ECE276C Assignment 4: State-of-Art baseline

Dimitri Schreiber , Anthony Simeonov (A12356253), Daniel Shak (A11330894)

June 5, 2018

Project Abstract

Submission Notes

We are collectively using Anthony's late days (7) and Dimitri's late days (7) for a total of 4 days past the original submission deadline.

Problem Statement

Control policies developed in simulation often fail or perform unsatisfactory when transitioned to the real-world. This is often caused by a mismatch between real-world dynamics and the simulated model.

Standard areas of mismatch for non-modeled dynamics include:

1. Inaccurate Model
    - Nonlinearities (friction, backlash)
    - Torque crosstalk
    - Incorrect system characterization
    - Sensor noise/drift
2. Uncontrolled disturbances
    - Collisions
    - Arm stiffness
3. Nonstationary dynamics
    - Wear and tear
    - Temperature

Our goal is to compensate for this mismatch. Common methods of compensation include:

- System Identification
- Finetuning on real world
- Dynamics Randomization
- Ensemble methods
- Data-driven compensator
- Hand tuning

These methods have different limitations including limited expressiveness due to their rigid model structure, high data or computation cost, significant engineer time, and/or poor performance.

Project Idea

We propose an ensemble of data-driven low dimensionality compensator to provide local improvements to compensate for model-controller mismatch. We will train an ensemble of MLP compensators while enforcing specialization within individual netowrks. This may be achieved through heirarchichal training or through the use of a discriminator.

Ideally, this method obtains low data usage, is model agnostic, and is interpretable. We will attempt to further extend this to perform automatic "mode switching" and tracking to compensate for nonstationary dynamics.

What metrics will you be using to evaluate improvement

We will test on three standard environments later in development: InvertedPendulum, HalfCheetah, and Ant. For both of these environments we will use the standard reward signals, and compensator training and evaluation will be applied to these environments with their dynamics modified. Improvement will be relative to the sample efficiency and final reward performance obtained with our method as opposed to our baseline state-of-art.

What is a closest state-of-art algorithm

We consider the recent ICRA 2018 paper "Model-plant Mismatch Compensation Using Reinforcement Learning", by Ivan Koryakovskiy et al. as the closest state of the art algorithm. Their approach involves learning an RL policy on top of a nominal controller (in their case a model predictive controller) to output a control signal which is summed with the output of the MPC to produce the net control input that is applied to the system. The policy learns to take actions which compensate for unmodeled dynamics that the MPC "doesn't know about". This allows the real system to perform well without having to redesign the entire control system from scratch, or modify any of the closed loop behavior of the black box control system.

They train their policy with the deterministic policy gradient algorithm and apply the methods in simulation and on a real robot designed to perform a repeated squatting task. Thus our baseline to implement will be developing a nominal control system on some idealized system/environment, modifying some aspects of the environment such that the nominal controller performs poorly, and then learn a single policy which acts as a compensator on the modified environment to gain back the good performance. Our proposed methods will extend upon this work by applying a different RL algorithm which incorporates neural networks, the ensemble of multiple compensation policies with it's accompanying novel training pro-

cess, and the application of RL policy compensators to more challenging environments (Half Cheeath, Ant in MuJoCo).

Point us to the repo online

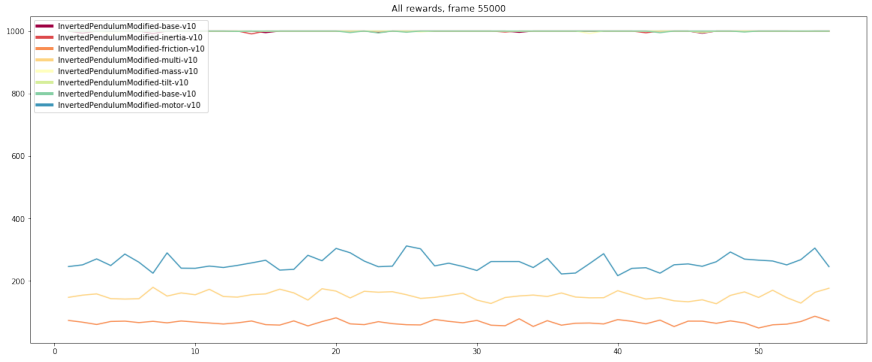
https://github.com/anthonysimeonov/276C-final
https://github.com/higgsfield/RL-Adventure-2 (for PPO implementation)

Baseline Results

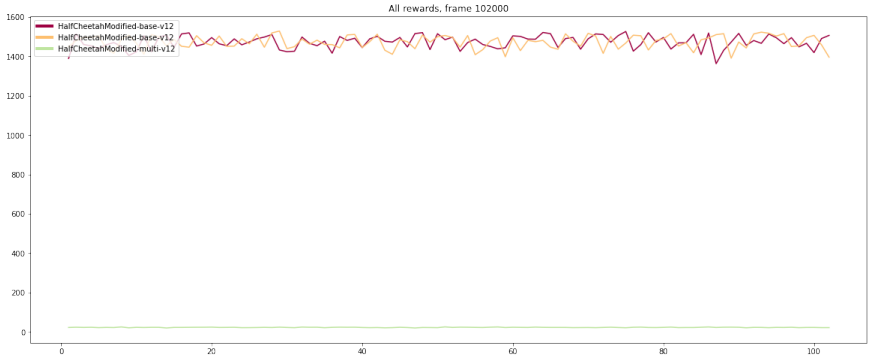
We demonstrate the process of simulation to real world controller transfer. A nominal controller is developed which performs ideally in simulation. We note that this is a black-box controller. In this baseline we use a trained reinforcement learning policy using Proximal Policy Optimization as the nominal controller, but in principal this can be any controller.

The transfer from simulation to real world is demonstrated by applying the same nominal controller to the same environment but where various properties of the dynamics have been modified. These modifications represent realistic physical system characteristics such as friction is various joints, actuator torque mismatch, geometric misalignment, and differences in mass and inertial distribution. The results (see Figure 1) show that a controller developed where these effects don't precisely match will likely perform poorly when deployed on the real physical system. **(*All plots are with reward on the y-axis and 1000 time steps on the x-axis*)**

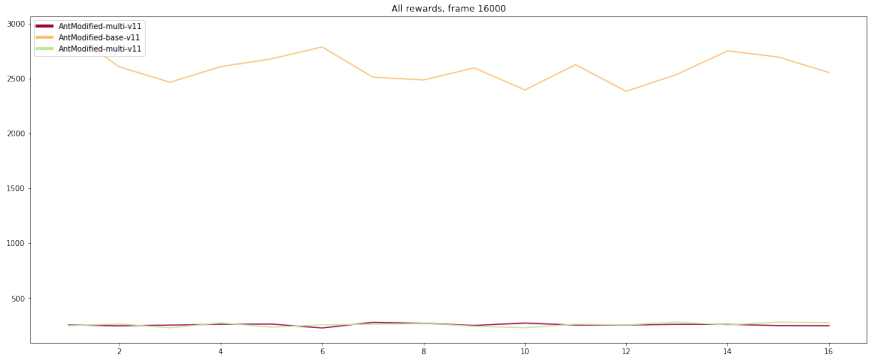
Our baseline for comparison is a single RL compensator policy which modifies the control input that is applied to the system in order to improve it's performance. We note that this leaves the nominal black-box control system unchanged, thus preserving any inherent properties (closed loop stability, etc.) that could be lost by modifying the controller itself. See Figure 2 for the training results of learning a single policy compensator on our evaluation environments. **(*All plots are with reward on the y-axis and 1000 time steps on the x-axis*)**



(a)

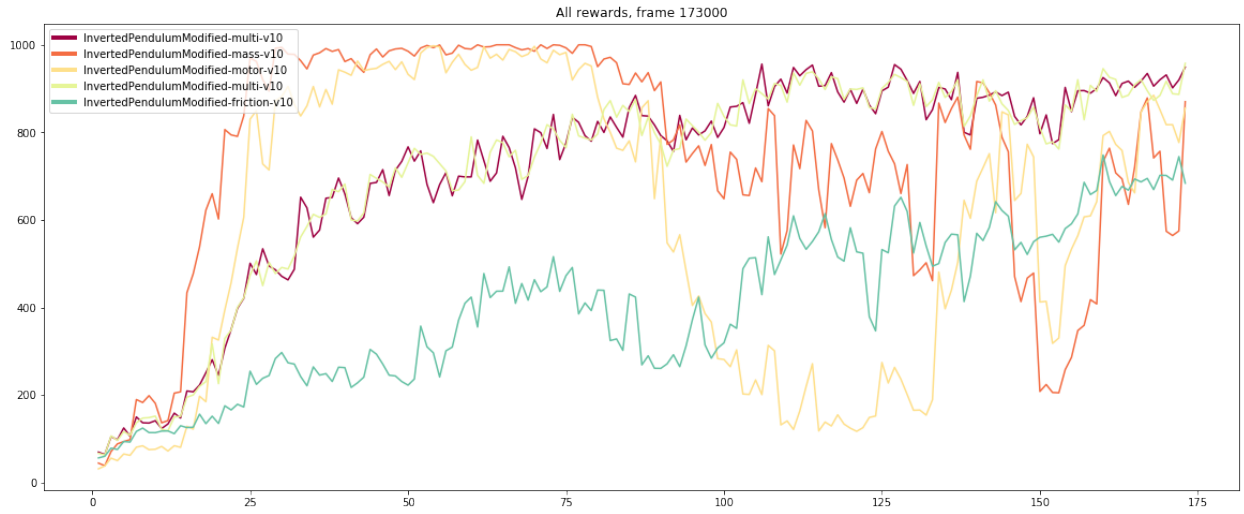


(b)

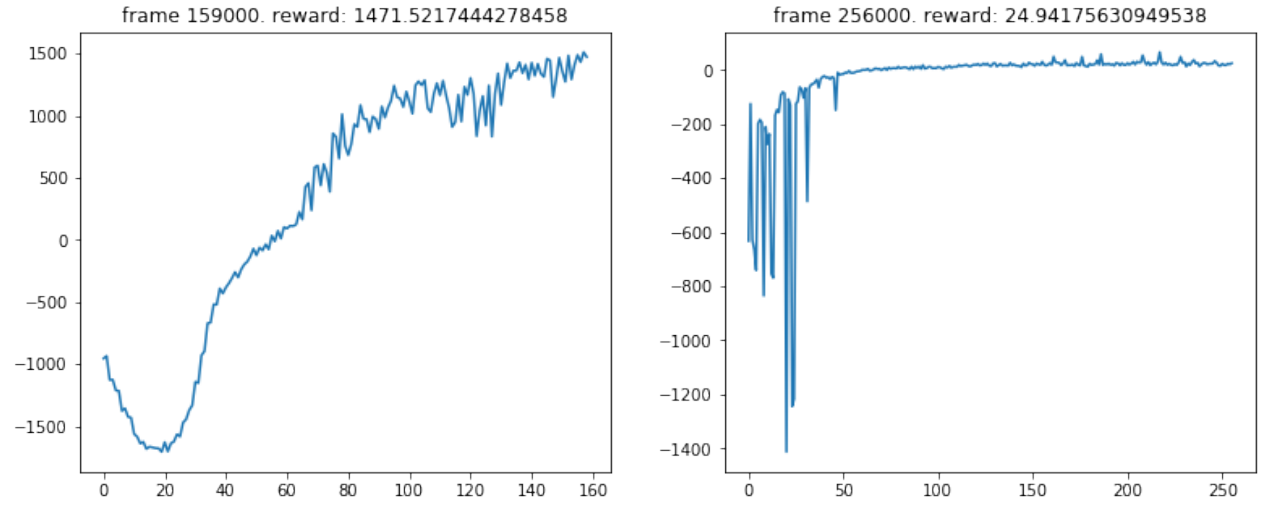


(c)

Figure 1: Nominal controller performance on unmodified (base) and modified (a) inverted pendulum (b) half cheetah (c) ant MuJoCo environments, respectively. The controller performs well on the unmodified env and some of the modified ones. However, there are several environments where the performance suffers significantly due to the different dynamics between the environment the controller was developed/trained on and deployed on.



(a)



(b) (c)

Figure 2: Single RL agent compensator trained using PPO algorithm in MuJoCo (a) Inverted Pendulum (b) Half Cheetah and (c) Ant. Policy is learned which takes actions that are summed with control inputs given by the nominal controller, such that performance is once again optimal on an environment that is similar but with important differences in the dynamics. We note that on the inverted pendulum experiment where the compensator was trained on a single modified environment with multiple modifications, but tested on environments with individual modifications, there are performance tradeoffs on the single-mod environments throughout training. This indicates the potential usefulness of incorporating multiple specialized compensators that are decoupled and don't have to trade off. We also note that this baseline solved the modified Inverted Pendulum, partially solved the Half Cheetah by cheating (flipping on back and vibrating forward) and took too long to train thus not solving Ant.