Learning Control Strategies and Design Parameters for Flexible-legged Locomotive Systems

C.R.A.W.LAB
http://www.ucs.louisiana.edu/~jev9637/

University of Louisiana Lafayette — Mechanical Engineering

Andrew Albright, Joshua Vaughan

andrew.albright1@louisiana.edu

Flexible Robotics

Figure 1: Boston Dynamics Atlas Robot [1]

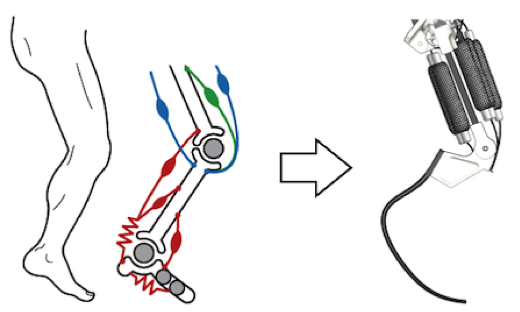

Figure 2: Athlete Robot [2]

Legged systems, like the robot seen in Figure 1, have many advantages compared to their wheeled counterparts. For example, if designed properly, they can more easily navigate uneven and unstable terrain. However, there are disadvantages as well, including dramatically less energy-efficient locomotion. In an effort to mitigate this disadvantage, research has been conducted that shows using flexible components within legged systems, like the ones seen in Figure 2, not only increases their efficiency but also their performance [3]. However, nonlinear models and controllers for flexible systems are difficult to develop using traditional methods.

Reinforcement Learning

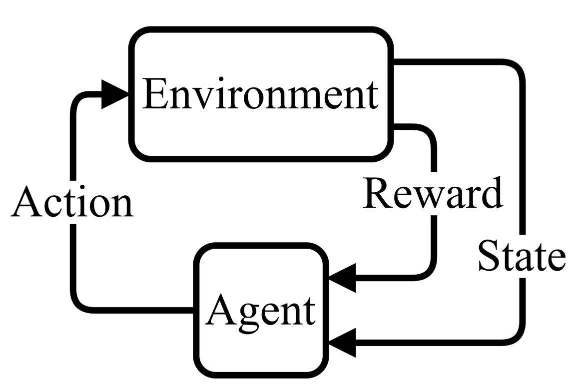

Figure 3: RL Block Diagram

A solution to combat the nonlinear difficulties is the use of Reinforcement Learning (RL), which takes advantage of neural networks to represent the nonlinear aspects of the control architecture. The RL approach does not require closed-form models of the system dynamics to develop a controller. Rather, the approach creates a machine-learning-based agent to interact with an environment collecting samples of rewards and actions that get mapped to system states, as shown in Figure 3. With this map, the agent defines inputs for the system based in its state. Previous work has shown that this method can be successful at defining both mechanical parameters and control strategies for rigid systems [4].

Reinforcement Learning for Flexible Robotics

In this work, Reinforcement Learning is used to develop control strategies leading to higher performance, both in desired movement and energy efficiency, for a flexible system. The pogo stick model shown in Figure 4 is used to train an agent to determine the acceleration of the mass, m_a, along the rod, m_l, causing the system to jump. The agent is tasked with learning a control strategy that jumps as high as possible in a single jump while also limiting power consumption. Two training strategies are deployed to accomplish this task. In the first strategy, the agent is rewarded based on the height it achieves, receiving a score directly proportional to jump height. For the second strategy, the agent is rewarded using a normalized positive score proportional to jump height, and punished a normalized negative score proportional to power use. The agent is able to learn both optimal jumping height strategies and power conserving strategies.

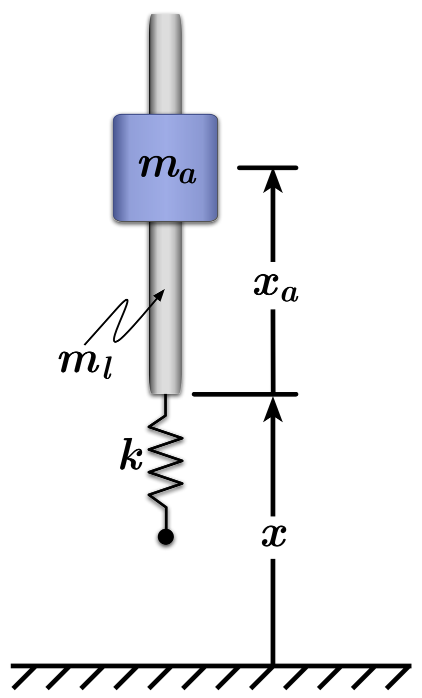

Figure 4: Model

Defining The Reward

Two reward functions are defined for training the two different agents; the first rewards the agent based on the height it reaches; the second seeks to balance reward height with power usage. The reward function for the second case is:

$$R = \frac{\omega_x \frac{x_t - x_{min}}{x_{max} - x_{min}} + m_a\, \omega_p \frac{a_t v_t - a_{max} v_{max}}{a_{min} v_{min} - a_{max} v_{max}} - \omega_p}{\omega_x + \omega_p} \tag{1}$$

where x_t, a_t and v_t are position, velocity and acceleration of mass m_a, respectively, and ω_x and ω_p are weights used to tune the relative contributions of jump height and power use.

Jumping Analysis

The jumping height data presented in Figure 5 shows that when training the agent to jump high regardless of power usage, the maximum jump height reached is higher than when training the agent to conserve power. The maximum height reached by the power conserving agent is 56% of the height reached by the agent trained to only jump high. However, the power consumption data in Figure 6 shows that the power conserving agent is more energy efficient, using 64% less power. Comparing the maximum jump height reached to the power used to get there, the the power conserving agent has learned a more efficient control strategy.

Evaluation Data

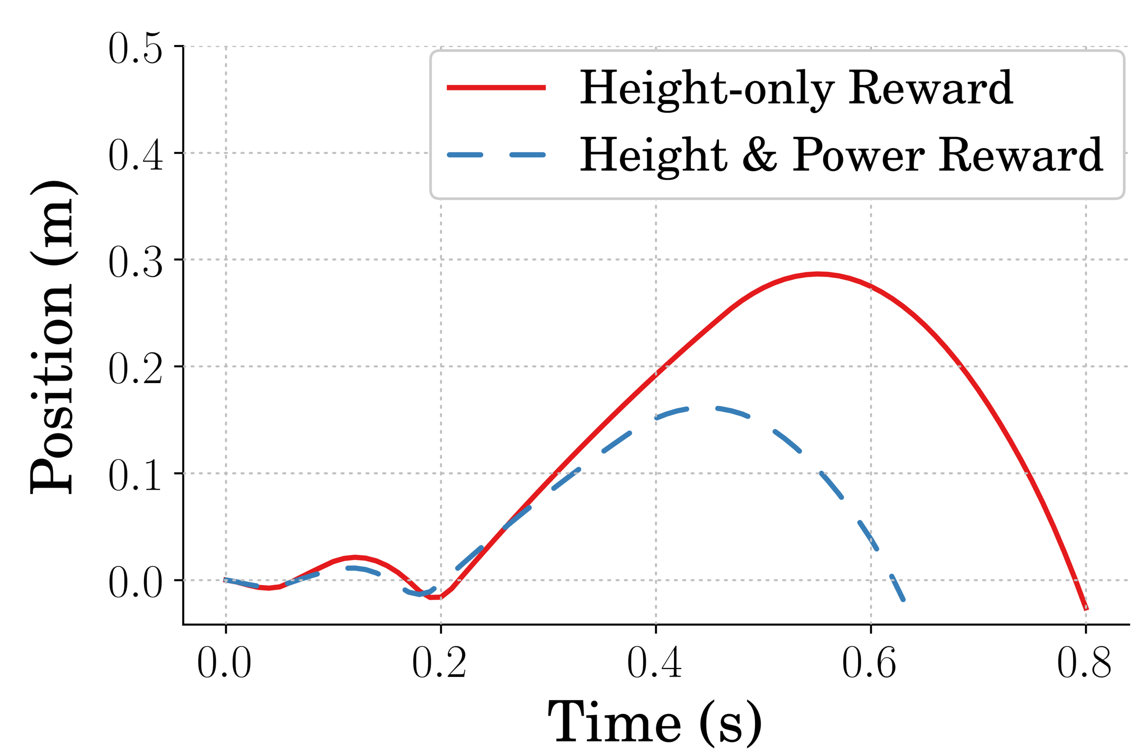

Figure 5: Jumping Height Comparison

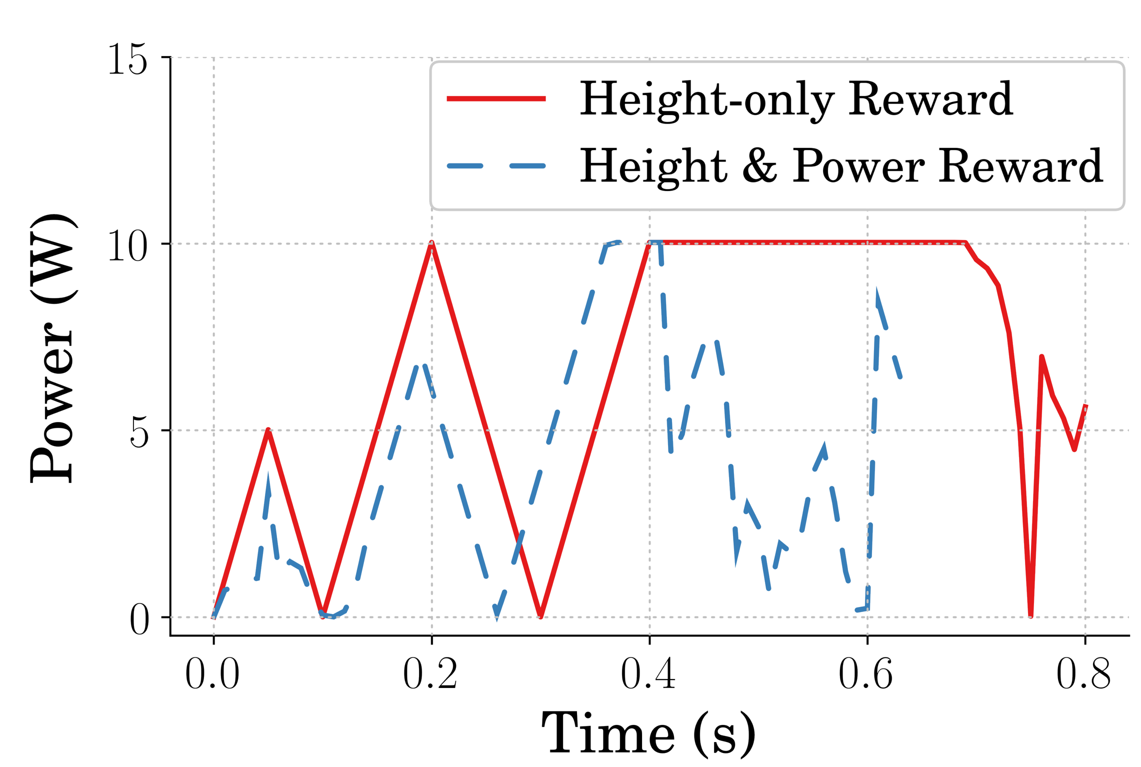

Figure 6: Power Usage Comparison

Implications of Results

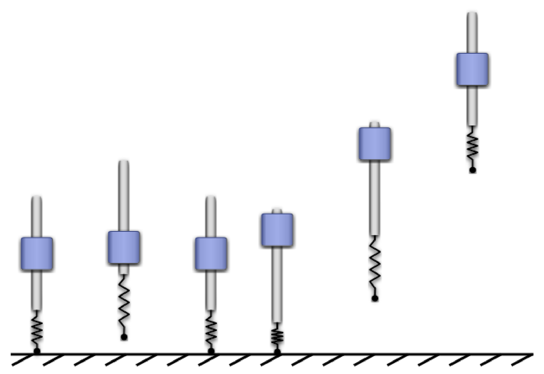

Figure 7: Stutter Jump [5]

The jumping profiles presented in Figure 5 show that the agent is controlling the pogo stick to jump with a profile matching a stutter jump profile, which can be seen in Figure 7. This kind of jump is optimal for reaching maximum heights [5]. The reinforcement learning agent learned optimal jumping strategies while minimizing power consumption for the simple model of a flexible legged locomotion system.

Conclusion

Using a reinforcement learning approach to define control strategies for flexible jumping systems leads to optimal performance. Using a correctly tuned reward function can further improve performance, leading to control strategies that require less power.

References

[1] Boston Dynamics.
Atlas® boston dynamics, 2021.

[2] designboom.
ryuma niiyama athlete robot, 2010.

[3] Yuuta Sugiyama and Shinichi Hirai.
Crawling and jumping of deformable soft robot.
2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4(c):3276–3281, 2004.

[4] David Ha.
Reinforcement learning for improving agent design.
Artificial Life, 25(4):352–365, 2019.

[5] Joshua Vaughan.
Jumping Commands For Flexible-Legged Robots.
2013.

Acknowledgements

Special thanks to the Louisiana Crawfish Promotion and Research Board for their support.