Online Adaptive Teleoperation via Incremental Intent Modeling

Xuning Yang[1], Koushil Sreenath[1,2], Nathan Michael[1]
[1]The Robotics Institute, [2] Dept. of Mechanical Engineering
Carnegie Mellon University, Pittsburgh, PA, USA
{xuning, koushils, nmichael}@cmu.edu

ABSTRACT

We present a task-independent framework that is amenable to long-duration teleoperation of a mobile robot in an unknown scenario, using incremental intent models. In particular, we focus on constructive teleoperation in performing an unknown task without knowledge of the environment, while allowing user to retain full control over the set of motions. We evaluate our method on a quadrotor flying in a lemniscate maneuver as a proof-of-concept experiment, and show reduced joystick entropy. We conclude with a discussion regarding our on-going investigation with this framework.

Keywords

Teleoperation; Adaptive Teleoperation; User Intent Prediction

1. INTRODUCTION

Teleoperation of mobile robots continues to be a superior choice when operating in an unknown environment, due to hindrances in computational complexity of goal identification, trajectory planning and optimization. While humans are efficient at identifying areas of interest and motion planning, we are imperfect controllers due to inexperience or fatigue. Assistive teleoperation is widely addressed in the context of manipulation [1] and wheelchair assistance [2], which assumes an episodic task structure with a terminating goal, as well as the availability of prior knowledge of the environment and user proficiency. For teleoperation of mobile robots in long-duration tasks, environment is typically not known a priori, and the set of goals could be very large and continuous.

In assistive teleoperation, user inputs are typically arbitrated with a policy that corresponds to some predicted goal or trajectory. These goal hypotheses are typically constructed over a set of discrete objects [6, 7], or a set of known trajectories in a fixed environment [2]. In most scenarios, linear arbitration is most widely used [7, 5, 4], although Bayesian arbitration has also been employed [3]. A key limitation in these approaches are the inherent assumptions of the environment and user. Linear arbitration parameters are typically selected depending on the proficiency of the user, and the set of goal states are reduced down to a computationally tractable size.

This paper extends the work in [12] and provides a more precise prediction by building local models for dynamic adaptation of the user intent over short temporal windows. The following sections describe our framework for a quadrotor aerial vehicle, and discusses the results for a proof-of-concept experiment for a quadrotor completing a lemniscate motion in simulation and in actual flight. Finally, future work regarding more precise and dynamic adaptation and further experimental validations are outlined.

HRI'17 Companion, March 06-09, 2017, Vienna, Austria
ACM ISBN 978-1-4503-4885-0/17/03.
DOI: http://dx.doi.org/10.1145/3029798.3038370

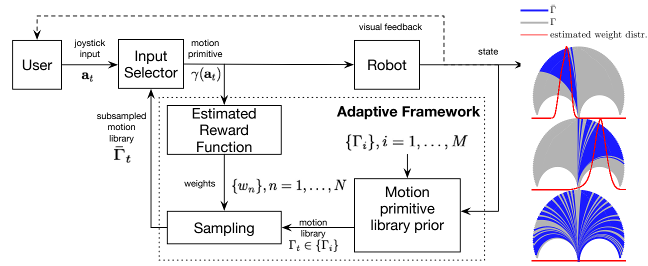

Figure 1: Left: System diagram of the proposed adaptive framework. Right: a graphical representation of the algorithm at time 0, 50, and 120 from bottom to top. The available set of the motion primitive to the user (blue) is sampled based on the inferred distribution (red) from the underlying dense set of motion primitives (grey).

2. PROPOSED FRAMEWORK

The framework consists of two components: First, system-specific action space is abstracted into the state space via a dense set of motion primitives. Then, a belief distribution of the user intent is inferred over the set of motion primitives. In this work, we assume that the operator is a rational agent that acts as an optimizing controller, such that an observable reward function is optimized regarding the system performance. A belief distribution is generated over the set of motions using incremental online regression of a linear reward function. Then, the available set of motions is constructed according to the predicted distribution. By construction, the user retains full control over the motion of the vehicle without arbitration of inputs, while having fine-grained control over the region of interest based on prior inputs. An overview of the system and algorithm is shown in Fig. 1.

2.1 Motion Primitives

We define an *action* to be a set of discrete input values. For q input dimensions, an action is denoted as $\mathbf{a} = \{a_1, \ldots, a_q\}$. A motion primitive $\gamma(\mathbf{a})$ is generated by parameterizing the action $\mathbf{a}$. Set of actions, $\{\mathbf{a}_i\}, i = 1, \ldots, N$, generate a *motion primitive library* (MPL), denoted by $\Gamma = \{\gamma(\mathbf{a}_i)\}, \; i = 1, \ldots, N$. The set of MPL is defined as a *motion primitive library collection*, which is denoted by $\{\Gamma_j\}, j = 1, \ldots, M$.

For the quadrotor aerial vehicle used in this work, the action space is composed of linear velocity, angular velocity, and thrust. Each input dimension is discretized finely to create a sufficiently dense set of actions. The parameterization of these inputs using a unicycle model results in a set of *forward-arc motion primitives* [9]. At each input time, the joystick interface selects a single motion primitive based on the action input via a selector function, which is then sent to the vehicle.

2.2 Intent Identification and Adaptation

Inspired by [8], we assume that the user is optimizing a linear reward function, with a set of qualitatively observable bases that is naturally optimized by the user. It is assumed that at each time instant, the user issues an action $\mathbf{a}$ that is in some neighborhood of

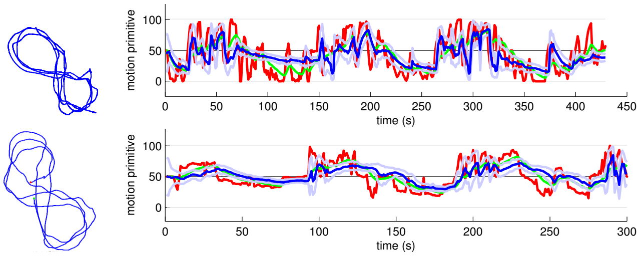

Figure 2: Comparison of simulation (top) vs. flight (bottom) results. Odometry (left) and selected motions (right) for each MPL containing 101 motion primitives. Red: actual input. Green: low-pass filter input. Blue: mean of predicted distribution with light blue highlighting upper and lower bounds based on the covariance.

$\mathbf{a}^*$, the optimal action that maximizes the linear reward function R_t:

$$\mathbf{a}^\star = \underset{\mathbf{a}}{\operatorname{argmax}}\, R_t(\gamma(\mathbf{a})) \approx \underset{\mathbf{a}}{\operatorname{argmax}} \sum_{i}^{P} \alpha_t^i \phi_t^i(\gamma(\mathbf{a})) \qquad (1)$$

Three *hindsight bases* $\phi^i, i = 1, 2, 3$, were chosen to model the reward function. These bases are purely based on user inputs and are evaluated using a window of past m motion primitives. These are: *smoothness*, evaluated as the sum of errors between the successive inputs; *time*, which is the inverse of the average linear velocity; and *orthogonality*, which penalizes the drastic deviation of γ_{t+1} from a smooth trajectory from time $t-m$ to t, denoted by $\gamma_{t-m:t}$.

We employ Locally Weighted Projection Regression (LWPR) [11] to estimate the reward function $R(\gamma_{t-m:t}, \gamma_{t+1})$. LWPR is a computationally efficient online method for approximating high dimensional nonlinear functions. However, it's key for computational efficiency is that it stores only the sufficient statistics in making incremental updates to the model. While older data can be neglected using a forgetting factor that decays the sufficient statistic over time for a single model, we find that a more precise control over the specific temporal window of data is required. To leverage the speed of LWPR (O(n)), we keep a queue of k LWPR models. At each input time, a model $\mathcal{M}_t$ is popped off the queue, and a new model $\mathcal{M}_{t+k}$ is added. Each model in the queue is updated with incoming data. The prediction $p(\hat{R}_t|\gamma_{t-m:t}, \gamma_{t+1})$ is then generated with model $\mathcal{M}_t$.

The belief distribution over the set of the motion primitive is constructed as follows. For every motion primitive at time $t+1$, the following is computed:

$$\begin{aligned} p(\gamma_{t+1}|\gamma_{t-m:t}, \hat{R}_t) &= \frac{p(\hat{R}_t|\gamma_{t-m:t}, \gamma_{t+1})p(\gamma_{t+1}|\gamma_{t-m:t})}{p(\hat{R}_t|\gamma_{t-m:t})} \\ &= \eta\, p(\hat{R}_t|\gamma_{t-m:t}, \gamma_{t+1})p(\gamma_{t+1}|\gamma_{t-m:t}) \end{aligned} \qquad (2)$$

where $p(\hat{R}|\gamma_{t-m:t}, \gamma_{t+1})$ is a distribution over the estimated reward function of the user, and η is a normalization weight.

The set of the available motion primitive is updated iteratively according to Eq. 2. At each time instance, we modify the set of available motion primitives by sampling the set of dense motion primitives via importance sampling with replacement according to a weight distribution of $w_n = p(\hat{R}|\gamma_{t-m:t}, \gamma_{t+1_n})$ for $n = 1, ..., N$ motion primitives.

3. EXPERIMENT

We test the proposed framework with a teleoperated quadrotor performing a lemniscate approximately 5m in length. A single operator is asked to perform 10 trials in simulation and flight experiments using apparatus shown in Fig. 3. We use an entropy-based measure, Joystick Steering Entropy (JSE) [10], to evaluate our method. In addition, we compare our result to a finite impulse low-pass filter on the raw inputs with a weighting factor of 0.9. In this experiment, we infer over the angular velocity component and provide no adaptation for linear velocity and thrust for clarity. A queue of $k = 10$ LWPR models is used to construct the belief distribution, which worked well for this maneuver.

Figure 3: The quadrotor and joystick used in the experiment and the quadrotor in flight.

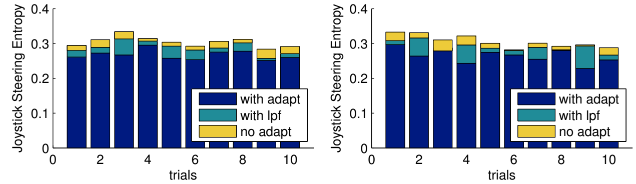

Figure 4: Joystick steering entropy with adaptation, without adaptation and with a low-pass filter for 10 trials in simulation (left) and in actual flight (right). Trials with adaptation performs comparatively with a low-pass filter approach. Lower entropy indicate smoother inputs, thus better.

Three trials of the simulation results are shown in Fig. 2. The graphs show the progression in the joystick inputs over time. The actual joystick input (red) is compared to the resulting mean of the prediction (blue) with the covariance (light blue). Smoother inputs are observed with adaptation overall, which resulted in final trajectories (left), performing comparatively with the low-pass filter (green). JSE comparisons are shown in Fig. 4, which indicate lower entropy with adaptation than without. We observe that our method is robust as compared to a low pass filter, but could improve in performance by increasing the precision in prediction.

4. CONCLUSION

In this work, we employed a queued LWPR approach in order to accommodate slight changes in the intent model, and demonstrated first results in a lemniscate motion in simulation and flight. We aim to further demonstrate the efficacy of this method with field testing of real applications such as exploration, with adaptation introduced to all multiple input dimensions. Furthermore, we aim to investigate how our framework performs with more aggressive maneuvers, which require a precise method to address changes in intent. To this end, our next steps include investigating computationally efficient and precise online regressors for dynamic modeling of intent, given limited data sets.

5. REFERENCES

[1] A. D. Dragan and S. S. Srinivasa. A policy-blending formalism for shared control. *Int. J. Robot. Res.*, 32(7):790–805, 2013.

[2] E. Demeester et al. User-Adapted Plan Recognition and User-Adapted Shared Control: A Bayesian Approach to Semi-Autonomous Wheelchair Driving. In *Autonomous Robots*, pages 193–211, 2008.

[3] E. Demeester et al. ML, MAP and Greedy POMDP Shared Control: Comparison of Wheelchair Navigation Assistance for Switch Interfaces. In *Proc. of the Int. Sym. on Robotics*, Taipei, Taiwan, Aug. 2012.

[4] A. Goil, M. Derry, and B. D. Argall. Using Machine Learning to Blend Human and Robot Controls for Assisted Wheelchair Navigation. In *IEEE Int. Conf. Rehabil. Robot*, Seattle, Washington, USA, June 2013.

[5] S. Jain and B. Argall. An Approach for Online User Customization of Shared Autonomy for Intelligent Assistive Devices. In *Proc. of the IEEE Int. Conf. on Robot. and Autom.*, Stockholm, Sweden, May 2016.

[6] S. Javdani, S. S. Srinivasa, and J. A. Bagnell. Shared Autonomy via Hindsight Optimization. In *Proc. of Robot.: Sci. and Syst.*, Rome, Italy, July 2015.

[7] K. Muelling et al. Autonomy Infused Teleoperation with Application to BCI Manipulation. In *Proc. of Robot.: Sci. and Syst.*, Ann Arbor, MI, Jan. 2015.

[8] K. Mombaur and A. Truong. From human to humanoid locomotion – an inverse optimal control approach. *Autonomous Robots*, pages 369–383, 2010.

[9] E. A. Nelson and N. Michael. Environment Model Adaptation for Autonomous Exploration. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, 2015.

[10] O. Nakayama, et al. SAE Technical Development of a Steering Entropy Method for Evaluating Driver Workload. Technical Report 724, 1999.

[11] S. Vijayakumar, A. D'Souza, and S. Schaal. Incremental Online Learning in High Dimensions. *Neural Computation*, 17(12):2602–2634, 2005.

[12] X. Yang, K. Sreenath, and N. Michael. A framework for efficient teleoperation via online adaptation. In *Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017. [Submitted].