Generating a Dynamic Controller for a Flamingo Inspired Robot using Deep Reinforcement Learning

Edward Lu, Nathan J. Kong, J. Joseph Payne, and Aaron M. Johnson
Mechanical Engineering Department, Carnegie Mellon University, Pittsburgh, PA, USA
Email: elu2@andrew.cmu.edu

SUMMARY

As the designs of robotic systems become more complicated over time, it becomes increasingly difficult to generate controllers that efficiently and effectively take advantage of the systems' capabilities. We would like to have a systematic way to improve and adjust aspects of our robot's control and design in order to choose optimal parameters for construction. Here, we explore our process of using deep reinforcement learning (DRL) to generate a dynamic controller for a simulated model of a robot, enabling us to swap out different design parameters (e.g. motor models) in simulation. We chose to use DRL over other methods of generating robot controllers because it is easier to implement in a robot-agnostic manner and requires less expert knowledge as an input. Additionally, the iterative nature of reinforcement learning allows us to adjust parameters of the robot's actuators without dramatically changing the model or the algorithm. Here, we discuss how we generated our controller and the effects it had on our design choices.

I. MOTIVATION

In prior work, we created a passively stable bipedal robot based off the skeletal structure of a flamingo [1]. This prototype was designed to minimize energy usage by relying on its mechanical structure to stand. Though stable when idle, the robot is quite unstable when moving and is unable to perform a dynamic walking gait. In order to address these issues, we decided to use DRL, which allows us to maximize the efficiency and develop a walk cycle without taking into account the robot's full dynamics, which are complex, arduous, and time consuming to optimize. DRL has also seen some success in generating bipedal locomotion in recent years [2]. Additionally, the system can be used to experiment with various motor parameters, such as gearing, torque limits, maximum velocities, etc., to determine which type of motor would be optimal for the next iteration of the robot.

II. METHODS

We started by creating an approximate model of the robot in MuJoCo (see Fig. 1) and ported it to OpenAI's Gym toolkit [3]. For the learning algorithm, we use Proximal Policy Optimization (PPO) with a generalized advantage estimation [4] because it is easier to implement and its hyperparameters are simpler to tune compared to other DRL methods. Our implementation uses two neural networks, one for the value function and one for the policy, each with three hidden layers. It is trained on an eight-core processor for 50,000 episodes. Training takes approximately 7-9 hours.

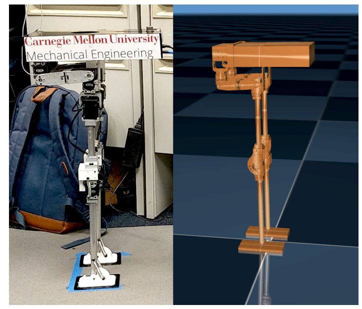

Fig. 1: Left: Real life prototype of the flamingo robot. Right: Robot model simulated in MuJoCo simulation environment.

The reward function used is:

$$R = \vec{v}_x + w_0 - w_1 \sum \vec{a}^2 - w_2 \Delta \vec{f}_z + w_3 \sum \Delta \vec{a}^2 \quad (1)$$

where R is the reward the model receives for each simulation step, $\vec{a}$ is a vector representing an action that the model took, $\vec{v}$ is a vector of the velocity of the center of mass, $\vec{f}$ is the position of the robot's foot from the ground, and w is a vector of weights.

III. RESULTS AND DISCUSSION

The reinforcement learning policy is able to get the simulated robot model to consistently walk dynamically for a long period of time. The DRL workflow also allows us to easily swap motor parameters and adjust the walking gait. We are able to find several brushless motors that fit the parameters of the best walk cycle.

In the future, we plan to make our models and policies more robust under unpredictable situations. To do so, we will implement dynamic randomization and add noise and delay to our observations to account for real world unpredictability and fluctuations. Though many of our policies are optimized in a simulation environment, we also plan to train on a real robot once we have an updated prototype.

REFERENCES

[1] J. J. Payne, N. J. Kong, and A. M. Johnson, "Flamingobot: a flamingo inspired minimal energy standing biped robot," in *Dynamic Walking*, Canmore, Canada, June 2019, workshop abstract.

[2] Z. Xie, G. Berseth *et al.*, "Feedback control for cassie with deep reinforcement learning," *CoRR*, vol. abs/1803.05580, 2018. [Online]. Available: http://arxiv.org/abs/1803.05580

[3] G. Brockman, V. Cheung *et al.*, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540

[4] J. Schulman, F. Wolski *et al.*, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347