Comparing Delay and Reflected Inertia for an Actuated Spring Loaded Inverted Pendulum Model

By
Kurt Reinschmidt

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Mechanical Engineering
(Honors Scholar)

Presented May 29, 2019
Commencement June 2019

AN ABSTRACT OF THE THESIS OF

Kurt Reinschmidt for the degree of Honors Baccalaureate of Science in Mechanical Engineering presented on May 29, 2019. Title: Comparing Delay and Reflected Inertia for an Actuated Spring Loaded Inverted Pendulum Model

Abstract approved:

Jonathan Hurst

Legged robots, while advancing quickly, have far to go before they can recover from major disturbances safely and consistently. Reflected inertia, which affects how quickly the actuators can physically respond to an environmental disturbance, and delay, which describes the time necessary for the controller to sense, re-plan, and respond to the disturbance, both make recovery harder as they increase. It is possible that their effect on disturbance response is comparable. This thesis will investigate the effects of reflected inertia and delay on disturbance response, using trajectory optimization on an Actuated Spring Loaded Inverted Pendulum model in order to compare these effects on the energy needed to recover from a disturbance. While the results proved to be inconclusive, the mistakes of the methodology are explained and a better methodology is suggested, so that the process can be learned from and the idea can be investigated in a better way.

Key Words: SLIP, disturbance, legged, running

Corresponding e-mail address: reinschk@oregonstate.edu

Comparing Delay and Reflected Inertia for an Actuated Spring Loaded Inverted Pendulum Model

By
Kurt Reinschmidt

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Mechanical Engineering
(Honors Scholar)

Presented May 29, 2019
Commencement June 2019

Honors Baccalaureate of Science in Mechanical Engineering project of Kurt Reinschmidt presented on May 29, 2019

APPROVED:

Jonathan Hurst, Mentor, representing Mechanical Engineering

Michael Hector, Committee Member, representing Mechanical Engineering

Joseph Davidson, Committee Member, representing Mechanical Engineering

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University Honors College. My signature below authorizes release of my project to any reader upon request.

Kurt Reinschmidt, Author

Acknowledgements

This thesis has been an incredible learning opportunity for me and I would like to thank those who helped me throughout the process. To Dr. Jonathan Hurst, who helped me to understand complex subject matter that did not come easily. To Mike Hector, whose mentoring was crucial throughout the coding and the writing process. To Kevin Green, who also acted as a mentor, helping me to answer questions related to dynamics, coding, and presenting results. To Dr. Joe Davidson, who helped me to stay on track throughout the writing process. Thank you all.

Contents

1 Introduction

In order for robots to complete a variety of tasks in tandem with humans, they must be able to work in unstructured environments and navigate human spaces. Human spaces consist of many features that, to a human, are of no matter, but to a robot can present an obstacle. A few examples would be a narrow hallway, stairs (especially if of different size and slope than standard), the floor changing from wood to carpet, or some debris on the floor. A robot must be able to navigate all of these in order to work collaboratively with humans.

While there are many approaches to creating a robot capable of these tasks, a promising subset are dynamic walking and running robots. These types of robots have legs and are dynamically unstable, meaning that during parts of their gait they are falling until they catch themselves with a new step. They are promising because having legs makes it easier for them to navigate human spaces. Human spaces are designed for humans, so it makes sense to design robots that move in a similar manner. They also exhibit relatively robust performance, already being able to navigate difficult terrain and recover from some disturbances by utilizing the passive dynamics of their legs, such as the bipedal robot Cassie [1].

An important factor in the design of dynamic walking and running robots is reflected inertia. Reflected inertia is a value that describes the relationship between the torque and acceleration output by a gearbox connected to an actuator. If reflected inertia is too high, the actuators cannot quickly react to changing signals. If it is low, which is a goal of robot design, the actuators can quickly respond to a changing signal, which makes recovering from a disturbance easier. However, reflected inertia cannot be minimized without trade offs and must be balanced with the gear ratio and the output torque required.

Another factor to consider when designing a robot is delay. Delay is not as important as reflected inertia since computers allow robots to quickly process information and react to it. However, it is still necessary to create the robot controller with update rate, or how fast it can take in information and re-plan, in mind. If robots cannot re-plan quickly, the delay would increase, making it more difficult to respond to and recover from disturbances.

Compared to robots, animals have a higher delay, yet are much more robust when it comes to navigating difficult terrain. This suggests two ideas. The first is that, with improved hardware and software, it is possible in the future for robots to possess the same robustness of performance. The second idea is that, if animals have a higher delay, they might excel over robots in another area, such as their 'actuation.' The muscles of animals and humans, while not having to overcome reflected inertia as robots do, could be able to output the desired motion much more quickly than the actuators of robots or utilize passive dynamics more efficiently, which allows them to 'make up' for their delay and achieve the performance that they possess. If quick 'actuation' can help overcome high delay, the two values could affect the response of an actuator (or muscle) similarly.

This thesis investigates the effects of reflected inertia and delay on disturbance response by evaluating the energy usage of an actuator in an Actuated Spring Leg Inverted Pendulum (ASLIP) model during disturbance recovery to see if the effects are similar or comparable. First, important related concepts will be covered in the Background section, then the methods used will be described in the Methodology section. The method used resulted in inconclusive results which, along with the faults of the method, will be covered in the Results & Discussion section, but a different method that may yield better insight is described in the Conclusion.

2 Background

2.1 Reflected Inertia

When designing dynamic legged robots, an extremely important consideration is designing around reflected inertia. Reflected inertia is a value that describes how the motor inertia is amplified by the gear ratio (N) in the following equation

$$I_{reflected} = I_{motor} * N^2$$

It relates the torque and acceleration output by the gearbox through the following equation

$$T = I_{reflected} * a$$

If the reflected inertia is higher then same amount of torque will result in lower acceleration compared to a system with lower reflected inertia. Lower acceleration means it takes longer for a change in motor signal to be achieved. By minimizing reflected inertia, less torque is needed for the actuator of the robot to accelerate a load, or with the same amount of torque the robot can respond more quickly to motor commands. However, since reflected inertia varies proportionately with the square of the gear ratio, it can't just be minimized. The gear ratio must be high enough so that the desired torque can be output, so a balance must be achieved [2].

Although reflected inertia cannot be used to compare actuators with biological muscle, it is apparent from seeing animal react to disturbances that they can accelerate limbs much more quickly than actuators. Because animals perform much more robustly than robots, this thesis will assume that animals, although they don't experience reflected inertia as robots do, would have a lower reflected inertia than robots if their muscles had a comparable parameter.

2.2 Delay

Delay impacts both animals and robots and describes the time necessary to sense, transmit, re-plan an action, and send the command to the actuator. There are three sub-types of delay, transmission delay, actuation delay, and response delay, which are important to consider. Transmission delay is the delay that describes the signals being communicated and received between the high-level controller and the control box. This would include high-level commands being sent to the control box and actions completed being sent from the control box back to the high-level controller and reflected in the high-level controller's memory. Actuation delay is the delay between the control box receiving a command, processing the command, calculating the required actuator input, and sending the command to the actuator. Response delay is the delay between the actuator command being executed and the motion being updated in the control box memory [3].

For the purpose of this thesis, the three delays will be combined into simulating a delay that includes sensing the disturbance, re-planning the trajectory, and the actuator receiving the new command. When modelling the delay, the value will be swept through a spectrum, starting at 0 ms (representing an idealized robot) and going to a value larger than a typical animal delay of 100 ms [3].

2.3 Dynamic Walking & Running Robots

There are different methods of locomotion used for robotic walking and running, two of which are dynamic walking and dynamic running. Dynamic walking describes a method of movement where the mass, or body of the robot, is unstable between steps, essentially falling and catching itself every step. An example of this is the robust bipedal robot Cassie (Figure 1). Dynamic running is similar to dynamic walking, except that there is a true flight phase, where after one step the robot has enough energy to leave the ground and then fall back down, catching itself with the other leg and repeating the process. An example of dynamic running is the robot ATRIAS (Figure 2), which could also move via dynamic walking.

For robots that utilize dynamic walking and running, passive dynamics are extremely important. Passive dynamics, in the simplest sense, describes how if designed correctly, and placed on a shallow slope, a mechanism can achieve rudimentary locomotion without any active control [6]. When applying this concept to an actively controlled robot, it can be used so that the robot can respond to disturbances without active control, thereby lessening the amount of energy used by its actuators during disturbance response [7].

Although passive dynamics allows the robot to use less active control, it still needs to be able to react quickly to disturbances. Both reflected inertia and delay are important to consider in order to minimize reaction time. If there is a disturbance, the robot could need to re-plan its trajectory, and re-planning causes delay. Afterwards, once the signal is sent to the actuator; the actuator acceleration is limited by the reflected inertia, so it takes time to output the desired actuator value. If either delay or reflected inertia is too high, the robot could fall. Most current robots have little problem with delay, and reflected inertia needs much more design effort to manage.

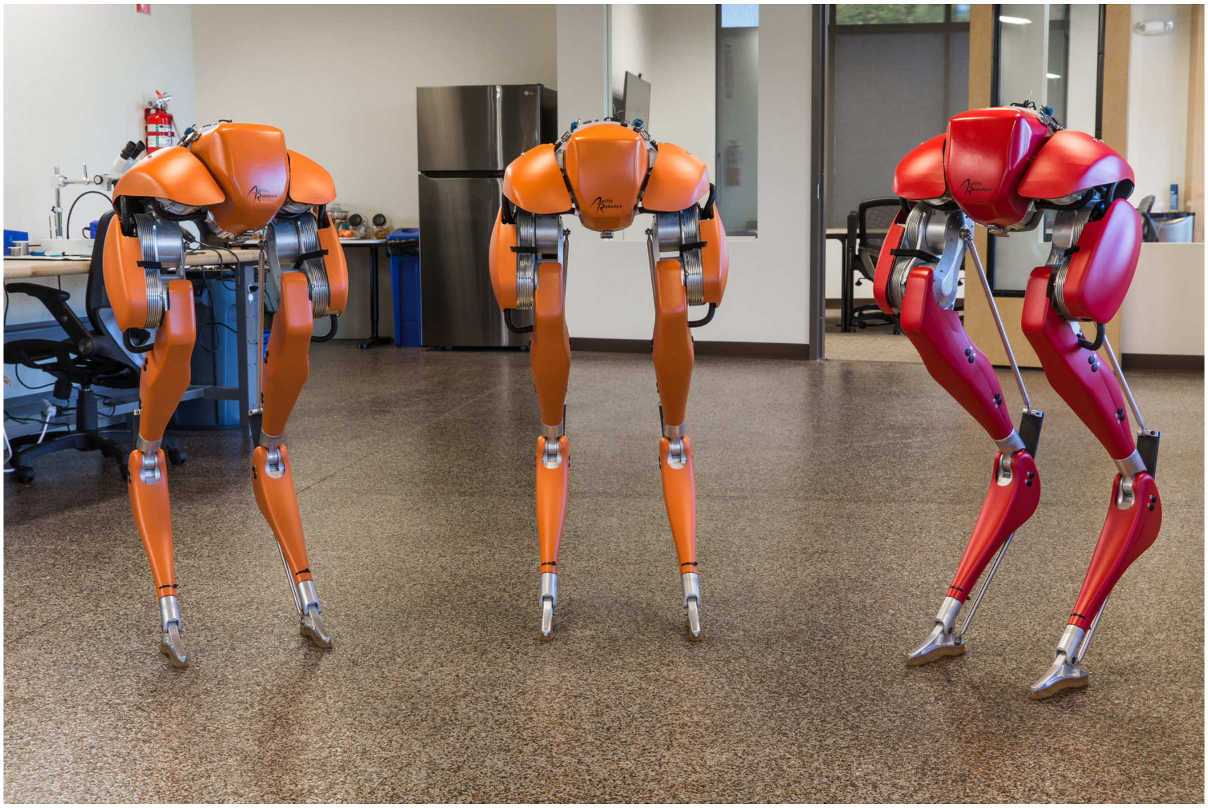

Figure 1: Cassie, a dynamic legged robot [4].

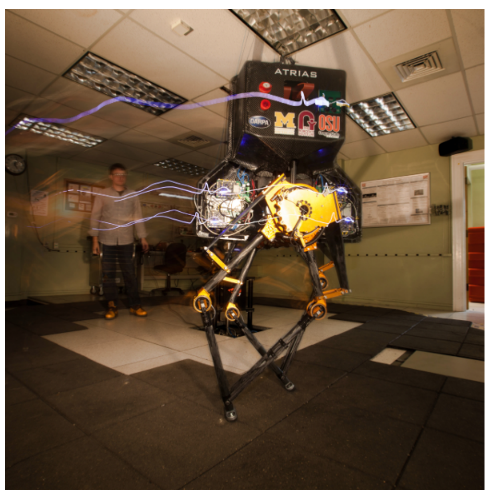

Figure 2: ATRIAS 2.1, a robot capable of dynamic running [5].

However, both values should be actively minimized to result in the best performance.

2.4 Actuated Spring Loaded Inverted Pendulum Model

In order to create control strategies for dynamic walking and running robots, a dynamic model is used to understand the motion of dynamic walking and running better. The Actuated Spring Loaded Inverted Pendulum Model is a reduced order model capable of modeling dynamic running, but it can also be applied to dynamic walking [7][8]. It has been used to help create control algorithms for both ATRIAS and Cassie, in addition to understanding how to better create the motion of dynamic movement, such as how leg thrust can best be utilized.

A traditional ASLIP model has a leg length actuator and moves in a plane. The leg length actuator changes the set point of the spring on the leg, which is in parallel with a damper. The model can also rotate around its center of mass so that it can choose the touchdown angle of the leg. This rotation is cost-less, as the leg is assumed to be mass-less and rotation around the center of a point mass means there is no rotational inertia. Touchdown angle is used to trade off between speed and hopping height. A physical example of the ASLIP model is the Raibert Hopper [8], and a model diagram can be seen in Figure 3.

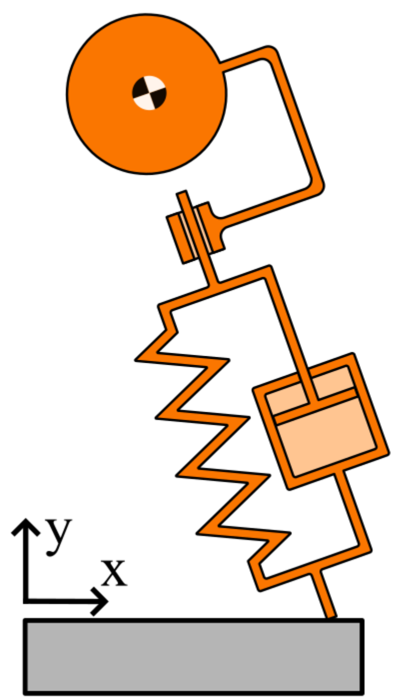

Figure 3: A Traditional ASLIP Model.

The ASLIP model is a hybrid dynamic system, which means it has two dynamic phases, each with its own set of equations. The equations that govern one of the phases, stance, are a coupled system of non-linear second order differential equations. This means that the ASLIP model cannot be directly solved, and other methods must be used to estimate the solution. The other phase, which is the flight phase, can be solved with ballistic flight equations.

2.5 Trajectory Optimization & Direct Collocation

Trajectory optimization describes a process that creates a trajectory while minimizing a desired quality of the trajectory. There are different methods of trajectory optimization, and it is important to choose a method that works well with the system for which trajectories are being created and optimized.

Direct collocation is a trajectory optimization method where the dynamics of the system are encoded into the constraints [9]. To use direct collocation it must be tailored to fit the model and the trajectory value that should be minimized by choosing the correct objective function and constraints. Direct collocation was chosen because the dynamics of the ASLIP model are complex, so it made sense to choose a trajectory optimization model where there is a way to easily turn system dynamics into constraints, so that the optimizer knows more about the system, making the optimization easier to solve.

To pose an optimization, there needs to be an objective, which is the value of the trajectory to minimize, and constraints, that ensure the trajectory fits the requirements of the system.

The objective function describes the objective, or value to be minimized, and results in a number that can be used to compare the trajectories created. The objective function is described with different state and control variables throughout the trajectory.

The control variables are the variables that the optimizer can control to affect the state variables through the system dynamics. The state variables are the variables that change between collocation points and describe the model at each point. The state and control variables can also be called decision variables, as they are variables that the optimizer can change to minimize the objective function. These variables also must stay within given upper and lower bounds, which can be used to keep them in a desirable range.

One group of constraints, called the collocation constraints, are the constraints that relate the decision variables at a collocation point with the decision variables of the next collocation point using the system dynamics.

Another group of constraints are the linear and nonlinear constraints. They are used to constrain the trajectories to other conditions, like initial and final constraints, and vary with the model and system is being optimized. This optimization only used nonlinear constraints.

2.6 Using Trajectory Optimization to Create Gaits for the ASLIP Model

Trajectory optimization can be used to learn more about dynamic walking and running through the ASLIP model. It has been used to confirm an energy-optimal control strategy, investigate the effect of using planned behaviours during disturbance recovery to mitigate the effect of delay, and evaluating limit cycle gaits, which describe a repeating gait without individual planning for each step.

Through the ASLIP model and trajectory optimization, fundamental truths about spring-mass running can be learned. In an early investigation into dynamic running using trajectory optimization, Sequential Quadratic Programming was used to generate energy-optimal trajectories for an ASLIP model traveling over soft ground. The energy used in these trajectories was compared to the energy used in the gait prescribed by the force control model, where the force at the toe of the leg is set to follow a certain trajectory [10], proving that force control is energy optimal [11]. This shows that this energy-optimal trajectory can be used to check ASLIP trajectories created via trajectory optimization to ensure that the optimizer is working correctly.

Trajectory optimization can also be used to simulate delay and evaluate how disturbance response is affected by different control methods or model parameters. Using direct collocation to create ASLIP trajectories, delay was simulated after a perturbation to investigate how prereflex behaviours (preflexes) could be used to decrease the energy necessary to recover from a disturbance. The preflexes were shown to decrease recovery energy when compared to a gait optimized for flat ground [12]. From this paper, direct collocation is shown to be good tool to investigate delay in an ASLIP model, as both the ASLIP dynamics and the delay can be input into the optimizer via constraints.

When using trajectory optimization to create and evaluate ASLIP trajectories, running the optimizer over several steps of the ASLIP model can drastically increase the time necessary for the optimizer to come to a solution. However, limit cycle gaits, where the ASLIP model repeats the same step for each new step, have been shown to be near energy-optimal [13]. To reduce time to solve for an optimizer, a single step of the ASLIP model can be optimized, which could then be theoretically repeated in a limit cycle. To encode this into an optimizer, an equilibrium gait is used, where the ASLIP model starts at a certain height during flight phase and must leave stance phase with enough energy to return to that height as shown in Figure 4.

From prior uses of trajectory optimization to create and evaluate ASLIP trajectories, lessons can be learned to aid in the creation of an optimizer. In this thesis, the optimizer can be verified by examining the force in the leg and ensuring it follows the trajectory described in the force-control method, direct collocation will be used so that the delay can be input as constraints, and constraints to enforce an equilibrium gait will be used to reduce the time the optimizer needs to solve the optimization.

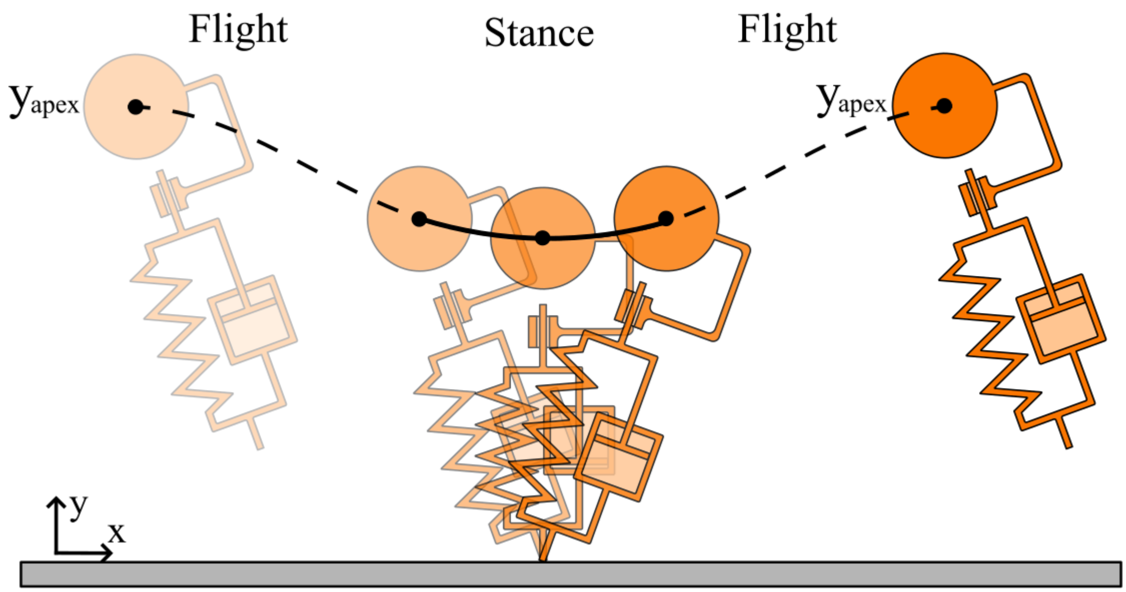

Figure 4: An Equilibrium Gait For a Traditional ASLIP Model.

2.7 Simplified ASLIP Model

The ASLIP model used in this paper varies from the traditional model by simplifying it to one dimension. This means that the model can only hop up and down, not forward and backwards. The model was simplified because this allows the variables and parameters relating to moving forward to be ignored. The leg-length actuator is more important when it comes to disturbance response, so the planar movement is not needed to compare how the reflected inertia and delay affect recovery. The model used can be seen in Figure 5. Although the model is simplified, it is still governed by hybrid dynamics.

One of the dynamic phases, the stance phase, is when the spring leg is in contact with the ground. It is governed by a coupled system of second order nonlinear differential equations. When in stance phase, two equations of motions govern the model used in this paper. One describes the leg spring set point acceleration and the other describes the acceleration of the mass in the y-direction. Both equation use the force in the leg, or the force in the spring, to calculate acceleration. The force in the leg is calculated as

$$F_{leg} = K * (r_0 - L_{spring}) + c * (dr_0 - V_{spring}) \tag{1}$$

where L_{spring} is the length of the spring and V_{spring} is the velocity with which the spring length is changing. For this model, because it is constrained to one dimension, Y and dY can be used instead. K is the spring constant of the leg spring and c is the damping constant of the leg damper.

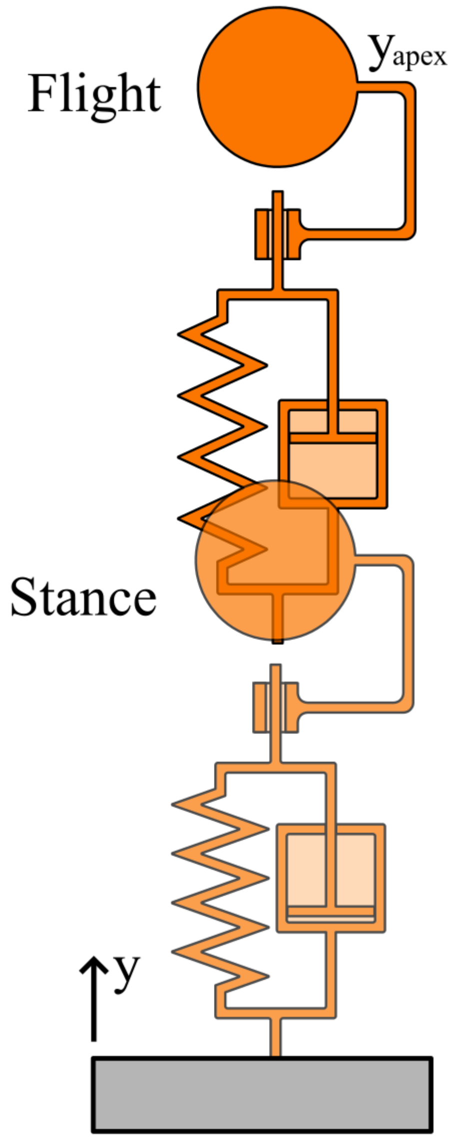

Figure 5: Simplified One-Dimensional ASLIP Model.

Leg set point acceleration is calculated as

$$ddr_0 = \frac{F_{motor} - F_{leg}}{M} \tag{2}$$

and the acceleration of the mass is calculated as

$$ddY = \frac{(F_{leg} - m * g)}{m} \tag{3}$$

where m is the mass of the robot, M is the reflected inertia, and g represents acceleration due to gravity. F_{motor} is a control variable.

The flight phase, when the spring-mass system is in the air, is governed by the equations describing projectile motion

$$ddY = -g \tag{4}$$

and can be solved for expressly.

For the above equations, the parameters are as follows:

K	$16\ N/m$
c	$4\ (N*s)/m$
m	kg
g	$1m/s^2$

Most of the parameters used are based off of the characteristics of a human leg and then non-dimensionalized by Froude number [14]. No value is given for M, the reflected inertia, because it will be varied in order to investigate the effect of reflected inertia on disturbance response.

3 Methods

3.1 Optimizing Trajectory with Direct Trapezoidal Collocation

Direct trapezoidal collocation was used to optimize the trajectory of one equilibrium gait step of the actuated SLIP model in order to find the motor forces of the leg length actuator that would result in the most energy efficient trajectory. One step is considered to be when the model begins in flight phase, goes through one stance phase, and then leaves stance to another flight phase. The flight phase follows the equations of projectile motion, so the state at the end of the two flight phases is solvable in closed form. Equilibrium gait means the peak height of the second flight phase is equal to the peak height of the first flight phase. The objective function is used to minimize energy usage and the constraints are used to constrain the solution of the optimizer to the dynamics of the actuated SLIP model. With the optimizer, a disturbance and disturbance recovery are modeled to investigate how reflected inertia and delay affect the energy efficient trajectory of the model.

3.2 Objective Function

To measure the energy efficiency of a trajectory, the positive mechanical work is summed with the electrical losses, as shown in the following equation.

$$J = \int_0^T \mid F_{motor} * dr_0 \mid dt + R_{leg} * \int_0^T F_{motor}^2 dt \tag{5}$$

Positive mechanical work is the integral of positive instantaneous mechanical power and gives a measure of the energy used in the leg length actuator throughout stance. The electrical losses, also called thermal losses, represent energy lost to heat in the motor. The thermal loss coefficient, R_{leg}, is measured from Cassie [14]. Each optimization will result in a single value, J, which describes the amount of energy used by the actuator throughout the stance phase.

Direct trapezoidal collocation results in discrete state points. To approximate the integrals, a trapezoidal Riemann sum was used. However, the command trapz in Matlab assumes unit spacing, so the trapezoidal Riemann sum was multiplied by the expression $\frac{T_{stance}}{N-1}$ where T_{stance} is the overall time of the stance phase and N is the number of collocation points.

3.3 Constraints

Constraints are used to bind the optimizer to the desired system dynamics, initial state, and final state.

The collocation constraints relate the decision variables at one collocation point to the decision variables at the following collocation point through the equations governing the system dynamics. The state variables are as follows:

Y	Position of Point Mass
dY	Y-velocity
r_0	Setpoint of Spring
dr_0	Setpoint Velocity

In addition to the state variables, there are two control variables. The first is the variable F_{motor}, which is the force applied by the motor to change r_0. The other variable is T_{stance}, which is used so that the optimizer can choose the overall stance time. The time at each following collocation point is calculated by assuming uniform spacing.

The collocation constraints are calculated with the following equation

$$Ceq = 0.5 * h * (f_{k+1} + f_k) - x_{k+1} + x_k \tag{6}$$

where the variables are given as

$$h = \frac{T_{stance}}{N-1} \tag{7}$$

$$x = \begin{bmatrix} Y \\ dY \\ r_0 \\ dr0 \end{bmatrix} \tag{8}$$

$$F = \begin{bmatrix} dY \\ ddY \\ dr_0 \\ ddr0 \end{bmatrix} \tag{9}$$

and the derivative of x, F, is calculated with equations 2 and 3.

This constraint is a nonlinear equality constraint, so the optimizer will minimize Ceq at each collocation point, resulting in state variables at each collocation point that stay true to the dynamics of the system.

The other constraints are used to ensure the system meets the desired initial and final states. The first set constrains the state variables at the first collocation point to the state variables at the end of the first flight phase. At the final collocation point, there is a constraint to ensure that the force in the leg, calculated with Equation (1), is equal to zero. This is the method for determining when stance ends, as when the force in the leg is zero, the model has just left the ground. There is also a constraint to ensure that the model has enough energy when leaving the stance phase to reach the peak height of the preceding flight phase, ensuring an equilibrium gait. This constraint is given through the following equation

$$Ceq = Y_{peak} - (Y_N + \frac{dY_N}{2 * g}) \tag{10}$$

The optimizer will attempt to minimize both of these constraints. The final constraint is to ensure that the model leaves the ground with a positive velocity. Occasionally, the trajectory optimization would find a solution that went above the equilibrium height and then dropped back down to the desired height. This constraint was implemented to make sure this was not a valid solution, and is simply given as

$$C = -dY_N \tag{11}$$

The optimizer will attempt to make this constraint less than or equal to zero, which results in a positive final velocity.

The last constraint in the trajectory optimization are the bounds put on the state variables and are as follows:

$$0.5m \leq Y \leq 1m$$
$$-\infty \leq dY \leq \infty$$
$$-\infty \leq r_0 \leq \infty$$
$$-\infty \leq dr_0 \leq \infty$$
$$-5N \leq F_{motor} \leq 5N$$
$$0.01s \leq T_{stance} \leq 3s$$

3.4 Implementation

To measure how reflected inertia and delay affected the energy used in disturbance recovery, the energy needed for recovery of a perturbed model was compared to the energy used by a standard model with no disturbance. Both the standard and perturbed model had the same reflected inertia during experiments, but it was varied across different optimizations. The disturbed model was affected by a delay, and different perturbation magnitudes were used.

In order to determine the energy efficiency of the disturbance response, a baseline energy must be found. The first step in the experiment is to run a standard optimization, where the model is dropped from the equilibrium height, to determine the energy optimal trajectory through stance phase that returns the model the equilibrium height. The force of the motor at each collocation point is then saved.

After the unperturbed optimization is run, the model is perturbed and then run through the optimizer again. To perturb the model, it is dropped from a different height than the standard optimization. The changed drop height represents a legged robot walking over smooth terrain, then encountering a ground height that differs. It thinks it will be contacting the ground at a certain point, but the ground is either raised (a speed bump) or lowered (a pothole). Dropping the model from a higher height represents a pothole while dropping it from a lower height represents a speed bump.

To ensure the optimizer is actually surprised by the perturbation, delay is simulated. To model delay, the motor forces from the standard optimization are used as constraints. At each collocation point in the perturbed model for the first X points (these points are the delay period), the motor force is constrained to the standard optimization motor force values. Additionally, the time step between collocation points is constrained to be the same as in the standard optimization. With these constraints, the model attempts to enact the standard force trajectory for an amount of time that varies with the number of control points (x) in the delay period. After the delay period has ended, it can re-plan its trajectory to find the energy optimal trajectory that takes the disturbance into account. It then must end stance with the necessary energy to get back to equilibrium height without overshooting or under shooting.

In both the standard and the perturbed optimization, the optimization was run four times for each set of variables. In between each optimization, white Gaussian noise was added to the decision variables and they were then fed back into the optimizer. The optimization with the lowest objective function evaluation after four cycles was used going forward. This ensures the solution does not get stuck in a local minima [9].

Finally, the desired variables were varied to get the data used in the results. The delay was varied between 0 and 5 control points, which corresponds to 0 to around 180 ms of delay. The high end of this range is not exact as the optimizer can choose stance time, which affects the time between each collocation point, which affects how long the delay period lasts. The reflected inertia was varied between three values, 0.05, 0.1563, and 0.25. These values were chosen because 0.1563 represents the approximate reflected inertia of Cassie non-dimensionalized to match the other parameters. 0.05 and 0.25 were then chosen as they gave a range of reflected inertia, and beyond these values the model broke down. The disturbance magnitude was varied between a simulated groundheight change of -10 cm, -5 cm, 5 cm, and 10 cm.

4 Results & Discussion

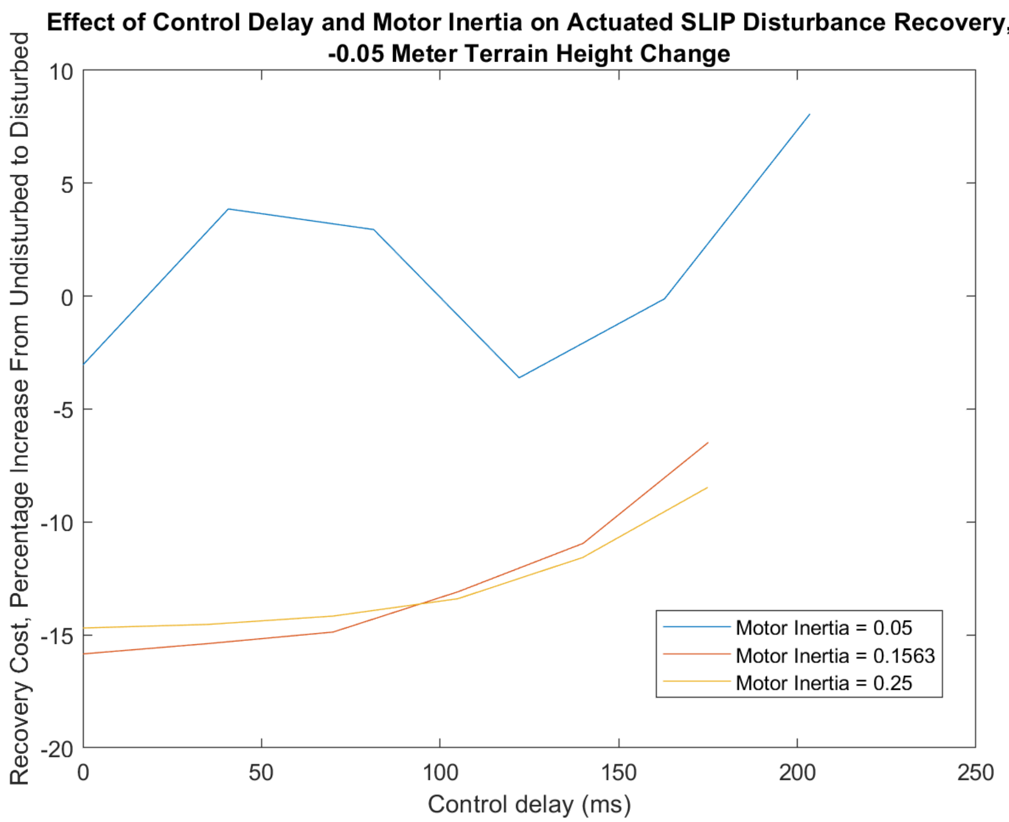

Figure 6: The results from one Pothole optimization, where the ground height was 5 cm lower than expected.

Neither the pothole or speed bump scenario resulted in any conclusive results. As can be seen in Figures 6, 7, 8 and 9, there is not enough data to support a conclusion that reflected inertia and delay can result in comparable effects on disturbance response. Upon reflection, the experimental setup was not the best method that could be used to investigate this question.

Comparing the energy used during the disturbance response of energy-optimal ASLIP trajectories while varying reflected inertia and delay was not a good way to evaluate how those values affect disturbance response because of how the passive dynamics of the ASLIP affect the energy used in disturbance recovery. This is highlighted in the pothole scenario, where the perturbed model used less energy to get

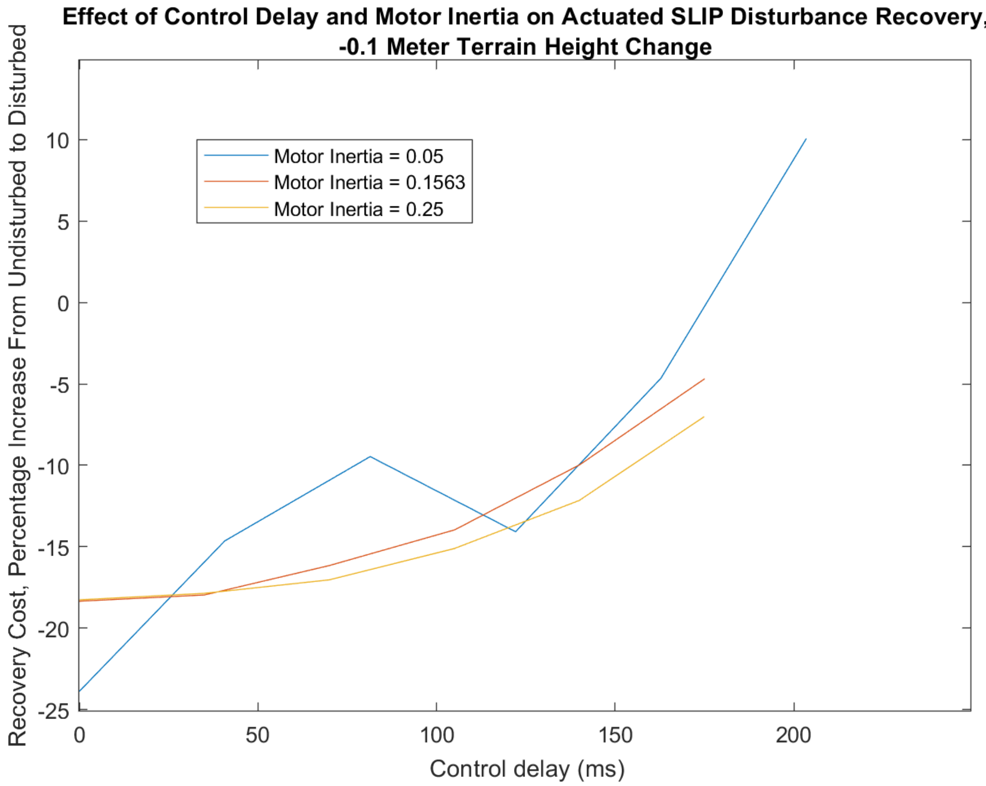

Figure 7: The results from one Pothole optimization, where the ground height was 10 cm lower than expected.

back to equilibrium gait than the standard model, as it entered stance with more kinetic energy. This means that both reflected inertia and delay didn't affect the stance phase much, as the actuator did not need to do much work. If the passive dynamics affected the pothole scenario, they affected the speed bump scenario too, however it is difficult to attribute any aspect of the results to the passive dynamics alone. In summary, the actuator response was diluted through the ASLIP model, so when evaluating disturbance response to investigate how different values affect the actuator, the effects were themselves affected by the ASLIP model, leading to inconclusive results.

5 Conclusion & Lessons Learned

Reflected inertia and delay are two factors that are important in the disturbance response of a dynamic legged robot. If either value is too high, the robot may not be able to effectively recover from a perturbation. While delay is typically low for robots, reflected inertia is often tougher to overcome, and dynamic legged robots still have far to go in robustness. Conversely, animals have much larger delays, but much

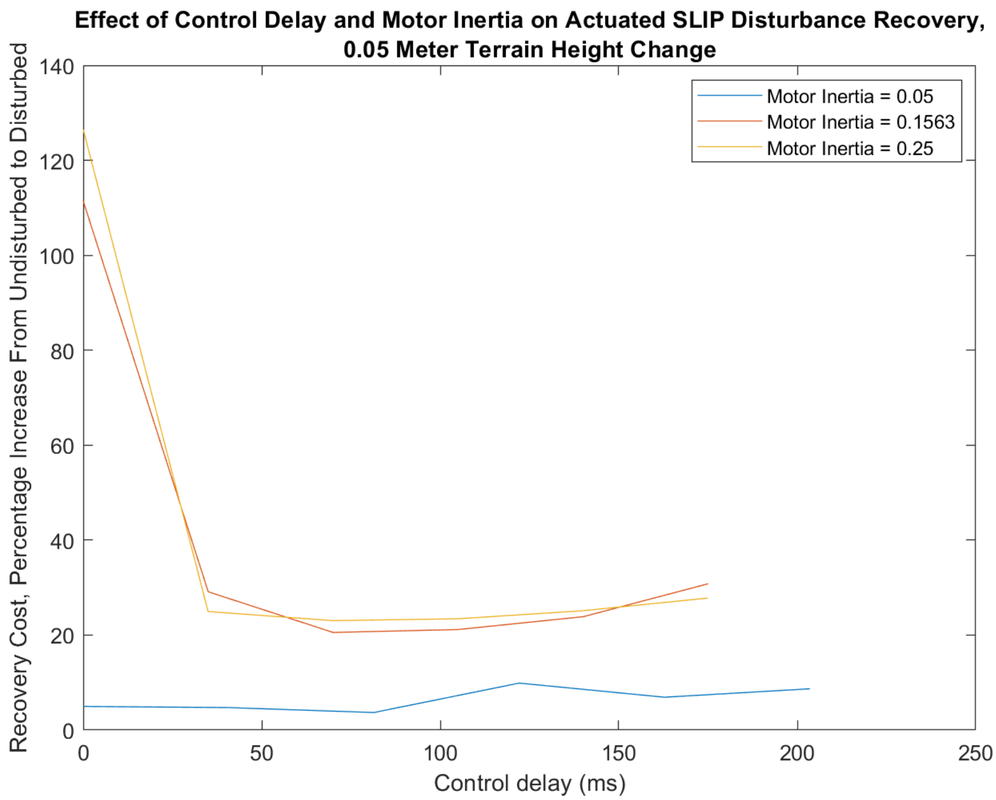

Figure 8: The results from one speed bump optimization, where the ground height was 5 cm higher than expected.

more robust performance, leading to the assumption that their muscles can respond much faster than a robot's actuators. If this is true, then it is possible that the effects on disturbance response of delay and reflected inertia are comparable.

To investigate if their effects are comparable, an ASLIP model was used in conjunction with trajectory optimization and looked how the energy efficiency during a disturbance response was affected by the reflected inertia and delay. This resulted in no conclusive results. In hindsight, the methodology used was not the best plan, as by investigating the effects of reflected inertia and delay through disturbance response evaluation, the effects were diluted by the system dynamics of the ASLIP model.

A better way of investigating the effect of reflected inertia and delay on disturbance response would have been to investigate actuator response, realizing that reflected inertia and delay affect disturbance response through their effect on actuator response. A possible methodology would be to use a simulated actuator and evaluate its response to different input signals and how the response is affected by reflected inertia and delay. To do this, an actuator model would be created, where there is an input command to an equation describing the the actuator output as a function of the current input signal, the actuators current state, and the actuator parameters. Then,

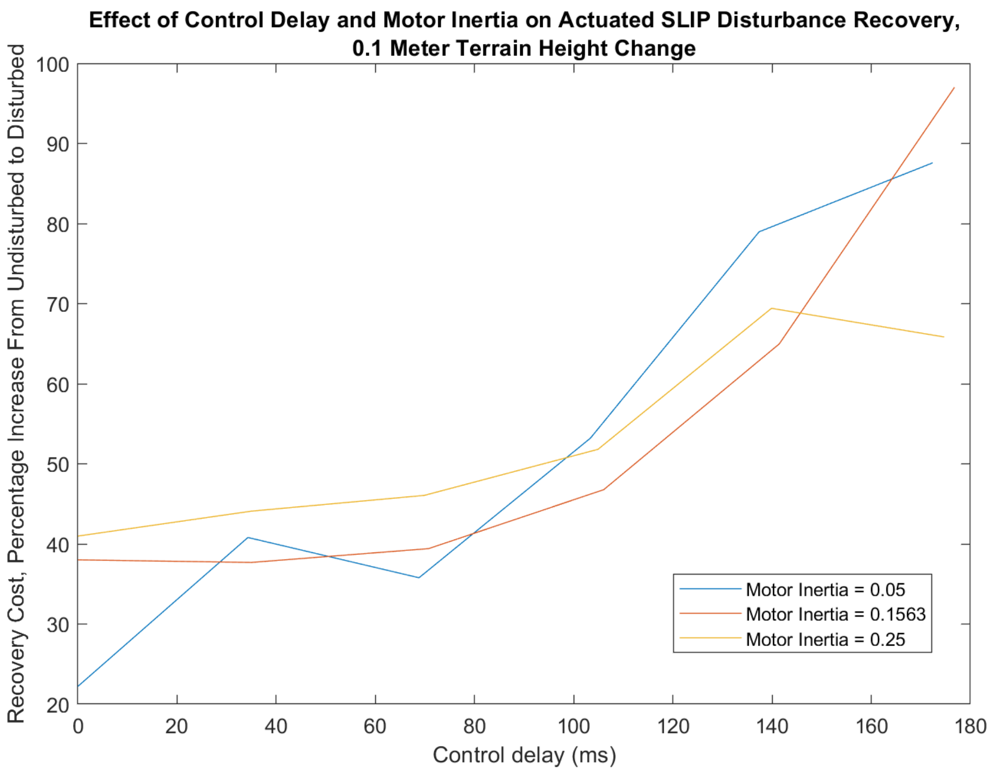

Figure 9: The results from one speed bump optimization, where the ground height was 10 cm higher than expected.

the input commands could be varied to reflect a robot re-planning its trajectory, disturbances could be modeled by varying resistance to the actuator's output (for example, allowing it to extend with no counter-force, simulating a ground height lower than expected for a dynamic legged robot), and delay and reflected inertia could also be varied through these scenarios to see if any patterns arose in the data.

References

[1] T. Apgar, P. Clary, K. Green, A. Fern, and J. Hurst, “Fast Online Trajectory Optimization for the Bipedal Robot Cassie,” in *Robotics: Science and Systems 2018*, (Pittsburgh), 2018.

[2] K. Knight, “Understanding Inertia and Reflected Inertia,” tech. rep., Motion Control & Motor Associatioin, 2015.

[3] T. T. Andersen, H. B. Amor, N. A. Andersen, and O. Ravn, “Measuring and modelling delays in robot manipulators for temporally precise control using machine learning,” in *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, pp. 168–175, 2016.

[4] Agility Robotics, “Cassie.”

[5] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, “ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot,” *International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[6] T. McGeer, “Passive Dynamic Wlalking,” *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[7] D. Renjewski, A. Sprowitz, A. Peekema, M. Jones, and J. Hurst, “Exciting Engineered Passive Dynamics in a Bipedal Robot,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1244–1251, 2015.

[8] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA: Massachusetts Institute of Technology, 1986.

[9] M. Kelly, “An introduction to trajectory optimization: How to do your own direct collocation,” *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[10] D. Koepl and J. Hurst, “Force control for planar spring-mass running,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 3758–3763, 2011.

[11] C. M. Hubicki and J. W. Hurst, “Running on Soft Ground: Simple, Energy-Optimal Disturbance Rejection,” in *International Conference on Climbing and Walking Robots*, 2012.

[12] J. Van Why, C. Hubicki, M. Jones, M. Daley, and J. Hurst, “Running into a trap: Numerical design of task-optimal preflex behaviors for delayed disturbance responses,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 2537–2542, 2014.

[13] C. Hubicki, M. Jones, M. Daley, and J. Hurst, "Do limit cycles matter in the long run? Stable orbits and sliding-mass dynamics emerge in task-optimal locomotion," in *IEEE International Conference on Robotics and Automation*, vol. 2015-June, pp. 5113–5120, 2015.

[14] M. Hector, K. Green, B. Sencer, and J. Hurst, "Ankle Torque During Mid-Stance Does Not Lower Energy Requirements of Steady Gaits," in *Intelligent Robots and Systems*, 2019.