Reinforcement Learning Power Conservative ~~Control~~ Strategies

Andrew Albright * **Joshua Vaughan** **

* *University of Louisiana at Lafayette, Lafayette, LA 70503 USA (andrew.albright1@louisiana.edu)*
** *University of Louisiana at Lafayette, Lafayette, LA 70503 USA (g.dang1@louisiana.edu)*

Abstract: Legged locomotive systems have many advantages over their wheeled counterparts, such as their ability to navigate rough terrain. They have the ability to deploy many navigating techniques to overcome obstacles, one of which is jumping. Still, there are disadvantages to overcome when using legged systems, such as their lack of energy efficiency. To combat this lack of efficiency, flexible links can be used to conserve energy that would otherwise be wasted during locomotion. Furthermore, control methods can be utilized which assist in both a jumping systems ability to jump high, and their ability to conserve power. Using reinforcement learning (RL) to create these controllers for flexible-legged jumping systems can lead to designs that outperform, in terms of jumping height, traditional optimization methods for jumping. Furthermore, using a power-conservative reward function to train the agent results in a control strategy which balances jump height and energy usage, leading to more efficient jumping control.

Keywords: Power Efficient, Reinforcement Learning, Flexible Robots, Legged Locomotion, Jumping

1. INTRODUCTION

Legged locomotive robots have many advantages over their wheeled counterparts, for example their ability to more easily navigate harshly uneven terrain Saranli et al. (2001), Park et al. (2017) Blackman et al. (2018) Seok et al. (2015). However, there are disadvantages as well, one of which is power consumption. There are several factors that contribute to this disadvantage. For example, in many cases several motors are required to actuate several linkages to move only a few feet, whereas a wheeled system needs only a single motor to turn a wheel to accomplish the same task. Another factor, one that is particularly prevalent when navigating uneven terrain, is the challenge of defining how a walking robot places its feet in such a way that reduces the level of wasted energy by harshly taking steps.

In an effort to alleviate the power consumption issues seen when using legged systems to accomplish locomotive tasks such as walking, running and jumping, research has been conducted which replaces rigid aspects of said systems with flexible ones. It has been shown that this not only leads to higher performance but also higher efficiency Sugiyama and Hirai (2004), Galloway et al. (2011), Hurst (2008), Seok et al. (2015).

While the introduction of flexible components within robotics systems solves some challenges related performance measures like power consumption, it raises other challenges. ~~Particularly the difficulties seen in~~ modeling ~~the~~ the systems become ~~prevalent~~ because the models become highly nonlinear. Different approaches have been taken to ~~circumnavigate and solve~~ these issues. A popular and successful example being ~~the~~ use of series elastic actuators instead of flexible links Iida et al. (2005), Ahmadi and Buehler (1997). Other solutions seek to create control methods which are ~~more~~ suited to non-rigid systems. One of those control methods is the use of reinforcement learning to define control strategies based on interactions with the environment. In this work, RL is used to train an agent (controller) which seeks to jump a simplified jumping robot modeled as a pogo stick. The RL agent is tasked with maximizing jump height while conserving power. It is shown that when tasked as such, the agent finds unique control strategies to maximize the jumping potential of the system, ~~as well as~~ balancing power usage.

* Road map for the rest of the paper * discuss related work in the field * detail the method used to test the idea * system * reward * system used to compare (DV's) * discuss results * conclude the results and discuss future work

2. PREVIOUS WORK

2.1 Reinforcement Learning for Legged Locomotion

Research has shown that using RL for defining control strategies for legged systems is a viable path for simple and even complex legged locomotive systems. Yang et al. (2019) shows that using a model-based method can require an order of magnitude less environment interactions comparing to the best performing model-free algorithms to train controlling agents. However, model-based methods expose other challenges. Learning a model often requires many interactions, and the inevitable limited environment

that is learned leads to limited exploration during controller training. As such, model-free methods are often of interest as they can be deployed directly on hardware to learn based on interactions with the environment, or in simulation. Peng et al. (2016) shows the viability of such an approach training many different agents to locomote navigating uneven terrain. Reda et al. (2020) shows similar successes using a host of different environments, while also displaying the apparent difficulties of defining an environment.

2.2 Reinforcement Learning for Flexible Systems

As discussed, flexible systems do have some advantages over their rigid counterparts, particularly ones which are used for locomotion tasks. Thuruthel et al. (2019) shows the use of model-based methods for controlling soft-robots where as a part of training, learning the model is required. In contrast to that work, Dwiel et al. (2019) shows that using model-free methods and algorithms such as DDPG citeDDPG prove effective for both locomotive and non-locomotive tasks where flexible systems are considered. Additionally, comparing RL control strategies to more traditional control strategies such as PD control,He et al. (2020) shows that the comparison results show that RL certainly is practically applicable for controlling flexible systems. Furthermore, work by Cui et al. (2019) shows that RL can be used for partial control of flexible systems solely to damp out vibration while a separate controller is used to determine general pose.

2.3 Flexible Robots Improved Performance

Seok et al. (2015), Sugiyama and Hirai (2004), Using flexible components within robotic systems has shown great potential for conserving power. Ahmadi and Buehler (1997) shows the results of a common technique which is using flexible joints (series elastic actuators) to increase energy efficiency. Folkertsma et al. (2012) shows that flexible joints are not the only way to increase efficiency, but a similar technique which seeks to emulate the tendons seen in nature can produce similar efficient results. In the work completed by Seok et al. (2015), Seok et al. (2013), design principles for efficient quadruped locomotion are discussed, and evidence supports the use of flexible components increases efficiency.

2.4 Control for Power Efficiency ***WIP***

Pace et al. (2017), Harper et al. (2017) In stead of relying solely on the design of a system to increase efficiency, Harper et al. (2019) shows that the use of model predictive control methods can lead to more efficient locomotion strategies alone. * Discussion leading towards talking about using RL for training agents to control flexible legged systems to be more efficient. Which is what we are doing.

3. POGO-STICK ENVIRONMENT

3.1 Define pogo-stick environment

Representing a flexible legged locomotive system is the jumping model shown in Fig. 1. Blickhan and Full (1993)

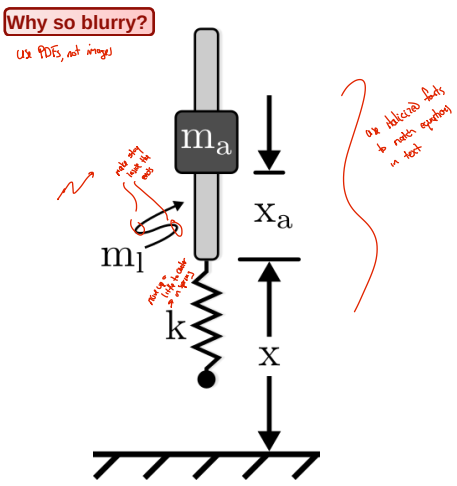

Fig. 1. Pogo-stick Model: A Simplified Flexible Legged Locomotive Jumping System

Table 1. Training and Evaluation Parameters

Model Parameter	Value
Mass of Leg, m_l	0.175 kg
Mass of Actuator, m_a	1.003 kg
Natural Frequency, ω_n	11.13 Hz
Spring Constant, k	200000 N/m
Actuator Stroke, $(x_a)_{\max}$	25 mm
Actuator Velocity, $(\dot{x}_a)_{\max}$	2.0 m/s
Actuator Acceleration, $(\ddot{x}_a)_{\max}$	10 m/s

show that this model can be used as a base representation for many different jumping animals which in turn represent many different kinds of jumping gaits. As such, this model is used as the environment that the control agents are trained and evaluated on to determine if reinforcement learning can define efficient strategies. During training, the agent is tasked with learning a control strategy with the input defined as the acceleration of the mass, (m_a), along the rod (m_l).

Variables m_a and m_l represent the masses of the actuator and leg respectively. Because this system is selected as a representation of flexible systems a spring with constant k is used as a flexible component. A damper (not shown in Fig. 1) is present within the model of the system as well and is represented by the variable c. The vertical position relative to the ground is represented by x, and the position which the actuator moves along the rod is represented by x_a. The values of many of these parameters are detailed in Table 1. Note that the system is defined to move along a vertical line so that the agent does not learn to keep the system balanced.

4. REINFORCEMENT LEARNING

For the experiments in this work, a traditional reinforcement learning setup is utilized. Wherein a Markov Decision Process (MDP) is defined by a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition probability function $p : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, which is used to define the probability of entering into state s_{t+1} from state s given action a is taken, a reward function $R : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which defines the reward $R(s_{t+1}, s_t, a_t)$ received by entering into state s_{t+1} from state s_t by taking action a, and a future discounting

factor $\gamma \in [0, 1]$ which weights the importance of estimated future rewards. To solve the MPD a policy $\pi_\phi : \mathcal{S} \rightarrow \mathcal{A}$ is employed where ϕ are defined so that all $s_0 \in \mathcal{S}$ have a defined maximum $J(\phi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_{t+1}, s_t, a_t)\right]$.

The algorithm used to define the weights ϕ is Stable Baselines written version of of Twin Delayed Deep Deterministic Policy Gradient (TD3) Fujimoto et al. (2018). This is a modern and state of the art, actor-critic, off-policy RL algorithm wherein the actor is represented by the policy π_ϕ taking the actions a, and the critic is represented by the estimated expected return of taking action a in state s and following the policy π from there after. The critic is a neural network itself with parameters θ that updated according to temporal difference error found between a set of twin target networks. These target networks are updated to follow the critic network every defined n updates of the critic network.

5. REWARD FUNCTION

5.1 Reward the Agent for Jumping High

To accomplish the task of jumping high, the reward function which the RL algorithm will seek to maximize, is defined to represent equating the height of the pogo-stick. The signal returned to the agent is normalized with a predefined maximum height so that if the pogo-stick is at or over that maximum, the agent is receiving maximum reward. This reward function is defined as:

$$r = \frac{x_t - x_min}{x_max - x_min} \tag{1}$$

where x_t is the height at the current time step, x_min is the minimum height of the system and x_max is the max height of the system. Note that x_max is chosen based on the theoretical maximum height discovered by solving the IVP...**WIP**

5.2 Reward the Agent for Jumping Efficiently

The second reward function used is one that seeks to balance jumping height with power consumption ~~during an episode~~. To do this, the reward signal returned is defined as the ratio of the height achieved at a given time step to the power used up to that time step. The power used at a given time step is defined by:

$$p_t = m_a \, a_{a_t} \, v_{a_t} \tag{2}$$

where m_a is the mass of the actuator, ~~as seen in Fig. 1,~~ a_{a_t} is the acceleration of the actuator at time t and a_{v_t} is the actuators velocity at time t. The efficiency at a given time step is then ~~expressed as~~:

$$e_t = \frac{x_t}{\sum_{t=0}^{t} p_t} \tag{3}$$

This efficiency value is then normalized where the max and min efficiency limits are determined based on experimental realizations of efficiency expectations. The max efficiency e_{max} is set to 0.002 and the min efficiency e_{min} is set to 0. The reward function for the case of maximizing agent efficiency can then be written as:

$$r = \frac{e_t - e_{min}}{e_{max} - e_{min}} \tag{4}$$

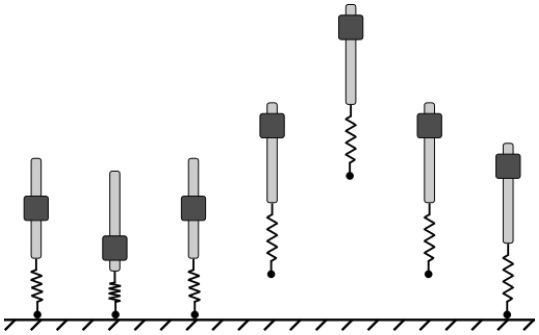

Fig. 2. Pogo-stick Single Jump ~~Representation~~

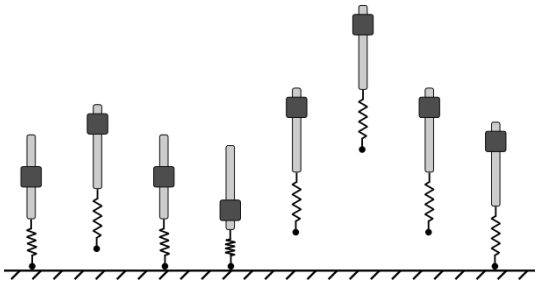

Fig. 3. Pogo-stick Stutter Jump ~~Representation~~

~~so that~~ getting any reward requires the agent to first get off the ground and maximizing reward requires the agent to maximize jump height while minimizing power used.

6. EXPERIMENTS

6.1 Define Agents Task

To evaluate if reinforcement learning, and more specifically the TD3 algorithm, can define more efficient control strategies, two types of jumping experiments are performed. For the first, the agent is tasked with learning both high jumping and efficient jumping strategies for single jumps within the pogo-stick environment. This kind of jump can be visualized within Fig. 2. Here it is expected that the agent will learn some form of initial spring compression leading towards launching the pogo-stick as high as possible.

The second jumping strategy the agent is tasked with learning both high jumping strategies and efficient strategies for is called a stutter jump and can be visualized in Fig. 3. Here, it is expected that the agent will learn to compress the spring, launch the pogo-stick, return and compress the spring farther and finally launch the agent as higher than the initial launch.

6.2 Define Agent Training

To find control solutions for each task, ten different agents are trained on each task individually. The agents ~~are given~~ the ~~initialized~~ pogo-stick environment with its initial position x at resting position so that the spring is ~~compressed according to the weight of the system~~. The actuator is initialized ~~half way along~~ its stroke a_a, so that it can move in either direction. The agent is to control the actuator until either the episode is terminated due to the agent accomplishing the task, or until the episode times out and the environment is reset.

Episode terminations are defined based on the two different tasks the agent is looking to accomplish. The first episodic termination stipulation is after the agent completes a single jump. This is defined as the rods position (x), being greater than zero, then returning to zero. Here, the episode is terminated and the pogo-stick is reset to its initial state. The second type of episode termination is based on the stutter jump motion. This is allowing the rods position to be greater than zero twice and then returning to zero. Note that if the two defined terminations do not occur, the episode is terminated after 500 time steps, or 5 seconds. Each agent ~~used~~ is trained for a total of 500k time steps. This results in a varying number of episodes, depending on the way ~~the~~ episode ~~is defined to~~ terminates.

7. RESULTS **WORKING ON THESE NOW, DOING SOME PLOTTING**

* Analyze jump height results from agents * Analyze power consumption results from agents * Analyse efficiency vs $omega_x$ for power efficiency agents specifically * Make comparisons of jump height reached to power consumed * Compare results to DV's paper

8. CONCLUSION

* Using RL leads higher jumping robot control designs * Using RL also leads to more energy efficient controller designs * Mention future work giving RL access to the design parameters to further optimize jump height and power consumption

~~A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.~~

ACKNOWLEDGEMENTS

Place acknowledgments here.

REFERENCES

Ahmadi, M., Buehler, M., 1997. Stable control of a simulated one-legged running robot with hip and leg compliance. IEEE Transactions on Robotics and Automation 13 (1), 96–104.

Blackman, D. J., Nicholson, J. V., Pusey, J. L., Austin, M. P., Young, C., Brown, J. M., Clark, J. E., 2018. Leg design for running and jumping dynamics. 2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017 2018-Janua, 2617–2623.

Blickhan, R., Full, R. J., 1993. Similarity in multilegged locomotion: Bouncing like a monopode. Journal of Comparative Physiology A 173 (5), 509–517.

Cui, L., Chen, W., Wang, H., Wang, J., 2019. Control of Flexible Manipulator Based on Reinforcement Learning. Proceedings 2018 Chinese Automation Congress, CAC 2018, 2744–2749.

Dwiel, Z., Candadai, M., Phielipp, M., 2019. On Training Flexible Robots using Deep Reinforcement Learning. IEEE International Conference on Intelligent Robots and Systems, 4666–4671.

Folkertsma, G. A., Kim, S., Stramigioli, S., 2012. Parallel stiffness in a bounding quadruped with flexible spine. IEEE International Conference on Intelligent Robots and Systems, 2210–2215.

Fujimoto, S., Van Hoof, H., Meger, D., 2018. Addressing Function Approximation Error in Actor-Critic Methods. 35th International Conference on Machine Learning, ICML 2018 4, 2587–2601.

Galloway, K. C., Clark, J. E., Yim, M., Koditschek, D. E., 2011. Experimental investigations into the role of passive variable compliant legs for dynamic robotic locomotion. Proceedings - IEEE International Conference on Robotics and Automation, 1243–1249.

Harper, M., Pace, J., Gupta, N., Ordonez, C., Collins, E. G., 2017. Kinematic modeling of a RHex-type robot using a neural network. Unmanned Systems Technology XIX 10195, 1019507.

Harper, M. Y., Nicholson, J. V., Collins, E. G., Pusey, J., Clark, J. E., 2019. Energy efficient navigation for running legged robots. Proceedings - IEEE International Conference on Robotics and Automation 2019-May, 6770–6776.

He, W., Gao, H., Zhou, C., Yang, C., Li, Z., 2020. Reinforcement Learning Control of a Flexible Two-Link Manipulator: An Experimental Investigation. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 1–11.

Hurst, J., 2008. The Role and Implementation of Compliance in Legged Locomotion. The International Journal of Robotics Research 25 (4), 110.

Iida, F., Gómez, G., Pfeifer, R., 2005. Exploiting body dynamics for controlling a running quadruped robot. 2005 International Conference on Advanced Robotics, ICAR '05, Proceedings 2005, 229–235.

Pace, J., Harper, M., Ordonez, C., Gupta, N., Sharma, A., Collins, E. G., 2017. Experimental verification of distance and energy optimal motion planning on a skid-steered platform. Unmanned Systems Technology XIX 10195, 1019506.

Park, H. W., Wensing, P. M., Kim, S., 2017. High-speed bounding with the MIT Cheetah 2: Control design and experiments. International Journal of Robotics Research 36 (2), 167–192.

Peng, X. B., Berseth, G., Panne, M. V. D., 2016. Terrain-Adaptive Locomotion Skills Using Deep Reinforcement Learning 35 (4), 1–12.

Reda, D., Tao, T., van de Panne, M., 2020. Learning to Locomote: Understanding How Environment Design Matters for Deep Reinforcement Learning. URL http://arxiv.org/abs/2010.04304%0Ahttp://dx.doi.o

Saranli, U., Buehler, M., Koditschek, D. E., 2001. RHex: A Simple and Highly Mobile Robot. International Journal of Robotics Research 20 (7), 616–631.

Seok, S., Wang, A., Chuah, M. Y., Hyun, D. J., Lee, J., Otten, D. M., Lang, J. H., Kim, S., 2015. Design principles for energy-efficient legged locomotion and implementation on the MIT Cheetah robot. IEEE/ASME Transactions on Mechatronics 20 (3), 1117–1129.

Seok, S., Wang, A., Chuah, M. Y., Otten, D., Lang, J., Kim, S., 2013. Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot. In: Proceedings - IEEE International Conference on Robotics and Automation. IEEE, pp. 3307–3312.

Sugiyama, Y., Hirai, S., 2004. Crawling and jumping of deformable soft robot. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 4 (c), 3276–3281.

Thuruthel, T. G., Falotico, E., Renda, F., Laschi, C., 2019. Model Based Reinforcement Learning for Closed Loop Dynamic Control of Soft Robotic Manipulators. IEEE Transactions on Robotics 35, 124–134.

Yang, Y., Caluwaerts, K., Iscen, A., Zhang, T., Tan, J., Sindhwani, V., 2019. Data efficient reinforcement learning for legged robots. arXiv (CoRL), 1–10.