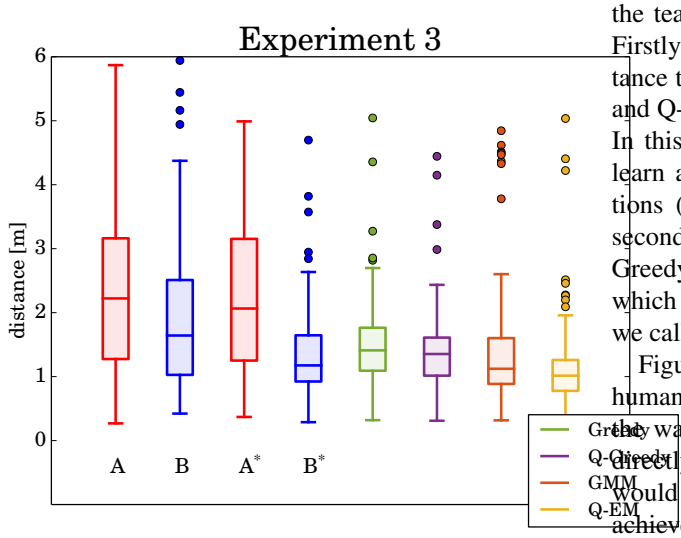


Figure 15: Distance travelled until the socket's edge is reached. Results corresponding to Experiment 3, Figure 12, *top right*. Again the Q-EM method is better, but at a less significant level.

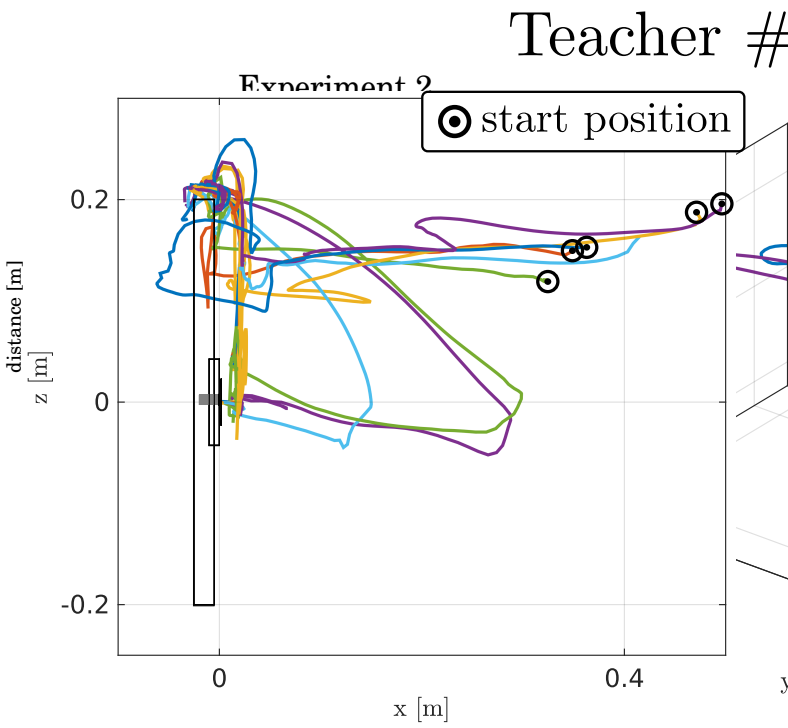


Figure 16: Demonstrations of teacher # 5. The teacher demonstrates a preference

5.3. *Importance of data*

We perform two tests to evaluate the importance of the teachers training data for learning a search policy. Firstly we take the worst two teachers in terms of distance taken to find the socket's edge and learn a GMM and Q-EM policy separately from their demonstrations. In this way we can evaluate whether it is possible to learn a successful policy given a few bad demonstrations (15 training trajectories for each policy). Our second evaluation consists of using a noisy explorative Greedy policy as a teacher to gather demonstrations which can then be used to learn a new policy, which we call Q-Greedy.

Figure 16 illustrates 6 trajectories of teacher # 5. The human teacher preferred to localise himself at the top of the wall before either proceeding to a corner or going directly towards the socket. Once localised, the teacher would reposition himself in front of the socket and try to achieve an insertion. This behaviour was not expected since by losing contact with the wall, the human teacher no longer had sensory feedback necessary to maintain an accurate position estimate.

Figure 17 illustrates the value function of the belief state learned from the data of teacher # 5. The states with the highest values seem to create a path going from

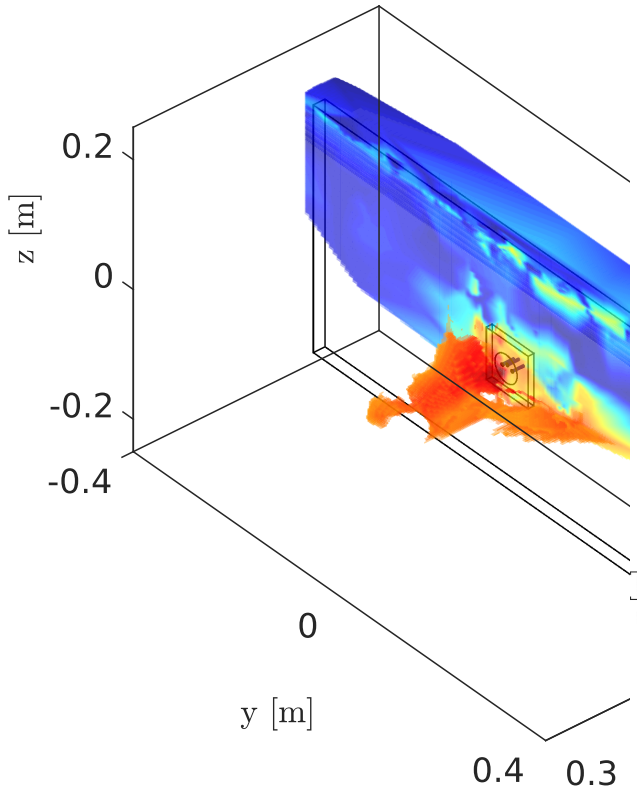


Figure 17: Value function learned from the 15 demonstrations of teacher #5. The value of the most likely state is plotted.

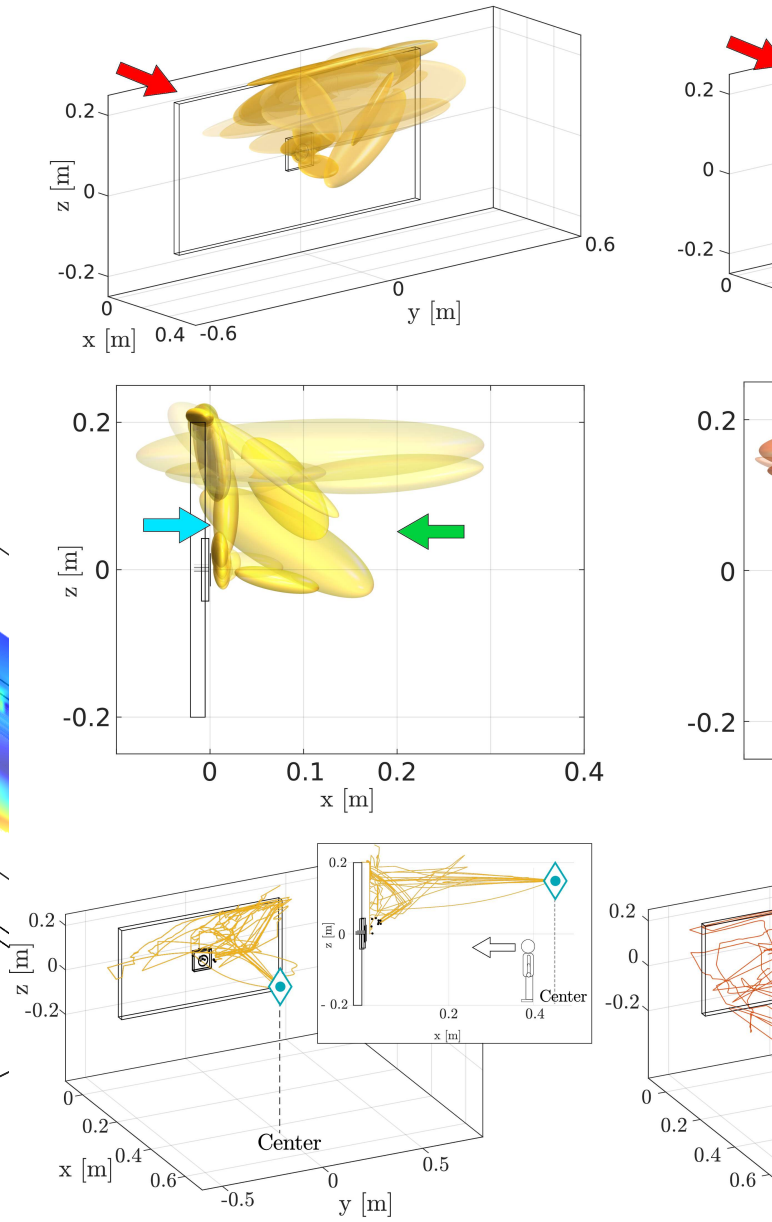


Figure 18: Marginalised Gaussian Mixture parameters of the GMM and Q-EM learned from the demonstrations of teacher #5. The illustrated transparency of the Gaussian functions is proportional to their weight. *Left column*: The Gaussian functions of the Q-EM have shifted from the left corner to the right. This is a result of the value function being higher in the top right corner region, see Figure 17. *Center column*: The original data of the teacher went quite far back which results in a Gaussian function given a direction which moves away from the wall (green arrow), whilst in the case of the Q-EM parameters this effect is reduced and moved closer towards the wall. We can also see from the two plots of the Q-EM parameters that they then follow the paths encoded by the value function. *Right column*: Rollouts of the policies learned from teacher #5. We can see that trajectories from the GMM policy have not really encoded a specific search patter, whilst the Q-EM policy gives many more consistent trajectories which replicate to some extent the pattern of making a jump (no contact with the wall) from the top right corner to the socket's edge.

the socket towards the right edge of the wall. We proceed as before to learn a GMM policy from the raw data and a Q-EM policy in which the data points are weighted by the gradient of the value function. In Figure 18, we illustrate the resulting Marginalised Gaussian Mixture parameters for both the GMM and Q-EM policies and we plot 25 rollouts of each policy starting at the *Center* initial condition used in Experiment 1. We note that the trajectories of the GMM policy have much variance in contrast to the Q-EM policy, resulting from an excess of variance in the 15 original demonstrations given by the teacher. Too much variance is not necessarily good, a random (uniform) policy in terms of generated trajectories will have the most variance and is as expected extremely inefficient in achieving a goal. Furthermore there is insufficient data to encode a pattern for the GMM model. In contrast, the Q-EM finds a pattern by combining multiple parts of the available data and as a result fewer data points are necessary to achieve a good policy. This effect is clear in Figure 19, showing the performance of the GMM and Q-EM algorithms under the same initial conditions as in Experiment 1. For all the conditions and for both teachers #5 and #7 the Q-EM policy always does better than the GMM.

We also tested whether we could use the Greedy policy as a means of gathering demonstrations in order to learn a value function and train a Q-Greedy policy. We used the Q-Greedy algorithm in combination with random perturbations applied to the Greedy velocity, to act as a simple exploration technique. We performed a maximum of 150 searches, which terminated once the socket was found and used these demonstrations to learn a value function and GMM policy which we refer to as Q-Greedy. Figure 14 illustrates the statistical results of the Q-Greedy policy for Experiment 1 and 3, showing that there is no difference between two policies. Our exploration method is probably too simplistic to discover meaningful search patterns and we could probably devise better search strategies which would result in a good policy. However we have shown that human behaviour does already have a usable trade-off between exploration and exploitation which can be used to learn a new policy through our RL-PbD-POMDP framework.

5.4. Generalisation

An important aspect of a policy or any machine learning methodology is to be able to generalise. So far we have trained and evaluated our policy within the same environment. To test whether our GMM policies can generalise to a new setting we changed the location of the socket to the upper right corner of the wall. The GMM was trained in the frame of reference of the

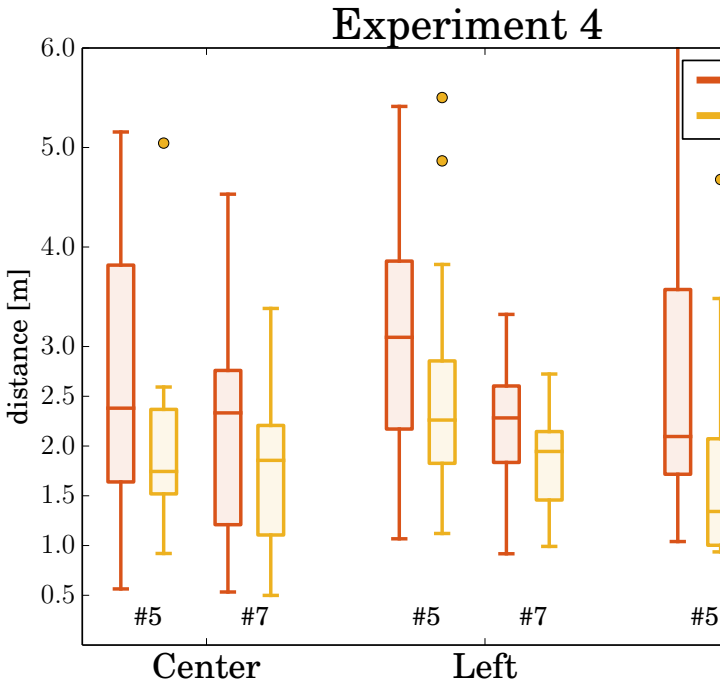
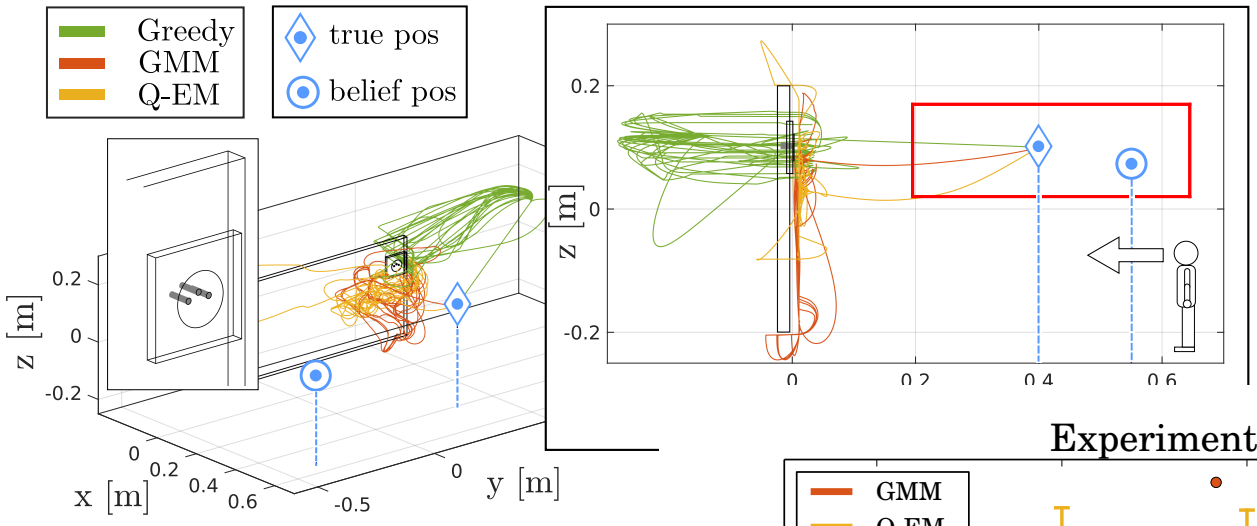


Figure 19: Results of a GMM and Q-EM policy under the same test conditions as Experiment 1. The Q-EM policy nearly always does much better than the GMM policy.

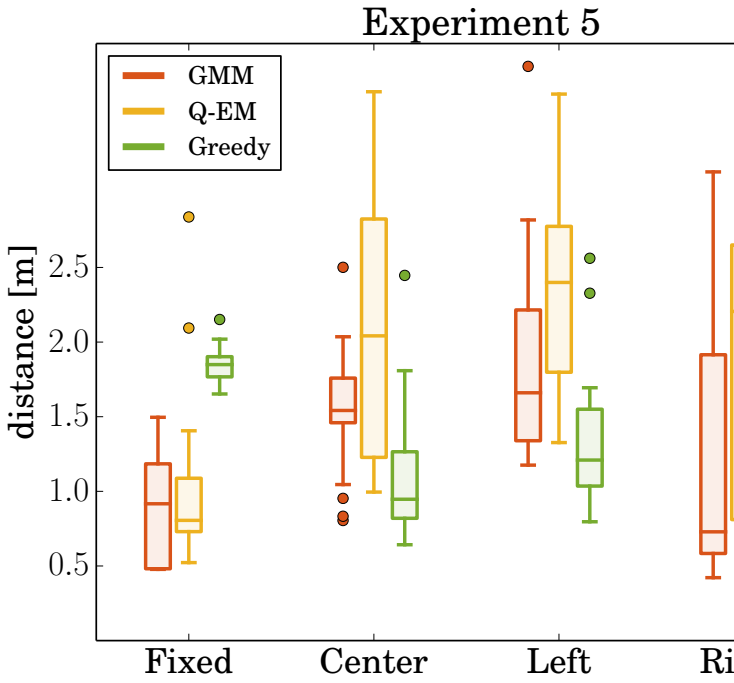


(a)

Figure 20: Evaluation of generalisation. The socket is located in at the top right corner of the wall. We consider a *Fixed* starting location for both the true and believed location of the end-effector. The red square depicts the extent of the initial uncertainty, which is uniform. (b) Distance taken to reach the socket's edge. For the Fixed setup (see (a) for the initial condition), both the Q-EM and GMM significantly outperform the Greedy. The other three conditions are the same as for Experiment 1.

socket and when we translated the socket's location it also translated the policy.

To evaluate the generalisation of our learned policy we use the same initial conditions of Experiment 1 with an additional new configuration named *Fixed*, in which both the true and believed location are fixed, blue triangle and circle. Figure 20 illustrates the trajectories of the three search policies for the *Fixed* initial condition. The Greedy policy moves in a straight line towards the top right corner of the table. As the true position is to the right, it takes the Greedy policy longer to find the wall in contrast to both the GMM and Q-EM policies. From the statistical results shown in Figure 21 we can see that for the *Fixed* and *Right* initial condition, which are similar, both GMM and Q-EM are better. However, for the *Center* and *Left* initial condition this is no longer the case. The Greedy method is better under this condition since the socket is close to informative features (it is located close to the edges of the wall). Once the end-effector has entered in contact with the wall the actions of the Greedy policy always result in a decrease of uncertainty, which was not the case when the socket was located in the center of wall. Thus in both the *Fixed* and *Right* initial condition the Greedy method does worse because it takes longer to find the wall.



(a)

Figure 21: Distance taken to reach the socket's edge. For the Fixed setup (see Figure 20) for the initial condition), both the Q-EM and GMM significantly outperform the Greedy.

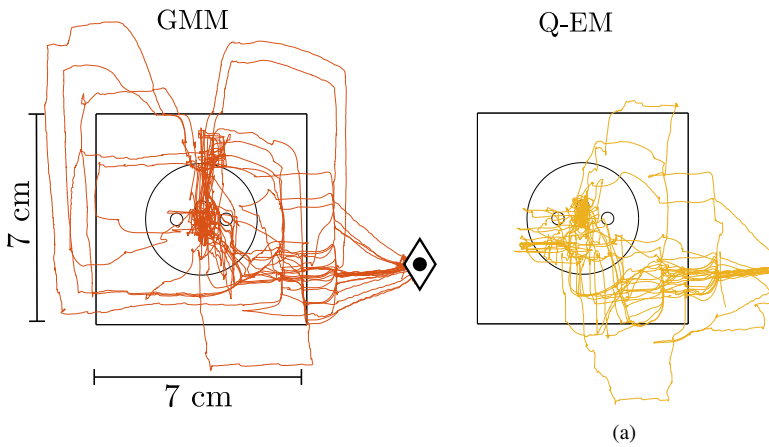
The GMM based policies are still able to generalise under different socket locations. In general, as the socket's location is moved further from the original frame of reference in which it was learned, the higher is the likelihood that the search quality degrades. We chose the upper right corner since it is the furthest point from the origin and the GMM and Q-EM policies were still able to find the socket. We note that the policy will always be able to find the socket once it has localised itself. This can be seen from the vector field of the GMM policy when the uncertainty is low, see Figure 9 on page 12. In this case the policy is a sink function with a single point attractor.

5.5. Distance taken to connect the plug to the socket

In this section we evaluate the distance taken for the policies and humans to establish a connection, after the socket has been found. We start measuring the distance from the point that the plug enters in contact with the socket's edge until the plug is connected to the socket. All the following evaluations are done on a KUKA LWR4 robot. The robot's end-effector is equipped with a plug holder on which is attached a force-torque sensor, the same holders used during the demonstration of the human teachers. In this way both the teacher and robot apprentice share the same sensory interface.

We chose to have the robot's end-effector located to the right of the socket and a belief spread uniformly along the z-axis. See Figure 23 for an illustration of the initial starting condition. This initial configuration was used to evaluate the search policies for the three different sockets, see Figure 1 on page 5 for an illustration of the sockets. The same initial configuration for the evaluation of the three sockets was kept in order to observe the generalisation properties of the policies. As a reminder we used only the training data from demonstrations acquired during the search with socket A. Socket B has a funnel which should make it easier to connect whilst socket C should be more difficult as it has no informative features on its surface.

For each of the sockets we performed 25 searches starting from the same initial condition. In Figure 22 we plot the trajectories of each of the search methods for socket A. The GMM reproduces some of the behaviour exhibited by humans, such as first localising itself at the top of the socket before trying to attempt to make a connection. The Q-EM algorithm exhibits less variation than the GMM and tends to pass via the bottom of the socket to establish a connection. The Greedy method in contrast is much more stochastic since it does not take into consideration the variance of the uncertainty but tries instead to directly establish a connection. Figure 24 (c) shows that for socket A both the Greedy and Q-EM are better than the GMM and the Q-EM has less variance in comparison to the Greedy searches. All three search methods are vastly superior, when compared to the human's performance see Figure 24. In Figure 23 illustrates a typical rollout of the GMM search policy for both socket A and C. Once a contact is made with the socket's edge the policy tends to stay close to informative features and tends to wander vertically up and down. Only when the uncertainty has been reduced does the GMM policy try to go towards the socket's connector.

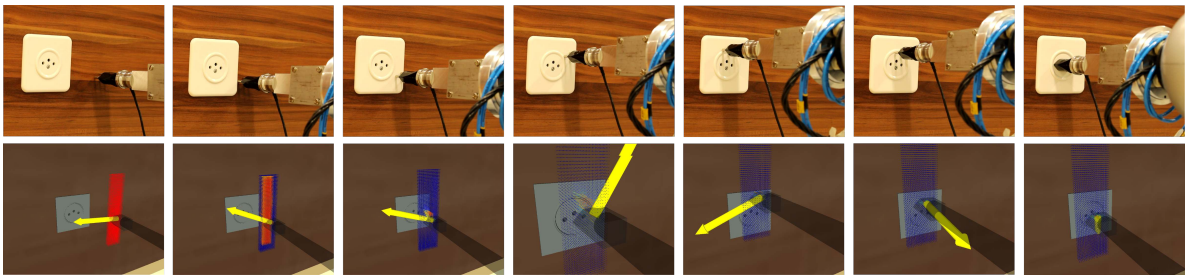


(a)

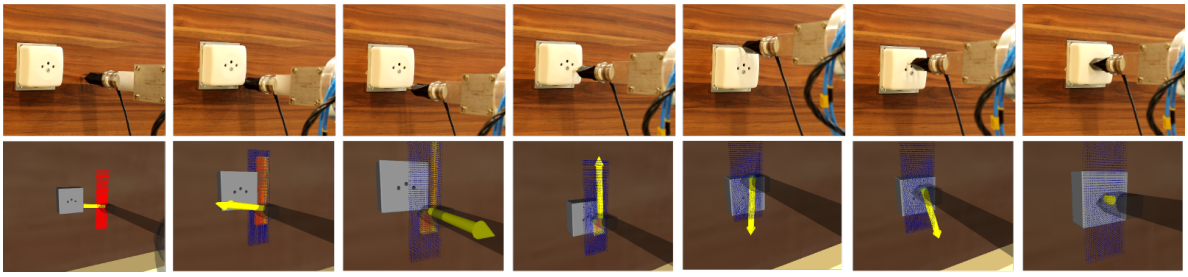
Figure 22: 25 search trajectories for each of the three search policies for socket A.

The GMM and Q-EM policies are able to generalise to both socket B and C, as the geometric shape and connector interface of the two sockets are similar to socket A. The local force modulation of the policy's vector field, which is not learned, allows the end-effector to surmount edges and obstacles whilst trying to maintain a constant contact force in the x-axis. This modulation makes it possible for the plug to get on top of socket C. Figure 24 (c) illustrates the statistics of the distance taken to establish a connection for all three sockets. The interesting point is that both the GMM and Q-EM algorithms perform better than the Greedy approach for socket C. Socket C has no informative features on its surface and as a result myopic policies such as the Greedy policy will perform poorly. However for socket A and B, the Greedy policy performs better as both of these sockets have edges around their connector point allowing for easy localisation. It can also be seen that most search methods perform better on socket B than A, since the funnel shape connector helps in maintaining the plug within the vicinity of the socket's holes.

The discrepancy between the humans performance



(a)



(b)

Figure 23: KUKA LWR4 equipped with a holder mounted with a ATI 6-axis force-torque sensor. (a) The robot's end-effector starts to the right of socket A. The second row shows screen captures taken of ROS Rviz data visualiser in which we see the Point Mass Filter (red particles) and a yellow arrow indicating the direction given by the policy. In this particular run, the plug remained in contact with the ring of the socket until the top was reached before making a connection. (b) Same initial condition as in (a) but with socket C. The policy leads the plug down to the bottom corner of the socket before going the center of the top edge, localising itself, and then making a connection.

and the search policies can be attributed to many causes. One plausible reason is that the PMF probability density representation of the belief is more accurate than the human teachers position belief. Also, the motion noise parameter was fixed to be proportional to the velocity and the robot moves at gentle pace ($\sim$ 1 cm/s) as opposed to some of the human teachers. In actuality, humans are far less precise than the KUKA which has sub-millimetre accuracy.

6. Discussion & Conclusion

In this work we learned search policies from demonstrations provided by human teachers for a task which consisted of first localising a power socket (either socket A, B or C) and then connecting it with a plug. Only haptic information was available as the teachers were blindfolded. We made the assumption that the position belief of the human teachers was initially uniformly distributed in a fixed rectangular region of which they were informed and is considered prior knowledge. All subsequent beliefs were then updated in a Bayesian recursion using the measured velocity obtained from a vision tracking system, and wrench acquired from a force torque sensor attached to the plug. The filtered probability density function, represented by a Point Mass Filter, was then compressed to the most likely state and entropy.

Two Gaussian Mixture Model policies were learned from the data recorded during the human teachers' demonstrations. The first policy, called Q-EM, was learned in an Actor-Critic RL framework in which a value function was learned over the belief space. This was then used to weight training datapoints in the M-step update of Expectation-Maximisation (EM). The second policy, called GMM, was learned using the standard EM algorithm, and considered all training data points equally, following in the footsteps of our initial approach [**?**]. Both the Q-EM and GMM policies were trained with data solely from the demonstrations of the search with socket A.

We evaluated 4 different aspects of the learned policies. Firstly, we evaluated which of three policies, Q-EM, GMM and a Greedy policy, took the least distance to find the socket. We concluded that across three different Experiments the Q-EM algorithm always performed the best. It was clear that the Q-EM policy was less random and more consistent than the GMM policy as it tried to enter in contact with the wall at the same height

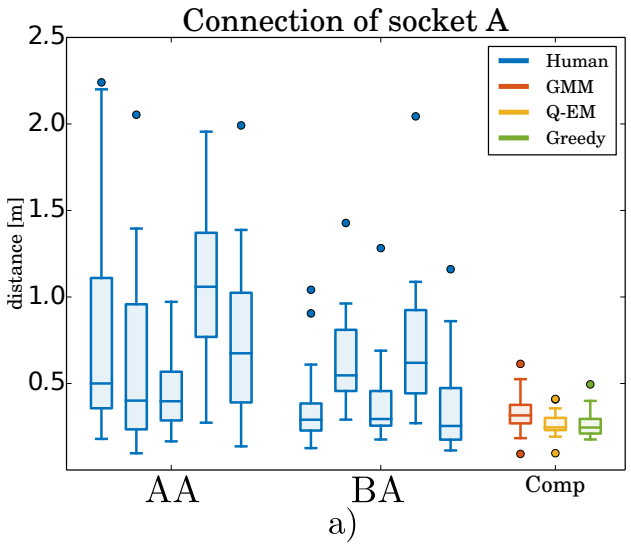


Figure 24: Distance taken to connect plug to socket once the socket is localised. (a) **Socket A**. The human Group A are the set of teachers who first started with socket A. They had no previous training on another socket beforehand. Group BA first gave demonstrations on Socket B before giving demonstrations on Socket A. Group BA is better than Group AA at doing the task. This is most likely a training effect. However all policy search methods are far better at connecting the plug to the socket. (b) **Socket B**. Both Groups AB and BB are similar in terms of the distance they took to insert the plug into the socket and as was the case for (a), the search policies travel less to accomplish the task.

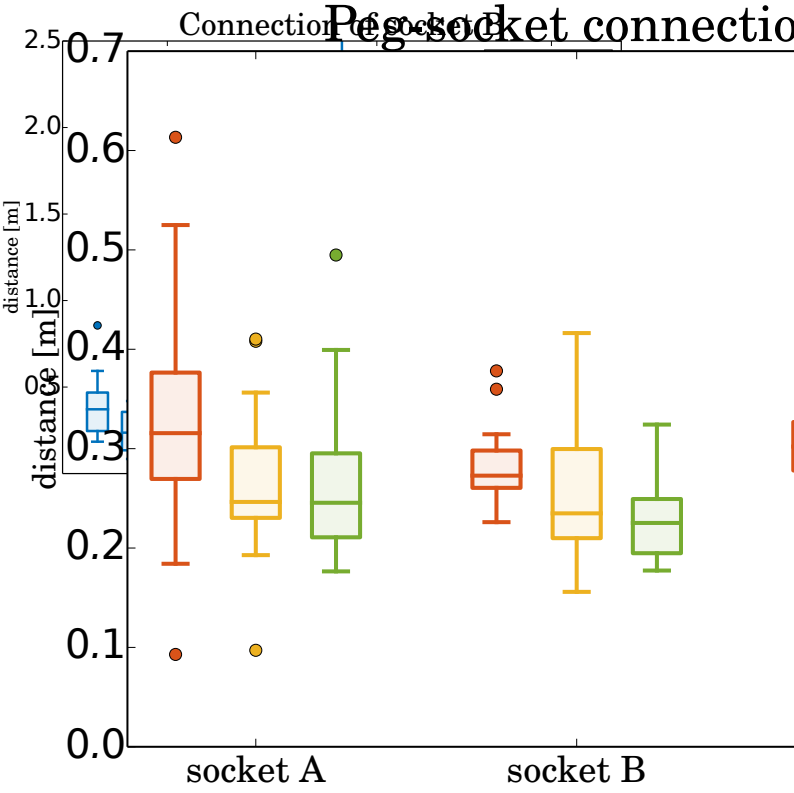


Figure 25: Distance taken (measured from point of contact of plug with socket edge) to connect the plug to the socket.

as the socket thus increasing the chances of finding the socket.

Secondly, we tested the importance of the data provided by the human teachers. We took the worst two teachers and trained an individual GMM and Q-EM policy for each of them. We found that the performance of the Q-EM was better than the GMM in terms of distance travelled to find the socket. When qualitatively evaluating the trajectories of the GMM with respect to the Q-EM for the worst teacher, it is clear that the Q-EM policy managed to extract a search pattern, which was not the case for the GMM policy. We also tried to learn a Q-EM policy from the data provided by a Greedy policy with explorative noise and we found no improvement. From these results we conclude that the exploration and exploitation aspects of the trajectories provided by the human teachers is necessary.

Thirdly, we tested whether the two policies (GMM and Q-EM) were able to generalise to a different socket location. Under a specific condition, which we called *Fixed*, both policies were significantly better than the Greedy policy. However for the *Center* and *Left* initial conditions the Greedy policy performed better. For the initial conditions in which the Greedy policy enters in contact with the wall at an early stage, it also performs better than the GMM and Q-EM. The reason for this is that the actions taken by the Greedy policy in this setting will always result in a decrease of entropy when the location of the socket is close to a corner, as opposed to being in the center of the wall.

Fourthly, we evaluated all three policies on the KUKA LWR4 robot and found that all the policies did better than the human teachers. For socket A, on which both the GMM and Q-EM policies were trained, there is no clear distinction between the Q-EM and Greedy policy. On socket B, which was novel, the Greedy policy performed better than the statistical controllers, which we hypothesize was a result of a funnel which would make it easier for a myopic policy. For socket C, both the GMM and Q-EM policies performed better than the Greedy, as socket C has no features on its surface, this being a disadvantage for a myopic policy.

We conclude by making the observation that by simply adding a binary reward function in combination with data provided by human demonstrations, with Fitted reinforcement learning, we can learn a better policy without the need to perform expensive exploration-exploitation rollouts traditionally associated with reinforcement learning and designing complicated reward functions. This is especially advantageous when only a few demonstrations are available.

[1] Abu-Dakka, F., Nemec, B., Kramberger, A., Buch, A. G., Krüger, N., Ude, A., 2014. Solving peg-in-hole tasks by human demonstration and exception strategies. Industrial Robot 41 (6), 575–584.

[] Atkeson, C. G., Moore, A. W., Schaal, S., 1997. Locally weighted learning. ARTIFICIAL INTELLIGENCE REVIEW, 11–73.

[2] Bdiwi, M., Winkler, A., Jokesch, M., Suchy, J., 2015. Improved peg-in-hole (5-pin plug) task: Intended for charging electric vehicles by robot system automatically. In: Proc. of 12th IEEE International Multi-Conference on Systems, Signals and Devices.

[] Bergman, N., Bergman, C. N., 1999. Recursive bayesian estimation: Navigation and tracking applications. thesis no 579. Tech. rep., Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation.

[] Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., Billard, A. G., June 2010. Learning and reproduction of gestures by imitation. IEEE Robotics Automation Magazine 17 (2), 44–54.

[] Chambrier, G. d., Billard, A., 2014. Learning search policies from humans in a partially observable context. Robotics and Biomimetics 1 (1), 1–16.
URL `http://dx.doi.org/10.1186/s40638-014-0008-1`

[3] Cheng, H., Chen, H., May 2014. Online parameter optimization in robotic force controlled assembly processes. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 3465–3470.

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

[] Deisenroth, M. P., Neumann, G., Peters, J., 2011. A survey on policy search for robotics. Foundations and Trends in Robotics 2 (1-2), 1–142.
URL `http://dx.doi.org/10.1561/2300000021`

[] Ernst, D., Geurts, P., Wehenkel, L., April 2005. Tree-based batch mode reinforcement learning. Journal of Machine Learning Research 6, 503–556.

[5] Fisher, W. D., Mujtaba, M. S., 1992. Hybrid position/force control: A correct formulation. The International Journal of Robotics Research 11 (4), 299–311.
URL `http://ijr.sagepub.com/content/11/4/299.abstract`

[6] Gullapalli, V., Barto, A. G., Grupen, R. A., 1994. Learning admittance mappings for force-guided assembly. In: Proceedings of the 1994 International Conference on Robotics and Automation, San Diego, CA, USA, May 1994. pp. 2633–2638.

[7] Kalakrishnan, M., Righetti, L., Pastor, P., Schaal, S., Sept 2011. Learning force control policies for compliant manipulation. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4639–4644.

[8] kook Yun, S., May 2008. Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion? In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. pp. 1647–1652.

[9] Meeussen, W., Wise, M., Glaser, S., Chitta, S., McGann, C., Mihelich, P., Marder-Eppstein, E., Muja, M., Eruhimov, V., Foote, T., Hsu, J., Rusu, R. B., Marthi, B., Bradski, G., Konolige, K., Gerkey, B., Berger, E., May 2010. Autonomous door opening and plugging in with a personal robot. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on. pp. 729–736.

[10] Nemec, B., Abu-Dakka, F. J., Ridge, B., Ude, A., Jorgensen, J. A., Savarimuthu, T. R., Jouffroy, J., Petersen, H. G., Krüger, N., Nov 2013. Transfer of assembly operations to new work-

piece poses by adaptation to the desired force profile. In: Advanced Robotics (ICAR), 2013 16th International Conference on. pp. 1–7.

[] Neumann, G., Peters, J. R., 2009. Fitted q-iteration by advantage weighted regression. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (Eds.), Advances in Neural Information Processing Systems 21. Curran Associates, Inc., pp. 1177–1184.

[11] Park, H., Bae, J.-H., Park, J.-H., Baeg, M.-H., Park, J., Oct 2013. Intuitive peg-in-hole assembly strategy with a compliant manipulator. In: Robotics (ISR), 2013 44th International Symposium on. pp. 1–5.

[] Peters, J., Schaal, S., 2008. Natural actor-critic. Neurocomputing 71 (7-9), 1180–1190.

[] Riedmiller, M., 2005. Neural Fitted Q Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method. pp. 317–328.
URL http://dx.doi.org/10.1007/11564096_32

[] Roy, N., Gordon, G. J., 2003. Exponential family pca for belief compression in pomdps. In: Becker, S., Thrun, S., Obermayer, K. (Eds.), Advances in Neural Information Processing Systems 15. MIT Press, pp. 1667–1674.
URL http://papers.nips.cc/paper/2319-exponential-family-pca-for-belief-compression-in-pomdps.pdf

[12] Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A., 2004. Learning movement primitives. In: International Symposium on Robotics Research (ISRR2003. Springer.

[] Sung, H., 2004. Gaussian mixture regression and classification. Ph.D. thesis, Rice University.

[] Sutton, R., Barto, A., 1998. Reinforcement learning: An introduction. Vol. 116. Cambridge Univ Press.

[13] Yang, Y., Lin, L., Song, Y., Nemec, B., Ude, A., Buch, A., Krger, N., Savarimuthu, T., 2014. Fast programming of peg-in-hole actions by human demonstration. IEEE, pp. 990–995.