*Chapter 4*

# Peg in hole

In Chapter 3, we demonstrated that we could learn a search policy from demonstrations of human teachers for a task consisting of locating a wooden object on a table and successfully transferring it to a robot apprentice. The motivation behind our approach is that the intuition and knowledge exhibited by the human teachers, during the search, consists of a good balance between exploration and exploitation actions which then can be encapsulated in a generative Gaussian Mixture Model (GMM) and subsequently used as a control policy. The approach is satisfactory if we are interested in extracting different behaviours present across the human teachers and reproducing them. If our objective is however to learn a policy with a unique behaviour which is optimal or at least close to optimal, then using the approach detailed in Chapter 3, as it is, will not necessarily result in a efficient policy. For the GMM w odel both good patterns exhibited by the human teachers and their mistakes. If the task is difficult and many possible solutions exist, such as was the case in the blindfolded search task of the previous Chapter, many demonstrations will be necessary for search patters to be present and encoded in the GMM. Otherwise it d have to be combined with another policy as showed in Chapter 3, for our Hybrid GMM-Greedy policy. There the task undertaken is implicitly encoded, as there is no cost function which is optimised, in the GMM and as a result the taught behaviour has to be goal orient d consistent, which is not always the case in a blindfolded search task.

To overcome the above mentioned limitations, in this Chapter we use a binary cost function as means of ranking demonstrations provided by the human teachers. We combine our PbD-POMDP approach with an Actor-Critic Reinforcement Learning (RL) framework which is close to Fitted-Value Iteration(FVI) and other experience repla ethods, wh e will refer to as RL-PbD-POMDP. Our objective is to avoid noisy explorative rollouts, common to all RL approaches and is their Achille's heel, and only rely on the data provided by the human teachers. We want to avoid any autonomous exploration common in RL for three reasons. Firstly it is time consuming and is typically only applicable to RL problems in which an exhaustive exploration of the entire state or parameter space is feasible, such as in traditional RL problems like the inverted pendulum or mountain cart. The universal exploration method, used

throughout RL, is state independent (sometimes state dependent) white noise which results in an entire exploration of the state space. This is neither practical o sible for the type of search problems we are considering. The second reason is that the exploration cannot be random as we are using a physical robotic system, and this would be dangerous. The third and most important reason is that we want to use the same amount of information for our RL-PbD-POMDP as we used for the PbD-POMDP approach. This is t rongly highlight the fact that we can obtain an improved policy without the need of any additional informat

We analyse our RL-PbD-POMDP approach on a power-socket Peg-in-Hole (PiH) search task. In this task, human teachers must demonstrate to a robot apprentice how to search for and connect a plug to a power socket. The first component of the task, the search for the socket, is similar to the table-wooden block setup in the previous Chapter. Here the conne n of the plug to the power socket, the PiH component, which requires a higher level of precision to be able to achieve.

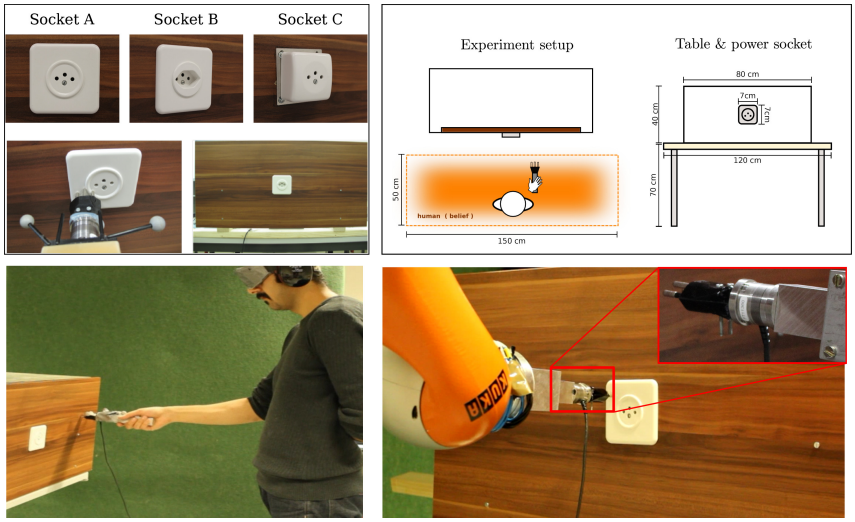



**Figure 4.1:** *Top-right:* The experimental setup. Blindfolded human teachers stand in the orange rectangle always facing the wall. Human teachers are informed of their starting location and told that they would always be facing the wall. *Top-left:* Three different power sockets. The plug is connected to a cylinder, to make it easy to hold for the human teachers. An ATI 6-axis force torque sensor (Nano25 Series) is between the cylinder and the plug. *Bottom-left* Human teacher performing the PiH search task. The human teachers wear goggles to remove sight and ear defenders to greatly diminish hearing.*Bottom-right:* The KUKA LWR robot equipped with the same force torque sensor and plug used by the human teachers.

In Figure 4.1 (*Top-right*), we illustrate the setup of the search task. The diagram shows the distribution of the initial starting position (in orange) of the human teachers. As was the case in our previous search experiment in Chapter 3, the teacher is disoriented before each trial but knowns that his initial heading will be the same, facing the wall. The human teachers are deprived of

visual information (they are blindfolded) and their hearing sense is significantly reduced (by wearing ear defenders). 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 4.1 (*Top-left*) for an illustration. The plug held by the human teachers is attached to a cylindrical handle with an ATI 6 axis force torque sensor (Nano25 [2]) fixed between the two. On top of the cylinder is a set of markers from which a motion capture system (OptiTrack[3]) provides both position and orientation, see Figure 4.1 (*Top-left*). In the *Bottom-left* quadrant we can see an example of a human teacher trying to accomplish the search and connection task, and to the *Bottom-right* the KUKA LWR robot apprentice is reproducing a search policy learned from the teacher.

We found that by learning a v function over the belief space using approximate dynamic programming (part of FVI) and using this as a Critic to update the parameters of our GMM policy (Actor) we were able to achieve an important improvement over our previous PbD-POMDP approach. We performed evaluations both in simulation and on the KUKA LWR robot where we tested policies ability to generalise to sockets for which no training data was provided and different socket locations. In all our evaluations the RL-PbD-POMDP approach proved to be always better. More importantly we demonstrate that the RL-PbD-POMDP approach performs significantly better when we use the training data from the worst teacher, which mitigates the **original assumption** that the teachers have to be consistently efficient at the task.

## 4.1 Background

### 4.1.1 Peg-in-hole

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 Chhatpar and Branicky (2001) and valves Cheng and Chen (2014). To be successful, the estimated position of the robot's end-effector and workpiece must be precise. Typically, the clearance the between and plug 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 in failure of 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.

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

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. 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. To increase the chances 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 academic circles) 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 so as to reproduce the same (F)orce/(T)orque profile as that 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 Meeussen et al. (2010) 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 is equipped with a checkerboard to facilitate pose estimation of the plug with respect to 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. When unsuccessful a spiralling outward motion is carried out with 2mm increments. This method achieved an overall success rate of 95%. The hybrid control paradigm Fisher and Mujtaba (1992) was used throughout the execution of the task.

In Yang et al. (2014) 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) Schaal et al. (2004) 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 results in the policy having similar force profiles to the human teachers. Such an approach was first proposed by Nemec et al. (2013) and further applications based on this method have been done Abu-Dakka et al. (2014) with the incorporation of a disturbance rejection policy. Reinforcement learning has also been used in combination with DMP to learn PiH policies. In Kalakrishnan et al. (2011) an DMP policy is initialised with kinesthetic demonstrations of a door opening and pen pick up task. The recorded Cartesian trajectories 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 110 trials a policy was found to be a 100% successful. In Gullapalli et al. (1994) a 18 dimensional input (sensed position, previous position and force) and 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 is successful and was able to generalise across different geometries and clearances.

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

In Chhatpar and Branicky (2001) different blind search policies for the insertion of a spline toothed hub into a forward clutch are analysed. 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 which ensure that all the points are visited are evaluated. It is found that paths following concentric circles gradually spiralling inwards were the most effective method in finding the hole. The concentric circle search strategy has been applied in many PiH tasks. For instance in M. Bdiwi (2015), 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 spiral search policy to achieve a connection and demonstrated their approach to be reliable.

In Park et al. (2013) the authors make the observation that humans lack the precision and sensing accuracy of robotic systems. But nevertheless, are more proficient than robots at PiH. The authors observe 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 kook Yun (2008). 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.

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 fine-tuning parameters of predefined policies. In Cheng and Chen (2014) 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 can be used to learn the parameters of the policy faster than offline methods, such as Design of Experiment (DOE) or Genetic Algorithms.

## 4.2 Experiment

The sockets are positioned at the center of a fake wall clamped to a table, see Figure 4.1 (*Top-left*) for an illustration of the environment. Each teacher is given the opportunity to familiarise himself with the environment. Before each trial the human teacher is placed on a chair and disoriented by the experimenter. Once disoriented, the teacher is allowed to stand and is signalled to start the search task by a light touch to the shoulder. Figure 4.1 (*Bottom-left*) illustrates a human teacher performing the task. The disorientation step is to induce the effect of an uniform prior over the teachers believed location. We make the assumption that the human's believed location can be presented by a probability density function, which is assumed to be known. All subsequent distributions can be obtained through the application of a Bayesian filter given that both the sensing and motion information are provided by the force torque and motion capture sensors.

In Figure 4.1 (*Top-right*) we illustrate the experimental setup. The orange area represents the teachers starting location and is assumed prior knowledge. The teachers are told and shown before hand the environmental setup and it is made explicitly clear to them that they will always be starting in the orange area facing the wall.

A group of 10 human teachers participated in the plug-socket experiment. Each teacher performed the search and PiH for sockets A and B. A teacher of group A (starting with socket A), would start by performing 15 times the search and connection with socket A and after a short break, during which socket A was replaced by socket B, would carry on to perform 15 trials with socket B.

Before the actual recording of the task, the teachers had a training period to familiarise themselves with environment and become comfortable in wearing the sensor deprivation apparatus. After each teacher felt sufficiently ready to carry out the task to the best of his ability, the experimenter proceeded to disoriented him through the usage of a chair as mentioned previously. The teachers were reminded that they were facing the direction of the fake wall and that the starting location would be always within the orange rectangular area. In Figure 4.2 we can see the time taken by the teachers to accomplish the task.

Both groups A and B took $9 \pm 10$s to find their socket. This was expected since the sockets are at the same location. However after the socket was found it took a further $8 \pm 7$s on average for group A to connect socket B and $12 \pm 10$s on average for group B to connect socket A. As we can see this is not a straight forward task when considering the sensory deprivation. See Figure 4.2 (*Right*) the time taken to connect the plug to the socket.

The location belief of the humans was represented by a probability density function. We made the assumption that after the disorientation step the human's believed location would be uniform and spread across the rectangular starting area. Although the mental state of the human remains unobservable, there is sufficient evidence to support our assumption that his belief state gets updated in a Bayesian fashion [citation].

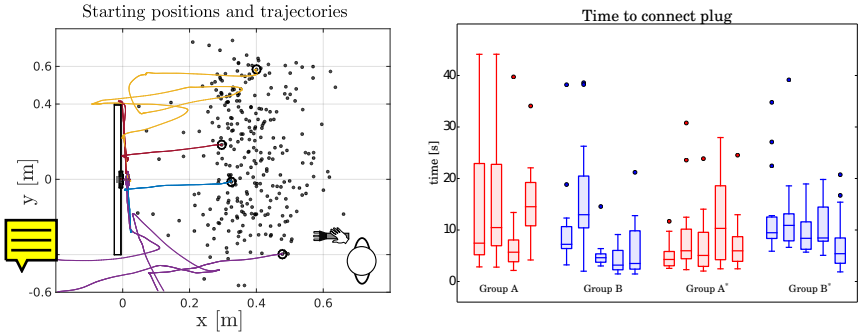



**Figure 4.2:** *Left*: Black points represent the starting position of the end-effector for all the demonstrations. Four trajectories are illustrated. *Right:* Time taken for the teachers to accomplish the PiH once the socket is localised. Group A and B are depicted in red and blue. The asterisk indicates that the group has changed sockets, so Group A$^*$ means that Group A is now performing the task with socket B and Group B$^*$ means that group B is now performing the task with socket A.

During each trial we recorded the position and orientation of the plug provided by the motion capture system, and the sensed force and torque, given by the ATI sensor.

## 4.3 Formulation

### 4.3.1 Belief probability density function

In our setting the belief probability density function is a Point Mass Filter (PMF) (Bergman and Bergman, 1999, p.87), which is a Bayesian filter. It is parametrised by a set of grid cells which contain valid probabilities. Our choice of a PMF, as means to represent the believed location of the plug, is motivated by the fact that 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 PMF gives a probability density, $p(x_t|y_{0:t}, \dot{x}_{0:t})$, which is recursively updated through the application of a **motion**, $p(x_t|x_{t-1}, \dot{x}_t)$ 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. This step essentially consists of applying a convolution kernel to the PMF where the covariance is proportional to the measured velocity. The measurement model indicates areas of the state space from which a measurement $\tilde{y}_t$ could have originated. Both the human teachers and the robot apprentice use the same sensor interface. This is a plug holder equipped with a 6-axis force torque sensor which provides a sensed wrench, $\phi \in \mathbb{R}^6$, which we call the **raw** measurement. We define the **actual** measurement to be a function of the sensed wrench, $\tilde{y}_t = h(\phi_t)$, which

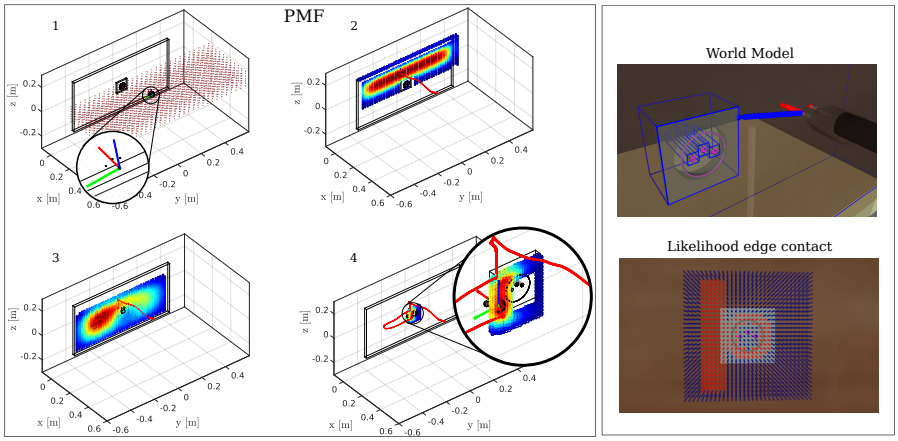



**Figure 4.3:** *Left:* Point Mass Filter (PMF) update of a particular human demonstration. (1) Initial uniform distribution spread over the starting region. Each grid cell represents a hypothetical position of the plug, the orientation is assumed to be known. (2) First contact, the distribution is spread 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 presented 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.

is a binary feature vector. The feature vector encodes whether a contact is present and the direction in which it occurs, which we discretized to the four cardinalities. In Figure 4.3 (*Right-bottom*) we illustrated the likelihood when an edge is sensed.

### 4.3.2 Belief compression

The probability density function $p(x_t|y_{0:t}, \dot{x}_{0:t})$ is high dimensional and it is impractical to directly learn a statistical policy $\pi_\theta : p(x_t|y_{0:t}, \dot{x}_{0:t}) \rightarrow \dot{x}_t$; therefore some form of compression is necessary. One possibility would be E-PCA [cite] which finds a set of representative basis functions (which are probability distributions). Although elegant this method requires a discretisation of the belief space which is computationally expensive. Instead we chose to compress the pdf to a belief space vector composed of the maximum a posteriori, $\hat{x}_t^{\text{MAP}} = \text{argmax}_{x_t}\, p(x_t|y_{0:t}, \dot{x}_{0:t}) \in \mathbb{R}^3$, 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, $b_t = [\hat{x}_t^{\text{MAP}}, U]^{\text{T}}$.

From the demonstrations we obtained 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 trajectory and subscript $1:T$ denotes the time steps during the trajectory from initialisation $t = 1$ until the end $t = T$. The data consisted of the plug's linear velocity, $\dot{x} \in \mathbb{R}^3$, angular velocity $\omega \in \mathbb{R}^3$, the sensed wrench $\phi \in \mathbb{R}^6$ (force-torque), and the belief state,

$b$, over the plug's location.

## 4.4 Learning Actor and Critic

In our approach we learn two data driven policies. The first policy maps from belief space to linear velocity $\pi_{\boldsymbol{\theta}_1} : b_t \mapsto \dot{x}_t$ and the second from angular sensed wrench to angular velocity, $\pi_{\boldsymbol{\theta}_2} : \phi_t \mapsto \omega_t$. We chose to learn the belief policy $\pi_{\boldsymbol{\theta}_1}$ in a Actor-Critic RL framework and the wrench policy $\pi_{\boldsymbol{\theta}_2}$ directly from the demonstrated data as was done in [cite], which proved to be efficient in overcoming jamming during the PiH. A POMDP solver's objective is to find a policy (Actor), $\pi_{\boldsymbol{\theta}_1} : b \mapsto u$, which maximises the value function (Critic) $V^{\pi_{\boldsymbol{\theta}_1}} : b \mapsto \mathbb{R}$ for an initial belief, $b_0$. The value function is the expected reward over an infinite time horizon.

$$V^{\pi_{\boldsymbol{\theta}_1}}(b_t) = \mathbb{E}\left\{ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | b_0 = b, \pi_{\boldsymbol{\theta}_1} \right\} \tag{4.1}$$

In an Actor-Critic setting, the temporal difference error, $\delta_t \in \mathbb{R}$, of the value function (the Critic) is used both as a learning signal to update simultaneously itself and the actor (the policy). In our setting we will learn two separate policies, one for the linear velocity and the other for the angular velocity, as the orientation remains the same during most of the search until it is time to connect the plug to the socket. When the plug has to be connected it is necessary to control for orientation to avoid jamming.

### 4.4.1 Actor

Both actors/policies are parametrised by a Gaussian Mixture Model (GMM), Equation 4.2.

$$\pi_{\boldsymbol{\theta}}(\dot{x}, b) = \sum_{k=1}^{K} w^{[k]} \cdot g(\dot{x}, b; \boldsymbol{\mu}^{[k]}, \boldsymbol{\Sigma}^{[k]}) \tag{4.2}$$

The parameters $\boldsymbol{\theta} = \{w^{[k]}, \boldsymbol{\mu}^{[k]}, \boldsymbol{\Sigma}^{[k]}\}_{1,\dots,K}$, 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 $\sum_k w^{[k]} = 1$, $\boldsymbol{\mu}_{\dot{x}}^{[k]} \in \mathbb{R}^3$ and $\boldsymbol{\mu}_{b}^{[k]} \in \mathbb{R}^4$.

A generative model of the angular velocity and wrench $\pi_{\boldsymbol{\theta}_2}(\omega, \theta)$ and a generative model of the linear velocity and belief state $\pi_{\boldsymbol{\theta}_1}(\dot{x}, b)$ 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 $\pi_{\boldsymbol{\theta}_1}$ can be adapted by the value function of the Critic.

### 4.4.2 Critic

The Critic (the value function, Eq. 4.1) evaluates the performance of the current policy. It is the expected future reward given the current belief state and policy. In our setting 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 Atkeson et al. (1997) (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 then the result of a weighted least square of the $p$ points. Equation 4.3 is the distance function used where $D$ is a diagonal matrix.

$$W_{i,i} = \exp\left(-\frac{1}{2}(b - b_i)^{\mathrm{T}} D^{-1} (b - b_i)\right) \tag{4.3}$$

A new value is queried according to Equation 4.4,

$$V^\pi(b) = b\,(B^{\mathrm{T}} W B)^{-1} B^{\mathrm{T}} W \mathbf{r} \tag{4.4}$$

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, $\mathbf{r} = (r_1, \cdots, r_p)^{\mathrm{T}} \in \mathbb{R}^{(p \times 1)}$

#### Fitted Policy Evaluation

To learn the value function we apply batch reinforcement learning Ernst et al. (2005a), 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.

**Algorithm 4.1** Fitted Policy Evaluation

**Inputs:**
  $K$, $\epsilon$, $\{(b_t^{[i]}, r^{[i]}, b_{t+1}^{[i]})\}_{i=1,\cdots,M}$
Initialise $\hat{V}_0^\pi = 0$
**for** $k = 0$ in $K$ **do**
  $\hat{V}_{k+1}^\pi(b_t) = \text{Regress}(b,\ r_t + \gamma \hat{V}_k^\pi(b_{t+1}))$
  **if** $||\hat{V}_{k+1}^\pi(b) - \hat{V}_k^\pi(b)|| < \epsilon$ **then**
    **return** $\hat{V}_{k+1}^\pi(b)$
  **end if**
**end for**

A wide spectrum of research has made use of batch RL methods to learn policies. Most of them have focused on learning the Q-value function directly

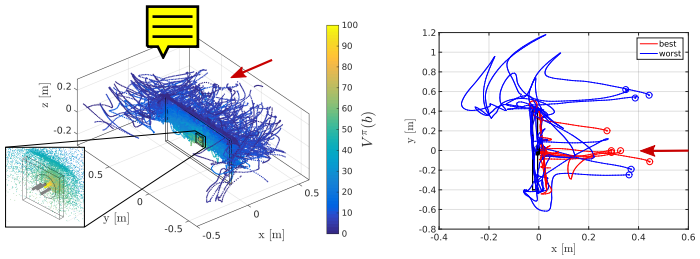



**Figure 4.4:** *Left:* LWR approximation of the value function $\hat{V}^\pi(b)$. *Right:* first five best and worst trajectories in terms of the accumulated value.

(Fitted Q-Iteration) Neumann and Peters (2009); Ernst et al. (2005a); Riedmiller (2005a). Although learning the Q-value function directly solves the control problem it often requires discretisation of the action space or assumes quantifiable actions. The reason is that doing Q-Bellman backups, $\hat{Q}(b_t, \dot{x}_t) \leftarrow \gamma \max_{\dot{x}_{t+1}} \hat{Q}(\dot{x}_{t+1}, b_{t+1})$, requires an optimisation over the actions space, $\dot{x}_{t+1}$, to find the best applicable action. Given the dimensionality and continuity of our problem we opted for an on-policy evaluation method which does not require an optimisation, but does require multiple *policy evaluation* and *policy improvements* iterations to achieve an optimal policy. We applied Algorithm 4.1 on our dataset until convergence.

Figure 4.4 (*Left*) shows the belief space with respect to the value function. As expected, the value function is high closest to the socket and low further away. Figure 4.4 (*Right*) shows the best and worst trajectories in terms of the accumulated value function. There is a close relationship between the best trajectories and the time taken to successfully connect the plug to the socket. We can see that the five best trajectories (red) tend to be aligned with the socket (star position in front of socket), whilst the five worst are towards the edges of the wall.

### 4.4.3 Actor update

The Temporal Difference (TD) error, is used to update the actor $\delta_t^\pi = r_{t+1} + \gamma V^\pi(b_{t+1}) - V^\pi(b_t)$, given by the critic(Sutton and Barto, 1998, Chap. 6). In our offline approach we first computed the value function of the belief state, $V^\pi(b)$, until convergence and then used the estimated value function to update the actor. This offline batch method has the advantage that no divergence will occur during the learning process.

We proceed to update the Actor given the Critic 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 (Deisenroth et al., 2013b, p.50).

The reward of a demonstrated trajectory (one episode) is given by the dis-

counted return, Equation 4.5.

$$R(b^{[i]}, \dot{x}^{[i]}) = \sum_{t=0}^{\mathrm{T}^{[i]}} \gamma^t \, r(b_t^{[i]}, \dot{x}_t^{[i]}) \tag{4.5}$$

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 4.6.

$$\begin{aligned} J(\boldsymbol{\theta}) &= \mathbb{E}_{\pi_{\boldsymbol{\theta}}(\dot{x},b)}\{R(b, \dot{x})\} \\ &= \sum_{i=1}^{N} \underbrace{\left( \prod_{t=0}^{T^{[i]}} \pi_{\boldsymbol{\theta}}(\dot{x}_t^{[i]}, b_t^{[i]}) \right)}_{\pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})} R(b^{[i]}, \dot{x}^{[i]}) \end{aligned} \tag{4.6}$$

To find the parameters which maximise the cost function, $\text{argmax}_{\boldsymbol{\theta}'} J(\boldsymbol{\theta}')$, the derivative is taken and set to zero. As this cannot be done directly, we maximise the logarithmic lower bound of the cost function. This results in Equation 4.7,

$$\nabla_{\boldsymbol{\theta}'} \log(J(\boldsymbol{\theta}')) = \sum_{i=1}^{N} \sum_{t=0}^{T^{[i]}} \nabla_{\boldsymbol{\theta}'} \log \pi_{\boldsymbol{\theta}'}(\dot{x}_t^{[i]}, b_t^{[i]}) \, Q^{\pi}(\dot{x}_t^{[i]}, b_t^{[i]}) \tag{4.7}$$

Setting the derivative of Equation 4.7 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 $Q^{\pi}$, see Appendix 4.8.1 for a more complete derivation. We use the TD error of the *Critic* as a substitute for $Q^{\pi}$. Assuming that our estimated value function, $\hat{V}^{\pi_{\boldsymbol{\theta}}}$, is close to the true value function $V^{\pi_{\boldsymbol{\theta}}}$, the TD error $\delta^{\pi}$ is an unbiased estimate of the advantage function, Equation 4.8 (see Appendix 4.8.3).

$$A^{\pi}(b_t, u_t) = Q^{\pi}(b_t, u_t) - V^{\pi}(b_t) = \delta_t^{\pi} \tag{4.8}$$

Using the advantage function as means of policy search is popular with examples such as Natural Actor Critic (NAC) Peters and Schaal (2008a).

Each data point has an associated weight, $\delta \in \mathbb{R}$, where $\delta^{[m]} \geq 0$ means that the state action-pair $x^{[m]}$ leads to an increase in the value function and $\delta^{[m]} \leq 0$ leads to a decrease in the value function. The 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, we rescale the TD error to be between 0 and 1.

## 4.5 Control architecture

We learned both a Gaussian Mixture Model for both the linear and angular

velocity. For most of the search and up until the plug is within the socket's hole, only the linear control policy is active. The orientation is kept constant. The direction to search is given by conditional, Equation 4.9,

$$\pi_{\boldsymbol{\theta}}(\dot{x}|b) = \sum_{k=1}^{K} w_{\dot{x}|b}^{[k]} \cdot g(\boldsymbol{\mu}_{\dot{x}|b}^{[k]}, \boldsymbol{\Sigma}_{\dot{x}|b}^{[k]}) \tag{4.9}$$

which is a distribution over the possible normalised velocities. The subscript $\dot{x}|b$ indicates that the parameters are the result of the conditional. The reader is referred to Calinon et al. (2010),Sung (2004) for a detailed derivation of the conditional of a GMM. In autonomous dynamical systems control, the velocity to is taken from the expectation of Equation 4.9. The model learned is multi-modal, as different search velocities are possible in the same belief state. Taking the expectation, which is weighted linear combination of the modes would result in unobserved behaviour or no movement if the velocities cancel out. In Figure 4.6 we illustrate the multi-modal vector fields of the conditional, Equation 4.9. As a result we use a modified version of the expectation operator which favours the current direction, Equation 4.10 - 4.11.

$$\alpha(\dot{x}) = w_{\dot{x}|b}^{[k]} \cdot \exp(-\cos^{-1}(< \dot{x}, \boldsymbol{\mu}_{\dot{x}|b}^{[k]} >)) \tag{4.10}$$

$$\dot{x} = \mathbb{E}_{\alpha}\{\pi_{\boldsymbol{\theta}}(\dot{x}|b)\} = \sum_{k=1}^{K} \alpha_k(\dot{x}) \cdot \boldsymbol{\mu}_{\dot{x}|b}^{[k]} \tag{4.11}$$

When a velocity mode being applied is no longer present (because we have moved into a region of belief space where the current applied velocity has not been seen) another direction mode is sampled. As an example, when the robot suddenly enters in contact with a feature, which greatly reduces the uncertainty, the current modes will dramatically change and cause a new search direction to be computed.

The above steps can control the general behaviour of the search but are insufficient for a successful implementation on a robotic system, such as the KUKA LWR. This search task is haptic and as a result the end-effector of the robot will always be 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. Our hybrid controller targets a sensed force $F_x$, in the $x$-axis, of 3N. The other two velocity components of the direction vector are given by Equation 4.11.This force by itself in insufficient to reliably surmount the edges of the socket and the robot will become stuck at the edges, unable able to surmount the friction as these right angle contacts. To overcome the edges we locally modulated the vector field of the GMM in $y$ and $z$-axis, Equation 4.12.

$$\dot{x} = R_y(c(F_z) \cdot \pi/2) \cdot R_z(c(F_y) \cdot \pi/2) \cdot \dot{x} \tag{4.12}$$

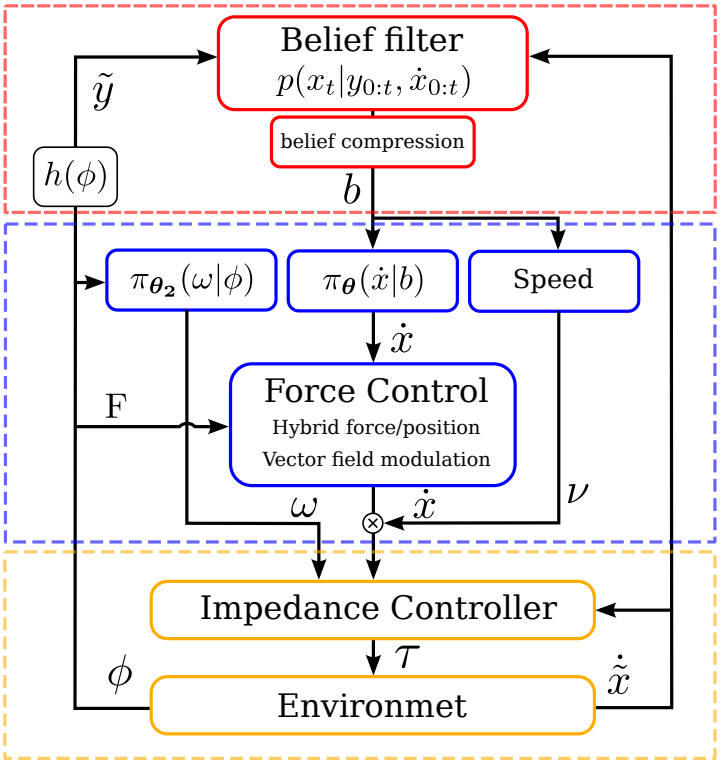



**Figure 4.5:** Control architecture. The PMF (belief) received a measured velocity, $\dot{\tilde{x}}$, and sensor feature $\tilde{y}$ and gets updated via Bayes rule. The belief is compressed and used by both the GMM policy and the proportional speed controller, Equation 4.13.

$R_y$ and $R_z$ are rotation matrix around the $y$ and $z$-axis, $c(F) \in [-1, 1]$ 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,

$$\nu = \max(\min(\beta_1, K_p(x_g - \hat{x}), \beta_2) \tag{4.13}$$

where the $\beta$'s are lower and upper amplitude limits, $x_g$ is the position of the goal, and $K_p$ the proportional gain which was tuned through trials. In Figure 4.5 we illustrate the complete control flow.

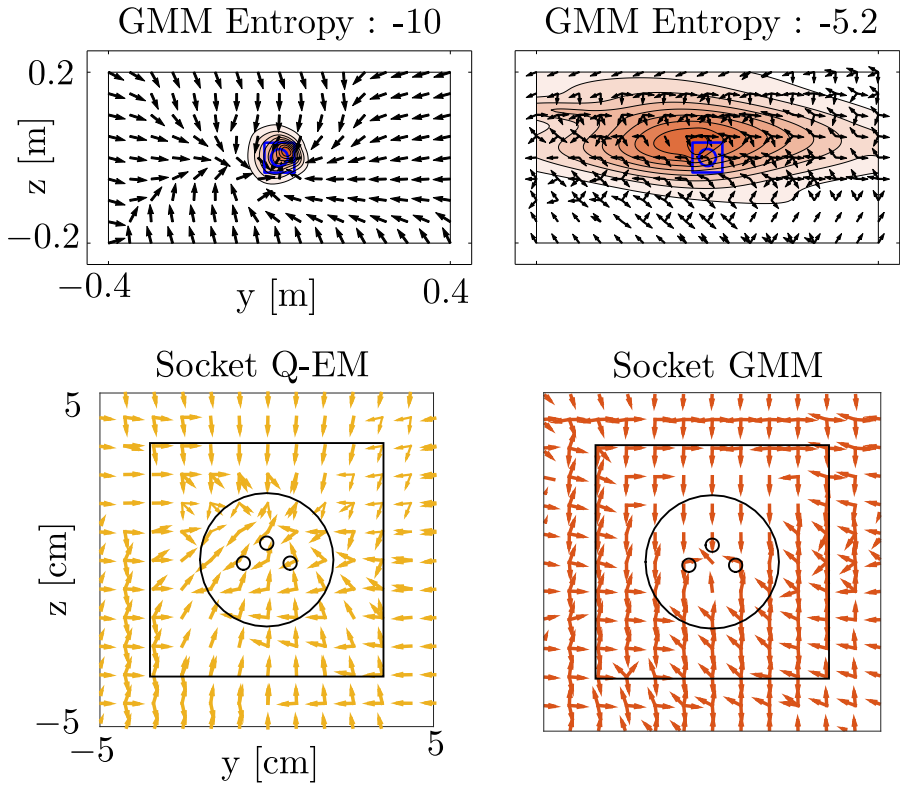



**Figure 4.6:** Q-EM and GMM policy vector fields. *Top*: The GMM policy is conditioned on an entropy of $-10$ and $-5.2$. For the lowest entropy level, most of the probability mass is close to the socket area since this level corresponds to very little uncertainty; we are already localised. We can see 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.

## 4.6 Results

We evaluate the following three properties 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 as a result gives an appreciation of the problem's difficulty.

2. **Importance of data** provided by human teachers. We evaluate whether it is possible to improve the Greedy using it as means of generating demonstrations, which we call Q-Greedy. This is to test whether a human teacher are necessary instead of using heuristics to gather demonstrations. We evaluate whether it is possible to obtain a good policy from the worst two teachers. 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 was located at the center of the wooden wall. We test the generalisation of the policy in finding a new socket location. We further test whether the policy can generalise to two new sockets which were not used during the training phase.

We evaluate the above properties under two separate conditions. In the **first condition** we consider the period between the start of the search until the socket is localised. In the **second condition** we consider the period from the point the socket is found until a connection has been established. In the first condition the evaluation is done in simulation whilst in the second condition, when the socket is found, we perform the evaluation with a physical robot, the KUKA LWR4.

This choice is motivated by the fact that the two parts of the task require different levels of precision. Finding the socket requires much less precision than establishing a connection. It is thus more informative to consider the performance of these two parts separately. Another aspect is that the search for the socket can be reliably evaluated in simulation since the physics of the interaction is simple. The connection phase is more complicated and a simulation would be unrealistic. For the evaluation of the connection of the plug to the socket we consider the search start point already within the vicinity of the socket.

### 4.6.1 Distance taken to reach the socket's edge (Qualitative)

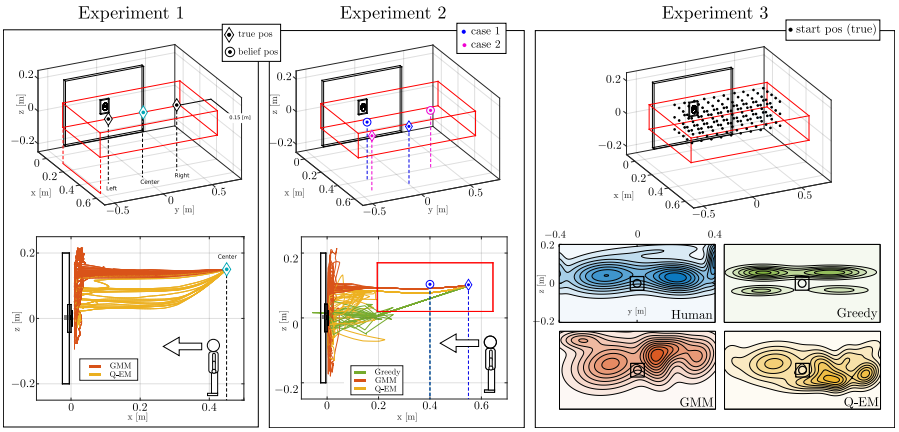



**Figure 4.7:** Three simulated search experiments. **Experiment 1:** Three start positions are considered: *Left*, *Center* and *Right* in which the triangles depict true position of the end-effector. The red cube illustrates the extent of the uncertainty. In the second row of Experiment 1, we illustrate the trajectories of both the GMM (orange) and Q-EM (yellow) policies. For each start condition a total of 25 searches were performed for each search policy. **Experiment 2:** Two cases are considered: *Case 1* blue, the initial belief state (circle) is fixed facing the left edge of the wall and the true location (diamond) is facing the socket. *Case 2* pink, the initial belief state (circle) is fixed to the right facing the edge of the wall and the true location is the left edge of the wall. In the second row, the trajectories are plotted for *Case 1*. **Experiment 3:** A 150 start locations are deterministically generated from a grid in the start area. In the second row, we plot the distribution of the areas visited by the true position during the search.

We consider three search experiments which we refer to as **Experiment 1, 2** and **3**, in order to evaluate the performance (distance travelled to reach the socket) of 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 4.7, *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 4.7, *Experiment 2*. A total of 25 searches are carried for each of the two conditions.

In **Experiment 3**, Figure 4.7, *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.

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

We can see a clear difference between the trajectories generated by the GMM

and Q-EM policies in Experiment 1, see Figure 4.7 *Experiment 1, second row*. 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 due to the initial starting condition being lower than in Experiment 1.

The trajectories of the Greedy policy depend on the chosen believed location (most likely state of the PMF). In the second experiment 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 is registered when the believed location reaches the wall as the true location is before the believed location, see Figure 4.7, *Experiment 2, second row*.

In Figure 4.7 *Experiment 3, second row*, both Human and GMM distributions of searched locations are similar. 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.

The distribution of the searched locations of the Q-EM policy is centred around the origin of the $z$-axis, see Figure 4.7 *Experiment 3, second row*. 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 whilst remaining in the vicinity of the socket. The Greedy's policy search distribution is multimodal 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 4.8 (*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.

### 4.6.2 Distance taken to reach the socket's edge (Quantitative)

In Figure 4.9 we illustrate the quantitative results of the distance taken to

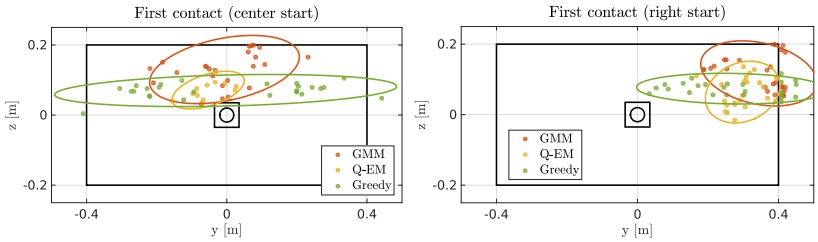


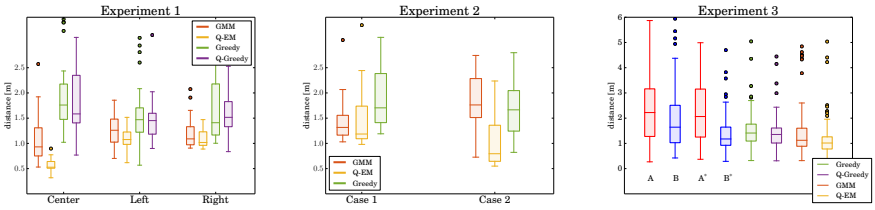

**Figure 4.8:** First contact with the wall, during experiment 1. (a) Contact distribution for initial condition "Center" . (b) Contact distribution for initial condition was "Right". The ellipses correspond to two standard deviations of a fitted Gaussian function.

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**, 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, given that the left edge of the wall is within close proximity. This on average results in an exploration in the upper left area of the wall, which explains why *Case 1* does 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 it does for Case 1, Figure 4.9 (b), as reason for the improvement over Case 1, is that in *Case 2* the true location of the end-effector is close to an edge which is an informative feature and results in a much faster localisation.

From **Experiment 3**, Figure 4.9 (c), it is clear that the three search policies have less variation in the distance travelled to find the socket's edge than the human teachers. 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




**(a)** **(b)** **(c)**

**Figure 4.9:** Distance travelled until the socket's edge is reached. (a) Three groups correspond to the initial conditions: Center, Left and Right depicted in Figure 4.7, *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 4.7, *top middle*, both the true position and most likely state are fixed. The Q-EM method always improves on the GMM. (c) Results corresponding to Experiment 3, Figure 4.7, *top right*. Again the Q-EM method is better, but at a less significant level.

by the GMM, the learned policy will perform poorly. We next evaluate the validity of this assumption and the importance of the training data provided by the human teachers.

### 4.6.3 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 4.10 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 has sensory feedback which is necessary to maintain an accurate position estimate.

Figure 4.11 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 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 4.12, 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

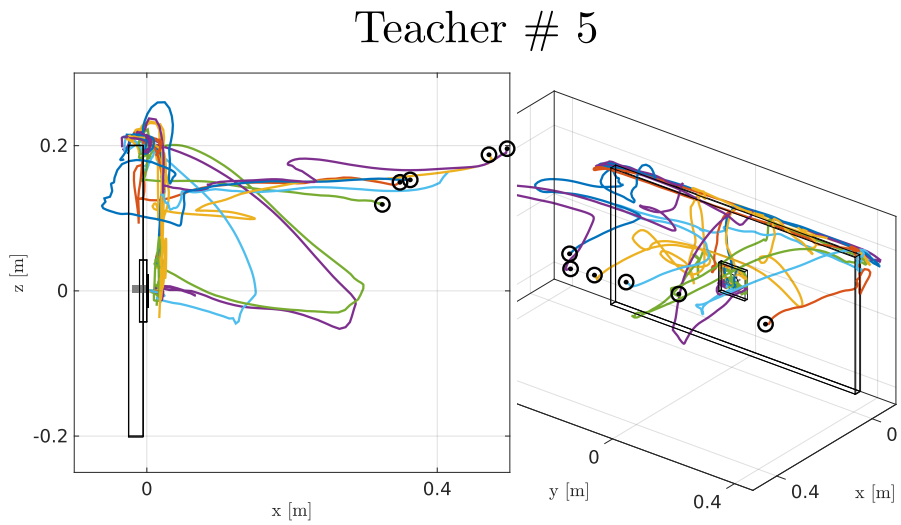



**Figure 4.10:** Original teacher # 5 demonstrations. The teacher demonstrates a preference to go first to the top of the wall. He then leaves contact with the wall to position himself in front of the socket before trying to find it

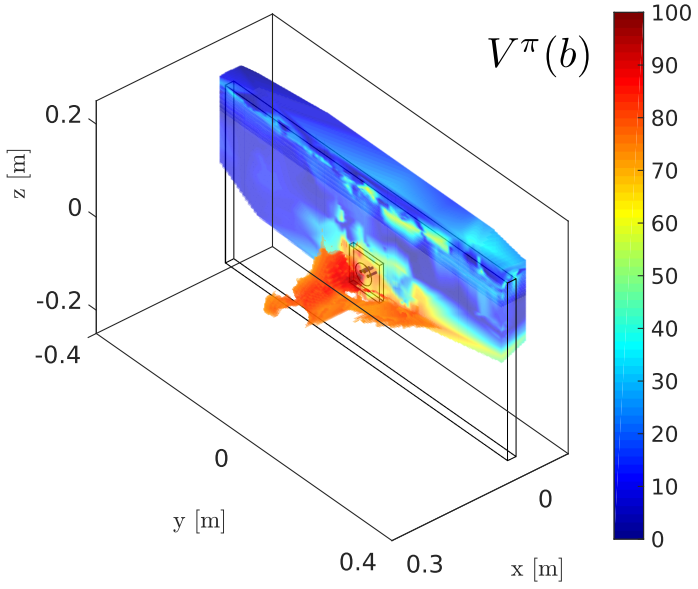



**Figure 4.11:** Value function learned from the 15 demonstrations of teacher #5.

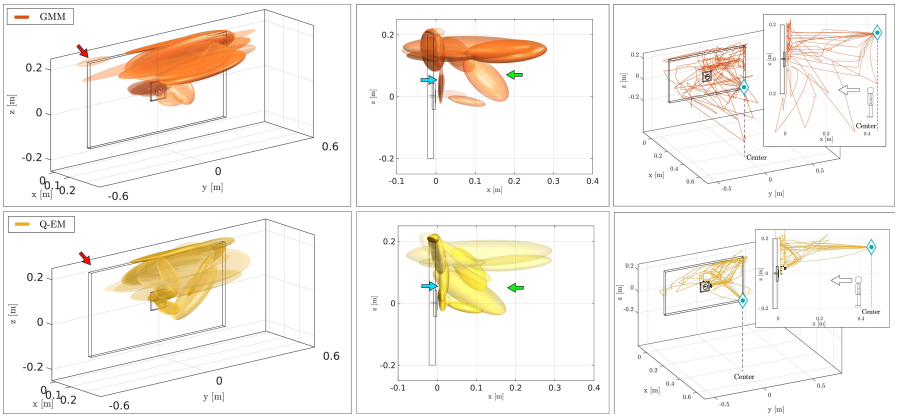



**Figure 4.12:** Marginalised Gaussian Mixture parameters of the GMM and Q-EM learned from the demonstrations of teacher #5. The illustrated transparency of the Gaussian functions is proportional to their weight. *Left column*: The Gaussian functions of the Q-EM have shifted from the left corner to the right. This is a result of the value function being higher in the top right corner region, see Figure 4.11. *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.

*Center* initial condition used in Experiment 1. We note that the trajectories of the GMM policy seem to have a lot of variance in contrast to the Q-EM policy, resulting from an average of variance amongst the 15 original demonstrations given by the teacher. 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 4.13, 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 4.9 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

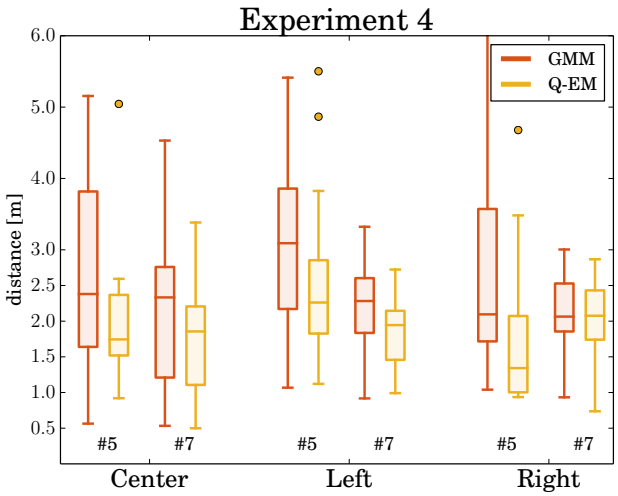


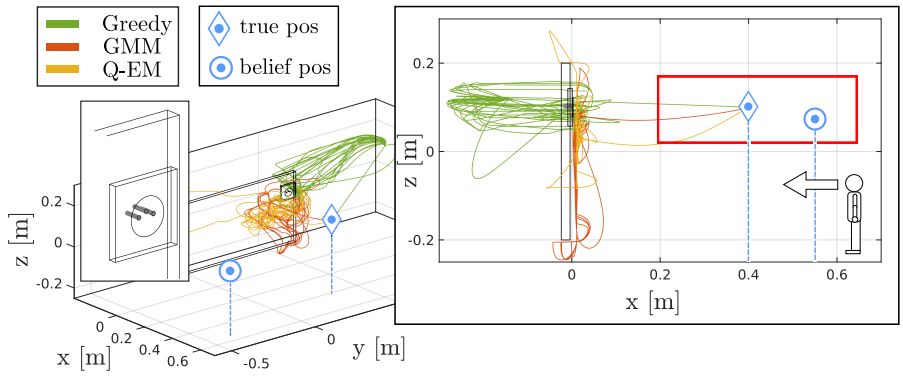

**Figure 4.13:** Results of a GMM and Q-EM policy under the same test conditions as Experiment 1. The Q-EM policy nearly always does much better than the GMM policy.

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.

### 4.6.4 Generalisation

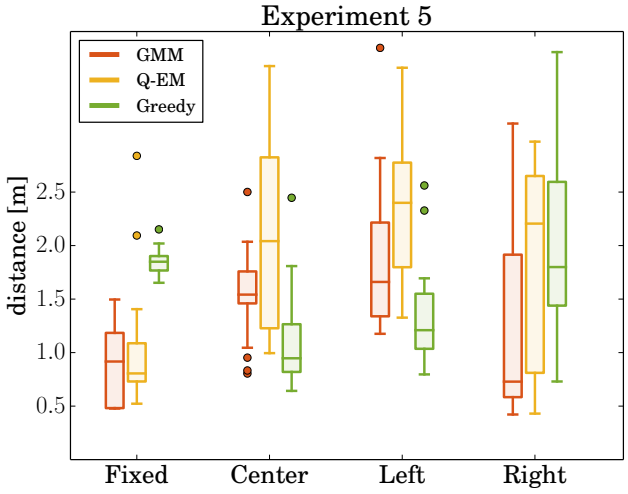
An important aspect of a policy or any machine learning methodology is to be able to generalise. So far we have trained and evaluated our policy within the same environment. To test whether our GMM policies can generalise to a new setting we changed the location of the socket to the upper right corner of the wall. The GMM was trained in the frame of reference of the 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, see Figure 4.14, which 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 4.15 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




**(a)**

**Figure 4.14:** 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.




**(a)**

**Figure 4.15:** Distance taken to reach the socket's edge. For the Fixed setup (see Figure 4.14) for the initial condition), both the Q-EM and GMM significantly outperform the Greedy.

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.

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 more likely will the search quality degrade. 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. The policy will always be able to find the socket once it has localised itself. This is can be seen from the vector field of the policy, see Figure 4.6, when the entropy is low. In this case the policy acts like a point attractor.

### 4.6.5 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 4.17 for an illustration of the initial starting condition. This initial configuration was used to evaluate the search policies for three different sockets, see Figure 4.16 (a) for an illustration of the sockets. We kept the same initial configuration for the evaluation of the three sockets so that we can observe the generalisation properties of the policies. As a reminder we only used 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 harder 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 4.16 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 instead tries

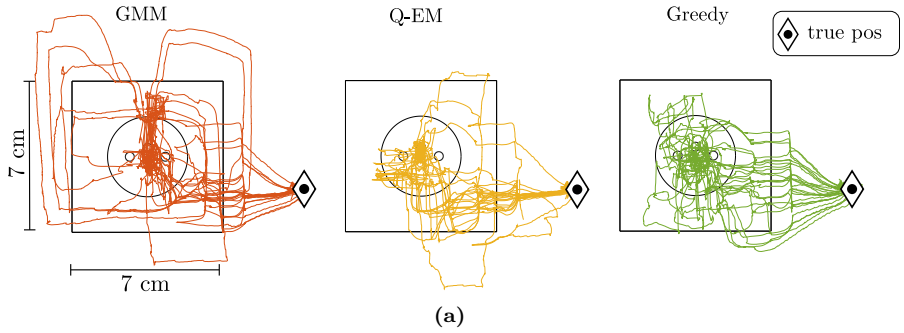
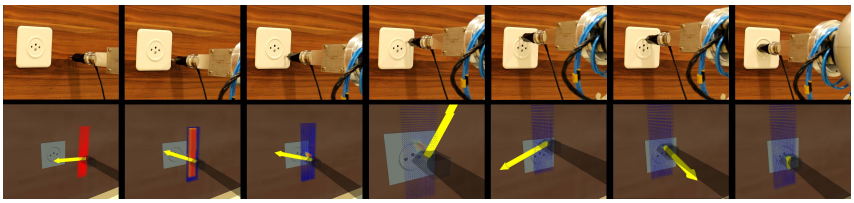



**Figure 4.16:** 25 search trajectories for each of the three search policies for socket A.

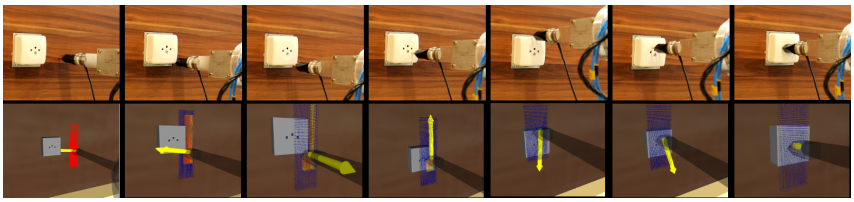
to directly establish a connection. In Figure 4.18 (c) we can see that for socket A both the Greedy and Q-EM are better than the GMM and the Q-EM has less variance in comparison to the Greedy searches. When compared to the human's performance, all three search methods are vastly superior, see Figure 4.18. In Figure 4.17 we illustrate 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 vertical up and down motions. Only when the uncertainty has been reduced does the GMM policy try to go towards the socket's connector.

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 isn't 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. In Figure 4.18 (c) we illustrate the statistics of the distance taken to establish a connection for all three sockets. The point of interest is that both the GMM and Q-EM algorithms do 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 in the Greedy case 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 search discrepancy between the performance of the humans and 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. 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, we are far less precise than the KUKA which has sub-millimetre accuracy.

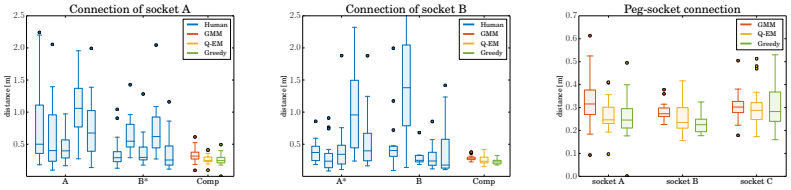


**(a)**



**(b)**

**Figure 4.17:** 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 are screen captures of the 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 makes a connection.




**(a)** **(b)** **(c)**

**Figure 4.18:** 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 B$^*$ first gave demonstrations on Socket B before giving demonstrations on Socket A. Group B$^*$ is better than group A 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 A$^*$ and B 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. (c) Distance taken (measured from point of contact of plug with socket edge) to connect the plug to the socket.

## 4.7 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 where learned from the data recordedj during the teaching by the human teachers . 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 weigh training datapoints in the M-step update of Expectation-Maximisation (EM). The second policy, called GMM, was learned using the standard EM algorithm, considering all training data points equally, following in the footsteps of our initial approach Chambrier and Billard (2014). 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 was always 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 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 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 was better. For the initial conditions in which the Greedy policy enters in contact with the wall at an early stage, it 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 the three policies on the KUKA LWR4 robot. First 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 do better than the Greedy, as socket C has no features on its surface, this being a disadvantage for a myopic policy.

We concluded 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 of doing 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.

## 4.8 Appendix

### 4.8.1 EM policy search

Steps taken to make a policy $pi_{\boldsymbol{\theta}}(\dot{x}, b)$ maximise the objective function, $J(\boldsymbol{\theta})$. The policy will be maximised with respect to the lower bound of the cost function $J(\boldsymbol{\theta})$:

$$\begin{aligned} J(\boldsymbol{\theta}') &= \sum_{i \in \mathbb{T}} \pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]}) \\ &= \sum_{i \in \mathbb{T}} \frac{\pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})}{\pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})}\, \pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]}) \end{aligned} \tag{4.14}$$

where $\mathbb{T}$ is the set of all rollouts. Next we take the logarithm and make use of Jensen's inequality and move the logarithm into the summation.

$$\log(J(\boldsymbol{\theta}')) = \log \sum_{i \in \mathbb{T}} \frac{\pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})}{\pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})} \pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]})$$
$$\geq \sum_{i \in \mathbb{T}} \log \left( \frac{\pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})}{\pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})} \right) \pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]}) \tag{4.15}$$

We take the derivative of the lower bound of $\log(J(\boldsymbol{\theta}'))$, Equation 4.15, with respect to $\boldsymbol{\theta}'$ and set it to zero so as to maximise the cost function.

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}'} \log(J(\boldsymbol{\theta}')) = \\
&\sum_{i \in \mathbb{T}} \nabla_{\boldsymbol{\theta}'} \log\left(\pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})\right) \pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]}) \\
&\underbrace{- \nabla_{\boldsymbol{\theta}'} \log\left(\pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\right) \pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]})}_{=0} \\
&= \sum_{i \in \mathbb{T}} \nabla_{\boldsymbol{\theta}'} \log\left(\pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})\right) \pi_{\boldsymbol{\theta}}(\dot{x}^{[i]}, b^{[i]})\, R(\dot{x}^{[i]}, b^{[i]}) \\
&= \mathbb{E}_{\pi_{\boldsymbol{\theta}}(\dot{x}, b)} \Big\{ \nabla_{\boldsymbol{\theta}'} \log\left(\pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]})\right) R(\dot{x}^{[i]}, b^{[i]}) \Big\}
\end{aligned} \tag{4.16}$$

$$\nabla_{\boldsymbol{\theta}'} \log(J(\boldsymbol{\theta}')) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}(\dot{x}, b)} \Big\{ R(b^{[i]}, \dot{x}^{[i]}) \sum_{t=0}^{T} \nabla_{\boldsymbol{\theta}'} \log \pi_{\boldsymbol{\theta}'}(\dot{x}^{[i]}, b^{[i]}) \Big\} \tag{4.17}$$
$$= \sum_{i=1}^{N} \sum_{t=0}^{T^{[i]}} \nabla_{\boldsymbol{\theta}'} \log \pi_{\boldsymbol{\theta}'}(\dot{x}_t^{[i]}, b_t^{[i]})\, Q^{\pi}(\dot{x}_t^{[i]}, b_t^{[i]}) \tag{4.18}$$

The reader is referred to Deisenroth et al. (2013a) for more details regarding Expectation-Maximisation and policy search in reinforcement learning.

### 4.8.2 Q-EM for GMM derivation

---

Making the substitution $x = (\dot{x}, b)^{\mathrm{T}}$ (small abuse of the notation) and insuring a positive Q-function, $Q^{\pi}(x^{[m]}) \geq 0$ and by setting the derivative of Equation 4.18 to zero and solving for the parameters $\boldsymbol{\theta} = \{w, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ we get a new weighted Maximisation update step in EM:

$$\nabla_{\boldsymbol{\mu}^{[k]}} \log J(\boldsymbol{\theta}) = \sum_{m=1}^{M} \alpha(z_{mk})\, Q(x^{[m]})\, {\boldsymbol{\Sigma}^{[k]}}^{-1} (x^{[m]} - \boldsymbol{\mu}^{[k]}) = 0$$

$$\boldsymbol{\mu}_{\mathbf{new}}^{[k]} = \frac{\sum_{m=1}^{M} \alpha(z_{mk})\, Q(x^{[m]})\, x^{[m]}}{\sum_{j=1}^{M} \alpha(z_{jk})\, Q(x^{[j]})} \tag{4.19}$$

where $\alpha(z_{mk})$ is the responsibility factor, denoting the probability that data point $m$ is a member of the Gaussian function $k$.

$$\alpha(z_{mk}) = \frac{w^{[k]} \cdot g(x^{[m]}; \boldsymbol{\mu}^{[k]}, \boldsymbol{\Sigma}^{[k]})}{\sum_{j=1}^{K} w^{[j]} \cdot g(x^{[m]}; \boldsymbol{\mu}^{[j]}, \boldsymbol{\Sigma}^{[j]})} \tag{4.20}$$

$$\boldsymbol{\Sigma}_{\mathbf{new}}^{[k]} = \frac{\sum_{m=1}^{M} Q(x^{[m]}) \alpha(z_{mk}) (x^{[m]} - \boldsymbol{\mu}^{[k]})(x^{[m]} - \boldsymbol{\mu}^{[k]})^{\mathrm{T}}}{\sum_{j=1}^{M} Q(x^{[j]})\, \alpha(z_{jk})} \tag{4.21}$$

$$w_{\mathbf{new}}^{[k]} = \frac{\sum_{m=1}^{M} Q(x^{[m]})\, \alpha(mk)}{\sum_{j=1}^{M} Q(x^{[j]})} \tag{4.22}$$

### 4.8.3 Unbiased estimator

The temporal difference error is an unbiased estimate of the advantage function:

$$\begin{aligned} \mathop{\mathbb{E}}_{\pi_{\boldsymbol{\theta}}} \{\delta_t^{\pi} | b_t, u_t\} &= \mathop{\mathbb{E}}_{\pi_{\boldsymbol{\theta}}} \{r_{t+1} + \gamma V^{\pi}(b_{t+1}) | b_t, u_t\} - V^{\pi}(b_t) \\ &= Q^{\pi}(b_t, u_t) - V^{\pi}(b_t) \\ &= A^{\pi}(b_t, u_t) \end{aligned} \tag{4.23}$$