Robot Module 4: Discrete-Action and Continuous-Action Actor-Critic Reinforcement Learning

Amir Samani*

***Abstract*— In Robot Module 1 [1], we built the basic tools required for real life system to test various applications of reinforcement learning (RL). In Robot Module 2 [2], we built General Value Functions (GVFs) on top of the experimental setup from Robot Module 1 and learned two on-policy predictions and one off-policy. Then, in Robot Module 3 [3], we extend upon Robot Module 2 and implemented the Horde architecture and answered several on-policy and off-policy GVFs in parallel. However, in Robot Module 4, we shift our focus from prediction to control. The goal of this module is to use actor-critic methods to select actions for the robot to maximize the expected sum of rewards.**

I. Introduction

In Robot Module 4, we try to implement actor-critic algorithm for both discrete-action-space and continuous-action-space for controlling the angular position of a servo motor to be at a specific position using a predefined reward signal (function). The outline for the rest of write-up is in the following. In Section II, we discuss experimental setup. Section III is dedicated to our discrete-action-space control experiment in which the actions are selected from a discrete set of possible actions. In Section IV, we design a similar experiment to the discrete-action-space control, while the actions are then continuous. Finally, Section V summarizes the write-up and point out the most noticeable conclusions.

II. Robot Setup

In this module we use the same robot setup as Robot Module 1. The final robot setup is illustrated in Figure 1. For the purpose of simplicity, all the experiments in this work only uses one of the servo motors. All of the codes for these experiments in the GitHub Repository of Robot Module 4 [1].

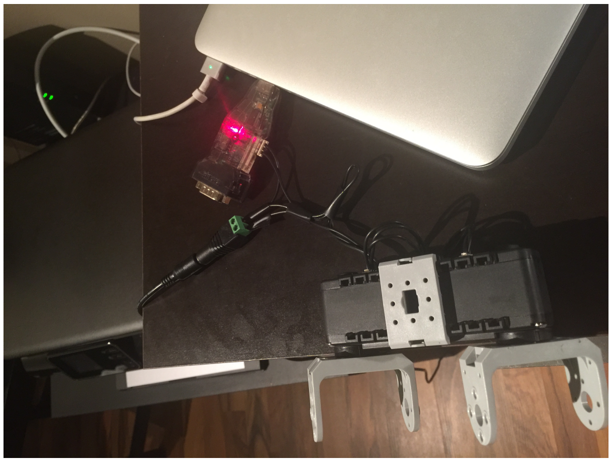

Fig. 1. Robot setup for the experiments.

III. Discrete-Action-Space Control Experiment

In this section, we design a simple control experiment for a servo motor. The goal is to set the angular position of the servo to 0 (angular position). The only possible actions are adding 0.1 to the current angular position or subtracting 0.1 from the current angular position. It is important to note that we design the robot to have -1.5 and +1.5 as the angular boundaries. It means that if the control algorithm tries to move beyond these boundaries (less than angular position -1.5 or more than angular position 1.5), it would not be possible. The actor-critic we are using is ”Actor-Critic with Eligibility Traces (continuing)” from the RL textbook [4]. Since we have a discrete-action-space, we consider policy parameterizations using the softmax in action preferences with linear action preferences. For the state representation, we use state aggregation (using bins) based on the angular position of the servo motor. The reason we do not need to include velocity (or the direction of current movement) is that we use *blocking* for robot movement. It means that, while the servo is moving to a specific angular position the program is frozen. While this might not be ideal, it helps to the control algorithm to learn the task easier (since the world will not change while the controller is deciding). To clarify, we divided the possible angular positions (between -1.5 and 1.5) to 60 different bins. We use the step size of 0.1 for the actor and the critic. The average reward step size is set to 0.01 and we use bootstrapping parameter, λ, of 0.4. Finally, the reward signal is calculated based on the angular distance between the current angular position and angular position 1 (the absolute value of current angular position subtracted by 0). The starting angular position is 1.

Figure 2 illustrates the angular position, reward and the selected action at early stages of learning and we can see the servo motor is moving around. Also, Figure 3 illustrates the actions preferences at beginning of learning. After around 2000 time steps we can see in Figure 4 that the servo angular position is moving back and forth close to angular position 0. Since there are only two possible actions, moving forward of backward (and not staying) the servo motor needs to move back and forth to keep itself close to angular position 0. Figure 5 illustrates the action preferences at time step near 2000.

*Amir Samani is with Faculty of Computing Science, University of Alberta, Edmonton, Canada samani@ualberta.ca

[1] available at: https://github.com/Amir-19/M4-ACRL

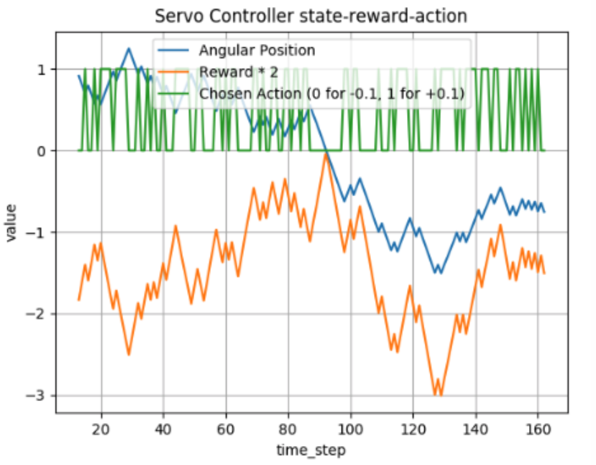

Fig. 2. Angular position, reward and the selected action at early stages of learning.

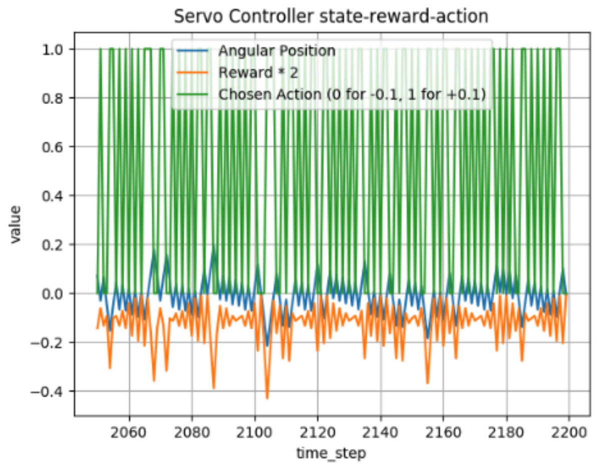

Fig. 4. Angular position, reward and the selected action at time step near 2000.

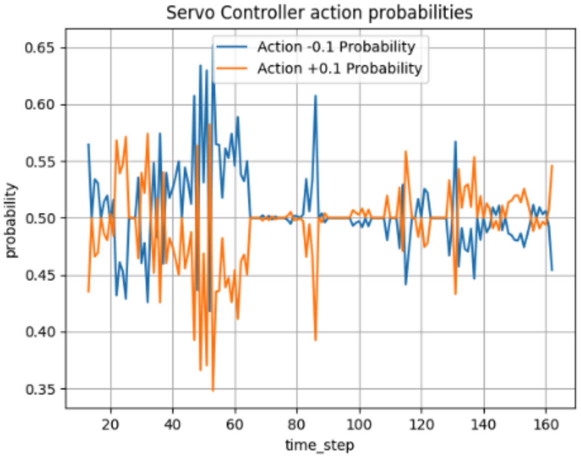

Fig. 3. Actions preferences at early stages of learning.

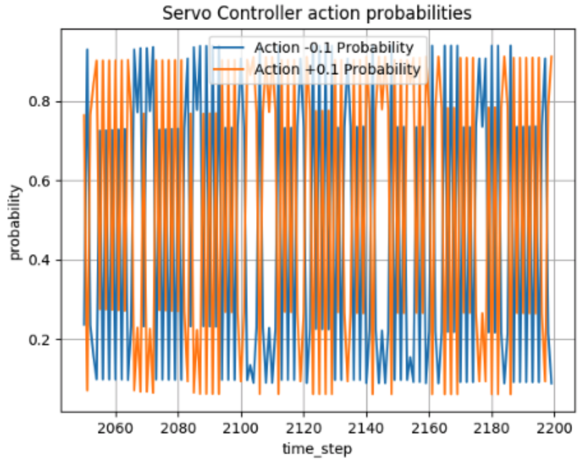

Fig. 5. Actiosn preferences at time step near 2000.

IV. Continuous-Action-Space Control Experiment

For the continuous-action-space experiment, we have the similar task to the previous experiment. While instead of a discrete set of actions, we parameterize a normal distribution to achieve the continuous-action-space. To clarify, all the detail of this experiment, including the reward signal design, the angular position boundaries, and the function approximation scheme are exactly the same as the previous experiment. We still use ”Actor-Critic with Eligibility Traces (continuing)” from the RL textbook [4]. However, since we are dealing with continuous-action-space, we have a normal distribution to take the actions from. The update rules for this setting is available in Exercise 13.4 of the RL textbook. Another important implementation detail is, we do not directly use the action chosen by the algorithm, but we divide the action selected by the actor-critic method by 10 (the reason is to make the magnitude of the action smaller). The critic step size is 0.1. For the actor we have two step sizes, one for the mean and one for the standard deviation. The mean step size is 0.1 and the standard deviation step size 0.01. The average reward step size is 0.1. we use bootstrapping parameter, λ, of 0.4.

Figure 6 illustrates the angular position, reward and the selected action at early stages of learning and Figure 3 illustrates the mean and standard deviation of the parameterized normal distribution. After around 3000 time steps the controller is performing much better at keeping the servo motor at angular position 0. Figure 8 illustrates the angular position, reward and the selected action time step near 3000. Figure 9 illustrates the mean and standard deviation of the parameterized normal distribution at time step near 3000 and we can see that mean value are learned and the standard deviation has decreased (lower standard deviation means we are closer to a fixed policy).

V. Summary

In this module we experiment robot servo control using actor-critic methods under both discrete-action-space and continuous-action-space settings. The task was to achieve a certain angular position and a reward signal calculated

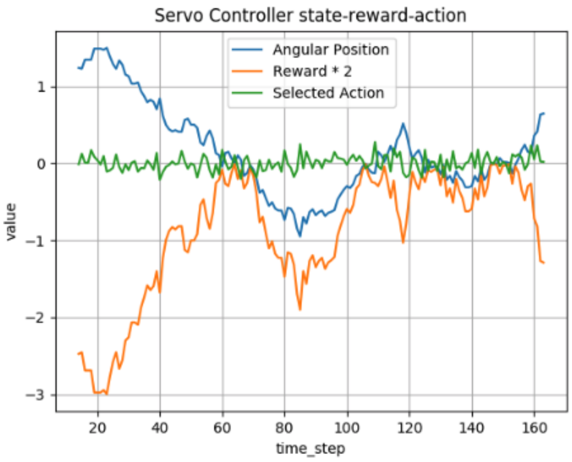

Fig. 6. Angular position, reward and the selected action at early stages of learning.

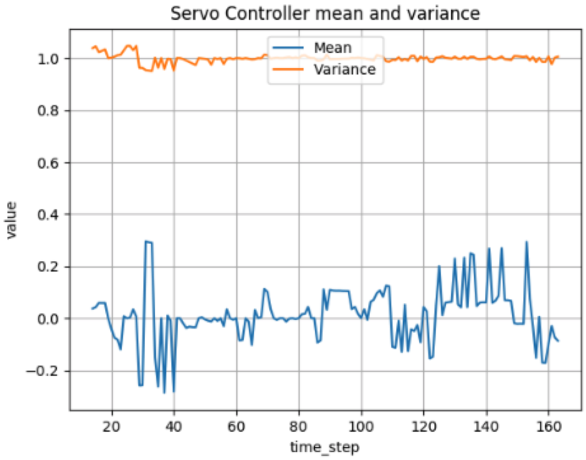

Fig. 7. Mean and standard deviation of the parameterized normal distribution at early stages of learning.

based on the distance between current angular position and the desired angular position. We showed actor-critic methods can successfully control the servo motor to achieve the goal of this tasks in both experiments. This work can be extended greatly by introducing better function approximation scheme. Also, a harder task, such as using two servo motors, with one following a simple policy and the other trying to mirror its angular position, can be an interesting experiment.

References

[1] Amir Samani. Signs of life. https://github.com/Amir-19/M1-Signs_Of_Life, 2018.

[2] Amir Samani. General value functions. https://github.com/Amir-19/M2_GVFs, 2018.

[3] Amir Samani. Horde architecture and pavlovian control. https://github.com/Amir-19/M3-Horde_Pavlov, 2018.

[4] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 2nd edition, 2018.

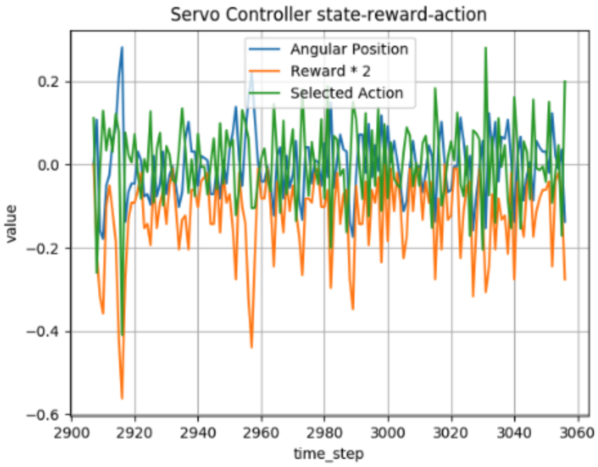

Fig. 8. Angular position, reward and the selected action at time step near 3000.

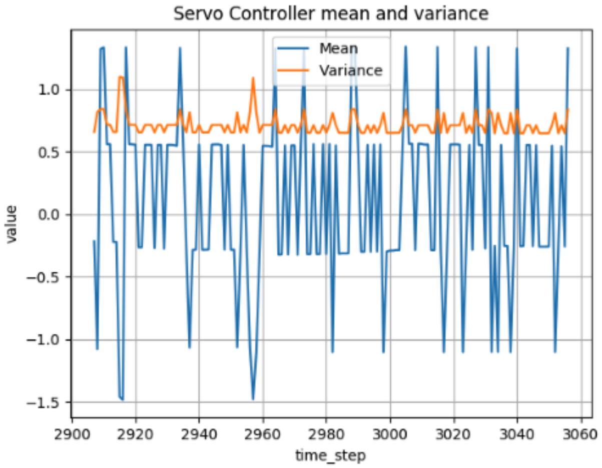

Fig. 9. Mean and standard deviation of the parameterized normal distribution at time step near 3000.