# Pouncing Pogo

Yu Jun Shen
MENG 443 Final Report

December 2022

## 1 Introduction

In this project, the hopping behavior of a bouncing stick robot is simulated with a Spring Loaded Inverted Pendulum (SLIP) model. The relevant robot locomotion parameters are the spring constant, incident angle from the ground, mass and leg length. We develop a physics-based simulation of the contact dynamics in Python and explore the effects of robot parameters on the longevity and distance travelled.

This simple "pouncing" model has real-world similarities with legged animal locomotion and airbag landing systems used by early Mars rovers [2, 4]. In SLIP, the spring behaves like a muscle that absorbs and releases kinetic energy upon impact with the ground.

## 2 Theory

### 2.1 Pogo architecture

Pogo is a one degree of freedom robot that moves by compressing and releasing a spring (Fig. 1). The main control input is the incident angle $\theta_I$; the release angle will be determined by model physics. The spring is characterized by Hooke's

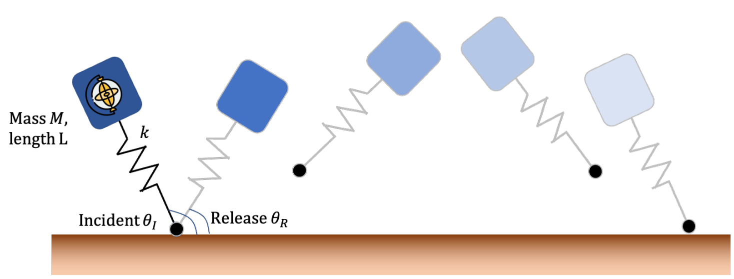


Figure 1: Illustration of proposed Pogo robot; $M$ is the tip (flywheel) mass, $k$ the spring constant, $\theta_I$ is the incident angle.

Law with a spring constant $k$. Since Pogo must swing forward while airborne, imagine a flywheel mechanism housed in the top to manage the angular momentum. The mass, length and spring constants are inspired by biological examples (bigger animals have bigger muscles) [1]. Pogo's motion can be separated into a sequence of "flight" and "contact" phases.

During Pogo's flight phase, it follows a ballistic trajectory under the influence of gravity alone. Following the notation in [2] (with $g$ as gravitational acceleration), the equations of motion for the mass are:

$$\ddot{x}(t) = 0 \tag{1}$$

$$\ddot{y}(t) = -g \tag{2}$$

From literature, the highest point reached by the top mass during flight is called the apex and defined when the vertical velocity is zero [2, 3]. Upon impact with the ground, the spring compresses until it reaches a minimum compression point, then starts expanding. Take-off occurs when the spring reaches its equilibrium point on the expansion cycle.

Relative to the origin (pivot) of the ground contact point, Pogo's tip mass is an inverted pendulum with coordinates given by:

$$x(t) = l(t)\cos\theta(t) \tag{3}$$

$$y(t) = l(t)\sin\theta(t) \tag{4}$$

From the kinetic, gravitational potential and spring elastic potential energies the Lagrangian is set up as:

$$L = \frac{M}{2}(l^2\dot{\theta}^2 + \dot{l}^2) - Mgl\sin\theta - \frac{k}{2}(l_k - l_{k,0})^2 \tag{5}$$

Equations of motion are obtained by applying the Euler-Lagrange equation to coordinates $l, \theta$ (instead of x, y) during contact:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q} \tag{6}$$

$$\Rightarrow \ddot{l} = -\frac{k}{M}(l_k - l_{k,0}) - g\sin(\theta) + l\dot{\theta}^2 \tag{7}$$

$$\Rightarrow \ddot{\theta} = -2\frac{\dot{l}}{l}\dot{\theta} - \frac{g}{l}\cos\theta \tag{8}$$

Eq. 7 and 8 are numerically simulated to solve for the contact dynamics. The acceleration of the total leg length equals that of the spring i.e. $\ddot{l} = \ddot{l_k}$. Spring length parameter is set as $l_k \leq l$ at the start of the simulation. At every timestep, the translational position, velocity and rotation values ($\theta(t)$ and $\dot{\theta}(t)$) are stored as Pogo's state vector.

## 2.2 Bounce termination rules

As sketched in Fig. 1, Pogo is launched with a fixed initial velocity and allowed to bounce repeatedly. Without a driving energy source, energy is fully conserved. The maximum distance travelled and timesteps before falling are metrics of performance. There are two termination conditions in the program:

- **Fall back:** During ground contact, if Pogo's angular velocity $\dot{\theta}(t) \geq 0$ while $\theta(t) > 90°$, it would have lost forward momentum while pivoting. A continuous torque from the top mass will then cause it to fall backwards.
- **Trip:** When airborne and moving upwards, if Pogo's foot height falls to zero (swinging like a pendulum), it would trip over having not jumped high enough.

$$y_{foot} = y_M(t) - L * \sin(\theta(t)) \leq 0 \tag{9}$$

The airborne phase only considers translational parabolic motion of the top mass (also the center of mass, assuming negligible spring and foot mass), subject to trip-over. This model ignores air resistance and supposes the flywheel enables Pogo to return to its initial $\theta_I$ incident angle.

The state transitions to contact mode after Pogo's foot hits the ground. The spring compresses to arrest the vertical velocity component while the horizontal inertia forces the top mass forward. We assume sufficient friction at the foot (point of contact pivot) to keep Pogo from sliding.

## 2.3 Hopping simulation framework

Pogo is instantiated as an object in Python with various methods such as flight, contact dynamics and contact/liftoff checking. Equations of motion 7 and 8 are time-stepped forward using Newton's method.

A finite state machine is implemented to toggle the flight and contact states. Since the two states are easier to compute in Cartesian and polar coordiante geometry respectively, we add a conversion function that executes every ground contact and liftoff moment. This is the first-contact function in code.

An outline of the pseudocode for the classic SLIP model is given below:

```
Initialize Pogo object, start state
while time < T and Pogo alive:
    while Pogo is airborne:
        Forward propagate ballistic path per Eq. 1 and 2
        Check for contact (foot height)
        Increment distance travelled
        Increment time
    while Pogo is in contact:
        Forward propagate stance motion per Eq. 7 and 8
        Compress spring and check extension
```

Increment time
If spring is back to original length:
Pogo goes airborne
**return** distance traveled

# 3 Results

## 3.1 Example simulation run

The results of an individual Pogo hop are presented in Fig. 2. The leg length and $\theta$ angle oscillate over the contact events, showing a distinctive oscillation. The robot state of $(k, l, m, x, \dot{x}, y, \dot{y}, \theta, \dot{\theta})$ were initialized as (200, 0.5, 0.3, 0, 0.2, 0.5, -0.3, 2, 0) respectively.

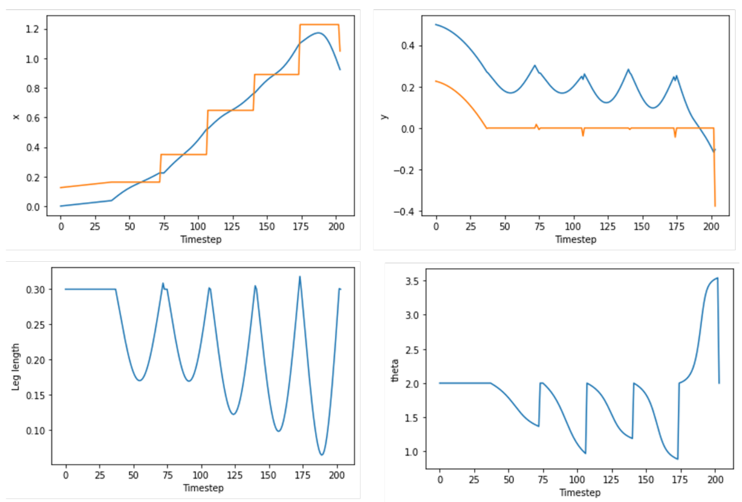


Figure 2: Generated statistics of a Pogo bounce at $(k, l, m, x, \dot{x}, y, \dot{y}, \theta, \dot{\theta})$ of (200, 0.5, 0.3, 0, 0.2, 0.5, -0.3, 2, 0)

The corresponding trajectory in this example is shown in Fig. 3.

## 3.2 Effect of Pogo length and spring parameters

A systematic investigation of the effects of leg length and spring constant was conducted to find the optimal characteristics of Pogo, in terms of both maximum survived timesteps (Fig. 4) and maximum distance travelled (Fig. 5). A short leg length and strong spring gave the best performance.

Finally, the effect of incident angle was investigated. The maximum timesteps alive are plotted in Fig. 6 and the distance travelled in Fig. 7. A slight incidence angle was the best to maximize survival time and distance travelled.

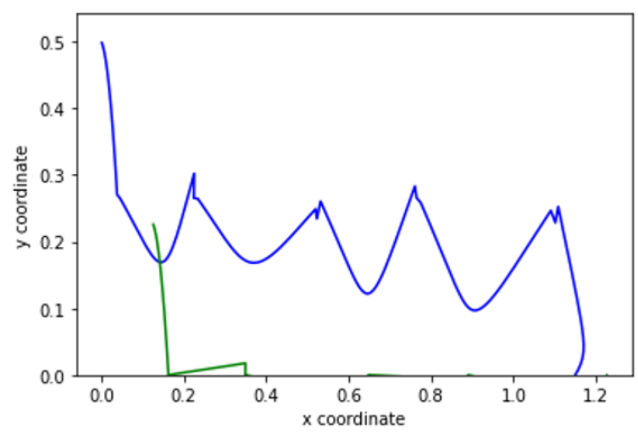


Figure 3: Generated path of a Pogo bounce at $(k, l, m, x, \dot{x}, y, \dot{y}, \theta, \dot{\theta})$ of (200, 0.5, 0.3, 0, 0.2, 0.5, -0.3, 2, 0)

# 4 Discussion

Intuitively, the leg angle cannot be too close to $\pi$ to prevent tripping. If it was too close to $\pi/2$, then forward motion is impeded. Thus the simulation optimum of around 2 radians is a sensible result.

Given the relatively small search space, a systematic investigation of the different length, spring constant and incident angle permutations was possible instead of a genetic algorithm initially proposed. The search nonetheless revealed that short, stiff legged robots performed better in time and distance travelled.

For future work, active control can be studied with a driven spring. Its frequency, phase and amplitude will be control parameters for optimization.

# 5 Conclusion

Simulations of the hopping behavior of a bouncing stick robot in Python show that a short robot leg, strong spring and slight incidence angle maximize the distance traveled and timesteps alive before falling. The robot was modelled as a Spring Loaded Inverted Pendulum (SLIP) and contact dynamics analyzed using the Lagrangian method. The relevant robot locomotion parameters are the spring constant, incident angle from the ground, mass and leg length. Simulation results are consistent with physics and shed light on legged locomotion.

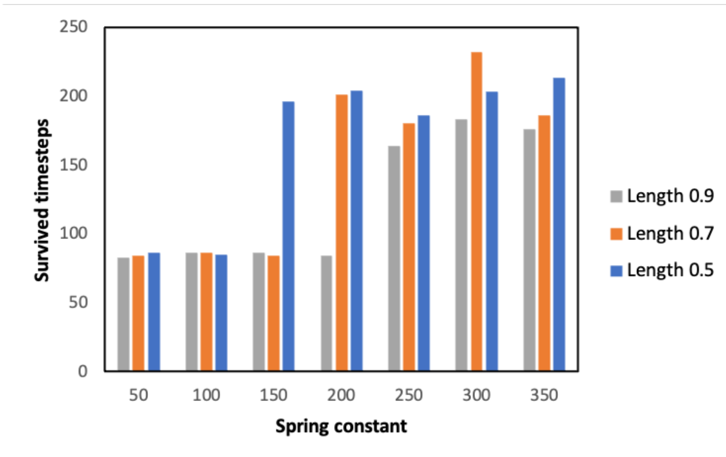


Figure 4: Maximum survived timesteps of Pogo simulations at different leg lengths and spring constants; the other parameters of $(m, x, \dot{x}, y, \dot{y}, \theta, \dot{\theta})$ were kept fixed at (0.3, 0, 0.2, 0.5, -0.3, 2, 0)

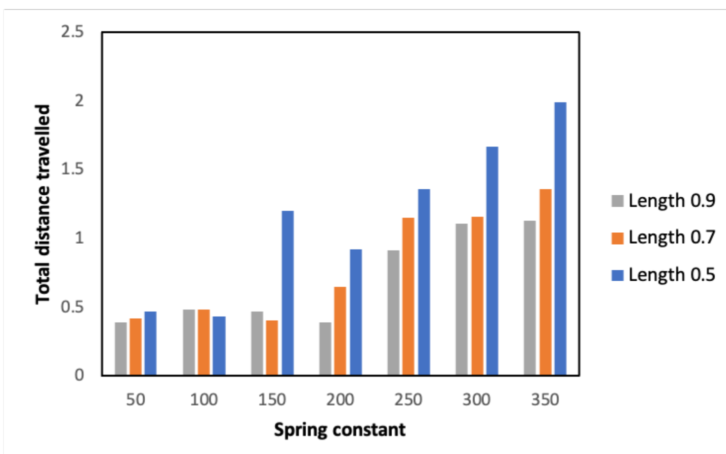


Figure 5: Maximum distance travelled in Pogo simulations at different leg lengths and spring constants; the other parameters of $(m, x, \dot{x}, y, \dot{y}, \theta, \dot{\theta})$ were kept fixed at (0.3, 0, 0.2, 0.5, -0.3, 2, 0)

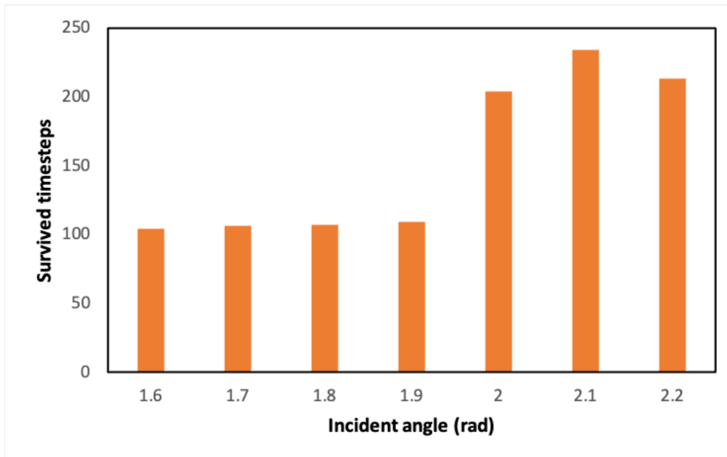


Figure 6: Maximum survived timesteps of Pogo simulations at different incident angles; the other parameters of $(k, l, m, x, \dot{x}, y, \dot{y}, \dot{\theta})$ were kept fixed at (200, 0.5, 0.3, 0, 0.2, 0.5, -0.3, 0)

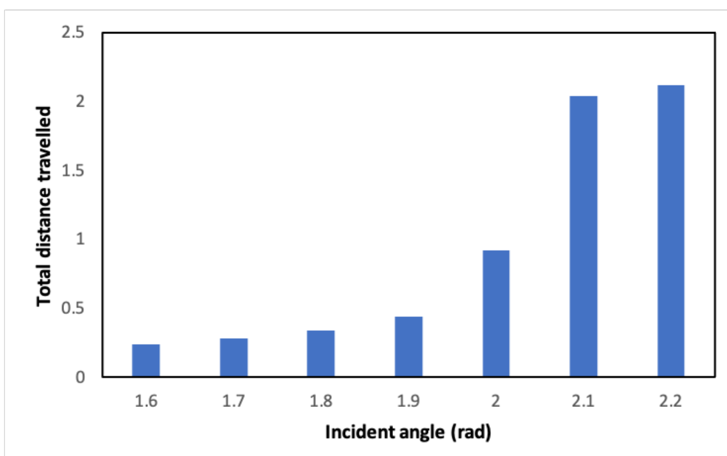


Figure 7: Maximum distance travelled in Pogo simulations at different incident angles; the other parameters of $(k, l, m, x, \dot{x}, y, \dot{y}, \dot{\theta})$ were kept fixed at (200, 0.5, 0.3, 0, 0.2, 0.5, -0.3, 0)

## References

[1] Ha D. (2019) Reinforcement learning for improving agent design. Artif Life; 25 (4): 352–365. doi: https://doi.org/10.1162/artl_a_00301

[2] Piovan G., Byl K. (2016) Approximation and control of the SLIP model dynamics via partial feedback linearization and two-element leg actuation strategy

[3] Walker K., Hauser H. (2018) Evolving optimal learning strategies for robust locomotion in the spring-loaded inverted pendulum model. IROS.

[4] NASA, Mars Exploration Rovers. https://mars.nasa.gov/mer/mission/overview/ Accessed 18 Oct. 2022.

[5] Geeksforgeeks. Genetic Algorithms. https://www.geeksforgeeks.org/genetic-algorithms/ Accessed 18 Oct. 2022.