

Learning assistive teleoperation behaviors from demonstration

Ioannis Havoutis and Sylvain Calinon

Abstract—Emergency response in hostile environments often involves *remotely operated vehicles (ROVs)* that are teleoperated as interaction with the environment is typically required. Many ROV tasks are common to such scenarios and are often recurrent. We show how a probabilistic approach can be used to learn a task behavior model from data. Such a model can then be used to assist an operator performing the same task in future missions. We show how this approach can capture behaviors (constraints) that are present in the training data, and how this model can be combined with the operator’s input online. We present an illustrative planar example and elaborate with a *non-Destructive testing (NDT)* scanning task on a teleoperation mock-up using a two-armed Baxter robot. We demonstrate how our approach can learn from examples task specific behaviors and automatically control the overall system, combining the operator’s input and the learned model online, in an assistive teleoperation manner. This can potentially reduce the time and effort required to perform teleoperation tasks that are commonplace to ROV missions in the context of security, maintenance and rescue robotics.

I. INTRODUCTION

Disaster scenarios require swift action and efficient use of available technology. Often the success of such operations depends on time-critical factors, while the lack of structure in such incidents makes the use of autonomous approaches difficult. In many cases, teleoperation of a Remotely Operated Vehicle (ROV) is a reliable choice, as this allows experts to assess the situation online from a safe distance.

In many disaster scenarios visual assessment of the situation is the first step taken. Nonetheless interaction with the environment, where the ROV is operating in, is often required. For this step, the response team uses an operator –a pilot experienced in ROV control– who is responsible for performing a set of specified tasks. The set of tasks relevant to disaster response can vary according to the situation at hand. Nonetheless, there are tasks that are common to such scenarios and are often recurrent.

Within the DexROV project [1] we are looking into ways of assisting operators performing such tasks, using probabilistic approaches to learn manipulation task representations from teleoperation data. We aim at assisting the operator by learning a set of behaviors that are appropriate for the tasks at hand, either as a training step before a mission or from prior teleoperation data. In such a setting, part of the control authority can be passed to the robot (ROV) and the resulting overall behavior of the system is a combination of the learned task behavior and the operator’s input.

The aim of such *mixed teleoperation* approach is to boost the efficiency of the system by reducing the time and effort



Fig. 1: The DexROV project’s simulated underwater environment. On the left, a rig with a set of manipulation test cases, i.e., a fish-tail handle plug, a hot-stub with standardized D-handle, a T-handle tool and a rotary ROV valve. The simulated ROV is rendered on the right. *Inset image*: a NDT probe used to scan surfaces for defects (cracks, breaks, etc.).

required to perform tasks that were previously encountered. Typical examples include a variety of ROV situations, ranging from inspection, maintenance and facility construction to decommissioning of these infrastructures.

We are interested in the use of ROVs in underwater activities (Fig. 1) that are not safe for human divers, either because of the great depth where ROVs need to operate (for example cable inspection, infrastructure maintenance, underwater biology and archeology) or other non-structured factors such as disaster response and recovery (oil & gas accidents and emergencies, pipes, platforms and drill site evaluation).

II. MOTIVATION

Consider a scenario where an accident at an off-shore site, for example an off-shore laboratory or a natural gas platform, has caused a fire and a series of small explosions. Let’s assume that all personnel has been evacuated safely from the facility and the fire was extinguished by the automatic fire suppression system. Now an expert team is called in for rapid and secure inspection of critical infrastructure, to assess the condition of the facility. For an off-shore site, this would include the assessment of the condition of the supporting infrastructure, something usually performed with means of non-destructive testing (NDT). NDT in small depths can be performed by divers in situations where there is no further risk, for example in routine structural inspection. In the case following an event,

Idiap Research Institute, Martigny, Switzerland. {ioannis.havoutis, sylvain.calinon}@idiap.ch

This work was supported by the DexROV project through the EC Horizon 2020 programme (Grant #635491).

an ROV can be used to perform the task, keeping the operator at a safe distance.

NDT of underwater infrastructure is typically a time consuming task for both human divers and teleoperated ROVs. It involves “scanning” a number of surfaces that are of interest while keeping the NDT probe at a certain angle against the surface and moving at a certain speed. Think of the end-effector state of a teleoperated arm performing the NDT scanning task. The constraint in this behavior is to keep the probe at a certain angle (low variance) against the surface while the position of the probe can take a range of values (high variance). The approach we present exploits this observation of difference in the signals (position and orientation of the scanning tool for the NDT task), and encodes them in a behavior used to assist the teleoperator in subsequent missions that require NDT scanning.

Learning of assistive (supportive) behaviors can be used for a large variety of tasks that are common in teleoperation scenarios and involve manipulation of tools or objects in the environment. For example a drilling task would involve the teleoperated arm to hold the drill perpendicular to the surface (low variance), with the option of drilling holes in a range of locations (high variance). Similarly, for plugging a connector to a socket, the orientation of the plug would remain approximately identical through different data samples of such a motion while the position of the plug can change substantially with respect to the base of the manipulator arm. If we now think of the position of the plug from the perspective of the socket –a simple change in the frame of reference– we see that, in this frame, the position of the plug is also converging to a particular point. By changing the frame of reference we can observe another low variance signal stemming from the constraint of the task.

Similar tasks include, pushing a switch, reaching for a tool (handle), turning an ROV valve, hooking a carabiner to a metallic bar, etc. The common thread is that all such tasks have a structure -or constraint- which we can observe in and learn from data, represent in a probabilistic model and use in support of the teleoperation. This way the system follows the operator’s lead in areas where, according to the learned task behavior, the task varies (high variance) and automatically converges to positions and/or orientations of observed task constraints (low variance) accordingly.

To learn and represent the behavior of a task, we use a task-parametrized version of a Gaussian Mixture Model (TP-GMM) [2]. To generate reference behavior according to a learned model we use Gaussian Mixture Regression (GMR) [2]. We represent the input of the operator as an extra frame in the TP-GMM formulation and compute the combination of the operator’s input and the learned task behavior using products of Gaussians.

The rest of the paper is structured as follows. Section III presents an overview of related work. In Section IV we present our approach alongside an illustrative planar example. Next, we show how this approach can scale to a realistic teleoperation scenario and provide an example using the arms of a Baxter robot as a teleoperation mock-up (Section V). Last

we conclude with a discussion on our results and give possible directions for future work in Section VI.

III. RELATED WORK

A variety of learning from demonstration (LfD) approaches exist in literature but are most often used for encoding motions (of joints or end-effectors) that are then autonomously performed. In this paper we are mostly interested in learning behaviors that can be combined with the input of an operator, providing support, assistance and guidance, with the aim of boosting the efficiency and performance of the overall system.

A popular *learning by demonstration* (LbD) approach is to use *Dynamic Movement Primitives* (DMPs) [3]. DMPs are dynamical systems with simple convergence behaviour that are coupled with a learned non-linear function that modulates their output. DMPs can provide adaptive motion representations that are easy to implement. One drawback of standard DMPs is that a sequence of radial basis activation functions needs to be allocated manually (usually spread equally in time with a shared variance), and that each DoF of the system is separately described (synchronized by a phase variable), sometimes leading to sets of DMPs that have difficulty in capturing joint synergies if too few basis functions are used.

Along the same direction, the Probabilistic Movement Primitive (ProMP) approach in [4], [5], uses a model-free approach to encode a distribution over trajectories and analytically derive a stochastic feedback controller to reproduce the given trajectory distribution. This allows for flexibility over the possible motion generation such as spatial and temporal rescaling, combination and blending of the modeled motion primitives (MPs).

A complementary research direction aims at moving away from the autonomous motion generation and execution paradigm by considering the human operator as part of the system. For example, the work in [6] proposes an approach for modeling of *virtual guides* in co-manipulation examples. They show how multiple guides can be used in pick-and-place tasks of heavy objects and how the combination and switching between virtual guides can reduce the need of manual tuning and *a priori* expert design.

The LfD approach that we selected is to encode demonstrated samples in a TP-GMM and use Gaussian Mixture Regression (GMR) to regenerate the motion or behavior [2]. With a TP-GMM representation, the reproduction of a reference movement or an average skill behavior (the focus of our work in this paper) can be formalized as a regression problem [7]. We showed that in robotics, GMR offers a simple solution to handle encoding, recognition, prediction and reproduction in robot learning [8]. GMR relies on basic properties of normal distributions such as linear transformation and Gaussian conditioning. It provides a probabilistic representation of movements or policies, in which the model can compute the next actions on-the-fly, with a computation time that is independent of the number of datapoints used to train the system.

In contrast to other regression methods such as *Locally Weighted Regression* (LWR) [9], *Locally Weighted Projection*

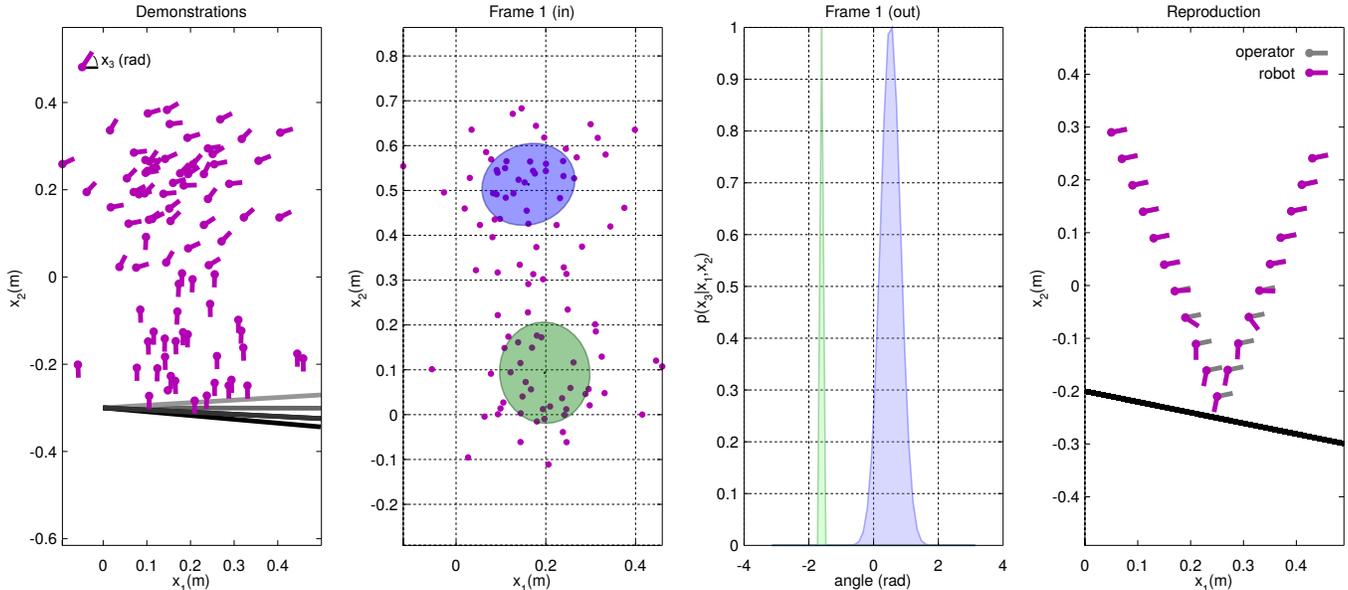


Fig. 2: Planar example of our approach to learning teleoperation behaviors from demonstration. *Left*: Demonstrated samples of tool orientation far and near the scanning surface (black lines) *Middle*: The input states of the model in position and the output orientation distribution. *Right*: The resulting overall behavior of the system produced by the combination of the operator’s input and the learned behavior (cf. Fig. 3). Please refer to Section IV for further details.

Regression (LWPR) [10], or *Gaussian Process Regression* (GPR) [11], [12], GMR does not model the regression function directly, but models the joint probability density function of the data. It then derives the regression function from the joint density model. This can be an advantage in robot applications where the input and output components are only specified at the very last step of the process, if input information are missing at some iterations, or if it is not relevant to retrieve the whole set of outputs (e.g. to predict or react as rapidly as possible). Density estimation can thus be learned in an off-line phase (with the TP-GMM estimation step), allowing the regression process to be computed rapidly.

IV. APPROACH

In this section we present our approach using a running example of a planar task, i.e. 2 position variables and 1 orientation. Consider the task of NDT testing in a planar world. The probe can move in the plane, along x_1 and x_2 , and the direction that the NDT probe is pointing at is given by the angle x_3 (with respect to the horizontal axis). A line in this setup represents the surface that we would like to scan, for simplicity by pointing the probe towards the line and getting a sensor reading. The constraint of this task is that the probe requires to be perpendicular to the surface (line), in order to get a clean reading. We are interested in learning the behavior of how to orient the probe from demonstration data.

Our method consists of two phases; first a model of the desired behavior is learned from demonstration, using data obtained as the operator is performing the task, and second – during the deployment phase – the learned model is used to assist the operator in efficiently completing the learned task.

We use a task-parametrized Gaussian Mixture Model (TP-GMM) to build a representation of the planar NDT task from demonstration [2]. The task-parametrized formulation allows us to attach frames (coordinate systems) to places that are relevant to the tasks we wish to model. Accordingly we attach a frame to the line representing the surface of interest. For this task we only use one such frame while complex tasks would typically require more frames acting as candidate frames of reference, e.g. landmarks on the robot(s) body, obstacles, tools.

Each frame is fully described by $\{\mathbf{b}_j, \mathbf{A}_j\}_{j=1}^P$, where P is the number of frames in the TP-GMM ($P = 1$ in this example). These are the task parameters of the model, that typically vary over time, while without loss of generality we assume static for this example. $\{\mathbf{b}_1, \mathbf{A}_1\}$ in our example describes where in the plane and with what angle the line is positioned (Fig. 2 *left*). With this, we can translate demonstrations (training samples) to a frame-local representation.

We ask the operator to perform the planar NDT task a number of times and collect a set of N datapoints $\{\boldsymbol{\xi}_n \in \mathbb{R}^{D \times N}$. For our running example $D = 3$ and $\boldsymbol{\xi}_n = [x_{1,n}, x_{2,n}, x_{3,n}]^T$, the position and angle of the NDT probe at each sample (Fig. 2 *left*).

Each sample is observed from the perspective of the frames of the model (a single frame for our example), hence samples collected in the global frame are transformed with

$$\mathbf{X}^{(j)} = \mathbf{A}_j^{-1}(\boldsymbol{\xi} - \mathbf{b}_j), \quad (1)$$

to the frame of the surface of interest (black line(s) in (Fig. 2 *left*).

A quick look at the data reveals the behavior of the planar NDT task. Samples that are close to the scanning surface (black line) are perpendicular to the surface, while samples

that are away from the line present a large variety of angles (orientations). In other words, near the line we consistently observe the task constraint (low variance across samples) and far from the line the probe orientation does not follow a consistent pattern (high variance across samples).

The parameters of a TP-GMM with K components are

$$\{\pi_i, \{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^P, \}_{i=1}^K,$$

where π_i are the mixing coefficients, $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ are the center and covariance matrix of the i -th Gaussian component in frame j . We estimate the priors, centers and covariances, of the model with an *expectation-maximization* (EM) algorithm [13], iteratively computed until model convergence, see [2]. The number of Gaussian components K is the only open parameter of the model and can also be estimated with a number of approaches (e.g., cross-validation, BIC, spectral learning, etc.). In this example, we set $K = 2$ empirically and estimate the model parameters.

The states of the TP-GMM represent the task behavior near and far to the surface. This learned model is drawn in the second plot of Fig. 2, note that this is the GMM displayed in the task frame (line) and ellipsoids represent 1 standard deviation isocontours of position variance. Next, we use Gaussian Mixture Regression (GMR) to compute and estimate the desired probe angle (x_3) conditioned on the position of the probe in the task frame. This procedure leads us to the second phase of our framework, that of model deployment, which is described next.

The superscripts \mathcal{I} and \mathcal{O} will be further used to describe the sets of dimensions that span input and output variables (that will be used as exponents in vectors and matrices). Also, we drop the frame exponent and introduce time as an index to show that this is the on-line system behavior.

At each time step t , the datapoint $\boldsymbol{\xi}_t$ can be decomposed as two subvectors $\boldsymbol{\xi}_t^{\mathcal{I}}$ and $\boldsymbol{\xi}_t^{\mathcal{O}}$ spanning for the input and output variables. For our example, \mathcal{I} corresponds to the position dimensions (x_1 and x_2), and \mathcal{O} corresponds to the output dimension describing the angle x_3 (orientation) of the probe.

With this notation, a block decomposition of the datapoint $\boldsymbol{\xi}_t$, vectors $\boldsymbol{\mu}_i$ and matrices $\boldsymbol{\Sigma}_i$ can be written as

$$\boldsymbol{\xi}_t = \begin{bmatrix} \boldsymbol{\xi}_t^{\mathcal{I}} \\ \boldsymbol{\xi}_t^{\mathcal{O}} \end{bmatrix}, \quad \boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{\mathcal{I}} \\ \boldsymbol{\mu}_i^{\mathcal{O}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{\mathcal{I}} & \boldsymbol{\Sigma}_i^{\mathcal{I}\mathcal{O}} \\ \boldsymbol{\Sigma}_i^{\mathcal{O}\mathcal{I}} & \boldsymbol{\Sigma}_i^{\mathcal{O}} \end{bmatrix}. \quad (2)$$

The TP-GMM estimated in the previous section encodes the joint distribution $\mathcal{P}(\boldsymbol{\xi}^{\mathcal{I}}, \boldsymbol{\xi}^{\mathcal{O}}) \sim \sum_{i=1}^K \pi_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ of the dataset $\boldsymbol{\xi}$. At each reproduction step t , $\mathcal{P}(\boldsymbol{\xi}_t^{\mathcal{O}} | \boldsymbol{\xi}_t^{\mathcal{I}})$ is computed as the conditional distribution

$$\mathcal{P}(\boldsymbol{\xi}_t^{\mathcal{O}} | \boldsymbol{\xi}_t^{\mathcal{I}}) \sim \sum_{i=1}^K h_i(\boldsymbol{\xi}_t^{\mathcal{I}}) \mathcal{N}(\hat{\boldsymbol{\mu}}_i^{\mathcal{O}}(\boldsymbol{\xi}_t^{\mathcal{I}}), \hat{\boldsymbol{\Sigma}}_i^{\mathcal{O}}), \quad (3)$$

$$\text{with } \hat{\boldsymbol{\mu}}_i^{\mathcal{O}}(\boldsymbol{\xi}_t^{\mathcal{I}}) = \boldsymbol{\mu}_i^{\mathcal{O}} + \boldsymbol{\Sigma}_i^{\mathcal{O}\mathcal{I}} \boldsymbol{\Sigma}_i^{\mathcal{I}}^{-1} (\boldsymbol{\xi}_t^{\mathcal{I}} - \boldsymbol{\mu}_i^{\mathcal{I}}), \quad (4)$$

$$\hat{\boldsymbol{\Sigma}}_i^{\mathcal{O}} = \boldsymbol{\Sigma}_i^{\mathcal{O}} - \boldsymbol{\Sigma}_i^{\mathcal{O}\mathcal{I}} \boldsymbol{\Sigma}_i^{\mathcal{I}}^{-1} \boldsymbol{\Sigma}_i^{\mathcal{I}\mathcal{O}}, \quad (5)$$

$$\text{and } h_i(\boldsymbol{\xi}_t^{\mathcal{I}}) = \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_t^{\mathcal{I}} | \boldsymbol{\mu}_i^{\mathcal{I}}, \boldsymbol{\Sigma}_i^{\mathcal{I}})}{\sum_k^K \pi_k \mathcal{N}(\boldsymbol{\xi}_t^{\mathcal{I}} | \boldsymbol{\mu}_k^{\mathcal{I}}, \boldsymbol{\Sigma}_k^{\mathcal{I}})}. \quad (6)$$

Eq. (3) represents a multimodal distribution. In many cases a single peaked output distribution is preferred. Likewise the

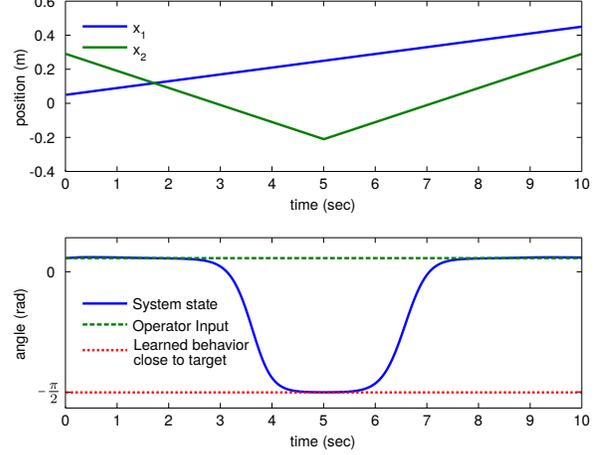


Fig. 3: State of the NDT probe corresponding to the last plot of Fig. 2. Note that as the position approaches the frame, the robot automatically aligns the output to be perpendicular to the surface of interest, as consistently observed in the demonstration samples.

output distribution can then be approximated by the Gaussian

$$\mathcal{P}(\boldsymbol{\xi}_t^{\mathcal{O}} | \boldsymbol{\xi}_t^{\mathcal{I}}) = \mathcal{N}(\boldsymbol{\xi}_t^{\mathcal{O}} | \hat{\boldsymbol{\mu}}_t^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}}). \quad (7)$$

Eq. (7) can be computed in a very rapid manner from the model parameters, allowing online regression. This makes GMR attractive for computing the desired behavior based on the learned task model. This is described in the third plot of Fig. 2 where we see that for the state far from the surface (blue) we have a large variance, while for the state near the surface we compute a highly peaked distribution around an orientation perpendicular to the task frame¹.

Up to now we have focused on learning the model of the task behavior and estimating the desired behavior given the current system state. Our system also integrates the input of the operator that is combined with the model of the task behavior. We model the input of the teleoperator as a second Gaussian $\mathcal{N}(\boldsymbol{\xi}_t^{\mathcal{I}} | \boldsymbol{\mu}_t^{\mathcal{I}}, \boldsymbol{\Sigma}_t^{\mathcal{I}})$, where $\boldsymbol{\mu}_t^{\mathcal{I}}$ is the current operator's input, while the variance $\boldsymbol{\Sigma}_t^{\mathcal{I}}$ is a measure of the operator's input weight. To combine these two inputs, we compute the Gaussian product of the two as

$$\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t) \quad \text{with} \quad \hat{\boldsymbol{\Sigma}}_t = \left(\hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}}^{-1} + \boldsymbol{\Sigma}_t^{\mathcal{I}\mathcal{O}} \boldsymbol{\Sigma}_t^{\mathcal{I}}^{-1} \right)^{-1}, \quad (8)$$

$$\hat{\boldsymbol{\mu}}_t = \hat{\boldsymbol{\Sigma}}_t \left(\hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}}^{-1} \hat{\boldsymbol{\mu}}_t^{\mathcal{O}} + \boldsymbol{\Sigma}_t^{\mathcal{I}\mathcal{O}} \boldsymbol{\Sigma}_t^{\mathcal{I}}^{-1} \boldsymbol{\mu}_t^{\mathcal{I}} \right), \quad (9)$$

which corresponds to the average and covariance for the solution of the minimization problem

$$\min_{\boldsymbol{\xi}_t} \left(d_{\hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}}}(\boldsymbol{\xi}_t, \hat{\boldsymbol{\mu}}_t^{\mathcal{O}}) + d_{\boldsymbol{\Sigma}_t^{\mathcal{I}}}(\boldsymbol{\xi}_t, \boldsymbol{\mu}_t^{\mathcal{I}}) \right), \quad (10)$$

with the squared Mahalanobis distance

$$d_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^{\top} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{y}). \quad (11)$$

¹Note that the pdf corresponds to reproductions where the line angle is -0.2rad , appearing in the fourth plot of Fig. 2.

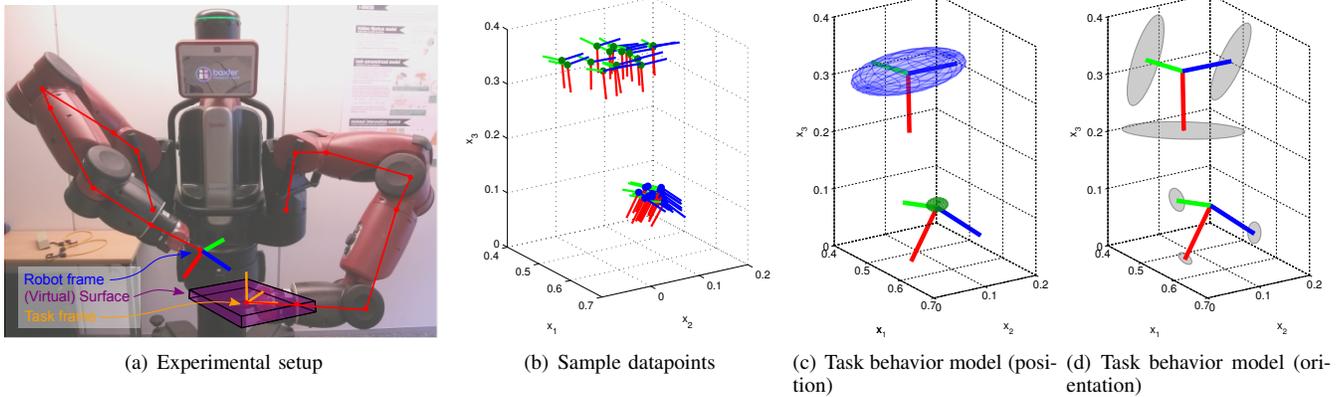


Fig. 4: (a) Teleoperation mock-up using the two-armed Baxter robot. The end-effector of the left arm is used as the task frame, representing the pose of the surface to be scanned. This can be arbitrarily positioned. The right arm is the teleoperated arm and the pose of its end-effector represents the pose of the NDT scanning probe. (b) Subset of the training data used to learn the NDT scanning behavior. (c),(d) The learned TP-GMM representing the NDT scanning behavior. See Section V for more details.

We track this reference using an infinite-horizon formulation of a Linear Quadratic Regulator (LQR). Within this setting we can use the estimated mean of the previous step as a reference, and the covariance as a weighting of the control cost, yielding a full stiffness and damping matrix at each time step. This way, we have already formulated a controller for the teleoperated robot, that acts according to the learned task behavior.

An example of the complete system is presented in the last plot of Fig. 2. Here we set the operator’s input to a linear motion towards the surface and away from it while also moving to the right. The operator’s input angle is kept constant throughout this reproduction (grey angle reference). The simulated behavior of the system is represented by the purple datapoints. As the NDT probe approaches the scan surface (black line), the robot automatically aligns the angle to be perpendicular to the task frame, as consistently observed in the demonstrations, “down-weighting” the input of the operator. When the position of the probe moves away from the task frame, the robot returns to the input angle of the operator. Fig. 3 presents the state evolution of this example in more detail.

V. EXPERIMENTAL EVALUATION

In this section we evaluate our approach on a similar task using the Baxter robot. This example demonstrates that our approach can be used with full pose information, using Cartesian position and quaternion orientation data to learn and generate an assistive task behavior from demonstration.

Building up on our previous example we set up a similar task using both arms of Baxter. We use the left arm of the robot as a way to represent the pose of the (virtual) surface to be scanned (Fig. 4(a)). Using the forward kinematics computation we calculate the position and orientation of the left end-effector, which we will regard as the frame of the surface of interest (similar to the line in the planar example). This way we can use a torque controller compensating for the effect

of gravity to move freely the frame of the task. We assume that the right arm is the robot arm that is holding the NDT probe. In this example, the probe is required to be at a 45° angle towards the surface frame, see Fig. 4(a) for a visual description of the task mock-up.

We begin by collecting samples from the task behavior that we wish to learn. Both arms are controlled with the gravity compensation controller to perform the NDT scanning task. We collect samples from a number of task configurations, by setting the left arm (scanning surface frame) to random poses. As in the planar example, we collect data in two approximate pose areas, near the surface and away from the surface. We learn a TP-GMM with one frame, describing the task position (surface), and the two states.

Fig. 4(b) shows a subset of the datapoints that are used to learn the task behavior. Near the top, we can observe the samples of the end-effector state when away from the task area. The samples here have high variance both in position and orientation. Close to the surface of interest (bottom part), the data is more concentrated to a particular area and the end-effector orientation is approximately identical for all poses (low variance).

Fig. 4(c) shows the TP-GMM learned from the data. The ellipsoids represent isocontours of one standard deviation, while the two axes’ positions and orientations represent the means of system states within the task frame. The model accurately captures the high variance in position and orientation for the far state, and the low variance of the state near the scanning surface (left arm endpoint frame).

For the reproduction of the behavior we use a constant operator’s input of an orientation that aligns the end-effector with the horizontal plane. We use a torque controller compensating the effect of gravity to allow the user to move the arm without effort while demonstrating the requested system behavior. We compute the product between the Gaussians predicted by the task behavior model and the operator’s input,

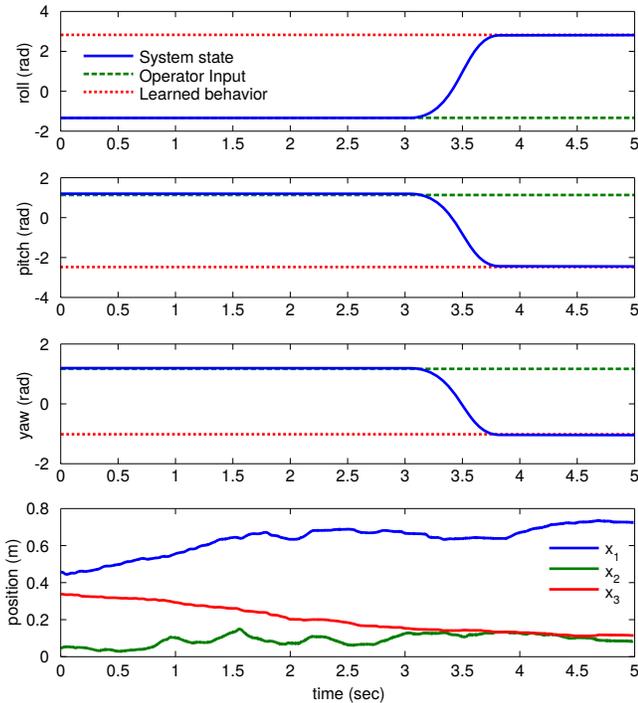


Fig. 5: NDT scanning trial on the Baxter robot mock-up, where we learn the tool orientation conditioned on the position. The top plots show the orientation of the end effector in axis-angle representation. Note that all computations are performed with a quaternion representation, while we plot the axis-angle values for clarity. The bottom plot shows the evolution of the position of the end-effector. See Section V for more details.

and use the result as reference to the infinite-horizon linear quadratic tracking formulation. This results in a reference pose that is tracked by a torque controller, active for the robot arm in question. An example trial is shown in Fig. 5. As the robot end-effector moves closer to the task frame, the orientation of the end-effector automatically changes to the required task specific orientation. In effect, the system smoothly changes from “trusting” the operator’s input to “trusting” the learned behavior, based on the covariance information captured when learning the task behavior from data². This has the effect of reducing the cognitive load of the teleoperator, who can concentrate on the high level parts of the task while delegating the recurring parts to the learning system (in this example, the operator does not need to worry about the precise alignment of the NDT probe with respect to the plane to analyse).

VI. CONCLUSION

In this paper we showed how a probabilistic model can be used to learn assistive teleoperation behaviors from demonstration. We presented a planar example to illustrate our approach, and a realistic NDT scanning example on a teleoperation mock-up using a two-armed Baxter robot. Our approach can potentially increase the efficiency of teleoperation tasks that

require interaction with the environment. The task behavior models can be learned from data of previous missions as such tasks are often recurrent. Our approach is general and can be used for various ROV tasks, such as drilling, using a screwdriver, pushing buttons, turning valves, cutting using a tool, hot-stabbing, etc.

Currently, we are looking into ways of incorporating more sensory dimensions in the learned behaviors. In particular, data such as force and torque are often crucial to task performance, and how to learn and represent sensorimotor behaviors is largely an open question. In future work, we also aim to move to semi-autonomous task execution, leading to more autonomous ROVs that need to only follow high-level commands and are capable of efficiently acting in unstructured environments, typical to emergency response scenarios.

REFERENCES

- [1] J. Gancet, P. Weiss, G. Antonelli, M. F. Pflingsthorn, S. Calinon, A. Turetta, C. Walen, D. Urbina, S. Govindaraj, C. A. Mueller, X. Martinez, T. Fromm, B. Chemisky, G. Indiveri, G. Casalino, P. A. Di Lillo, E. Simetti, D. De Palma, A. Birk, A. Tanwani, I. Havoutis, A. Caffaz, L. Guilpain, and P. Letier, “Dexterous undersea interventions with far distance onshore supervision: The DexROV project,” in *10th IFAC Conference on Control Applications in Marine Systems (CAMS)*, Trondheim, Norway, September 2016.
- [2] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.
- [3] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [4] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, “Probabilistic movement primitives,” in *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2013, pp. 2616–2624.
- [5] G. Neumann, C. Daniel, A. Paraschos, A. Kupcsik, and J. Peters, “Learning modular policies for robotics,” *Frontiers in Computational Neuroscience*, vol. 8, no. 62, 2014.
- [6] G. Raiola, X. Lamy, and F. Stulp, “Co-manipulation with multiple probabilistic virtual guides,” in *Proc. IEEE/RSSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, September 2015, pp. 7–13.
- [7] Z. Ghahramani and M. I. Jordan, “Supervised learning from incomplete data via an EM approach,” in *Advances in Neural Information Processing Systems (NIPS)*, J. D. Cowan, G. Tesauro, and J. Alsppector, Eds., vol. 6. San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc., 1994, pp. 120–127.
- [8] S. Calinon, F. D’halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, “Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression,” *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, June 2010.
- [9] S. Schaal and C. G. Atkeson, “Constructive incremental learning from only local information,” *Neural Computation*, vol. 10, no. 8, pp. 2047–2084, 1998.
- [10] S. Vijayakumar, A. D’souza, and S. Schaal, “Incremental online learning in high dimensions,” *Neural Computation*, vol. 17, no. 12, pp. 2602–2634, 2005.
- [11] D. Nguyen-Tuong and J. Peters, “Local Gaussian process regression for real-time model-based robot control,” in *Proc. IEEE/RSSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2008, pp. 380–385.
- [12] D. B. Grimes, R. Chalodhorn, and R. P. N. Rao, “Dynamic imitation in a humanoid robot through nonparametric probabilistic inference,” in *Proc. Robotics: Science and Systems (R:SS)*, 2006, pp. 1–8.
- [13] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

²Please see the accompanying video for a demonstration of this behavior.