Online Incremental Learning of Manipulation Tasks for Semi-Autonomous Teleoperation

Ioannis Havoutis[1], Ajay Kumar Tanwani[1,2], Sylvain Calinon[1]

I. Motivation

Remotely operated vehicles (ROVs) are becoming more commonly used for underwater tasks. Such tasks range from inspection and maintenance of underwater cables and pipelines, to underwater archaeology and marine biology. Most of such tasks require the use of the ROV's manipulator arms, in most cases in combination with the use of a tool.

Often ROVs are directly teleoperated from an off-shore supporting vessel, something that makes the cost of utilizing an ROV prohibitively high. This cost can be reduced by moving the teleoperation infrastructure to an on-shore facility and communicating with the ROV remotely. Current satellite communications technology suffers from large latencies and deems traditional *direct* ROV teleoperation infeasible.

We are developing a novel teleoperation paradigm within which no direct teleoperation of the ROV's DOFs is required but control is locally handled (onboard) using a probabilistic representation of task/skill primitives. Such a representation can adapt to task variations and is robust against intermittent communication. Within the DexROV project [1], we are investigating an efficient encoding of manipulation tasks that is learned via programming by demonstrations. We *incrementally* build our representation from demonstrated motions as *hidden semi-Markov models* (HSMM), using an online DP-Means algorithm [2], and generate motion plans by stochastically sampling from the learned model (see [3] for the batch version in contrast). The generated motion is tracked by an infinite-horizon linear quadratic regulator (LQR) that yields smooth trajectories with varying stiffness/compliance characteristics learned from the demonstrations. With this approach we are able to add datapoints incrementally, without the need to re-train the model in a batch fashion, and by discarding the demonstration datapoints after observation. We show how such skills can be learned and how this model can decouple the local control from the teleoperation setup. In fact, with our approach only a small set of model parameters needs to be communicated from the operator side to the teleoperated system. This makes the overall method robust to intermittent or failing communication.

II. Approach

We developed a method that leverages the advantages of an online HSMM building algorithm, the DP-means clustering [2], to arrive to a hard clustering approach. A HSMM is incrementally built by incorporating datapoints $\boldsymbol{\xi}_t \in \mathbb{R}^D$, where D is the dimensionality of the problem at hand. For each new datapoint, the squared Euclidean distance to the HSMM cluster centers is calculated. If this distance is greater than a threshold based on the size/range of the motion, a new cluster is added to the HSMM. If the distance is lower, then the cluster components, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ (and cluster prior π_i), are updated according to the MAP estimate described in [4]. This results in a model that is incrementally expanded with more clusters if the need arises, can be incrementally built –for example in a number of demonstrations– and is built online, i.e. no batch processing step is needed, while no data is stored. This makes the approach particularly appealing for incrementally growing sets of demonstrations.

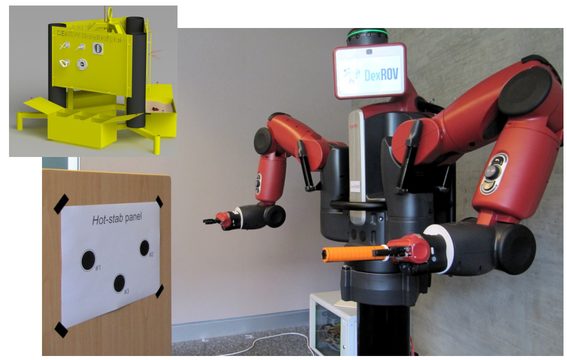

Fig. 1. The Baxter robot being taught how to perform *hot-stabbing* motions. The orange cylinder is a mock-up of the hot-stab plug with three receptacles mock-up shown on the hot stab panel. The inset image shows a rendering of the underwater evaluation panel used in DexROV, housing hot-stabs with different handles and rotational switches.

The parameters of an HSMM are described by $\boldsymbol{\Theta} = \{\{a_{i,j}\}_{j=1}^K, \Pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}}\}_{i=1}^K$. Here, we optimize the parameters *online* based on the DP-means estimate. For each datapoint that is added to the HSMM, we estimate to which Gaussian component it most likely belongs. This way, for each datapoint $\boldsymbol{\xi}_t$ we can estimate the state q_j and the previous state q_i. To build up the transition probabilities, $a_{i,j}$, we keep a matrix $c_{i,j}$, $\boldsymbol{c} \in \mathbb{R}^{K\times K}$, that counts the number of state transitions that are not self-transitory. The initial emission probabilities Π_i are estimated in a similar manner, by keeping track of the starting component of each demonstration sequence. A univariate Gaussian distribution $\mathcal{N}(\mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}})$ is used to model each state duration by keeping statistics over the state transitions and bypassing the computationally expensive batch training procedure. This way, as demonstrations are being performed and the underlying HSMM is being built, we keep track of each state duration and accordingly update the statistics of each state. This is done using a running statistics method to compute the mean and variance for each state duration. This requires that we only keep track of the total number of samples while we incrementally add new values. Consequently, our approach

[1]Idiap Research Institute, Martigny, Switzerland. {ioannis.havoutis, ajay.tanwani, sylvain.calinon}@idiap.ch, [2]Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. This work was in part supported by the DexROV project through the EC Horizon 2020 programme (Grant #635491).

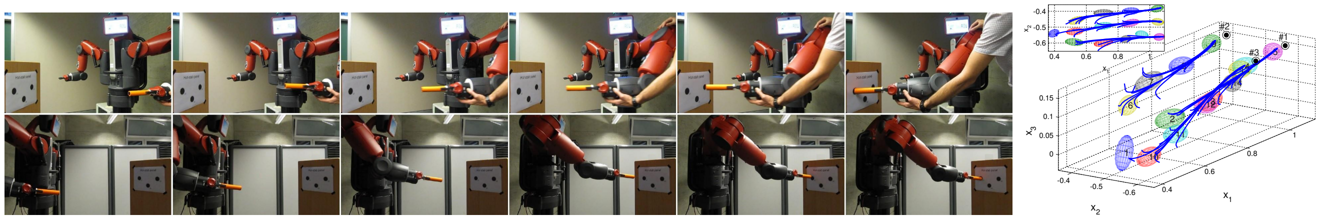

Fig. 2. *Left*: Kinesthetic teaching and motion generation for the hot-stabbing skill. *Top row:* The left arm of Baxter is used to teach the hot-stabbing motion. *Bottom row:* Execution of the hot-stabbing motion, using the learned HSMM model, on the teleoperated side (right arm). *Right*: Evaluation trials, 10 motions per receptacle in blue lines, starting from randomized initial states.

TABLE I

RMSEs of multiple averaged trials.

Receptacle Number	#1	#2	#3
Reproductions, RMSE	0.77cm	0.74cm	0.99cm

does not need to store any datapoint while all learning is performed online.

By stochastically sampling from the HSMM for T time steps, we obtain a sequence of states to be visited, $q_1 \ldots q_T$. This step-wise reference trajectory $\mathcal{N}(\hat{\boldsymbol{\mu}}_{q_t}, \hat{\boldsymbol{\Sigma}}_{q_t})$ can be smoothly tracked using an infinite-horizon linear quadratic regulator with a double integrator system [5].

III. Experimental Setup & Evaluation

We use the two-armed Baxter robot as a working example of a teleoperation system. We control Baxter's arms by compensating the effect of gravity, allowing us to demonstrate motions kinesthetically. We use the left arm as the operator's side and the right arm as the teleoperated end.

To demonstrate our approach, we chose one of the most frequently executed tasks in underwater ROVs. This is the task of inserting a hot-stab plug to a hot-stab receptacle. Hot-stabbing is used to provide hydraulic actuation to most of the tools used in underwater facilities.

The operator kinesthetically demonstrates the hot-stabbing skill. 6 hot-stabbing demonstrations –2 for each goal– are provided, starting from a similar neutral joint angle configuration and reaching three previously defined goal positions on the hot-stab panel mock-up as shown in Fig. 1.

As the demonstrations are performed, the model of the skill is being built. As new demonstrations are made available, the model incrementally grows. The transition probabilities $a_{i,j}$ and the state duration parameters $\mu_i^{\mathcal{D}}$ and $\Sigma_i^{\mathcal{D}}$ are incrementally updated accordingly. In the hot-stabbing experiment, a model with 14 Gaussians was learned. Fig. 2 (*left*) presents the learned model along with 30 generated motions used for evaluation.

The model parameters are then communicated to the reproduction side for motion generation. The subsequent motion control is handled locally on the remote side, while only the model parameters need to be communicated at intermittent time. The remote arm (right) then performs the manipulation motion, while computed varying stiffness and damping profiles of the controller allow the task to be regulated in accordance with the required precision. In essence this allows us to control *lazily* along task directions that do not matter and *accurately* along the important task directions, by following a minimal intervention principle [6].

We simulate a failure in communication as the operator directly teleoperates the remote arm. As soon as the communication fails, our system samples the learned model, starting from the current state, and generates a motion that continues the execution of the hot-stabbing task. Note that the model here is mirrored for the teleoperated side, i.e. the right arm of the robot. Trial snapshots are presented in Fig. 2 (left).

To evaluate the efficiency of the learned model we compare the end position of the generated hot-stabbing motions against the demonstrations. We chose this metric as the position of the hot-stabbing plug at the end of the motion should reach the receptacle entrance –up to some allowed variance– while the path to this state can vary reasonably. Fig. 2 (*right*) shows the 30 hot-stabbing motions, 10 per receptacle target, that were generated by sampling from the learned model, beginning at randomized initial states.

Table I summarises the results of the evaluation trials. The motions that are generated by the learned model accurately position the plug according to the demonstrations. The outcome is highly repeatable, as in practice all motions to a particular receptacle converge to the same end-point at the end of the motion regardless of the starting state. As all reproduction RMSEs are bellow 1cm, we conclude that all 30 trials are successful in hot-plugging to the different receptacle targets.

IV. Conclusion

We presented an online and incrementally learned HSMM representation for encoding manipulation tasks in semi-autonomous underwater teleoperation scenarios. We demonstrated how such representation can be learned from incremental piece-wise demonstrations, without the need to store demonstration data, and how motions can be regenerated from the learned model. We evaluated the performance of our approach with a common ROV task and showed how a learned model can reproduce motions with high accuracy and repeatability.

References

[1] J. Gancet, et. al., "DexROV: Dexterous undersea inspection and maintenance in presence of communication latencies," in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*, 2015,

[2] B. Kulis and M. I. Jordan, "Revisiting k-means: New algorithms via Bayesian nonparametrics," in *Proc. ICML*, 2012,

[3] A. K. Tanwani and S. Calinon, "Learning Robot Manipulation Tasks with Task-Parameterized Semi-Tied Hidden Semi-Markov Model,", in IEEE Letters on Robotics and Automation, pp. 235-242, 2016,

[4] J. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and Audio Processing, IEEE Transactions on*, 1994,

[5] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Proc. ICRA*, 2014,

[6] E. Todorov and M. I. Jordan, "A minimal intervention principle for coordinated movement," in *Advances in NIPS*, 2002,