Robot Module 3: Horde and Pavlov

Amir Samani*

***Abstract*— In Robot Module 1, we built the basic tools required for real life system to test various applications of reinforcement learning (RL). In Robot Module 2, we built General Value Functions (GVFs) on top of the experimental setup from Robot Module 1. We designed two on-policy and one off-policy questions in GVF framework and use raw sensorimotor data stream from the robot to answer those questions. In Robot Module 3, we try to extend the ideas from Robot Module 2 to make a system capable of forming a horde of GVFs and try to answer them in real-time. In addition to the horde module, we try to use Pavlovian control to change the behavior of the robot based on the prediction of a GVF, while the new behavior is hand crafted and not learned like control systems in RL context.**

I. Introduction

In Robot Module 2, we showed how to use GVFs as an approach to representing predictive knowledge [1]. In this module we want to take GVFs one step further and build a horde of GVFs to predict various aspect of robot sensorimotor data stream. In fact, we form a horde of ten GVFs (eight on-policy GVFs and two off-policy GVFs) and try to answer their corresponding questions with temporal difference learning methods (TD(λ) for on-policy GVFs and GTD(λ) for off-policy GVFs). 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 design of horde architecture. In Section IV, we design eight on-policy GVF questions and two off-policy GVF question. In Section V we use our horde to answer the corresponding questions about the robot sensorimotor data stream. In addition, we discuss our Pavlovian control experiment as a simple, yet effective way to change the behavior of the robot using predetermined control rules. Last but not least, Section VI 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 [1]. The final robot setup is illustrated in Figure 1. Most of the experiments in this module only use one of the servo motors expect for the Pavlovian control experiment.

III. Architecutre

As illustrated in Figure 2, we get the raw sensorimotor data stream from the robot. The horde contains several GVFs, at each time step, the raw sensorimotor data stream is processed to create the required cumulant and termination for each GVF. After all of the GVFs are updated using their temporal difference learning method, we calculate their RUPEE and UDE [1]. The GVFs predictions can be used as the input of the Pavlovian control component. Pavlovian control component may change the behavior of the policy based on the predictions of one or more GVFs in the horde.

Fig. 1. Robot setup for the experiments.

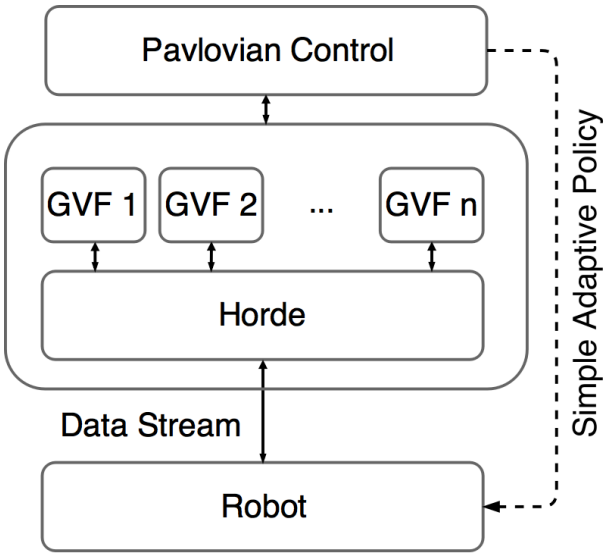

Fig. 2. The implemented horde architecture in this work.

IV. Questions

In order to form our horde, we designed eight on-policy questions and two off-policy questions. We explain each of

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

[1]available at: https://github.com/Amir-19/M1-Signs_Of_Life

them and their corresponding cumulant and termination in the following. However, before going over the questions, it is important to mention that, the behavior policy of the robot for these experiments are similar to Robot Module 2[2], which is rotating between angle -1.5 and +1.5.

1) Next angular position (on-policy): In order to predict next angular position, we use the angle of the servo as the cumulant with termination 0, as we want to predict the immediate angular position, so we only need to ask the question for one time step (γ = 0). This question is on-policy, which means that we are answering the question based on the policy the servo motor is following.

2) Next load (on-policy): To predict the next load, we use the load of the servo as the cumulant with termination 0, as we want to predict the immediate load, so we only need to ask the question for one time step (γ = 0). This question is also on-policy.

3) Next temperature (on-policy): Since we want to predict the next temperature, we use the load of the servo as the cumulant with termination 0, as we want to predict the immediate temperature, so we only need to ask the question for one time step (γ = 0). This question is also on-policy.

4) Next voltage (on-policy): Since we want to predict the next temperature, we use the load of the servo as the cumulant with termination 0, as we want to predict the immediate temperature, so we only need to ask the question for one time step (γ = 0). This question is also on-policy.

5) Number of steps to reach angular position -1.5 (off-policy): As the first off-policy question, we want to know how many steps we are away from angular position -1.5 if we were to always move towards angular position -1.5. The cumulant is always 1, unless we are at angle -1.5. The termination is also always 1 until we reach angular position -1.5.

6) Time to reach angular position +1.5 (off-policy): As the second off-policy question, we want to know how long it is going to take for the servo to reach angular position +1.5 if the servo is to always move towards angular position +1.5. The cumulant is the time difference between each time step (calculated in real time). The termination is always 1 until we reach angular position +1.5.

7) Number of steps to reach angular position -1.5 (on-policy): Similar to the first off-policy question, but this is the on-policy version of the same question (so target policy and behavior policy are the same). We want to know how many steps we are away from angular position -1.5. The cumulant is always 1, unless we are at angle -1.5. The termination is also always 1 until we reach angular position -1.5.

8) Time to reach angular position -1.5 (on-policy): In this question we want to know how long it is going to take to reach angular position +1.5 (on-policy). The cumulant is the time difference between each time step (calculated in real time). The termination is always 1 until we reach angular position -1.5.

[2]available at: https://github.com/Amir-19/M2_GVFs

9) Load of cumulative ten steps into the future (on-policy): Similar to the question 2, while this question is not about the next load but the load over the next ten time steps (so the termination, γ, is 0.9). The cumulant for this question is similar to the question 2, the load.

10) Angular position of cumulative ten steps into the future (on-policy): Same as the previous question but this prediction is about the angular position over the next ten time steps. The termination is 0.9 and the cumulant is the angular position.

V. Experiments and Discussion

In this section we use our designed horde in the mentioned robotic system to answer the corresponding questions. There are three experiments in this section, discussed in the following. Before going further into the experiments, we first discuss all the answer part variables in our system. First, for the simplicity, the state representation is a linear function approximation based on the angular position. We use 15 bins which means we divided the angle -1.5 to +1.5 into 15 bins and we used a vector of size 16 (in order to include all before -1.5 into bin 0). All the GVFs use the same parameters. The step size, α, is 0.1. The step size for second set of weights in GTD(λ), β, is 0.001. The eligibility traces parameter, λ, is 0.9 and all the weights are initialized to zero. For the RUPEE we used $\beta_0 = (1 - \lambda) * \alpha/30$ and its step size is $5 * \alpha$. Last but not least, UDE uses the step size of $0.5 * \alpha$ in our experiments[3].

A. Normal sequence experience

In this experience, the robot is following its regular behavior policy (rotating between angular position -1.5 and +1.5) and we try to answer the GVFs in the horde. The predictions after around 1300 time steps are illustrated in Figure 3. Figure

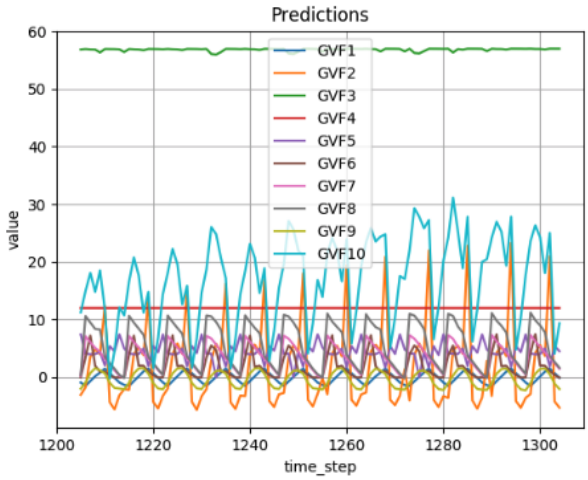

Fig. 3. Predictions of the robot in normal sequence experience at the end of learning.

[3]code for these experiments available at: https://github.com/Amir-19/M3-Horde_Pavlov

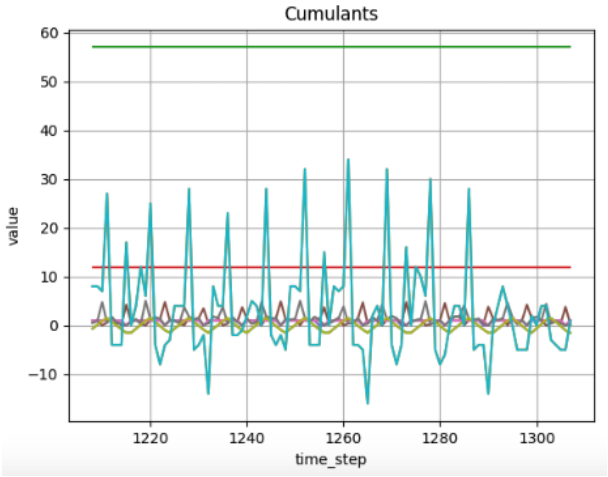

Fig. 4. Cumulants of the robot in normal sequence experience at the end of learning.

In order to measure the GVFs answer correctness we use RUPEE and UDE measures. Figure 5 illustrates the RUPEE measure at the beginning of learning. As we further learn the answer of the GVFs questions we can see the RUPEE measure reduced significantly, illustrated in Figure 6. Also at the end of the learning sequence, we can see in Figure 7 that the RUPEE measure is at a low level. For the UDE measure, similarly, we have high values at the beginning and low values at the end of the learning sequence, as illustrated in Figure 8 and Figure 9.

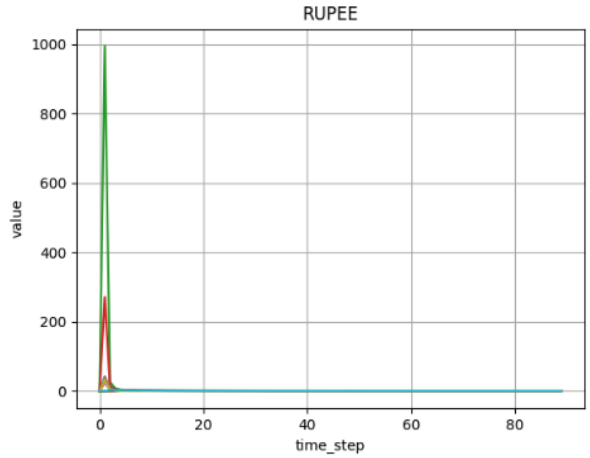

Fig. 5. RUPEE measure at the beginning of learning.

B. Surprise the robot

In this experience, we follow the same procedure as the previous experiment. However, at the time step 250 we make the cumulant of the first GVF (prediction of the next angular position) 10 times larger. Figure 10 illustrates the change in the cumulant. The dark blue line indicates the cumulant of the first GVF and as shown, from the time step 250, it is significantly larger. Since this is a huge change for the environment we expect high TD error for this GVF. Figure

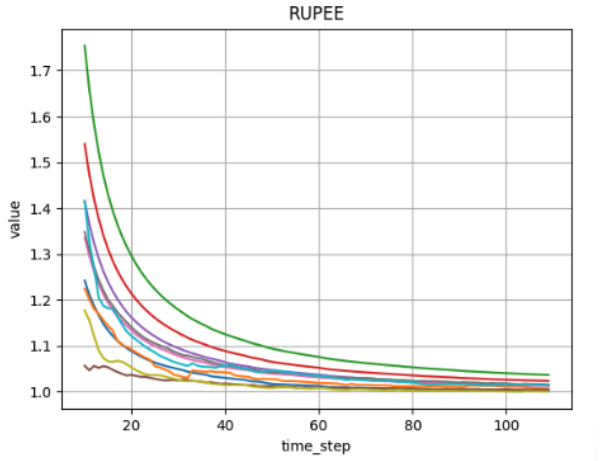

Fig. 6. RUPEE measure after a while

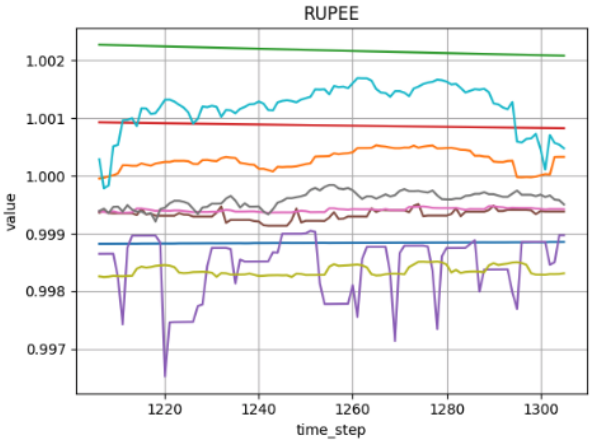

Fig. 7. RUPEE measure at the end of learning

11 illustrates the TD error. This situation is well suited to see whether RUPEE and UDE are able to capture this drastic change in the environment or not. Figure 12 illustrates the RUPEE measure for time step around 250 and onward. As shown, the RUPEE measure increases after the change and then decreases since we are capturing this change and the TD(λ) starting to learn this new signal. This is also true for the UDE measure as illustrated in Figure 13. In addition, we can see in Figure 14 that the prediction for the first GVF (dark blue) is adapting to the new signal.

C. Pavlovian control

With Pavlovian control we can use the prediction from the first GVF (prediction of the next angular position) and use a predetermined policy to adapt the robot behavior. For simplicity, we used the second servo motor as the out of the Pavlovian control, the Pavlovian control rule is designed to be simple. It will rotate the second servo (between angular position -1.5 and +1.5) if the prediction of the first GVF is more than 9. After making the cumulant of the first GVF ten times larger at time step 100, we can see the prediction for this GVF is adapting to the new signal (Figure 15). Each time the first GVF predict the angular position will be more

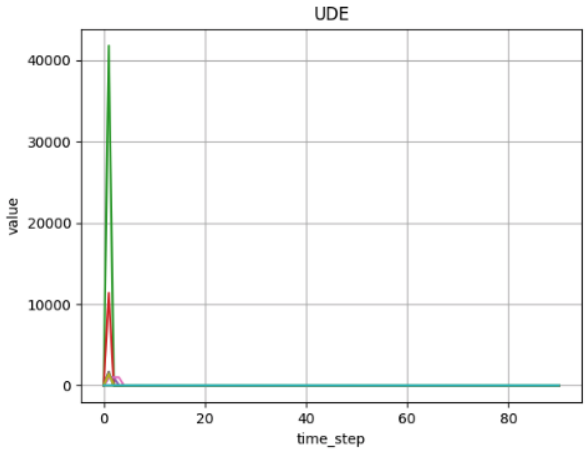

Fig. 8. UDE measure at the beginning of learning.

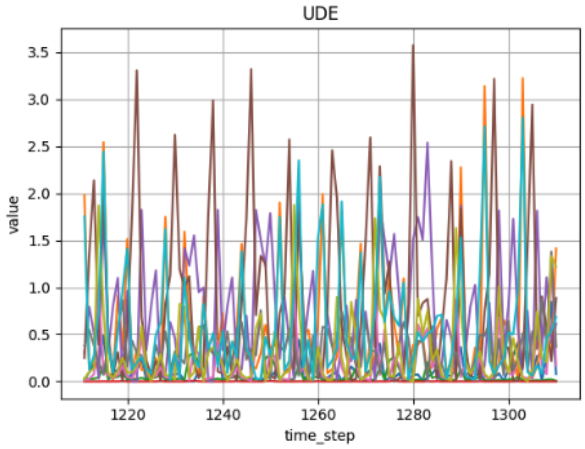

Fig. 9. UDE measure at the end of learning.

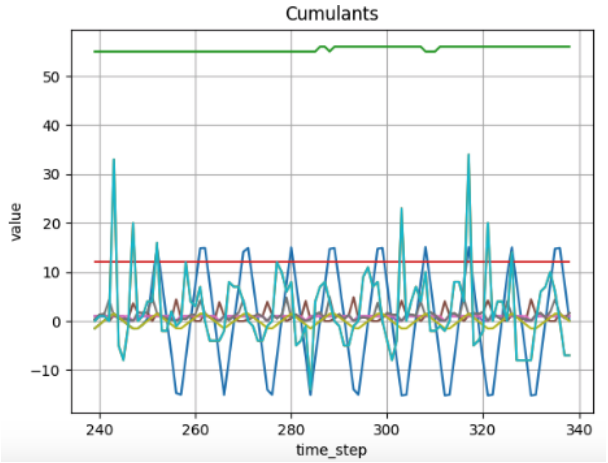

Fig. 10. Cumulant after surprise (dark blue line).

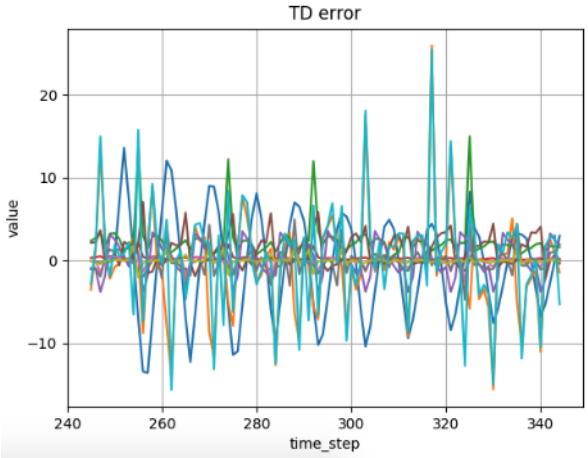

Fig. 11. TD error after surprise (dark blue line).

than 9, we can see rotation in the second servo. The video of this experiment is also included for better demonstrations [4].

VI. Summary

In this module we formed a horde of GVFs and showed how we can use the robot raw sensorimotor data stream to answer them. We designed ten GVFs (eight on-policy and two off-policy) and answered their questions using methods of temporal differences. We also showed the effect of the robot surprise (by changing the cumulant signal) in UDE, RUPEE, and the TD error, so it can be used to identify when the robot is dealing with change in its environment. Moreover, we demonstrated how we can use a prediction by one or more GVFs to change the robot behavior based on a predetermined adaptive policy. There are several aspect of this module that can be extended. From the experiments results we can see that we need better and more complex function approximation scheme such as tilecoding or artificial neural networks. Also, the mentioned architecture is used only ten GVFs while in real world robotic systems we may need to have millions of these predictions and we need more investigation on a better horde architecture.

References

[1] Adam White. *Developing a predictive approach to knowledge*. PhD thesis, University of Alberta, 2015.

[4] available at: https://www.icloud.com/iclouddrive/0nN6rGVn4IZZWjID0ed86Lp0Q#Pavlov.mov

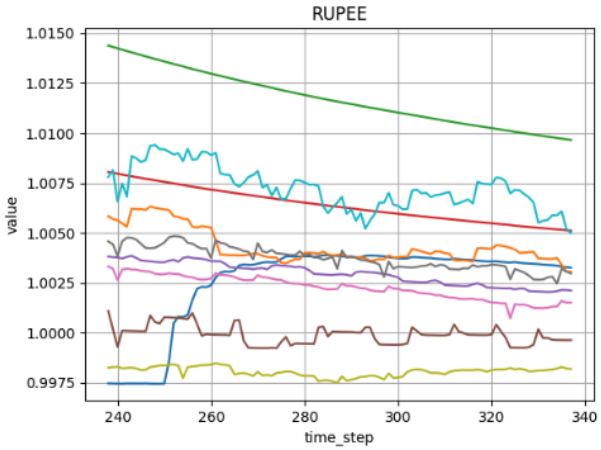

Fig. 12. RUPEE measure after surprise (dark blue line).

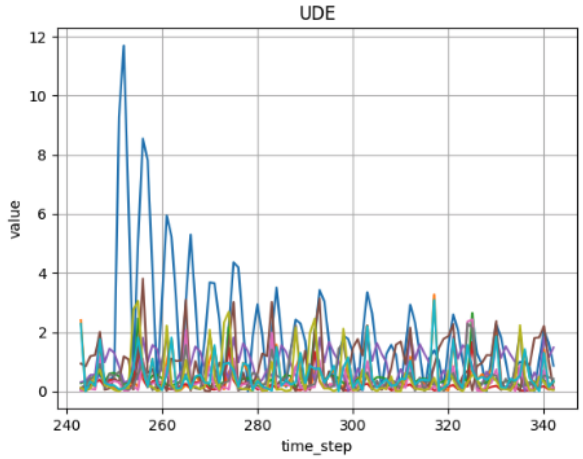

Fig. 13. UDE measure after surprise (dark blue line).

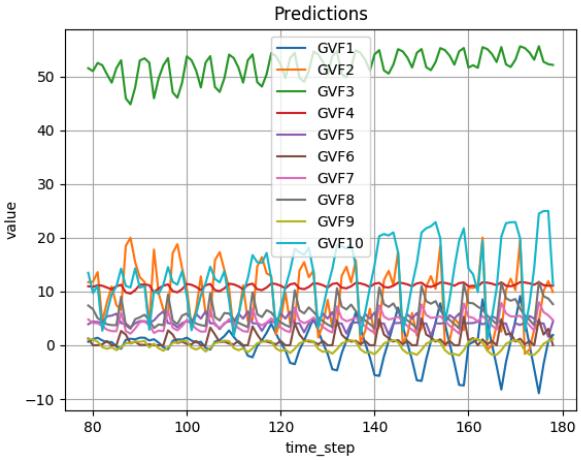

Fig. 15. Predictions adapting to the change in the cumulant signal (dark blue line).

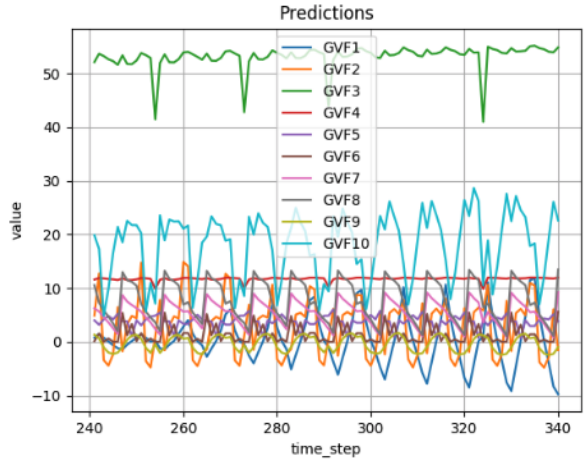

Fig. 14. Predictions after surprise (dark blue line).