Evaluating the Optimality of Reinforcement Learning Using Input Shaping

Albert Author[1] and Bernard D. Researcher[2]

***Abstract*— Legged mobile robotic systems are often deployed because of their many advantages over wheeled or tracked systems. They have been shown to be able to traverse more treacherous terrain deploying jumping-like techniques to overcome otherwise not traversable landscapes. It is difficult however to define robust and efficient control strategies for these types of systems using traditional methods however. Using input shaping techniques, it has been shown that optimal jumping strategies can be defined for flexible systems. In this work, input shaping is used to validate if reinforcement learning can find similar optimal jumping techniques using a monopode jumping system.**

I. INTRODUCTION

A. Legged Locomotion

Using legged mobile systems as compared to wheeled or tracked systems is often advantageous in terms of system capability. These systems have been used for their ability to navigate harsh, uneven and unpredictable terrain []. Defining control strategies for these systems is challenging however particularly when flexible components are introduced into the morphology. Many techniques exist to define locomotion strategies still, many of which are considered traditional []. Modern machine learning methods such as reinforcement learning have also shown promise in their ability to define proper control strategies.

B. Reinforcement Learning

Reinforcement learning, a form of machine learning, has been shown to be a viable method for defining controllers for many different types of systems []. In particular, the method has proven useful for learning complex control inputs for robotics systems /cite. Though many different types of algorithms have been evaluated, the ones that are most prevalent in regards to the ability to consistently learn without strenuous hypermeter tuning are actor-critic and gradient decent type algorithms.

The challenge of tuning an algorithm to find control strategies is not negated all together though, particularly when training a robust controller expected to perform with unexpected physical changes to the system. Training utilizing domain randomization has been shown to improve controller performance where system mechanical parameters may experience inaccuracies []. In addition, this technique has been shown to be useful when performing the simulation-to-real process []. In this work it is of interest to discover how the utilization of domain randomization changes the controller's output when changing system parameters.

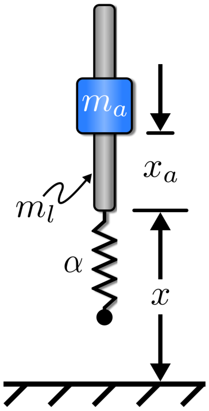

Fig. 1. Pogo-stick System

TABLE I
POGO-STICK MODEL PARAMETERS

Model Parameter	**Value**
Mass of Leg, m_l	0.175 kg
Mass of Actuator, m_a	1.003 kg
Spring Constant, $\alpha_{nominal}$	5760 N/m
Natural Frequency, ω_n	$\sqrt{\frac{\alpha}{m_l+m_a}}$
Damping Ratio, $\zeta_{nominal}$	1e-2 $\frac{\text{N}}{\text{m/s}}$
Gravity, g	9.81 m/s^2
Actuator Stroke, $(x_a)_{\text{max}}$	0.008 m
Max. Actuator Velocity, $(\dot{x}_a)_{\text{max}}$	1.0 m/s
Max. Actuator Acceleration, $(\ddot{x}_a)_{\text{max}}$	10.0 m/s^2

Of the different algorithms currently considered cutting-edge in research, Twin Delayed Deep Deterministic Policy Gradient (TD3) is one which has shown its ability for robotics control systems []. This algorithm, being built from the popular Deep Deterministic Policy Gradient (DDPG) [], and is of the actor-critic type. TD3 was the algorithm of choice and was used to train all the controllers in this work. Details regarding the specifics behind the algorithm and the training parameters are highlighted in the **Appendix**.

C. Input Shaping

Some background information on input shaping. Some words from DVs paper.

*This work was not supported by any organization

[1] Albert Author is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands `albert.author@papercept.net`

[2] Bernard D. Researcheris with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA `b.d.researcher@ieee.org`

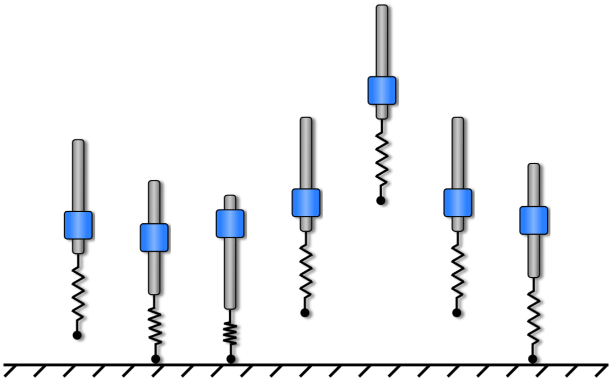

Fig. 2. Example Stutter Jump

II. ENVIRONMENT

Controlling the system is done so by accelerating the mass of the actuator m_a along the mass of the rod m_l. A nonlinear spring with constant α is modeled to represent system flexibility along with a damper (not shown in figure). The variables x and x_a represent the rod's position from zero and the actuator's position along the rod, respectively.

The equations which govern system behavior in simulation are:

$$\ddot{x} = \frac{\gamma}{m_t}\left(\alpha\, x + \beta\, x^3 + c\,\dot{x}\right) - \frac{m_a}{m_t}\,\ddot{x}_a - g \qquad (1)$$

where $\ddot{x}$, $\dot{x}$ and x are the rod's acceleration, velocity and position, respectively; $\ddot{x}$ represents the actuator acceleration and system input. The system's total mass along with the actuator's mass are represented by m_t and m_a, respectively. Constants c and a represent the damping ratio and spring constant, and β is set to $1e8$. Constant γ is in place to prevent the system's spring and damper from acting when the system is not in contact with the ground and can be expressed as:

$$\gamma = \begin{cases} -1, & x \leq 0 \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

The spring compression limit, or the systems position in the negative x direction, is limited to 0.008m. Additionally, the system is confined to move only vertically in regards to Fig. **??** so that controlling ballance is not required.

III. METHOD

A. Environment Setup

One hundred controllers were trained for each case discussed below, each with a different network initialization based on a set of seeds. This was to ensure the methods were reliable and repeatable. The network seeds used in this work can be found in the **Appendix**. The controllers were trained to jump in a manner matching what is seen in Fig. **??**. This type of jump has been called a stutter jump and can be described as the system jumping firstly to gain energy and secondly to gain height.

The state space and action space which the TD3 algorithm was provided was:

$$\mathcal{S} = [x_{a_t}, \dot{x}_{a_t}, x_t, \dot{x}_t] \qquad (3)$$

$$\mathcal{A} = [\ddot{x}_{a_t}] \qquad (4)$$

where x_{a_t} and $\dot{x}_{a_t}$ are the actuator's position and velocity, respectively, and x_t and $\dot{x}_t$ are the rod's position and velocity, respectively. $\ddot{x}_{a_t}$ is the actuator's acceleration and the system input.

Table of System Initial Position

At the start of each training episode, the system was initialized according to states provided in Tab. **??**. The rod's position was set accounting for spring compression due to system mass so that the system was at rest. The actuator was initialized randomly along its full length stroke. See Section. **??** and Section. **??** for further details on the mechanical parameters changed during simulation initialization.

A training episode was considered completed if one of two parameters where met. Either the system completed a stutter jump, where the rod's position became greater than zero and then returned to zero twice, or the episode reached the maximum number of steps. In this work, the maximum number of steps was set to 500 which represented a 5 second simulation.

B. Non Robust

The controllers trained in a non-robust fashion were done so without deploying domain randomization. Therefore the system was initialized with the same set of mechanical parameters for each episode. Those parameters are highlighted in Tab. **??**. The rewards passed to the TD3 algorithm were design to test two learning scenarios: learning to jump as high as possible and learning to jump as high as possible efficiently. The rewards used were:

$$R_{height} = x_t \qquad (5)$$

$$R_{efficiency} = \frac{x_t}{\sum P_t} \qquad (6)$$

where, x_t and P_t were the rod's height and the system't power used at each time step, respectively. Power usage in this work was defined mechanically as the product of the actuator's mass, acceleration and velocity.

- Deploy TD3 (maybe PPO as well) to train agents to
- Jump to maximum height
- Jump to specified height
- Jump to max/specified height efficiently

C. Robust

The controllers trained to be able to combat changes in system parameters were trained employing domain randomization. The spring constant, k, was randomly initialized at the start of each episode such that the natural frequency of the system could change by $\pm 30\%$. Additionally, the damping ratio, ζ, was randomly set between the ranges of 0.00 and 0.01, where the expectation was lower ζ values would lead to better performance both in height and efficiency. The rewards used to train the robust controllers were the same as the non-robust controllers and can be seen in Eq. **??** and Eq. **??**.

- Deploy TD3 (maybe PPO as well) to train agents to
- Jump to maximum height

- Jump to specified height
- Jump to max/specified height efficiently

D. Input Shaping

How were the commands generated?

- Jump to maximum height
- Jump to specified height
- Jump to specified height efficiently
- Perform different jump types
- Single jump
- Stutter jump

IV. Results

A. Controller Trained to be Non-Robust

- Figure: Input of all three methods
- Figure: Jumping Height of all three methods
- Figure: Power usage scatter chart of all three methods

B. Controller Trained to be Robust

- Figure: Input of all three methods
- Figure: Power usage scatter chart of all three methods
- Figure: Jumping Height of all three methods

V. Conclusion

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.

Appendix

Twin Delayed Deep Deterministic Policy Gradient (TD3)

TD3 is an actor-critic, off-policy reinforcement learning algorithm wherein the controller is represented by the actor which is formally know as the policy π_ϕ. In the deep learning case, the actor is a neural network. The actor takes actions, $\mathcal{A}$, at each time step during simulation which are the system's inputs. The critic is represented by the estimated expected return of taking action $\mathcal{A}$ in state $\mathcal{S}$ and following the policy π from then after. The critic is also a neural network, and it is updated according to the 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. The hyperameters used for the experiments in this work can be found in Table **??**.

Table of Hyperparameters

Network Initialization Seeds

When training a neural network from scratch, once the network design is defined, the node weights must be initialized. In this work, 100 different controllers were trained each with a different network initialization based on a set of seeds. The seeds are listed in Tab. **??**.

Table of Network Seeds

Acknowledgment

The authors would like to thank the Louisiana Crawfish Promotion and Research Board for their support of this work.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

References

[1] G. O. Young, 'Synthetic structure of industrial plastics (Book style with paper title and editor),' in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15'64.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123'135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, 'An approach to graphs of linear forms (Unpublished work style),' unpublished.

[5] E. H. Miller, 'A note on reflector arrays (Periodical style'Accepted for publication),' IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, 'Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style'Submitted for publication),' IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, 'Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style),' IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740'741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, 'Infrared navigation'Part I: An assessment of feasibility (Periodical style),' IEEE Trans. Electron Devices, vol. ED-11, pp. 34'39, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, 'A clustering technique for digital communications channel equalization using radial basis function networks,' IEEE Trans. Neural Networks, vol. 4, pp. 570'578, July 1993.

[12] R. W. Lucky, 'Automatic equalization for digital communication,' Bell Syst. Tech. J., vol. 44, no. 4, pp. 547'588, Apr. 1965.

[13] S. P. Bingulac, 'On the compatibility of adaptive controllers (Published Conference Proceedings style),' in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 8'16.

[14] G. R. Faulhaber, 'Design of service systems with priority reservation,' in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 3'8.

[15] W. D. Doyle, 'Magnetization reversal in films with biaxial anisotropy,' in 1987 Proc. INTERMAG Conf., pp. 2.2-1'2.2-6.

[16] G. W. Juette and L. E. Zeffanella, 'Radio noise currents n short sections on bundle conductors (Presented Conference Paper style),' presented at the IEEE Summer power Meeting, Dallas, TX, June 22'27, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, 'An analysis of surface-detected EMG as an amplitude-modulated noise,' presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, 'Narrow-band analyzer (Thesis or Dissertation style),' Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, 'Parametric study of thermal and chemical nonequilibrium nozzle flow,' M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, 'Nonlinear resonant circuit devices (Patent style),' U.S. Patent 3 624 12, July 16, 1990.