Let's keep working to make sure this is different than MECC, even in background / intro sections

**Proceedings of IMECE International Mechanical Engineering Congress & Exposition**
**IMECE 2021**
**November 01-November 05, 2021, Virtual Conference, Online, USA**

# IMECE2021-70012

Are you using evolution?

# MECHANICAL EVOLUTION OF A FLEXIBLE-LEGGED JUMPING SYSTEM WITH REINFORCEMENT LEARNING

I think we might want to be more explicit / focused here.

**Andrew Albright**
Department of Mechanical Engineering
University of Louisiana at Lafayette
Lafayette, Louisiana 70504
andrew.albright1@louisiana.edu

**Joshua Vaughan**
Department of Mechanical Engineering
University of Louisiana at Lafayette
Lafayette, Louisiana, 70504
joshua.vaughan@louisiana.edu

## ABSTRACT

*Legged systems have many advantages when compared to their wheeled counterparts. For example, they can more easily navigate extreme, uneven terrain. However, there are disadvantages as well, including dramatically lower energy efficiency. ~~To mitigate this performance issue,~~ research has shown that using flexible components not only increases efficiency but also performance. Flexible systems are highly nonlinear, however, and are difficult to develop controllers for using traditional methods. Because of the difficulties encountered in modeling flexible systems, control methods such as reinforcement learning can be used ~~which use neural networks to represent the nonlinear model and controller of the systems~~. Instead of tasking a reinforcement learning algorithm with learning or controlling the system, it can instead be tasked with learning mechanical parameters of a system to match a control input.*

a little awkward

Key results?

## 1 INTRODUCTION

The use of flexible components within legged locomotive systems has proved useful for both reducing power consumption and increasing performance [1–3]. However, designing controllers for these systems is difficult as the flexibility of the system generates ~~highly~~ nonlinear models. As such, employing series-elastic-actuators (SEA) instead of flexible links is an attractive and popular solution, ~~where~~ since the models of the systems become more manageable [2,4,5]. Still, the use of SEAs do not represent the full capability of flexible systems. ~~and~~ As a result, ~~such~~ other methods ~~are used which~~ that use flexible tendon-like materials meant to emulate more organic designs have been proposed [6]. Even these, however, are not representative of fully flexibly links like the ones used in [7], which...

I really don't like the "... in [#]" reference style. Just say what's in the paper. The citation itself says the "in" part.

Control methods have been developed ~~which are designed and~~ that work well for flexible systems like the ones mentioned [8,9]. Still, these are challenging to develop as the systems become more complex. As such, work has been done which uses neural networks and methods such as reinforcement learning to develop controllers for flexible systems [10,11]. ~~Specifically,~~ For example, ~~[12] showed the use~~ of RL ~~for training~~ has led to higher-performing control strategies for flexible-legged locomotive systems ~~compared to rigid legged systems.~~ [13]

need some transition / explanation why, specifically, are they difficult. (beyond nonlin)

In addition to the work done using RL to develop controllers for flexible systems, Work has been completed which shows that techniques such as RL can be used to concurrently design the mechanical aspects of a system and a controller to match said system [13]. These techniques have even been used to define mechanical parameters and control strategies where the resulting controller and hardware were deployed in a sim-to-real process, validating the usability of the technique [14]. Using this technique for legged-locomotion has also been studied, but ~~is~~ thus far has been limited to the case of rigid systems [15].

As such, this paper starts the discovery of using RL for concurrent design of flexible-legged locomotive systems. A simplified flexible jumping system is used where, for the initial work, the control input is held fixed so that the RL algorithm is tasked with only learning optimized mechanical parameters. The rest of the paper is ~~broken down~~ organized such that in the next section, similar work will be discussed. In Section 3, a brief explanation of the RL algorithm used will be presented. Following that, in Sec-

Stay formal

tions 4 and 5, the environment structure will be explained and the experiments will be broken down. Then, in Section 6, results will will be displayed. Finally, in Section 7, a conclusion will be made with implications based on the results.

Conclusion doesn't make conclusions. It just summarizes what was provided in the paper.

## 2 Related Work

### 2.1 Flexible Locomotive Systems

The use of flexible components within robotics systems has shown improvements in performance measures, specifically ones were power consumption are important [3]. Specifically, flexible systems have shown advantages in locomotion applications where crawling and jumping are employed [1]. Previous work has show that the use of flexible components ~~within~~ in the legs of legged locomotion systems increase performance while decreasing power consumption [7]. Contrasting the use of flexible links within legged systems, much work has been done showing the uses of series-elastic-actuators for locomotive systems [5]. Much of this work involving human interaction with robotic systems where rigidity is not always ideal [4]. The studies of flexible systems are challenging, however, as the models which represent them are often highly nonlinear and therefore difficult to develop control systems for. As such, there is a need for solutions which can be deployed for developing controllers for these nonlinear systems.

awkward

awkward

fragment

### 2.2 Controlling Flexile Systems Using RL

too vague. what kind of tasks?

Control methods developed for flexible systems have been shown to be effective for certain tasks [8, 16]. However, because of system model nonlinearities, using traditional methods to control flexible systems is not always advantageous. As such, research has been completed that shows the viability of using reinforcement learning to develop the control strategies for flexible systems [10]. As is with most reinforcement learning applications, different techniques are applied depending on the application. The technique has been used in simple planar cases, where it is compared to a PD control strategy for vibration suppression and proves to be a higher performing method [17]. Additionally, it has also been shown to be effective at defining control methods for flexible legged locomotion ~~where [12] used~~ Deep Deterministic Policy Gradient [18] was used to train running strategies for a flexible legged quadruped [12]. Much of the research is based in simulation, however, and often the controllers are not deployed in a sim-to-real fashion which leads to the question on weather or not these are practically useful techniques.

redundant. You've said this a few times already.

Proofread

### 2.3 Concurrent Design

Defining an optimal controller for a system can be challenging due to things like mechanical and electrical design limits. This is especially true when the system is flexible and the model is nonlinear. A solution to this challenge is to concurrently design a system with the controllers so that the two are jointly optimized ~~and designed for each other~~. Recent work has been completed which used advanced methods such as evolutionary strategies to define robot design parameters [19]. Additionally, reinforcement learning has been shown to be a viable solution for concurrent design of locomotive systems in 2D simulation locomotive [13]. This is further proved to be a viable method by demonstrating more complex morphology modifications in 3D reaching and locomotive tasks [15]. However, these techniques have not been applied to flexible type systems for locomotive tasks.

need some transition here

?

Did it really prove?

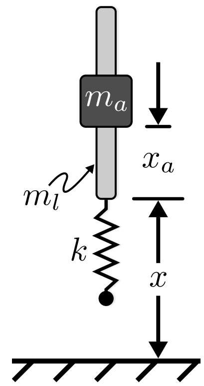


**FIGURE 1.** Pogo-stick System

**TABLE 1.** POGO-STICK MODEL PARAMETERS

| Model Parameter | Value |
|---|---|
| Mass of Leg | 0.175 kg |
| Mass of Actuator | 1.003 kg |
| Natural Frequency | 11.13 Hz |
| Spring Constant, $k$ | 200000 N/m |
| Actuator Stroke, $(x_a)$max | 25.0 m |
| Actuator Velocity, $(\dot{x}_a)$max | 1.0 m/s |
| Actuator Acceleration, $(\ddot{x}_a)$max | 10.0 m/s$^2$ |

This is varying right? Is this the nominal value?

Seems just a little too large.

## 3 Pogo-stick Model Description

The pogo-stick model show in Figure 1 has been shown to be useful as a representation of several different running and jumping gaits [20]. As such, it is used in this work to demonstrate the ability of reinforcement learning for the initial steps of concurrent design. The models parameters are summarized in Table 1.

The variable $m_a$ represents the mass of the actuator, which moves along the rod with mass $m_l$. A non-linear spring with constant $k$ is used as the representation of flexibility. A damper (not shown in Figure 1) is parallel to the spring. Variables $x$ and $x_a$ represent the systems vertical position with respect to the ground and the actuators position along the rod, respectively. The system is additionally constrained such that it only moves vertically so the reinforcement agent is not required to balance the system.

The equation of motion describing the system ~~are~~ is:

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

where $x$ and $x_a$ are position and velocity of the rod respectively, the acceleration of the actuator, $x_a$, is the control input, and $m_t$ is the mass of the complete system. Ground contact determines the value of $\alpha$, so that the spring and damper do not supply force while the leg is airborne:

$$\alpha = \begin{cases} -1, & x \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

## 4 Reinforcement Learning

stay formal & precise with your language

Is it or is it not?

$$$

use math mode for variables in text

The algorithm used for this work is Twin Delayed Deep Deterministic Policy Gradient (TD3) [21]. This is an actor-critic algorithm wherein there exists two main neural networks and a set of twin trailing networks. The first main network is the actor, which is essentially the control policy. This network takes in the systems state, S, and outputs the action (control input), A, based on the state. The critic is an estimator of the value of being in a state and is used to determine the difference between expected and estimated value to update the actor network during training. It takes in the systems state S, and outputs the expected future reward from being in that state, R. The twin trailing networks are used to find the temporal difference error against the critic network which is used to update the critic network.

stopped here

## 5 Experiments
### 5.1 Jumping Types

Two different jump types were used to evaluate if reinforcement learning can generate different designs for different tasks. The first jump type can be seen in Fig 2. Here, the goal of the agent was to learn a spring constant such that given the control input from [22], the system would jump as high as possible. The second jump type can be seen in Fig. 3, where the goal of the agent was to learn a spring constant that would allow the agent to jump as high as possible in two jumps given a similar input from [22].

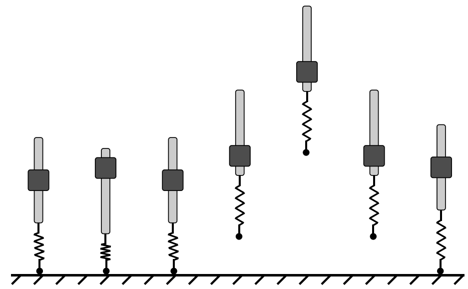

**FIGURE 2.** Example Single Jump

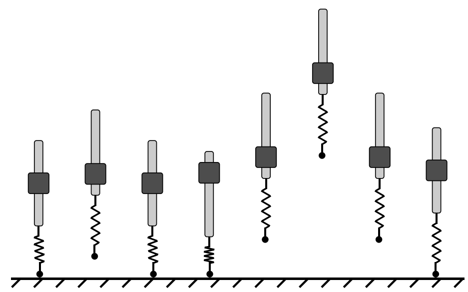

**FIGURE 3.** Example Stutter Jump

### 5.2 Finding Optimal Spring Constant

A reinforcement learning environment was modeled according to OpenAI Gyms standards [23]. Because the control input is set and not part of the agents actions, at each time step during training, the agents action is instead selecting a spring constant which is then evaluated in simulation. To be continued...

## 6 Results

Results.

## 7 Conclusion

Conclusion.

## ACKNOWLEDGMENT

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

## REFERENCES


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

[2] Buondonno, G., Carpentier, J., Saurel, G., Mansard, N., De Luca, A., and Laumond, J. P., 2017. "Actuator design of compliant walkers via optimal control". *IEEE International Conference on Intelligent Robots and Systems,* ***2017-Septe***, pp. 705–711.

[3] Hurst, J., 2008. "The Role and Implementation of Compliance in Legged Locomotion". *The International Journal of Robotics Research,* ***25***(4), p. 110.

[4] Zhang, T., and Huang, H., 2019. "Design and Control of a Series Elastic Actuator with Clutch for Hip Exoskeleton for Precise Assistive Magnitude and Timing Control and Improved Mechanical Safety". *IEEE/ASME Transactions on Mechatronics,* ***24***(5), pp. 2215–2226.

[5] Pratt, G. A., and Williamson, M. M., 1995. "Series elastic actuators". *IEEE International Conference on Intelligent Robots and Systems,* ***1***, pp. 399–406.

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

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

[8] Luo, Z.-h., 1993. "Direct Strain Feedback Control of Flexible Robot Arms: New Theoretical and Experimental Results".

[9] Modeling, A. M., 2003. "Gain Adaptive Nonlinear Feedback Control of Flexible SCARA / Cartesian Robots". pp. 1423–1428.

[10] Bhagat, S., Banerjee, H., Tse, Z. T. H., and Ren, H., 2019. "Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges". *Robotics,* ***8***(1), pp. 1–36.

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

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

[13] Ha, D., 2019. "Reinforcement learning for improving agent design". *Artificial Life,* ***25***(4), pp. 352–365.

[14] Chen, T., He, Z., and Ciocarlie, M., 2020. "Hardware as Policy: Mechanical and computational co-optimization using deep reinforcement learning". *arXiv*(CoRL).

[15] Schaff, C., Yunis, D., Chakrabarti, A., and Walter, M. R., 2019. "Jointly learning to construct and control agents using deep reinforcement learning". *Proceedings - IEEE International Conference on Robotics and Automation,* ***2019-May***, pp. 9798–9805.

[16] Ahmadi, M., and 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), pp. 96–104.

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

[18] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., 2016. "Continuous control with deep reinforcement learning". *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

[19] Wang, T., Zhou, Y., Fidler, S., and Ba, J., 2019. "Neural graph evolution: Towards efficient automatic robot design". *arXiv*, pp. 1–17.

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

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

[22] Vaughan, J., 2013. "Jumping Commands For Flexible-Legged Robots".

[23] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W., 2016. "OpenAI Gym". pp. 1–4.