Learning Quadrotor Dynamics with Limited State Variables via Simulation as a Basis for Experiment

Nathan Lambert, Saavan Patel, Vikram Sreekanti; University of California, Berkeley

***Abstract*—This project will explore the capabilities of learning model-based dynamics in robotics without highly accurate state variables provided from systems such as VICON. Current work in the area feeds all the state variables into the model to train the system and execute tasks. This project will explore the minimum quality and quantity of state variables needed to perform basic tasks. The lower bound on state information will provide insight into the sensors needed for training novel robotics platforms with potentially unmodeled dynamics to learn tasks. The project will begin with a quadrotor simulation environment to provide the foundation before potentially moving onto experiments with a quadrotor testbed.**

I. Introduction

Reinforcement learning (RL) conveys the opportunity for new agents to accomplish tasks in unknown environments without pre-described instructions, which has clear value to numerous applications. RL is often broken down into two categories: model-free learning, where the system is given goals and attempts to learn a policy via a reward function directly and model-based learning, where a system attempts to learn a dynamics model for future use. Current state of the art algorithms in model-free RL, such as OpenAI's PPO [1], often focus on digital environments where the cost of obtaining data is low, but are still encouraging to future prospects. Currently, a branch of this research focuses on methods to learn dynamics and abilities to complete tasks on low amounts of training data, which is needed when the cost of each experiment or test is high. Deisenroth and Rasmussen showed the feasibility of learning simple physical tasks rapidly with PILCO [2]. Other work in this area involves building on learned dynamics with effective control, which commonly is in the form of MPC [3] or a learned policy.

Building on PILCO and algorthms for low sample dynamics learning, numerous applications in exploring new platforms for these algorithms exists, such as gait discovery on quadrupedals [4], [5]. Work in quadrotors is of interest to see how new algorithms can perform on tasks that have been extensively studied in control. Bansal et al. shows the capabilities of learning generalized trajectories on quadrotors without explicitly training on trajectories of the same complexity [6]. Online Bayesian optimization can be used to learn dynamics in real time that give best control performance, proposed as aDOBO [7].

N. Lambert works with the Department of Electrical Engineering and Computer Science, University of California, Berkeley CA, 94720 USA email: nol@berkeley.edu.

II. Proposed Plan

The majority of our work for this project will be in the simulation phase, and if the time permits, it will continue into experimental analysis on a quadrotor testbed. we will begin by analyzing the convergence of dynamics models for simulated flight in a 12-dimensional python dynamics simulator. we chose python because of the existence of the dynamics model and the ease of integrating with existing machine learning solutions. Obviously, the convergence of data in simulation will be much higher than in experiment, but this simulation phase conveys many benefits on the course project side of research. It will allow me easier implementation of these algorithms and to test more configurations. This understanding will be crucial when continuing onto experiment, and doubly helpful when we move on from the quadrotor to training control of novel robot platforms.

See below for a formulation of our plans for quadrotor dynamics model and controller design. we will explore the convergence of various simple neural network architectures and their accuracy as estimating dynamics. The controllers will work with the learned dynamics model to minimize a cost function.

A. Dynamics Model

We will formulate the problem as fitting a discrete time, potentially non-linear, state update function f

$$s_{t+1} = f(s_t, u_t) \tag{1}$$

Here, the state at time t is $s_t \in \mathbb{R}^d$, where d is a design parameter, the dimension of the state space, and $u_t \in \mathbb{R}^l$ being the inputs, which will depend on the robot and it's control scheme.

Here, the quadrotor dynamics model will begin as the following 12-dimensional state vector:

$$s := \begin{bmatrix} p \\ v \\ \zeta \\ \omega \end{bmatrix} \tag{2}$$

Where $p := (x, y, z)$ is the inertial reference position, $v := (\dot{x}, \dot{y}, \dot{z})$ is the inertial frame velocity vector, $\zeta := (\phi, \theta, \psi)$ is attitude Euler angle vector, and $\omega := (\omega_x, \omega_y, \omega_z)$ is the body frame rotational velocity vector.

B. Model-Based Learning

The premise of this project is to train different f_i to fit the dynamics while trying to achieve the control discussed below. The indicator i will be connected with a vector $c_i \in \mathbb{R}^{12}$ which will be a vector of 1s and 0s indicating if that state variable will be included in the quadrotor's readout. The second knob to turn, which has very real experimental implications, is the change the update rate of all or some of the state variables. This could involve having two decoupled dynamics models for different dimensions, as done in [6], or repassing state variables that are not updated yet into the dynamics model.

C. Control

There are multiple directions of control to explore in this project. The standard, while computationally intensive, is model based control (MPC) based on the learned dynamics model. This works by looking through a range of random actions in the learned dynamics state, and taking the action that minimizes the desired cost function. Alternatively, we can employ a second round of learning to generate a policy for minimizing a cost function.

Initially, our cost function for control and goals will be centered around hovering an ionocraft. Without loss of generality, if the quadrotor is hovering around the origin, a quadratic cost function would be:

$$J(s_t, u_t) = \|p_t\|_2^2 \tag{3}$$

This would penalize the distance from the origin. we will explore future costs and trajectories, but we will begin with simple analysis.

D. Connection with other course, research

Nathan will also be using this project for a course project in the courses EECS289a: Introduction to Machine Learning with Professors Jennifer Listgarden and Anant Sahai *and* EE291E: Hybrid Systems and Intelligent Control with Professor Claire Tomlin. The learning of dynamics models clearly falls into the scope of EECS289a, and the quadrotor control and exploration overlaps with current work in Professor Tomlin's Hybrid Systems Laboratory (HSL). For later stages in the project, help of other graduate students in HSL would be extremely beneficial when testing crazyflie dynamics. This project is an open research question and will continue into future work and hopefully publication(s).

To be completed in project.

To be completed in project.

III. Conclusion

The long term goal of this project is to learn the requisite state variables needed to have simple robots learn their dynamics and how to act in a new environment. we work with Professor Kris Pister to control new micro-robots. The work with quadrotors will build the foundation for first training hovered flight of the ionocraft [8], [9], and then the framework can be used to investigate other robots.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[2] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, 2011, pp. 465–472.

[3] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1714–1721.

[4] A. Nagabandi, G. Yang, T. Asmar, G. Kahn, S. Levine, and R. S. Fearing, "Neural network dynamics models for control of under-actuated legged millirobots," *CoRR*, vol. abs/1711.05253, 2017. [Online]. Available: http://arxiv.org/abs/1711.05253

[5] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning," *ArXiv e-prints*, Aug. 2017.

[6] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 4653–4660.

[7] S. Bansal, R. Calandra, T. Xiao, S. Levine, and C. J. Tomiin, "Goal-driven dynamics learning via bayesian optimization," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 5168–5173.

[8] D. Drew, D. S. Contreras, and K. S. Pister, "First thrust from a micro-fabricated atmospheric ion engine," in *Micro Electro Mechanical Systems (MEMS), 2017 IEEE 30th International Conference on*. IEEE, 2017, pp. 346–349.

[9] D. S. Drew and K. S. Pister, "First takeoff of a flying microrobot with no moving parts," in *Manipulation, Automation and Robotics at Small Scales (MARSS), 2017 International Conference on*. IEEE, 2017, pp. 1–5.