Perturbed policy training for Probabilistic-Movement Primitives

Theodore Morales, Alejandro Balderas, Jeff Gordon, Aditi Jain

Abstract*— *We attempt to develop a methodology for the creation of probabilistic movement primitives to be applied on a rethink robotics sawyer robot arm. These Pro-MPs are created through a process by which we use Deep Deterministic Policy Gradients to choose trajectories that are successful in completing their given task. We employ the technique of environment perturbation to develop a Probabilistic Movement Primitive that is robust enough to perform well in any given environment. The use of this perturbation is primarily to address the 'reality gap' that causes policies that have been learned in simulation, to not perform as predicted when deployed in reality.

I. Introduction

Creating effective policies for the movement of an agent utilizing probabilistic-movement primitives requires a very large amount of training. When this training takes place using a single robot and a single dimension of time, the feasibility of developing robot agents that can interact with dynamic and unseen environments without needing long periods of time to train before being usable diminishes. To address this, researchers have employed physics simulators to conduct their training in. Training in simulation over reality is advantageous due to the ability to perform training using hardware models of which you have no physical access. the ability to train multiple models in parallel, and the ability to perform training at speeds far faster than would be possible during training in reality. While this seems like a promising solution to the issue of the expensive task of training in reality, it has not come without issue of its own. Researchers employing these techniques have run into a complication called the 'reality gap'.

The reality gap describes the event in which a policy for robotic movement has been developed and optimized during simulation in a physics engine and is then deployed in reality on a physical robot and fails to perform as expected. This complication is the issue of which we seek to address through our research. If we wish to reach a point where we can efficiently and quickly build robots and deploy them to perform tasks to aid humans in homes and workplaces, it is imperative that we design methods of policy training that are as inexpensive and productive as possible. Our answer to this is the use of Deep Deterministic Policy Gradients to develop Probabilistic Movement Primitives that are trained in a multitude of distinct and unique environments with the hope that the resulting Pro-MPs are robust and general enough to perform well in reality even though we may not be directly training our policy over the environment variable values present in reality.

II. Method

A. Probabilistic-Movement Primitives

When equipping a robotic agent with the ability to perform complex motor-skill based tasks, it is imperative that the agent has the ability to complete the task in the presence of environmental interference. If one were to instruct an agent to move along a single trajectory, they would soon be met with the issues of:

1. Joint wear and tear causing differences in resulting movement when force is applied to the joints.

2. Environmental interference causing an agent that is knocked off of its trajectory to lack the ability to recognize how far away it is from the goal and take the necessary movements to redirect itself onto the correct path.

3. Introducing objects that obstruct the trajectory would cause the robot to either collide with the obstruction or not take the necessary steps to avoid the obstruction and fail to complete its task.

In order to equip an agent with the ability to withstand interference during the completion of its assigned task, a Probabilistic-Movement Primitive is constructed. Probabilistic-Movement Primitives provide agents with a distribution of trajectories that have been previously identified as satisfying the criteria of the task given to the agent. A Pro-MP is assumed to have a Gaussian Distribution with the mean of the distribution being the trajectory that is initially followed by an agent. When an agent is pushed off the course of the mean trajectory of the distribution, it applies force to counter the environmental force such that a smaller amount of counter force is applied when the agent's trajectory is a substantial distance from the bounds of the distribution and a larger amount of counter force is applied when the agent's trajectory is closer to the bounds of the distribution. The aforementioned environmental force that is applied to the robotic agent to knock it off of its intended trajectory may be present in the form of an outside physical force but also the miscalculated resulting trajectory that comes from applying force to joints

that do not have the same amount of resistance as the joints that the Probabilistic-Movement Primitive was designed to function on. This introduction of a distribution of possible trajectories that an agent may employ to achieve it's given task widely increases the adaptability of the agent.

B. Deep-Deterministic Policy Gradient

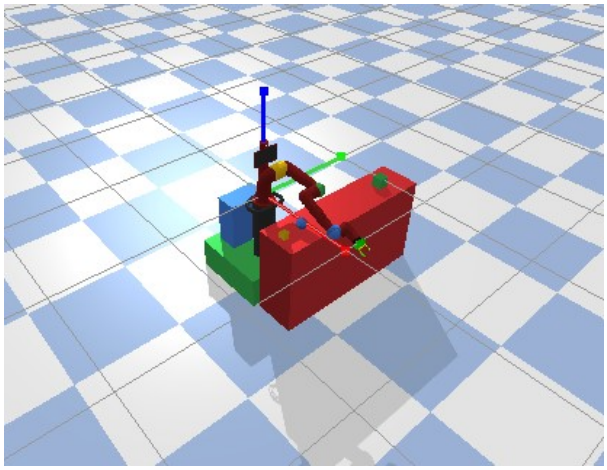
A key feature that contributes to the effectiveness of a Probabilistic-Movement Primitive is that the distribution contains trajectories that satisfy the criteria for a successful execution of a task. In order to identify successful trajectories in dynamic robot movement tasks, one could obviously observe each trajectory and identify whether each independent trajectory was satisfactory in completing the task. This would result in an incredibly time-consuming endeavor considering that Probabilistic-Movement Primitives require a very large set of successful trajectories in the construction of its distributions. An alternative to this method of individually identifying successful distributions is to use reinforcement-learning, specifically Deep-Deterministic Policy Gradient or DDPG as it will be referred to from this point forward.

A DDPG algorithm can be thought of as a form of Deep Q-learning that is engineered for use on continuous action spaces. Q-learning operates such that in a given state, after learning an optimal action-value function $Q^*(s, a)$ that yields a Q-value proportionate to the attractiveness of an action *a* in a given state *s*, an agent is then instructed to choose the action yielding the maximum Q-value at any given state, leading to a successful completion of a task in a discretized task space with accurate action-value functions. The issue with using vanilla Q-learning in a continuous task space arises in the computation of determining the optimum action from a given state. Attempting to compute the maximum Q-value over a continuous action space is an unfeasible and expensive routine that must be invoked every time an agent would like to make an action. In order to sidestep this expensive computation, DDPG makes use of a gradient based learning policy $\mu(s)$ that exploits the differentiability of the action-value function with respect to *a*. This learning policy is then used in estimating the Q-value, $maxQ(s, a) \approx Q(s, \mu(s))$. DDPG contains two neural networks, an actor and critic network. The actor network takes the current state of the agent as input and outputs the agents best choice action given the policy. The critic network takes the actors action and resultant state to compute a *Temporal Difference* error signal which is used by both the critic and actor network to derive gradients that are used to update the critic network and in turn the policy by which the actor critic operates.

C. Our Experiment

To tackle the reality gap, we aim to generate Probabilistic-Movement Primitives for unique environments through perturbation of environment variables and policy training using DDPG. This will result in a collection of DDPG generated trajectories for each environment encountered in simulation by the agent. We will then, to varying degrees, generalize and combine Pro-MPs for similar environments with the intent of creating Probabilistic-Movement Primitives that are robust enough to allow an agent to complete a task for an unseen environment due to the inclusion of multiple perturbed environment trajectories that may be similar to the new environment in the distribution.

We decided that in order to accurately measure the success of our experiment, which aims to address the reality gap issue by creating probabilistic-movement primitives that are robust enough to enable successful task completion in unseen environments, it was imperative to use a simple experiment. Therefore, our experiment consists of the agent being a Rethink Robotics Sawyer whose task is to, with a cube gripped to function as a club, employ a trajectory that uses the club to hit a sphere into another cube that will function as the goal or hole.



We conduct this experiment in the PyBullet physics engine. We chose this physics engine because it allows us to perform our environment perturbation in the most efficient manner when compared to other physics engines. The following are the environment variables we chose to perturb.

- Joint Dampening
- Friction between ball and table
- Gravity

The perturbation is done by selecting random values for the three variables within a reasonable range.

After a random environment has been selected, we use DDPG to derive optimal 'swings' or trajectories for the robot to employ with a positive reward achieved when the sphere hit by the agents club collides with the goal. An increasingly negative reward is then achieved as the sphere that the agent hits gets further away from the goal.

Next, using the policy learned during DDPG training, we simulate and record trajectories that result in a positive reward or success, defined by a collision of the ball with the goal. These trajectories are then used to form the distribution of

Identify applicable sponsor/s here. *(sponsors)*

a Probabilistic-Movement Primitive that is mapped to the environment. After creating Pro-MPs for each environment, we plan to create separate more general Pro-MPs for environments that are similar to each other in terms of values taken on for the perturbed variables.

III. Related Work

In the paper Training Deep Networks on Domain Randomized Synthetic X-ray Data for Cardiac Interventions, researchers, conducted experiments that measured performance of 3D/2D cardiac model-to-X-ray registration of a convolutional neural network artifical agent. Their results showed that Domain Randomization (perturbation of environment variables) resulted in significantly more consistent transfer from simulation to reality. [1] This work showed quantitative results on image processing via Convolutional Neural Networks, we hope to identify the same benefit of Perturbed training with robotic movement.

In the paper Asymmetric Actor Critic for Image-Based Robot Learning, researchers employ a method similar to ours in their use of actor-critic networks where the actor receives only RGBD images of the simulated environment. The researchers combined domain randomization with the actor-critic network and produced a simulation to real world transfer without training on any real world data. [2] The perturbation in this experiment was performed on the RGBD values as opposed to physical environment variables that we chose in our experiment.

The paper we used as the main source of information for Probabilistic-Movement Primitives was Using Probabilistic Movement Primitives in Robotics[3]. This article was introduced to us by our research adviser and presented a methodology by which to create Probabilistic-Movement Primitive for use in robotics.

As a source of information on Deep-Deterministic Policy Gradients we used a blog called Deep-Deterministic Policy Gradients in Tensorflow[4]. This source provided us with information on actor-critic model-free algorithms along with an example of a DDPG algorithm using the pendulum environment on OpenAI gym.

IV. Evaluation

While we did not complete our experiment, we plan to continue this project over the next semester and propose a means of measurement in which we train two agents, a control trained on a static environment and an agent which is trained over a variety of perturbed environments. We will develop our Pro-MPs using only synthetic data and after completion of training will employ them on sawyers in a physical reality with an environment as similar as possible to the environment that was presented as a static environment to the control agent. We will then measure efficiency of both agents with the hypothesis that the agent trained over a perturbed environment will perform more consistently.

V. Conclusion

In this article, we proposed a methodology for constructing Probabilistic-Movement Primitives using Deep-Deterministic Policy Gradients to produce the trajectories that are present in the distribution of the Pro-MP. We directly address the reality gap by introducing training data that is created in perturbed environments in which we change the variables of the environment in hopes of creating a model robust enough to encounter a new environment in reality and see it as merely a variation of the environments of which it was trained under.

References

[1] *Vercauteren, Tom, et al. "CAI4CAI: The Rise of Contextual Artificial Intelligence in Computer-Assisted Interventions." Proceedings of the IEEE (2019)*

[2] Pinto, Lerrel, et al. "Asymmetric actor critic for image-based robot learning." arXiv preprint arXiv:1710.06542 (2017).

[3] Paraschos, Alexandros, et al. “Using Probabilistic Movement Primitives in Robotics.” *Autonomous Robots*, vol. 42, no. 3, 2017, pp. 529–551., doi:10.1007/s10514-017-9648-7.

[4] “Deep Deterministic Policy Gradients in TensorFlow.” *Deep Deterministic Policy Gradients in TensorFlow*, 21 Aug. 2016, https://pemami4911.github.io/blog/2016/08/21/ddpg-rl.html.