Southern Methodist University
SMU Scholar

Mechanical Engineering Research Theses and Dissertations

Mechanical Engineering

Spring 5-19-2018

Feedback Linearization of Inertially Actuated Jumping Robots

Adam Cox
Southern Methodist University, acox@smu.edu

Yildirim Hurmuzlu
Southern Methodist University, hurmuzlu@lyle.smu.edu

Follow this and additional works at: https://scholar.smu.edu/engineering_mechanical_etds

Recommended Citation

Cox, Adam and Hurmuzlu, Yildirim, "Feedback Linearization of Inertially Actuated Jumping Robots" (2018). *Mechanical Engineering Research Theses and Dissertations*. 5.
https://scholar.smu.edu/engineering_mechanical_etds/5

This Thesis is brought to you for free and open access by the Mechanical Engineering at SMU Scholar. It has been accepted for inclusion in Mechanical Engineering Research Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.

FEEDBACK LINEARIZATION OF INERTIALLY ACTUATED JUMPING ROBOTS

Approved by:

Dr. Yildirim Hurmuzlu
Profesor

Dr. Edmond Richer
Associate Professor

Dr. Tindaro Ioppolo
Assistant Professor

FEEDBACK LINEARIZATION OF INERTIALLY ACTUATED JUMPING ROBOTS

A Thesis Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Master of Science in Mechanical Engineering

with a

Major in Mechanical Engineering

by

Adam Cox

B.S.M.E., Mechanical Engineering, Southern Methodist University

May 19, 2018

Copyright (2018)

Adam Cox

All Rights Reserved

ACKNOWLEDGMENTS

Cox, Adam B.S.M.E., Mechanical Engineering, Southern Methodist University

Feedback Linearization of Inertially
Actuated Jumping Robots

Advisor: Dr. Yildirim Hurmuzlu
Master of Science in Mechanical Engineering degree conferred May 19, 2018
Thesis completed April 17, 2016

The focus of robotics recently is in automated navigation. It seems that the engineering community has accepted that the discovery of all means of locomotion has been concluded with quad-copters and other similar drones. However, Inertially Actuated Jumping Robots provide a promising new means of locomotion.

The difficulty of IAJR is the hybrid nature of the ground contact/flying dynamics. This combined with the complexity of 3-dimensional translation, can make IAJR very complex. In this paper, a Nonlinear Feedback Linearization controller is introduced to provide controllability in this complexity. The controller design is based on invariant sets. By reducing the divergence from the invariant set, a greater response can be achieved. Within the available power of Kashki's Basketball Robot, the controller in this paper was able to achieve the greatest response to date for the Basketball Robot at a maximum jump height of 0.25 meters. Further simulation shows that without restricting physically or electrically available power, the robot can achieve a jump height of 0.6 meters!

The design paradigm used on the basketball robot was extend to a tapping robot. The tapping robot achieved a stable average forward velocity of 0.0773 meters/second in simulation and 0.157 meters/second in experimentation.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

Nomenclature

μ_k	Kinetic Friction Coefficient
μ_s	Static Friction Coefficient
$\delta\theta$	Experimental Sensor Angle Offset
$\Delta_s[t]$	The compression of the spring
η	Motor Efficiency
ω_c	Critical Spinner Frequency. This is the frequency that the Basketball robot begins jumping
ω_r	Resonant Spinner Frequency.
$\psi_m[t]$	Main Body tilt angle
τ	Input Torque of Motor
$\theta[t]$	Spinner Angle with respected to Main Body
$\xi[t]$	Basketball Height, under The Constant Angular Acceleration assumption
ξ_m	Basketball Height, under The Constant Angular Acceleration assumption, steady state response magnitude.
$\xi_r[t]$	Basketball Height, under The Constant Angular Acceleration assumption, steady state response resonance magnitude.
$\xi_s[t]$	Basketball Height, under The Constant Angular Acceleration assumption, steady state response.
$\xi_t[t]$	Basketball Height, under The Constant Angular Acceleration assumption, transient response.

b	Spring Damping Constant
C_t	Basketball Height, under The Constant Angular Acceleration assumption, transient response Constant.
F_f	Friction Force Between the ground and the tapping robot
F_N	Normal Force Between the ground and the tapping robot
F_s	Spring Force Between the ground and the tapping robot
g	Gravitational Acceleration
h_m	y-Distance from Tapping Robot Pivot Point to the Center of Mass of Tapping Robot
h_p	y-Distance from the Center of Mass of Tapping Robot to the pivot point of the spinner
I_m	Main Body Rotational Inertia
I_p	Spinner Rotational Inertia
k	Spring Elastic Constant
K_v	Motor Speed Constant
l_m	x-Distance from Tapping Robot Pivot Point to the Center of Mass of Tapping Robot
l_p	Spinner Length
l_s	x-Distance from Tapping Robot Center of Mass to the spring connection
m_m	Main Body Mass
m_p	Spinner Mass
R_g	Motor Gear Ratio
R_x	Reaction force between the spinner and the main body in the x-direction
R_y	Reaction force between the spinner and the main body in the y-direction

$S_c[t]$	Spring Contact Switch (Boolean)
S_f	Uncompressed spring length
T_c	Kinetic Energy while in Contact with Ground
T_f	Kinetic Energy while in Flight
T_s	Speed/Torque Gradient
$u[t]$	Input PWM Signal to Motor Driver $\in$(-100%,100%)
$u_v[t]$	Virtual Input, after linearization
V_b	Battery Voltage
V_c	Potential Energy while in Contact with Ground
V_f	Potential Energy while in Flight
X	Space Fixed x-coordinate
x	Main Body x-coordinate
$X_m[t]$	Main Body x-position (Space Fixed Coordinate Frame)
$x_m[t]$	Main Body x-position (Main Body Coordinate Frame)
Y	Space Fixed y-coordinate
y	Main Body y-coordinate
$y_G[t]$	Center of Mass Height of Basketball Robot
$y_m[t]$	Main Body y-position (Main Body Coordinate Frame)

This is dedicated to all my friends and family who pushed me and pulled me along.

Chapter 1

INTRODUCTION

1.1. Motivation

Inertially actuated jumping robotics is an important issue because of the need for maneuverability, speed, and efficiency. In light of recent drone technology, highly dynamic robots have revealed themselves to achieve performance unreachable by direct human control.

The benefit of land vehicles is their maneuverability and response. Li [11] showed this in a spherical robot. Flying drones, such as quad-copters, must change their orientation and use aerodynamics to change the flight path. This creates a delay. jumping robots do not have this delay. Inertially actuate jumping robots are able to change their jumping frequency to change the percent of jumping in the air and on the ground. A jumping robot is able to follow complicated paths by changing directions while the springs are in contact with the ground. While in the air, jumping robots can travel at high speeds. By changing the physical parameters of the robot design and the virtual parameters of the controller, a jumping robot will be able to achieve complicated dynamics that no other means of locomotion can achieve. Other research has focused on land locomotion by following biological organisms [1, 4–6, 12–14, 18, 19, 23–25, 27, 28]

The benefit of air vehicles is their ability to travel at high speeds. This is where drones excel. However, jumping robots promise the same performance. A drone is limited by the maximum speed that the thrusters can provide. A jumping robot is propelled by gravity. Hale [2] showed this efficiency in lower gravity. Gravity is more powerful than the propellers powered by brushless DC motors, and gravity is conservative.

Flying drones are able to carry only light loads and have short fly times. Inertially actuated jumping robots do not have this restriction. The stablizing dynamics of drones are highly non-conservative. In an inertially actuated robot, the energy is stored in springs. The only significant non-conservative forces are damping in the springs and friction in the spinner assembly.

At the intersection of flying locomotion and land locomotion is traversing rough terrain. Song [21] showed that this intersection can be used to produce high speed maneuverability on stairs.

1.2. Robots analyzed

Three systems are analyzed in this paper. The first system is a "Fixed Pivot," second is a "basketball Robot" and the third is a "Tapping Robot." These systems are the foundational systems of Inertially Actuated Jumping Robotics (IAJR).

1.2.1. Fixed Pivot

In IAJR, the fundamental actuation is a spinning mass. This spinning mass is called a "spinner." The natural first step to using a "spinner" is to understand and control its dynamics. A "spinner" is simply a pendulum. There is a heavy mass at the end of a rod. As the mass rotates, the inertia of the spinner creates reaction forces at the pivot point. The reaction forces can be attached to any robot. These reaction forces allow IAJR to work.

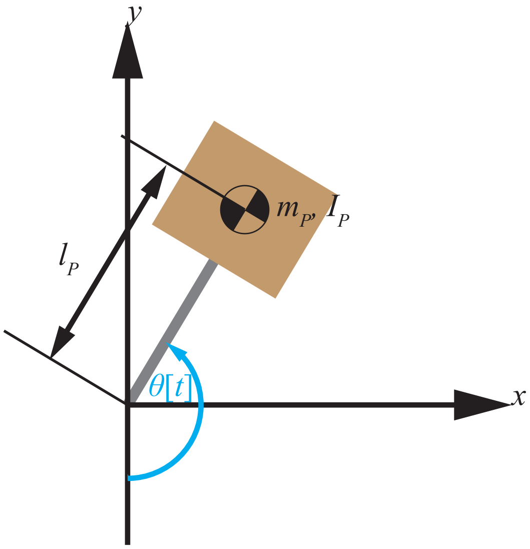

Figure 1.1. Fixed Spinner Diagram

1.2.2. Basketball Robot

Mohammad Kashki [9] introduced the first Nonlinear Controller for IAJR (See figure 1.2). The robot was mounted on a vertical guide to fix the main assembly to vertical motion only. In the assembly, a motor was attached to a spinner, to create an input force. The controller used an adaptive feedback controller to stabilize the jumps and to control the maximum jump height.

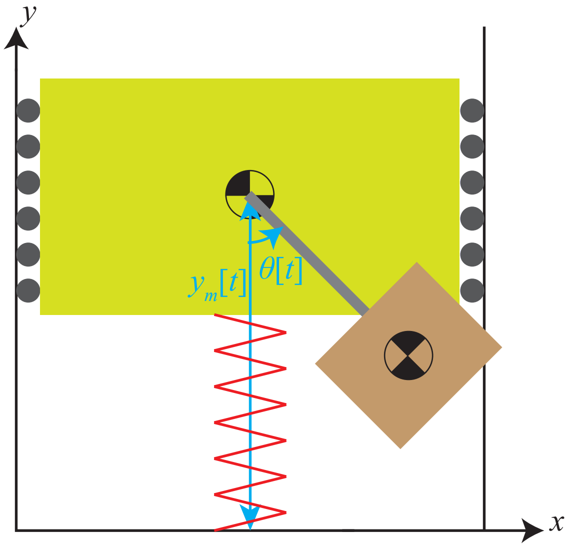

Figure 1.2. Kashki's Basketball Robot

1.2.3. Tapping Robot

Following the controller design from the "Basketball Robot," the vertical actuation was fixed to a frame to test the controller's dynamics with a robot that can move both vertically and horizontally. The resulting robot is called a "Tapping Robot." The "Tapping Robot" jumps on a hybrid nonlinear path. The simulation and experimental results provided insight into how effective the controller is, presented in this paper.

The "Tapping Robot" can be seen in figure 1.3. In reality, a robot that is only vertical and horizontal would fall to its side, assuming it is not on a rail guide. To stabilize it, the robot has two back legs and two front springs. The center of mass and the input force were positioned in the middle of the supports and springs to balance the robot for only forward and backward movement. The robot's springs were custom made such that the two springs have opposite windings.

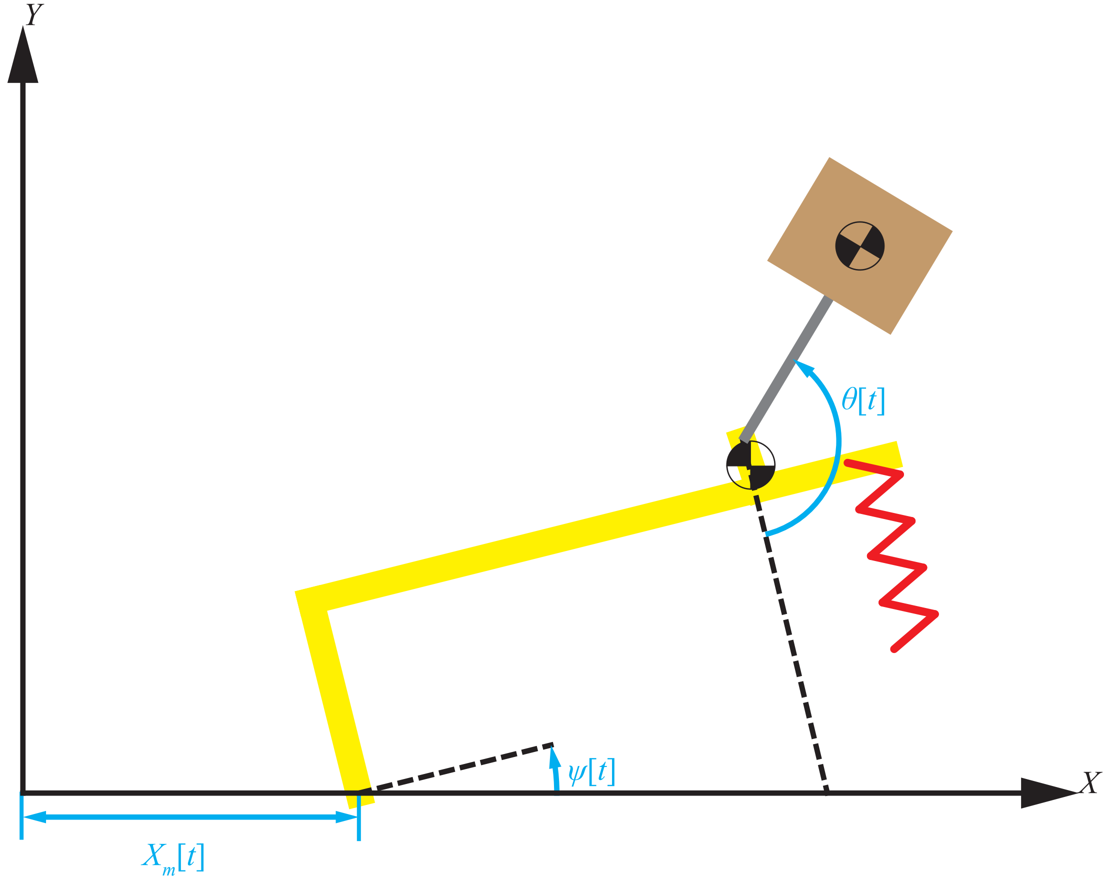

Figure 1.3. Adam's Tapping Robot

1.3. Literature Review

The jumping robot design provided in this paper is derived from Zoghzoghy's Pony robots [30]. The Pony robots suffered from several limitations. For the first limitation, the controller used is a simple linear PID controller. This works for the Pony robots because the rotational velocity of the spinners is far below the saturation of the motors. This means that there is enough power to overcome nonlinearities with such a controller. The robot design in this paper uses an adaptive feedback linearization controller to identify the parameters of the spinner. Then, a feedback lineariztion was used to created a more robust controller.

Another area of improvement for the Pony robots is the actuation force. The basic premise of the inertially actuated robots is that they jump at the same frequency as the spinners. When the robot is in contact with the ground, the system can be approximated as simple spring-mass-damper system. Therefore, there is a resonant frequency for the system. If the spinners are spun at this frequency, then the robot should have its maximum response. Okubo [16] studied the use of springs to store energy for jumping robotics. In the pony robots, the stiffness of the springs are low enough such that the Pony robot almost has saturation of potential elastic energy in the springs. Meaning, the springs almost fully compress, creating an impact. The Pony Robot has a spring stiffness of 700 N/m. The tapping robot in this paper has a spring stiffness of 3200 N/m. The stiffer springs in this paper can store over 4.5 times the potential energy of the Pony Robot. This promises a more powerful response.

In 1984, Raibert proposed a basic hopping machine. [14] The robot Raibert designed is based on the continuation of linear momentum, stabilized by a single jumping leg. The jumping motion is powered by changing the spring stiffness in the leg and the robot navigates by changing the position and orientation of the actuation force with respect to the center of mass. Raibert laid down the grown work of inertially actuated robotics. Specifically the accumulation of kinetic energy in a spring. Prosser [17] developed a control algorithm that does not require full state feedback.

R. Hayashi and S. Tsujio built a simple jumping robot that used a spinner to ”throw” a static frame into the air. [3] This ground breaking use of spinners, that enabled a robot to ”jump,” inspired the use of spinners in the Pony Robot.

Zoghzoghy’s work is based on a baton system. Started by the theory of Tavakoli and Hurmuzlu [22], inertial actuation was applied to a baton system [29].

Inertial actuation has also been used in controlling orientation for locomotion. Kashki used it for a pivot walking robot [8].

1.4. Problem Statement

What is essential to IAJR is controllability. In controllability, the control engineer needs to know the relationship between input and output. Once the ”plant” has been identified, the control engineer can design a feedback system to improve the response. The problem attacked in this paper is finding a linear relation between the input velocity of the spinners and the output response of the robot. In the "Basketball Robot," the desired output is a specific jump height. In the "Tapping Robot" the desired output is a specific horizontal velocity.

Chapter 2

DYNAMICS OF THE SYSTEM

2.1. Equations of Motion

2.1.0.1. Fixed Pivot

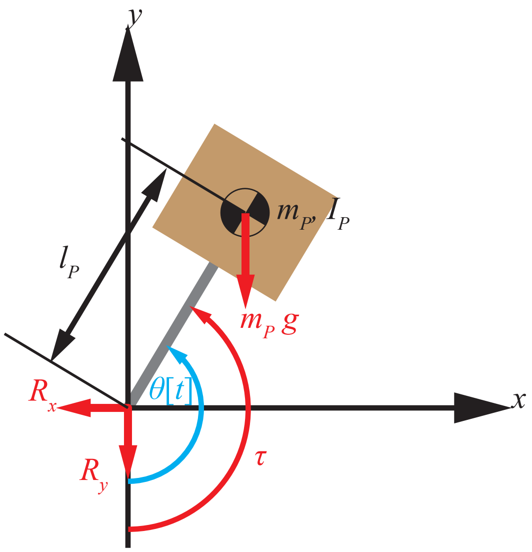

Figure 2.1. Fixed Pivot Forces Diagram

The equation of motion for the Spinner:

$$\theta''[t] = \frac{\tau + (-g + y''[t])\; l_P\; m_P\; Sin[\theta[t]] - l_P\; m_P\; Cos[\theta[t]]x''[t]}{l_P^2\; m_P + I_P} \tag{2.1}$$

$$R_y = -m_P(g{+}l_P\; Cos[\theta[t]]\; \theta'[t]^2{+}Sin[2\,\theta[t]]\; x''[t]{+}Cos[2\,\theta[t]]\; y''[t]{+}l_P\; Sin[\theta[t]]\; \theta''[t]) \tag{2.2}$$

$$R_x = m_P(l_P\ Sin[\theta[t]]\theta'[t]^2 - Cos[2\theta[t]]x''[t] + Sin[2\theta[t]]y''[t] - l_P\ Cos[\theta[t]]\theta''[t]) \tag{2.3}$$

$$\tau = \frac{\eta\ R_g\ (u[t]\ \frac{V_b}{100}\ K_v\ \ - g\ \theta'[t])}{T_s}$$

The terms $x''[t]$ and $y''[t]$ are the accelerations of the pivot point of the spinner. In a fixed pivot, we are considering these to be zero.

2.1.0.2. Basketball Robot

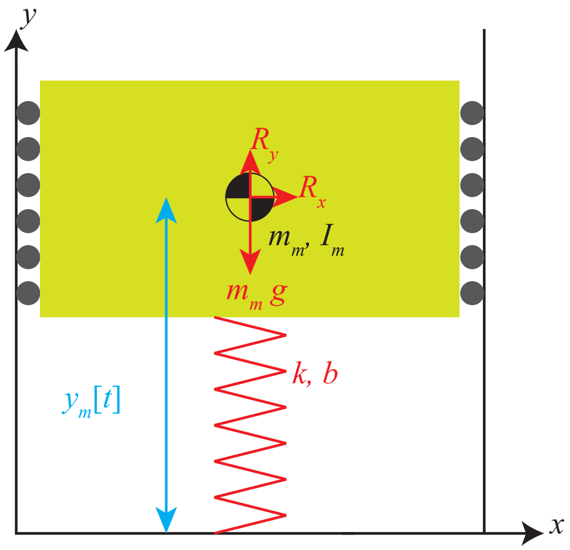

Figure 2.2. Basketball Robot Main Mass FBD

The equation of motion for the Basketball Robot is:

$$m_M\ y''_m[t] + (b\ y'_m[t] + k\ y_m[t])S_c[t] + m_M\ g = R_y \tag{2.4}$$

Where the reaction forces R_x and R_y are the reactions forces with a spinner attached at the center of mass.

2.1.0.3. Tapping Robot

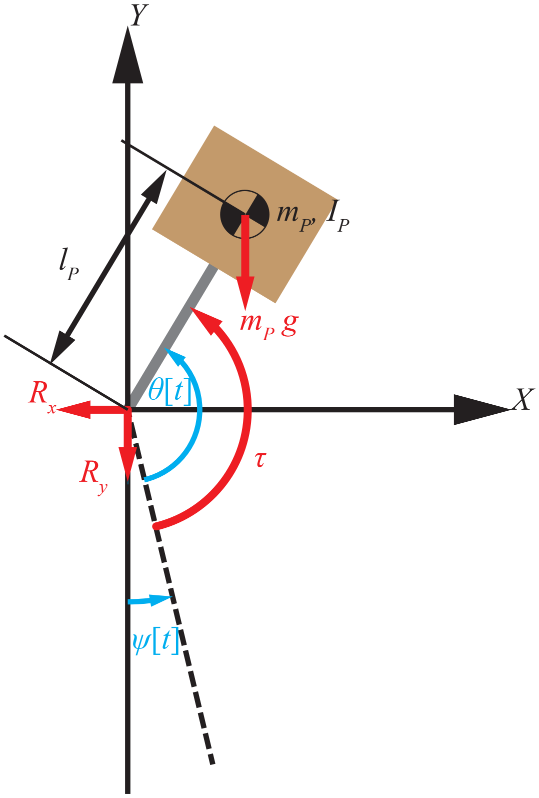

Figure 2.3. Adam's Robot's Spinner FBD

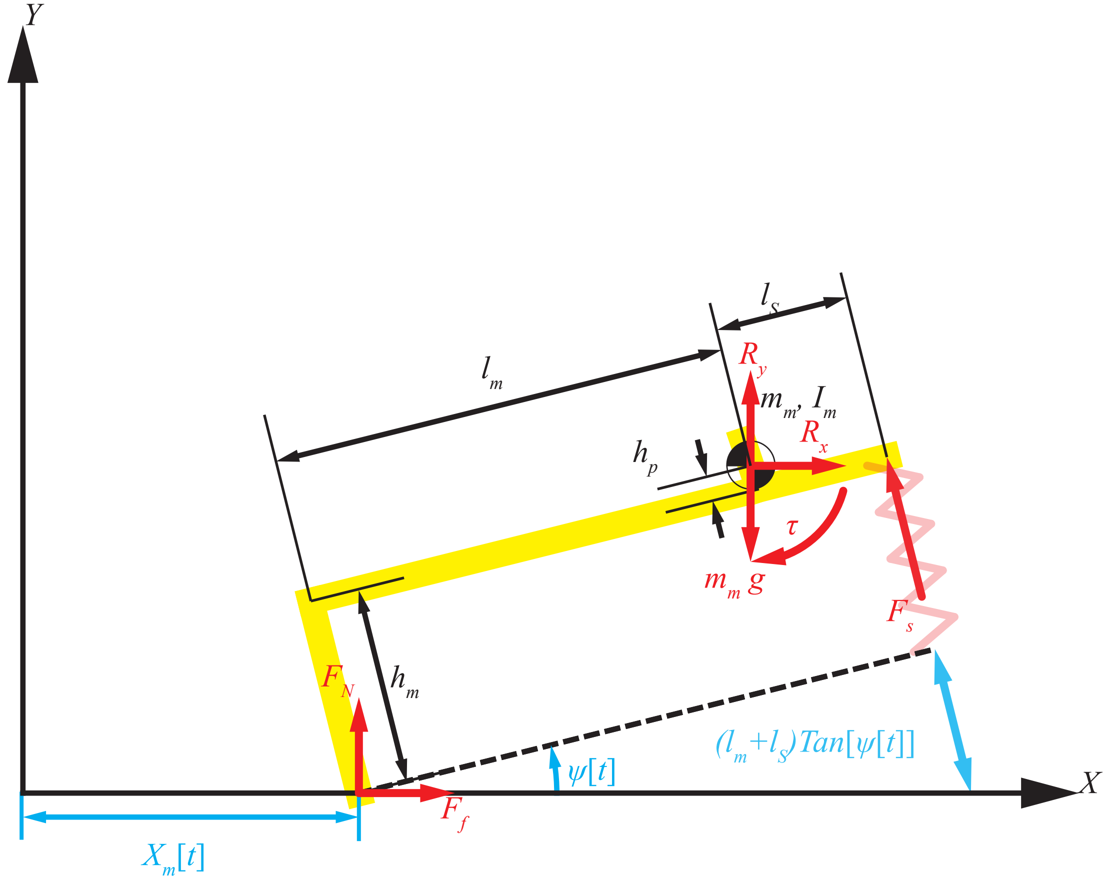

Figure 2.4. Adam's Robot's Main Mass FBD

We are assuming there are two states. The first state occurs when the back pivot point of the robot is fixed and $X'_m[t] = 0$. The second state occurs when the robot's back pivot point is sliding and has overcome static friction at the back pivot point.

Tapping Robot Spinner Dynamics:

$$R_y = -m_P(g + l_P\ Cos[\theta[t] + \psi[t]]\ (\theta'[t] + \psi'[t])^2 + Sin[2(\theta[t] + \psi[t])]\ x''[t]... \\ ... + Cos[2\ (\theta[t] + \psi[t])]\ y''[t] + l_P\ Sin[\theta[t] + \psi[t]]\ (\theta''[t] + \psi''[t])) \tag{2.5}$$

$$R_x = m_P(l_P\ Sin[\theta[t] + \psi[t]](\theta'[t] + \psi'[t])^2 - Cos[2(\theta[t] + \psi[t])]x''[t]... \\ ... + Sin[2(\theta[t] + \psi[t])]y''[t] - l_P\ Cos[(\theta[t] + \psi[t])](\theta''[t] + \psi''[t])) \tag{2.6}$$

Case 1: The robot is stationary ($X'_m[t] = 0$):

$$\psi''[t] = \frac{(l_m + l_s)F_s - \tau - A_1\ Cos[\psi[t]] + A_2\ Sin[\psi[t]]}{I_m + ((h_p + h_m)^2 + l_m^2)m_m} \tag{2.7}$$

$$A_1 = g\ l_m\ m_m + (h_p + h_m)R_x - l_m\ R_y$$

$$A_2 = (-l_m\ R_x\ + (h_p + h_m)(g\ m_m - R_y))$$

$$F_s = -S_c[t](k\ ((l_m + l_s)Tan[\psi] + h_m - S_f) + b\ (l_m + l_s)Sec[\psi[t]]^2\psi'[t])$$

Case 2: The robot is moving ($X'_m[t] \neq 0$):

$$X''_m[t] = \tfrac{1}{t_M} * (\tfrac{A_3}{2m_m} - \tfrac{A_4 * Sin[\theta[t]]}{1} + \tfrac{A_5 * Cos[\theta[t]]}{1} + \tfrac{A_6 * Sin[2\theta[t]]}{2} - \tfrac{A_7 * Cos[2\theta[t]]}{2})$$

$$A_3 = 2F_f(I_m + ((h_p + h_m)^2 + l_m^2)m_m) + ...$$

$$...(2I_m + ((h_p + h_m)^2 + l_m^2)m_m)R_x$$

$$A_4 = ((h_p + h_m)^2 - l_m l_s)F_s + (I_m F_s)/m_m + l_m \tau...$$

$$... + (h_p + h_m)(I_m + ((h_p + h_m)^2 + l_m^2)m_m)\psi'[t]^2$$

$$A_5 = (h_p + h_m)((l_m + l_s)F_s - \tau) + ... \tag{2.8}$$

$$...l_m(I_m + ((h_p + h_m)^2 + l_m^2)m_m)\psi'[t]^2$$

$$A_6 = h_p^2 g\ m_m + 2\ h_p\ g\ h_m\ m_m\ + g\ h_m^2 m_m\ ...$$

$$... - g\ l_m^2 m_m - 2\ h_p\ l_m\ R_x...$$

$$... - 2\ h_m\ l_m\ R_x - (h_p + h_m - l_m)(h_p + h_m + l_m)R_y$$

$$A_7 = 2\ h_p\ g\ l_m\ m_m + 2\ g\ h_m\ l_m\ m_m + h_p^2\ R_x...$$

$$... + 2\ h_p\ h_m\ R_x + h_m^2 R_x - l_m^2 R_x - 2(h_p + h_m)l_m\ R_y$$

$$\begin{gathered}
\psi''[t] = \tfrac{1}{I_1} * (A_8 + A_9 * Sin[\theta[t]] + A_{10} * Cos[\theta[t]] - A_{11} * Sin[2\theta[t]] + A_{12} * Cos[2\theta[t]]) \\
A_8 = l_m\ F_s + 2\ l_s\ F_s - 2\tau \\
A_9 = 2(F_f\ l_m + (h_p + h_m)(g\ m_m - R_y)) \\
A_{10} = 2(F_f\ (h_p + h_m) + l_m(-g\ m_m + R_y)) \\
A_{11} = (h_p + h_m)F_s ... \\
... + (h_p + h_m - l_m)(h_p + h_m + l_m)m_m\ \psi'[t]^2 \\
A_{12} = l_m(F_s + 2(h_p + h_m)m_m\ \psi'[t]^2)
\end{gathered} \tag{2.9}$$

$$\begin{gathered}
F_N = \tfrac{1}{t_M}(A_{13} - A_{14}\ Sin[\theta[t]] - A_{15}\ Cos[\theta[t]] - A_{16}\ Sin[2\theta[t]] + A_{17}(-Sin[\theta[t]]^2 +\ Cos[\theta[t]]^2)) \\
\\
A_{13} = I_m(g\ m_m - R_y) \\
A_{14} = m_m((h_p + h_m)(l_s\ F_s - \tau) + l_m\ I_m\ \psi'[t]^2) \\
A_{15} = I_m\ F_s + l_m\ m_m(-l_s\ F_s + \tau) + (h_p + h_m)I_m\ m_m\ \psi'[t]^2 \\
A_{16} = \tfrac{F_f(h_p + h_m - l_m)(h_p + h_m + l_m)m_m}{2} \\
A_{17} = F_f(h_p + h_m)l_m\ m_m
\end{gathered} \tag{2.10}$$

Where:

$$t_M = I_m + m_m(l_m Cos[\psi[t]] - (h_p + h_m)Sin[\psi[t]])^2$$

$$I_1 = 2\ I_m + ((h_p + h_m)^2 + l_m^2)m_m...$$

$$... - (h_p + h_m - l_m)(h_p + h_m + l_m)m_m\ Cos[2\psi[t]]...$$

$$... - 2(h_p + h_m)l_m\ m_m\ Sin[2\psi[t]]$$

$$F_f = -(F_N + F_s\ Cos[\psi[t]])\ \mu_k\ Sign[X'_m[t]]$$

$$F_s = -(k * \Delta_s[t] + b * \Delta'_s[t])$$

$$\Delta_s[t] = ((l_m + l_s)Tan[\psi] + h_m) - S_f$$

$$\Delta'_s[t] = (l_m + l_s)Sec[\psi[t]]^2\ \psi'[t]$$

2.2. Change in Variables

2.2.1. Basketball Robot

In analysis of the Basketball Robot, changing the states given by the Newton-Euler method to the states $\theta''[t]$ and $y_g''[t]$ can make analysis more clear. By definition of the Center of Mass:

$$y_g[t] = y_m[t] - y_m[0] - \frac{l_P\ m_P\ Cos[\theta[t]]}{m_m + m_P} \tag{2.11}$$

$$y_m[0] = S_f - \frac{g(m_m + m_P)}{k} \tag{2.12}$$

$$y_g'[t] = \frac{l_P\ m_P\ Sin[\theta[t]]\theta'[t]}{m_m + m_P} + y_m'[t] \tag{2.13}$$

$$y_g''[t] = \frac{l_P\ m_P\ (Cos[\theta[t]]\theta'[t]^2 + Sin[\theta[t]]\theta''[t]}{m_m + m_P} + y_m''[t] \tag{2.14}$$

If equations 2.11 to 2.14 are substituted for the equations of motion for the basketball robot in equations 2.1 and 2.4:

$$
\begin{aligned}
y_g''[t] &= -g + S_c[t]\tfrac{F_1}{(m_m+m_P)^2} \\
\theta''[t] &= I_2\ (\tau + \tfrac{l_P^2 m_P^2 Sin[2\theta[t]]\theta'[t]^2}{2(m_m+m_P)} - S_c[t]\tfrac{l_P\ m_P\ F_1\ Sin[\theta[t]]}{(m_m+m_P)^2})
\end{aligned}
\tag{2.15}
$$

where:

$$F_1 = (m_m + m_P)(g(m_m + m_P) - k\ y_g[t] - b\ y_g'[t]) + ...$$

$$...l_P\ m_P(-k\ Cos[\theta[t]] + b\ Sin[\theta[t]]\theta'[t])$$

$$I_2 = \frac{2(m_m + m_P)}{2\ m_m(I_P + l_P^2 m_P) + m_P(2\ I_P + l_P^2 m_P) + l_P^2 m_P^2 Cos[2\theta[t]]}$$

The advantage of this realization is that the vertical state, $y_g''[t]$, is independent of the torque, τ, and the angular acceleration, $\theta''[t]$.

2.2.2. Constant Angular Velocity Assumption

In this paper we will define two steady states for the spinner. We will call the ideal steady state the point in time that the spinner reaches a constant velocity and the rotational acceleration is zero, meaning $\theta''[t] = 0$.

We will call the practical steady state the point in time that the spinner reaches practically a constant velocity and the rotational acceleration is practically zero, meaning $\theta''[t] \approx 0$. This occurs when the nonlinearities are small enough to be ignored.

Our end goal is to reach an ideal steady state, where the spinner is rotating at the resonant frequency of the system to produce the most power. Once the maximum power has been reached, then we will analyze how best to control this power. The design methodology will be partial feedback linearization. The drawback of feedback linearization is a lack of robustness. However, partial feedback linearization is robust in this case because parameter uncertainty slows the system down and does not cause instability. The advantage of feedback linearization is that there is no need for a sensor that can acquire the vertical velocity of the basketball robot. This speeds up the controller because a filter is not required to calculate the vertical velocity. This reduces the cost of the robot because the sensor can have more noise.

Let us define a new state:

$$\xi[t] = y_m[t] - y_m[0] \tag{2.16}$$

Assuming the basketball robot has reached the ideal steady state, the dynamics of the basketball robots height simplify to:

$$(m_m + m_p)\ \xi''[t] + b\ \xi'[t] + k\ \xi[t] = l_P\ m_P\ Cos[\theta[t]]\ \theta'[t]^2 \tag{2.17}$$

With the fact that the ideal angular velocity of the spinner, ω_d, is constant, at the ideal steady state, equation 2.17 becomes:

$$(m_m + m_p)\ \xi''[t] + b\ \xi'[t] + k\ \xi[t] = l_P\ m_P\ Cos[\omega_d\ t]\ \omega_d^2 \tag{2.18}$$

Equation 2.18 mathematically shows what has been intuitively understood. The rotational velocity of the spinners creates a sinusoidal input in the spring-mass-damper system. In a linear system, according to BIBO (Bounded Input Bounded Output), a sinusoidal input creates a sinusoid output. The closer the frequency of the input is to the resonant frequency of the output, the greater the response.

2.2.2.1. Constant Angular Velocity Solution

Equation 2.18 is an ordinary linear differential equation. This can be solved using different basic methods, such as using laplace transforms. For the sake of precision, Mathematica's DSolve function was used with equations 2.19 and 2.20 as initial conditions.

$$\xi[0] = 0 \tag{2.19}$$

$$\xi'[0] = 0 \tag{2.20}$$

As expected for a simple second-order ordinary linear differential equation, the solution is of the following form:

$$\xi[t] = \xi_t[t] + \xi_s[t] \tag{2.21}$$

$$\xi_t[t] = C_t * e^{-\frac{b}{2(m_m+m_P)}t} Sin[\omega_b * t] \tag{2.22}$$

$$\xi_s[t] = \xi_m * Sin[\omega_d * t + \phi] \tag{2.23}$$

$$\xi_m = \sqrt{\frac{l_P^2 m_P^2 \omega_d^4}{b^2\omega_d^2 + (k - (m_m + m_P) * \omega_d^2)^2}} \tag{2.24}$$

$$\phi = Tan^{-1}[\frac{k - (m_m + m_P)\omega_d^2}{b\ \omega_d}] \tag{2.25}$$

Now we can find the resonant frequency of the system using the first and second derivative tests.

$$\frac{d\xi_m}{d\omega_d} = \\ \frac{(\omega_d^2(b^2-2k(m_m+m_P))+2k^2)\frac{l_P^2 m_P^2 \omega_d^4}{b^2\omega_d^2+(k-\omega_d^2(m_m+m_P))^2)^2}^{\frac{3}{2}}}{l_P^2 m_P^2 \omega_d^5} \tag{2.26}$$

Solving:

$$\frac{d\xi_m}{d\omega_d} = 0 \tag{2.27}$$

$$\omega_r = \frac{k}{\sqrt{-(b^2/2) + k(m_m + m_P)}} \tag{2.28}$$

Substituting in the ω_r into the second derivative:

$$\frac{d^2\xi_m}{d\omega_d^2} = \\ \frac{4l_P^2 m_P^2(b^2-2k(m_m+m_P))^3}{b^4(b^2-4k(m_m+m_P))^2\sqrt{-\frac{k^2 l_P^2 m_P^2}{b^4-4b^2k(m_m+m_P)}}} \tag{2.29}$$

The second derivative at the ω_r is always negative because of the material properties of the metal springs used in Interially actuated robots. Because $k >> b$, $2k(m_m + m_P) >> b^2$. Therefore, to find the maximum response for the system, we substitute the ω_r into ξ_m:

$$\xi_r[t] = 2\sqrt{-\frac{k^2 l_P^2 m_P^2}{b^4 - 4b^2 k(m_m + m_P)}} \tag{2.30}$$

Equation 2.30 will be used to compare the actually experimental results to the estimated response from these assumptions. By comparing the maximum response to the simulated results and Equation 2.30, we can measure how close the dynamics of the actual system are to idealized dynamics in this analysis.

Using the parameters from the Basketball robot (Appendix 1), the theoretical maximum response is at 25.7 rad/s, with a maximum response of 0.164 meters. In case the reader has forgotten, this is assuming that the basketball robot never jumps and the springs are always engaged. Obviously, the response changes when the robot loses contact with the ground. When the robot loses contact with the ground, the response is different because the springs no longer affect the dynamics and the effect of the $y_m''[t]$ on the spinner is great enough that the robot never reaches the practical steady state response of this analysis.

One use of this analysis is that our assumptions hold when the robot's response is not sufficiently large to jump. The forces of the main mass on the spinners at this point is insignificant. With this analysis, we can identify what frequency will cause the robot to start jumping or whether the robot will jump at all. If we set the steady state response of equation 2.21 to the spring length, then we can find what frequency will cause the robot to start jumping:

$$\begin{gathered}
\omega_d > \sqrt{\frac{2gk^2(m_m+m_P)}{g(m_m+m_P)(-b^2+2k(m_m+m_P))+\sqrt{B_1+B_2+B_3}}} \\
B_1 = 4k^4 l_P^2 m_P^2 \\
B_2 = b^4 g^2 (m_m + m_P)^2 \\
B_3 = -4b^2 g^2 k (m_m + m_P)^3
\end{gathered} \tag{2.31}$$

For the basketball robot, $\omega_d > 17.4\ rad/s$.

2.3. Tapping Robot: Linearization around the static equilibrium point

The constant angular acceleration assumption can be applied to the tapping robot. The challenge of using feedback linearization on the tapping robot is the complexity of the dynamics. First there are two modes. For a full feedback linearization the controller must be able to switch between the two. Second, the sliding mode, not to be confused with a sliding mode controller, is too complicated to be analyzed without a computer. Three, the spring-mass-damper system's spring force is nonlinear, where the basketball robot moves on a vertical linear guide. Therefore, compromises must be made. The dynamics will be explained according to these compromises.

The first compromise is that the controller will only be designed according to the first mode. In the first mode, the tapping robot is fixed on the back to the ground. We are assuming that the difference between the two modes in dynamics is negligible. In the first mode, The dynamics of the tapping robot, in terms of $\psi''[t]$:

$$\begin{gathered}
\psi''[t] = \tfrac{1}{I_m+((h_p+h_m)^2+l_m^2)m_m}((l_m+l_s)F_s ... \\
... - \tau + g(m_m+m_P)(\sqrt{(h_p+h_m)^2+l_m^2}Sin[\psi[t] ... \\
... - ArcTan[l_m/(h_p+h_m)]] + m_P((-(h_p+h_m)^2 - l_m^2)\psi''[t] ... \\
... + l_P Cos[\theta[t]](-l_m(\theta'[t]+\psi'[t])^2 + (h_p+h_m)(\theta''[t]+\psi''[t])) ... \\
... - l_P Sin[\theta[t]]((h_p+h_m)(\theta'[t]+\psi'[t])^2 + l_m(\theta''[t]+\psi''[t]))))
\end{gathered} \tag{2.32}$$

Where:

$$F_s = -S_c[t](k\ ((l_m+l_s)Tan[\psi] + h_m - S_f) + b\ (l_m+l_s)Sec[\psi[t]]^2\psi'[t])$$

The following transformations, symbolic calculations, and linearization are applied to equation 2.32:

1. We set $\theta''[t]$ and τ to zero because in the constant angular velocity assumption the angular acceleration is zero and, accordingly, the input torque is zero.

2. We replace $\theta'[t]$ as ω_d, which is the desired angular velocity achieved at the ideal steady state.

3. We replace $\theta[t]$ as $\omega_d\ t$ because the angular velocity is constant.

4. Solve equation 2.32 with the replacements from steps 1 to 3.

5. Replace F_s with its definition.

6. Linearize the resulting solution about $\psi[t] = 0$ because this is approximately the static equilibrium state for the tapping robot.

The resulting equation:

$$\begin{gathered}
\psi''[t] = (S_f\ k\ (l_m + l_s) - h_m\ k\ (l_m + l_s) - g\ l_m\ (m_m + m_P)... \\
... - l_m\ l_P\ m_P\ \omega_d^2 Cos[t\omega_d]... \\
... + (-k\ (l_m + l_s)^2 + g\ (h_p + h_m)(m_m + m_P))\psi[t]... \\
... - b\ (l_m + l_s)^2 \psi'[t])/(I_m + ((h_p + h_m)^2 + l_m^2)(m_m + m_P))
\end{gathered} \tag{2.33}$$

Following the steps of section 2.2.2.1, the transition frequency of the tapping robot is 35.3 rad/s and the resonant frequency = 49.0 rad/s.

Chapter 3

CONTROLLER DESIGN

3.1. Design Specification

The design specification for the 3 systems are:

3.1.1. Fixed Pivot

Identify the parameters of the dynamics for a total feedback linearization.

3.1.2. Basketball Robot

Achieve a stable limit cycle that results in a constant jump height.

3.1.3. Tapping Robot

Achieve a stable limit cycle that results in a constant average translational velocity.

3.2. Off-Line Spinner Parameter Identification

The success of the controllers in this paper depends on the accuracy of the estimated parameters of the partial feedback linearization. The first step in estimating the required parameters is to find the parameters isolated to the spinner. To find these parameters, a Self-Tuning Adaptive Controller is used. One adavantage of the control scheme provided in this paper is that angular acceleration is not used. The desired trajectory of the spinners is a constant velocity. Therefore, only terms in the spinners' dynamics that effect the steady state angular velocity are required to be known.

In finding the spinner's parameters, we assume the spinner's pivot is fixed and the dynamics of the spinner are equation 2.1. Meaning the dynamics of the spinner can be approximated to:

$$(I_p + l_p^2 m_p)\theta''[t] + l_p m_p g Sin[\theta[t]] = \tau \tag{3.1}$$

In experimentation, there is an offset of origin for $\theta[t]$, such that $\theta[t] = \hat{\theta}[t] + \delta\theta$. The state $\hat{\theta}[t]$ is the output of angle measured by the encoder. The dynamics can be expanded as follows:

$$Sin[\hat{\theta}[t] + \delta\theta] = Cos[\hat{\theta}[t]]Sin[\delta\theta] + Cos[\delta\theta]Sin[\hat{\theta}[t]] \tag{3.2}$$

Substituting the offset into the dynamics and solving for the u[t] signal from the MCU as the input:

$$\boldsymbol{\phi}^T \boldsymbol{f[t]} = u[t] \tag{3.3}$$

Where:

$$\boldsymbol{\phi} = \begin{bmatrix} \frac{I_p^*}{C_1 C_2} \\ \\ \frac{gR}{C_2} \\ \\ \frac{g l_p m_p Cos[\delta\theta]}{C_1 C_2} \\ \\ \frac{g l_p m_p Sin[\delta\theta]}{C_1 C_2} \end{bmatrix} \tag{3.4}$$

$$\boldsymbol{f[t]} = \begin{bmatrix} \theta''[t] \\ \\ \theta'[t] \\ \\ Sin[\theta[t] - \delta\theta] \\ \\ Cos[\theta[t] - \delta\theta] \end{bmatrix} \tag{3.5}$$

Where:

$$C_1 = \frac{\eta R_g}{T_s}$$

$$C_2 = \frac{V_b}{100} K_v \tag{3.6}$$

$$I_p^* = I_p + l_p^2 m_p$$

Let us consider the following lemma [15]:

Lemma 1: Consider two signals e and $\boldsymbol{\phi}$ related by the following dynamic equation:

$$e[t] = H[s](k\boldsymbol{\phi^T[t]}\boldsymbol{v}[t]) \tag{3.7}$$

where $e[t]$ is a scalar output signal, $H[s]$ is a strictly positive real transfer function, k is an unknown constant with known sign, $\phi[t]$ is an $m \times 1$ vector function of time, and $\boldsymbol{v[t]}$ is a measurable $m \times 1$ vector. If the vector $\boldsymbol{\phi}$ varies according to

$$\boldsymbol{\phi'[t]} = -sgn[k]\gamma e[t]\boldsymbol{v}[t] \tag{3.8}$$

with γ being a positive constant, then $e[t]$ and $\boldsymbol{\phi}[t]$ are globally bounded. Furthermore, if $\boldsymbol{v}[t]$ is bounded, then $e[t] \to 0$ as $t \to \infty$.

Let us define the input $u[t]$ as:

$$u[t] = \hat{\boldsymbol{\phi}}^T \boldsymbol{f[t]} - \lambda e[t] \tag{3.9}$$

Where:

$$\hat{\boldsymbol{\phi}} = \begin{bmatrix} 0 \\ \hat{a}_2[t] \\ \hat{a}_3[t] \\ \hat{a}_4[t] \end{bmatrix} \tag{3.10}$$

$$e[t] = \hat{\theta}'[t] - \omega_d \tag{3.11}$$

Let us substitute the input in equation 3.9 into equation 3.3:

$$\lambda e[t] = \tilde{\boldsymbol{\phi}}[t]^T \boldsymbol{f}[\boldsymbol{t}] \tag{3.12}$$

Where:

$$\tilde{\boldsymbol{\phi}}[t] = \hat{\boldsymbol{\phi}}[t] - \boldsymbol{\phi} \tag{3.13}$$

Let us define an adaptation law:

$$\tilde{\boldsymbol{\phi}}'[t] = \boldsymbol{\gamma} e[t] \boldsymbol{f}[\boldsymbol{t}] \tag{3.14}$$

Where:

$$\boldsymbol{\gamma} = \begin{bmatrix} 0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_2 \end{bmatrix} \tag{3.15}$$

$\boldsymbol{\gamma}$ are the feedback gains of the adaptation law. γ_2 is repeated because the adaptation of $\hat{a}_3[t]$ is very similar to $\hat{a}_4[t]$, as the sine function is the cosine function with a shift of $\frac{\pi}{2}$. Therefore, the difference in learning settling time of $\hat{a}_3[t]$ and $\hat{a}_4[t]$ should be insignificant.

Lemma 1 can be applied to this controller. $H[s]$ and k are unity. Therefore, with this controller, $\theta'[t] \to \omega_d$ as $t \to \infty$. Even more important, because $\theta'[t]$, $Sin[\theta[t]]$, and $Cos[\theta[t]]$ are independent of each other, having a desired trajectory that is a constant speed results in a convergence of $\hat{\boldsymbol{\phi}}$ to the actual parameters of $\boldsymbol{\phi}$, excluding the first element.

By using this controller, off-line of the actual operation, the parameters for the partial feedback linearization of the control law in equation 3.9 can be found!

3.3. Partial-State Feedback Linearization Controller Design for During Spring Contact

To reach a practical steady state, a state feedback linearization must be used to cancel out the significant nonlinearities.

Let us consider when the robot is in contact with the springs. In section 3.2, we found the parameters dependent only on the spinner. With $\hat{\boldsymbol{\phi}}$, we can cancel out the spinner-dependent nonlinearities. A block diagram of this feedback linearization controller is shown in figure 4. [20]

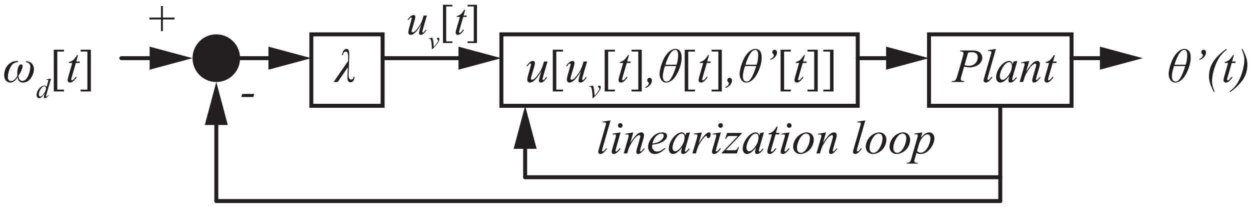

Figure 3.1. Spring Contact Basketball Robot Controller

Where:

$$u_v[t] = \lambda(\omega_d[t] - \theta'[t])$$
$$u[t] = \hat{\boldsymbol{\phi}}\boldsymbol{f[t]} - u_v[t] \tag{3.16}$$

The controller takes the angular position and velocity of the spinner and feeds into the input voltage. This cancels out the gravity term from the spinner's dynamics and the internal damping of the motor. Let us assume the effect of the $mHeight[t]$ is negligible, meaning $mHeight''[t] \rightarrow 0$. The resulting dynamics of the spinner are:

$$\theta''[t] + \lambda^*(\theta'[t] - \omega_d) = 0 \tag{3.17}$$

Where:

$$\lambda^* = \frac{C_1 C_2}{I_p^*}\lambda \tag{3.18}$$

The parameters in equation 3.18 correspond to the parameters in equation 3.5.

Assuming that we have perfect knowledge of the parameters of the system and $bodyX''[t]$ and $bodyY''[t]$ are insignificant, the spinner's dynamics become linear. At steady state, the response in equation 2.21 can be used.

3.3.1. Stability Analysis when in Contact with Spring

We will use the Direct Lyapunov Energy Method to analyze the stability of the basketball robot, while in contact with the springs. Wu [26] used a similar method in analyzing a two-link robot. Let equation 3.19 be the total mechanical energy of the basketball robot while in contact with the springs:

$$E[t] = T_c + V_c \tag{3.19}$$

$$T_c = \frac{1}{2}(m_m + m_p)y_G'[t]^2 + \frac{l_p^2 m_p^2 Cos[\theta^{[}t]]^2 \theta'[t]^2}{(m_m + m_p)^2} + \frac{1}{2}I_p \theta'[t]^2 \tag{3.20}$$

$$V_c = (m_m + m_p)g y_G[t] + \frac{1}{2}k y_m[t]^2 \tag{3.21}$$

The second term in equation 3.20 is the horizontal kinetic energy of the spinner.

Let us guess that the total energy of the system is an invariant set. Meaning:

$$\frac{dE[t]}{dt} = 0 \tag{3.22}$$

Taking the integral:

$$\int_{t[0]}^{t} \frac{dE[t]}{dt} dt = 0 \tag{3.23}$$

$$E[t] - E[0] = 0 \tag{3.24}$$

Let us consider the following Lyapunov Candidate:

$$V[t] = (E[t] - E[0])^2 \tag{3.25}$$

The Lyapunov Candidate corresponds to the distance from the "guessed" invariant set. Because it is squared, it is positive definite. Let us test its convergence by taking the derivative.

$$\frac{dV[t]}{dt} = 2(E[t] - E[0])E'[t] \tag{3.26}$$

We know that $(E[t] - E[0])$ is positive definite. For $V[t]$ to show stability, $E'[t]$ must be negative definite.

$$E'[t] = k(-S_f + y_m[t])y'_m[t] + (m_m + m_p)y'_G[t]\,(g + y''_G[t]) + I_p\theta'[t]\theta''[t] \tag{3.27}$$

For the basketball robot, the following is $E'[t]$ with the parameters substituted in:

$$\begin{gathered} E'[t] = -5y_G'(t)^2... \\ ... + \theta'(t)\sin(\theta(t))\left(0.0889y_G'(t) - 21.0y_G(t) - 0.374\cos(\theta(t)) + 0.314\right) \end{gathered} \tag{3.28}$$

Because the dynamics of the system are nonlinear, equation 3.28 can only be analyzed numerically. The simulations will show that:

1. $\theta'[t]$ is always positive.

2. If $Sin[\theta[t]] > 0$, then $y_G[t] > 0$ and $y_G'[t] < 0$.

3. If $Sin[\theta[t]] < 0$, then $y_G[t] < 0$ and $y_G'[t] > 0$.

Therefore, equation 3.26 is negative definite. The total energy is an invariant set!

3.3.2. Stability Analysis when Flying

The total energy when the basketball robot has lost contact with the ground and is flying:

$$E[t] = T_f + V_f \tag{3.29}$$

$$T_f = \frac{1}{2}(m_m + m_p)y_G'[t]^2 + \frac{1}{2}I_p\theta'[t]^2 \tag{3.30}$$

$$V_f = (m_m + m_p)gy_G[t] \tag{3.31}$$

This is positive definite.

The power of the basketball robot when flying:

$$E'[t] = I_p\theta'(t)\theta''(t) \tag{3.32}$$

From the change of variables in equation 2.15, we know that the dynamics of the basketball robot are only accessible through $\theta''[t]$. When flying, the basketball robot no longer converges to the invariant set found when in spring contact. Without the effect of the spring, $y''_G[t]$ and $\theta''[t]$ are no longer being affected by the springs. This leaves the spinner to accelerate to the steady state speed. The lack of convergence can be seen in the difference between the power, while in spring contact, in equation 3.27 and the power, while flying, in equation 3.32.

This problem can be fixed! Though we cannot artificially create a spring force with the spinner, we can artificially change the power of the basketball robot to converge to the invariant set. This method is actually ideal, because the robot is able to have a much larger response without the springs and jump.

Due to the complexity of the calculations and equations used here, the results are only provide for this subsection. All calculations, symbolic and numerical, were performed on Mathematica.

First, we set the power of the basketball robot during flight and in contact with the springs to be equal. We also define two different spinner dynamics. We have $\theta''_c[t]$ for the spinner dynamics while in contact with the springs. We have $\theta''_f[t]$ for the spinner dynamics while in flight

$$\begin{gathered} k(-S_f + y_m[t])y'_m[t] + (m_m + m_p)y'_G[t]\,(g + y''_G[t]) + I_p\theta'[t]\theta''_c[t] \\ = \\ I_p\theta'(t)\theta''_f(t) \end{gathered} \tag{3.33}$$

Next, we take equation 2.15 and define $\theta''_c[t]$ and $\theta''_f[t]$:

$$\theta''_c[t] = rI(\tau + \frac{l_p^2 m_p^2 Sin[2\theta[t]]\theta'[t]^2}{2(m_m + m_p)} - \frac{l_p m_p F_1 Sin[\theta[t]]}{(m_m + m_p)^2}) \tag{3.34}$$

$$\theta_f''[t] = rI(\tau + sp[\theta[t], \theta'[t], y_G[t], y_G'[t]] + \frac{l_p^2 m_p^2 Sin[2\theta[t]]\theta'[t]^2}{2(m_m + m_p)}) \tag{3.35}$$

$sp[\theta[t], \theta'[t], y_G[t], y_G'[t]]$ is a "spring compensator" that will correct the difference between the two powers.

Now, we solve for $sp[\theta[t], \theta'[t], y_G[t], y_G'[t]]$.

$$sp[\theta[t], \theta'[t], y_G[t], y_G'[t]] = \frac{kl_p m_p(S_f - y_m(t))\sin(\theta(t))}{m_m + m_p} \tag{3.36}$$

In solving equation 3.33, any terms being multiplied by I_p were dropped, because in the simulation we have $I_p = 0$. Also, b = 0 because b is related to $mHeight'[t]$ and we are choosing to ignore this term because of sensor noise.

Now, when the basketball robot looses contact with the ground, the spring compensator, $sp[\theta[t], \theta'[t], y_G[t], y_G'[t]]$, is engaged and the system does not drift as far from the invariant set. Due to the fact that this system is highly nonlinear, the performance of the spring compensator can only be fully analyzed by simulations, for a given system.

This leaves the final feedback linearization controller for the basketball robot:

$$u[t] = \hat{\boldsymbol{\phi}}\mathbf{f[t]} + (1 - S_c[t])\frac{kl_p m_p(S_f - y_m(t))\sin(\theta(t))}{m_m + m_p} - u_v[t] \tag{3.37}$$

The parameters are found offline. The parameters for $\hat{\boldsymbol{\phi}}$ are found using the adaptive feedback system identification from section 3.2. The parameters for the spring compensator are found by tuning.

3.4. Controller Applied to Tapping Robot

A shortcut to the feedback linearization designed in the previous section (section 3.3.1) is to simply replace the spring force term in the spinner dynamics with a virtual spring force, when the springs have lost contact with the ground. In order to do so, the following design algorithm must be used:

1. Assign two different systems. The first system, $\theta_c''[t]$ is the dynamics of the spinner, while in contact with the ground. The second system, $\theta_f''[t]$ is the dynamics of the spinner, while the springs are not in contact with the ground.
2. Define the input as:

$$u[t] = \hat{\boldsymbol{\phi}}\mathbf{f[t]} + (1 - S_c[t])sp[\theta[t], \theta'[t], \Delta_s[t]] - u_v[t]$$

3. Substitute the input into the two systems, with $S_c[t]$ defined accordingly.
4. Solve for $sp[\theta[t], \theta'[t], \Delta_s[t]]$.
5. Set the damping constant to zero in the solution because we are not sensing for $\psi'[t]$.

For the tapping robot:

$$\begin{gathered} sp[\theta[t], \theta'[t], \Delta_s[t]] \\ = \\ -((k(l_m + l_s)(-S_f + h_m + (l_m + l_s)tan(\psi(t)))(-l_p m_p(h_p + h_m)cos(\theta(t))... \\ ... + l_m l_p m_p sin(\theta(t)) + I_p + l_p^2 m_p))... \\ .../(C_1 K_v(m_m((h_p + h_m)^2 + l_m^2) + m_p((h_p + h_m)^2 + l_m^2 + l_p^2)... \\ ... - 2l_p m_p(h_p + h_m)cos(\theta(t)) + 2l_m l_p m_p sin(\theta(t)) + I_m + I_p))) \end{gathered} \tag{3.38}$$

For the tapping robot, with the parameters substituted:

$$sp[\theta[t], \theta'[t], \Delta_s[t]] = -\frac{(6250(tan(\psi(t))+0.0560)(-2.76sin(\theta(t))+cos(\theta(t))-0.978)}{-2.76sin(\theta(t))+cos(\theta(t))-132)}\% \tag{3.39}$$

Chapter 4

SIMULATION RESULTS

Wolfram Mathematica was used for simulations.

4.1. Off-Line Spinner Parameter Identification

In the simulation of the off-line parameter identification, the parameters for the spinner in the tapping robot were used (See appendix A.1). Other parameters include:

γ_1	0.02
γ_2	0.2
λ	10
ω_d	5 rad/s
$\theta(0)$	0 rad
$\theta'(0)$	0 rad/s

Mathematica's NDSolve function was used for numerically solving the differential equation. All options were used at their default setting. The simulation time was set for 150 seconds. This is the amount of time the experiment took to finish. The sensor feedback and control law was continuous. The performance of the simulation is desired to be similar to the performance of the experiment.

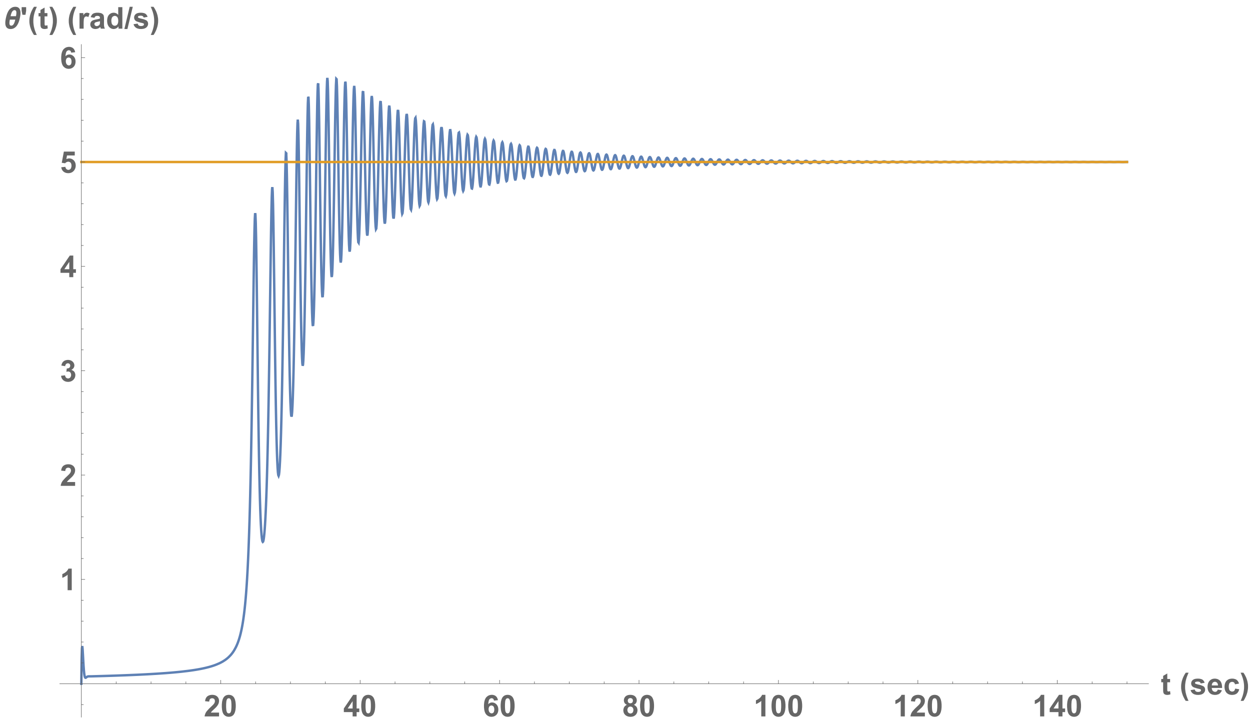

Figure 4.1. Spinner Velocity Convergence Simulation

Figure 4.1 shows the spinner velocity. The spinner velocity oscillates because the force of gravity oscillates with the rotation. The oscillation is eliminated as the feedback linearization cancels gravity. At about 120 seconds the velocity converges to the desired velocity.

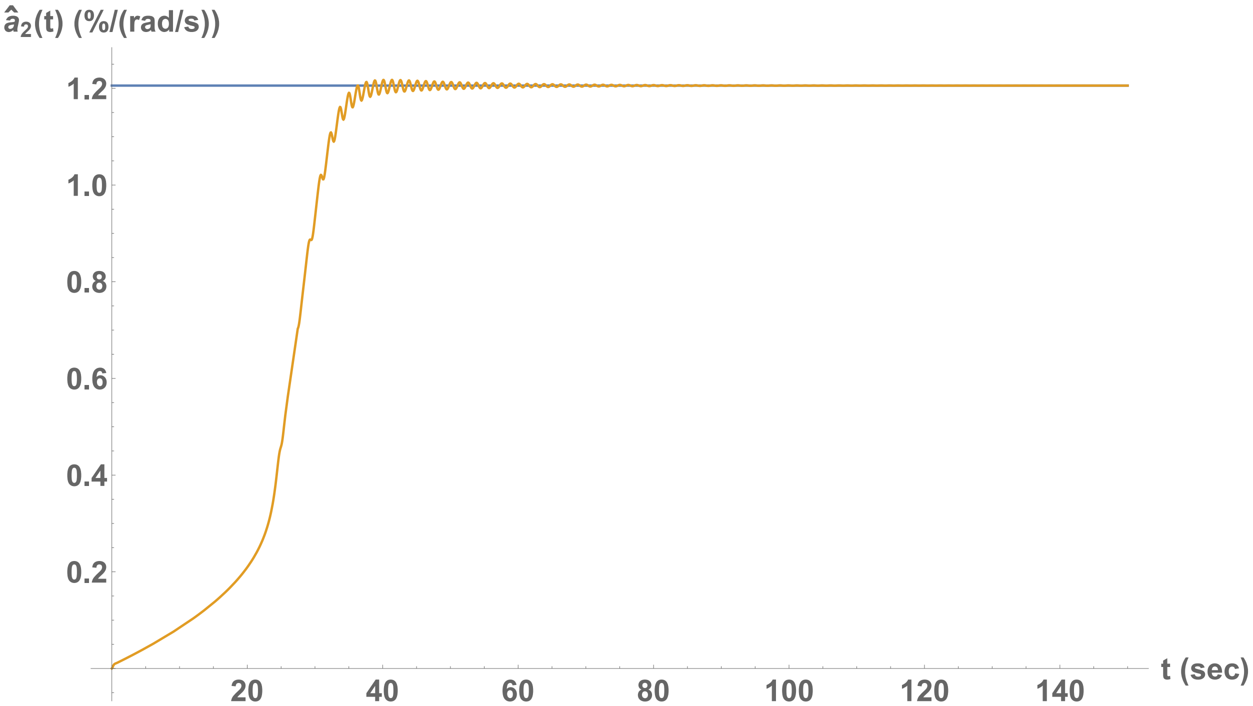

Figure 4.2. $\hat{a}_2(t)$ Convergence Simulation

Figure 4.2 shows the estimated $\hat{a2}(t)$ parameter. This parameter represents the effect of damping of the spinner. In the model, the only damping considered was the internal damping of the motor. The damping of the motor converges at about 80 seconds.

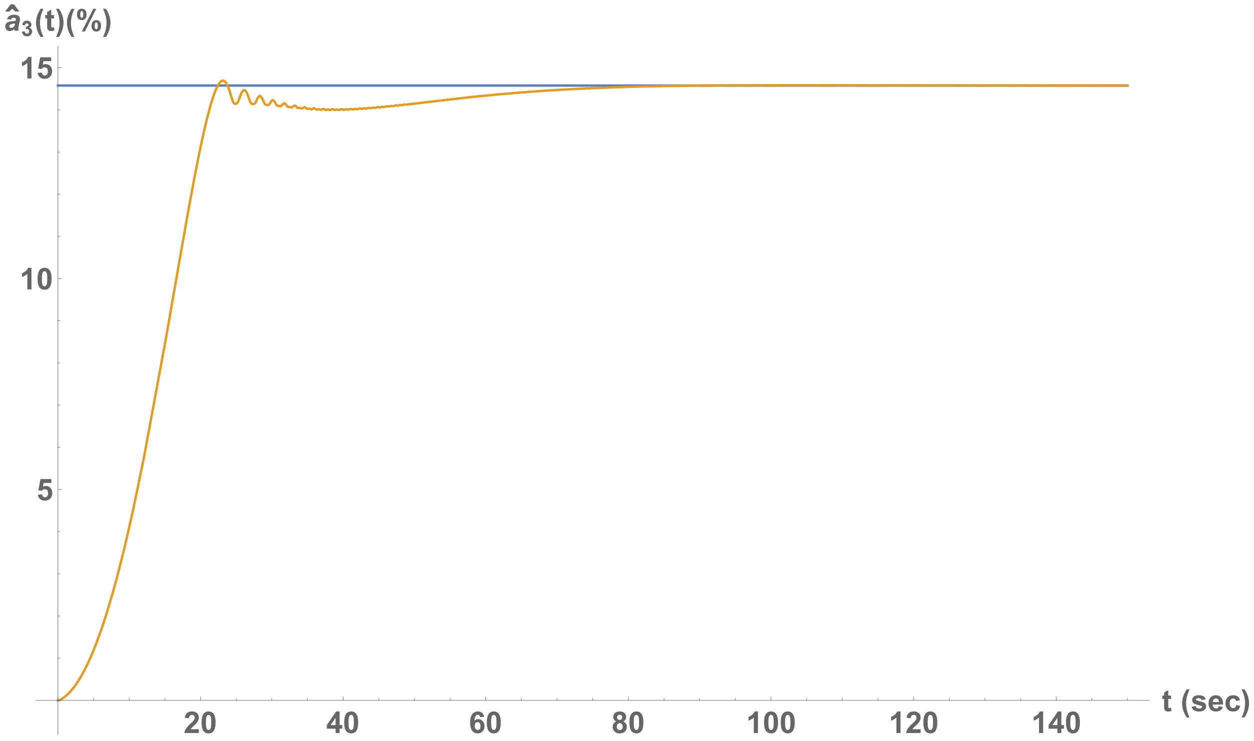

Figure 4.3. $\hat{a}_3(t)$ Convergence Simulation

Figure 4.3 shows the estimated $\hat{a3}(t)$ parameter. This parameter represents the effect of the $Sin[\theta[t]]$ term of the spinner. The parameter of the $Sin[\theta[t]]$ term converges at about 90 seconds.

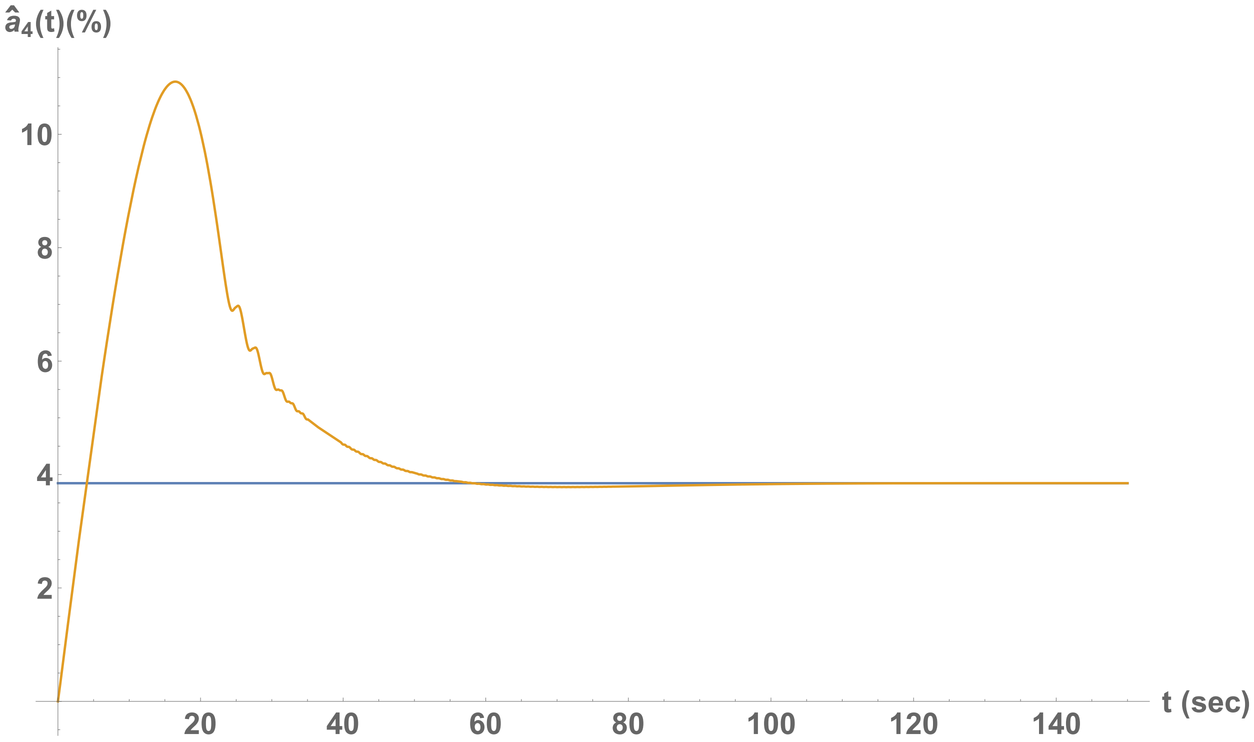

Figure 4.4. $\hat{a}_4(t)$ Convergence Simulation

Figure 4.4 shows the estimated $\hat{a}4(t)$ parameter. This parameter represents the effect of the $Cos[\theta[t]]$ term of the spinner. The parameter of the $Cos[\theta[t]]$ term converges at about 100 seconds. The large jump in the beginning was caused by an intial period of acceleration. Once the spinner was able to reach a certain minimum velocity, the $Cos[\theta[t]]$ term converges.

4.2. Basketball Robot Simulation Results

The simulation section of the Basketball Robot is structured according to increasing the sophistication of the controller design until the jumping of the Basketball Robot is stabilized.

4.2.1. Effects of Partial Linearization, While In Contact With The Springs

Let us first analyze the dynamics of the Basketball Robot while it is in contact with the springs, meaning it never jumps. The first simulation shows the results of the Basketball robot when the $\hat{\boldsymbol{\phi}}\mathbf{f}[\mathbf{t}]$ term is ignored. The second simulation shows the results when the partial linearization term is not ignored. The two simulations are combine into one plot in figure 4.5.

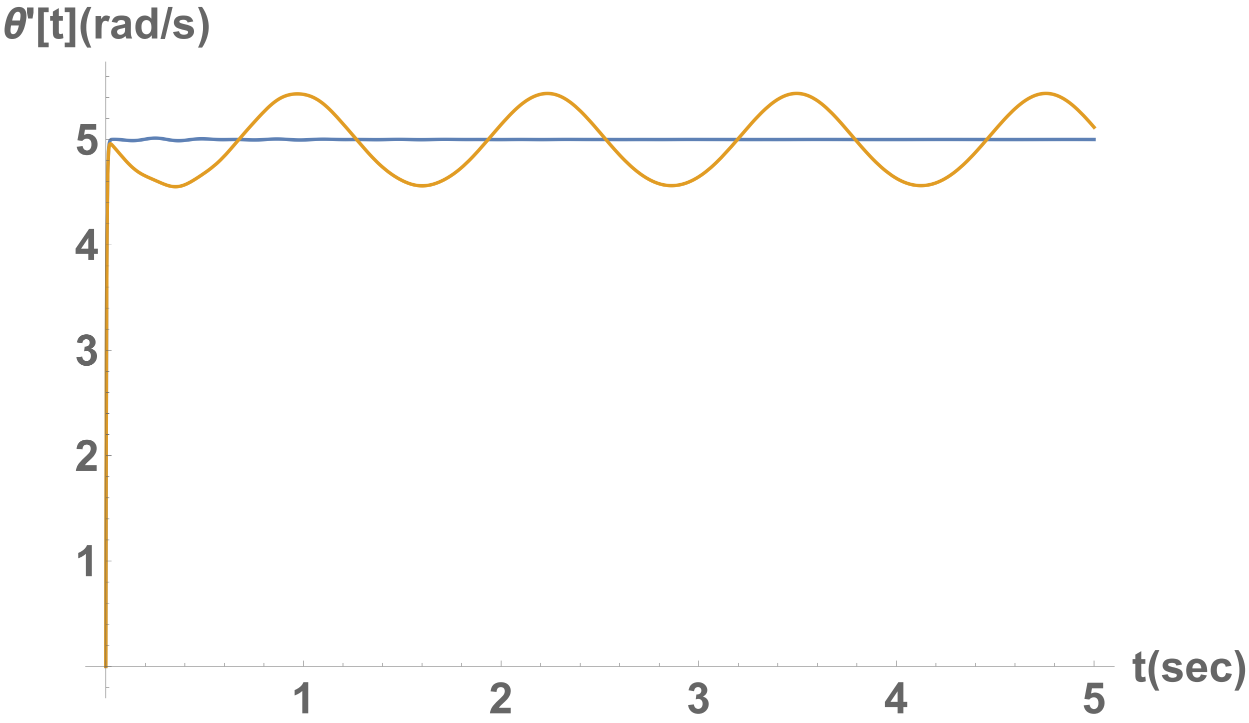

Figure 4.5. Comparison of Rotational Velocity at Low Speeds

For the simulations in figure 4.5, $\omega_d = 5\ rad/sec$ and $\lambda = 280$. The yellow curve is the response of the Basketball Robot when the partial feedback linearization is ignored, and the blue curve is when it is not ignored. Even at a high proportional controller gain, the effect of gravity is evident. However, the effect is less at higher speeds.

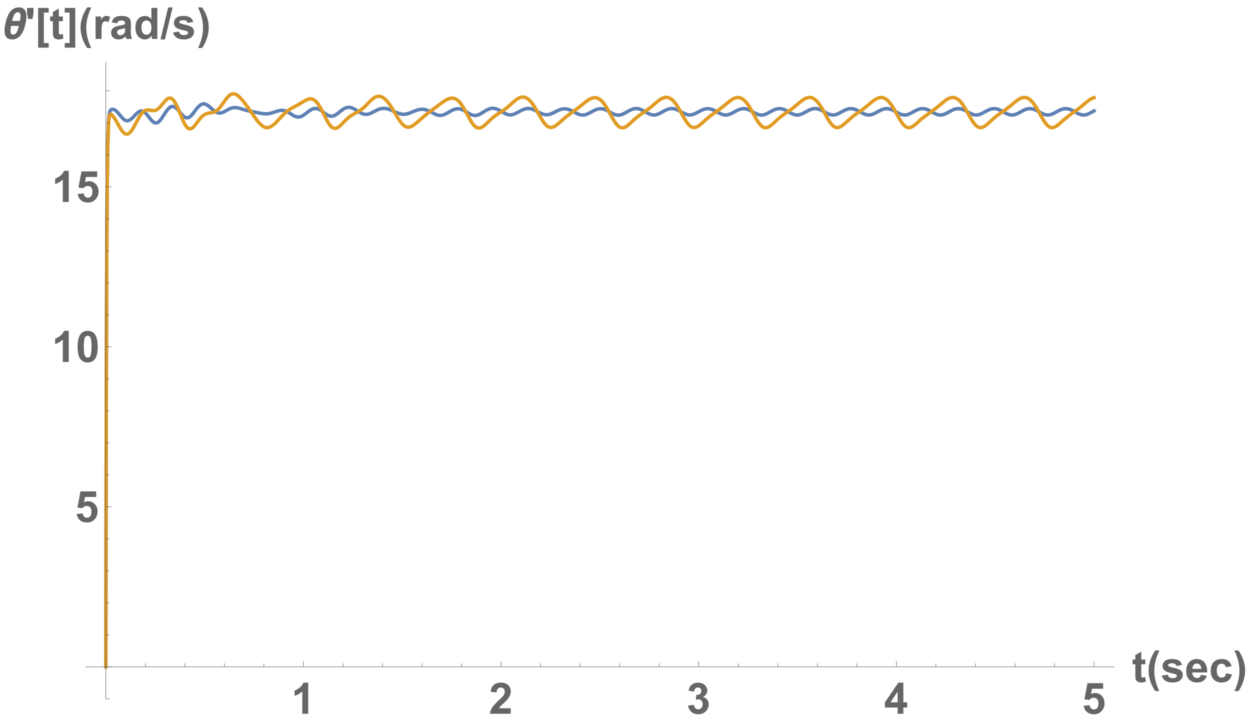

Figure 4.6. Comparison of Rotational Velocity at High Speeds

In figure 4.6, the basketball robot's controller was set for $\omega_d = 17.4\ rad/sec$ and $\lambda = 280$. This plot shows that at high speeds the effect of gravity on the spinners is much less.

4.2.1.1. Response at transition to jump

Let us now look at how accurate our simplified model is to a simulation.

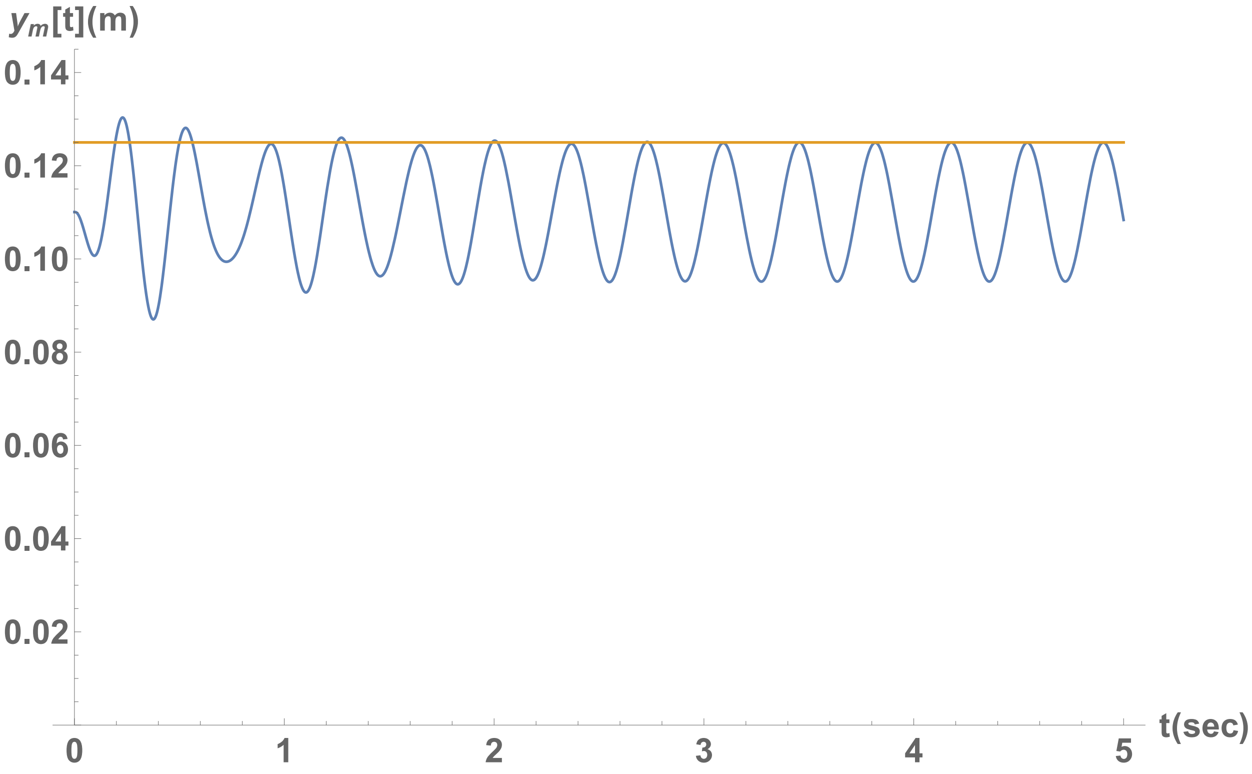

Figure 4.7. Basketball Height Response at Spring Contact Transition Frequency

Figure 4.7 shows the sinusiodal response of the basketball robot. The yellow line is the FSL. Figure 4.7 and Figure 4.6 are from the same simulation. Figure 4.7 shows how close our simplified model is to the actual response of the basketball robot. Our model fails in the transient because our model assumed the robot to be instantaneously at ω_d. In the reality, the acceleration of the spinners creates a transient second input to the system. The transient response only occurs with in one second and is not considered significant to the overall response of the robot.

The ω_d chosen for the second simulation in this section was calculated from equation 2.31. The actual response amplitude is 0.1250 meters. The response deviates from the expected response, which is the S_f, by $2.31*10^{-5}$ meters. This is a 0.0185% error. This is insignificant.

Let us look at the total mechanical energy. In section 3.3.1, we predicted that the total mechanical energy, defined in the section, would be an invariant set. Figure 4.8 shows the total mechanical energy simulated.

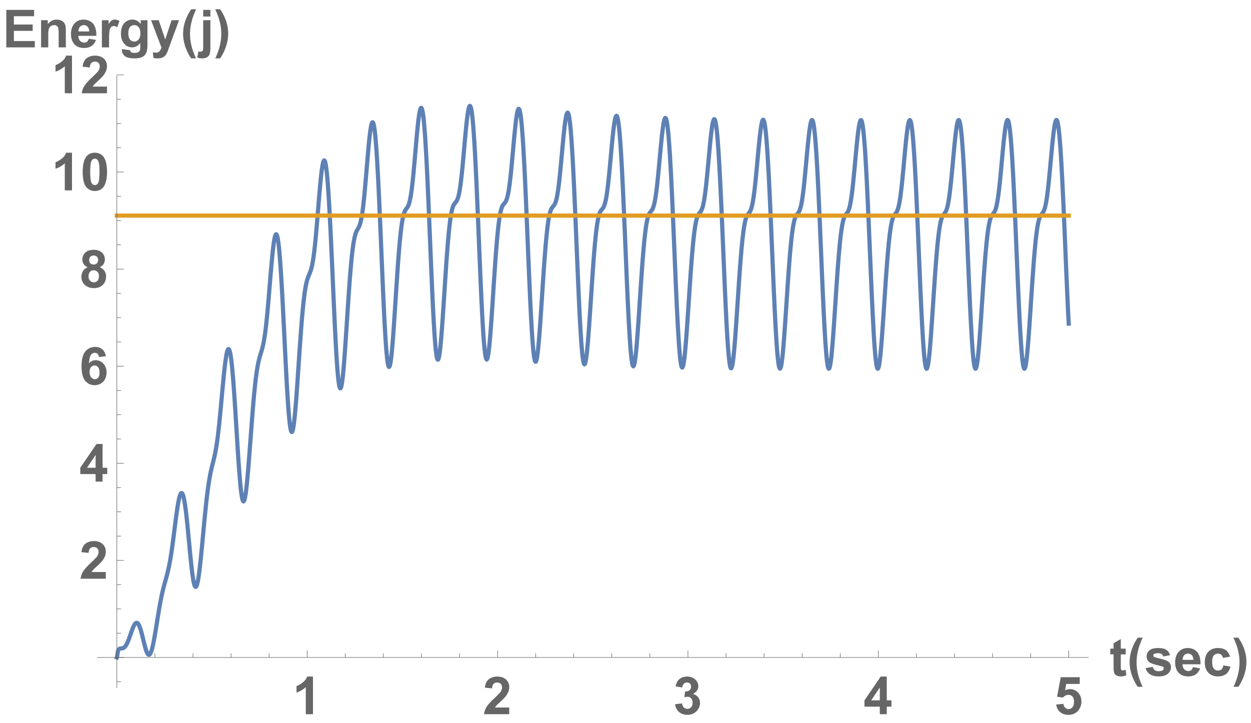

Figure 4.8. Total Mechanical Energy Response at Spring Contact Transition Frequency

The simulation results in figure 4.8, show the accuracy of this prediction. The total mechanical energy does converge, but it converges to an oscillation. Therefore, the invariant set defined in section 3.3.1 is not exact. The results show that there is an invariant set, but we have yet to find it and there is future work to be done to find it.

4.2.1.2. Response at resonant frequency, and why our controller fails

When we change ω_d to 25.7 rad/s, which is the resonant frequency calculated from equation 2.30, we get the following response for the spinners velocity and $mHeight[t]$:

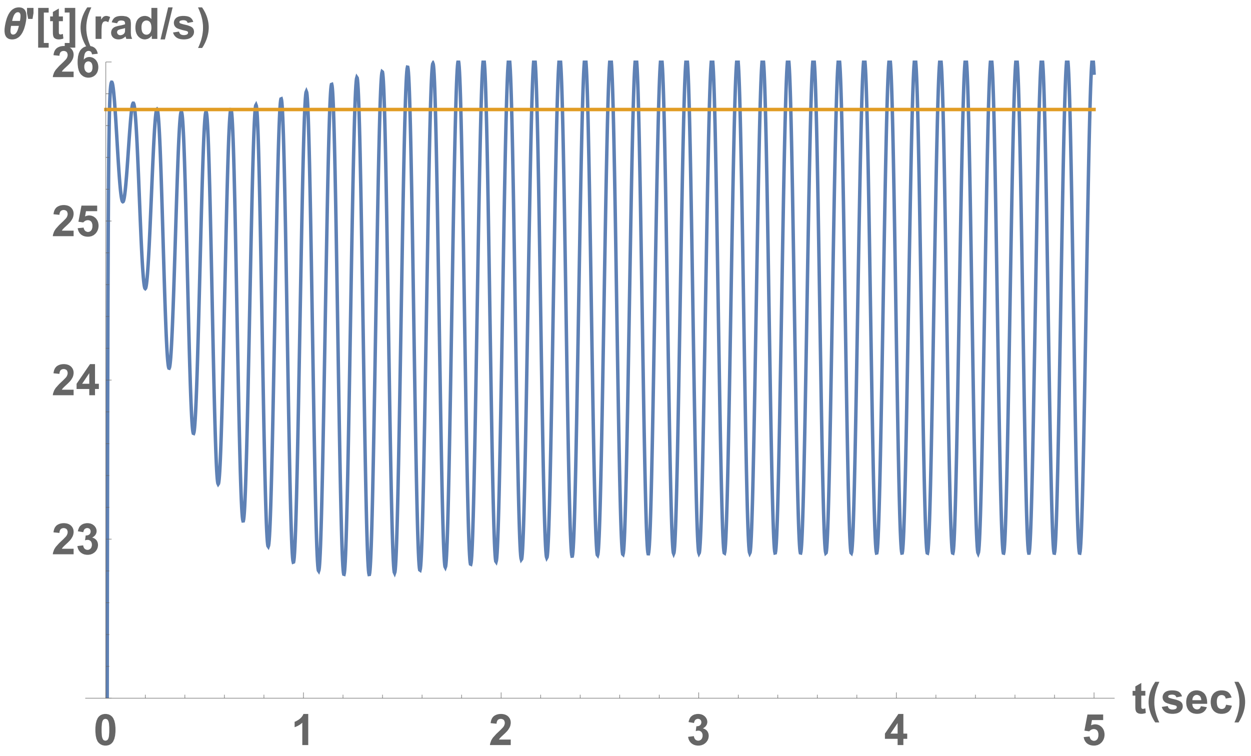

Figure 4.9. Spinner Response at Resonant Frequency

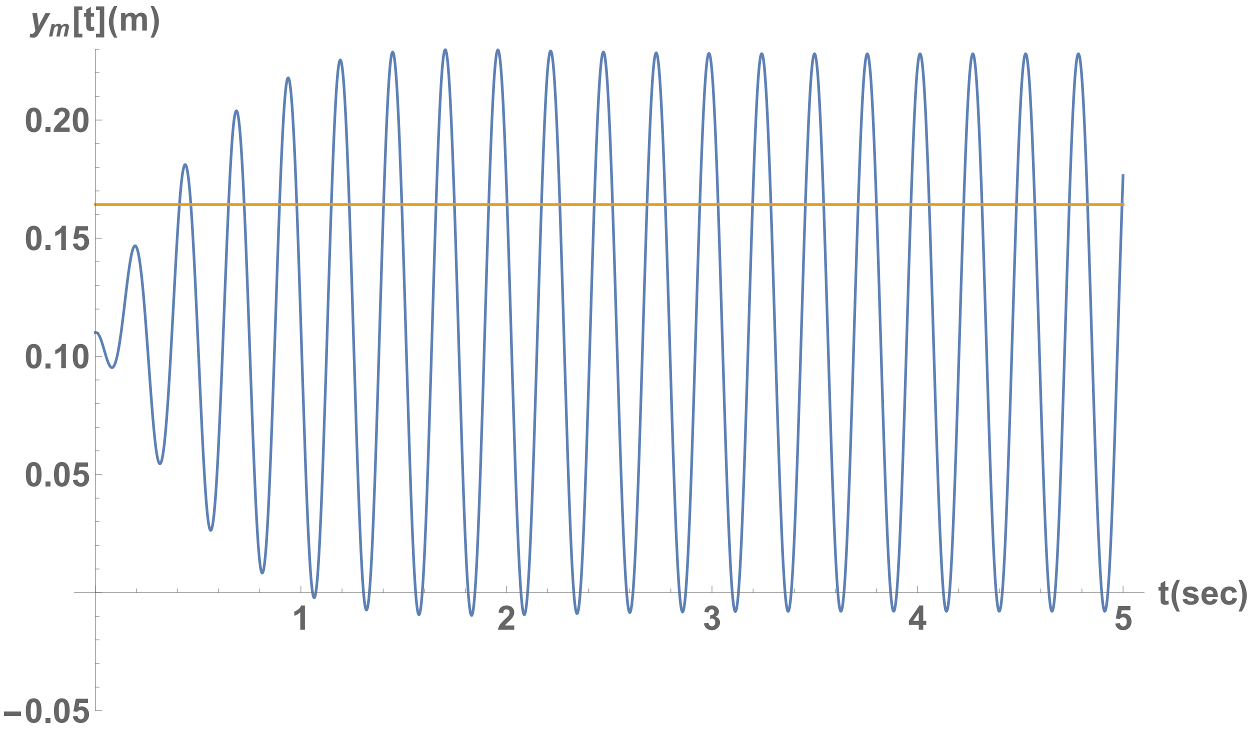

Figure 4.10. Basketball Height Response at Resonant Frequency

Figure 4.9 shows how much the actual spinner velocity deviates from the model. The yellow line shows ω_d. In this case, the spinner has a maximum error of 2.79 rad/s or 12.2% of the modeled response.

Figure 4.10 shows how much the actual $mHeight[t]$ deviates from the model. The yellow line is the maximum response amplitude calculated from equation 2.30. One can see that the basketball robot greatly outperforms the model. The robot has a maximum error of 0.0638 meters and 28.%. This shows that at a certain point, the rotational acceleration of the spinners, as an input to the $mHeight[t]$ dynamics, adds to the response. Therefore, to calculate the expected response, numerical simulations must be performed.

4.2.2. Effects of Partial Linearization, Jumping, With No Spring Compensation

As discussed in section 3.3.2, model predicts that the robot diverges from the invariant set when it loses contact with the ground. This divergence results in unpredictable behavior.

However, if the divergence is small enough, corresponding to a small jump, then the behavior becomes more predictable. The following simulation shows this:

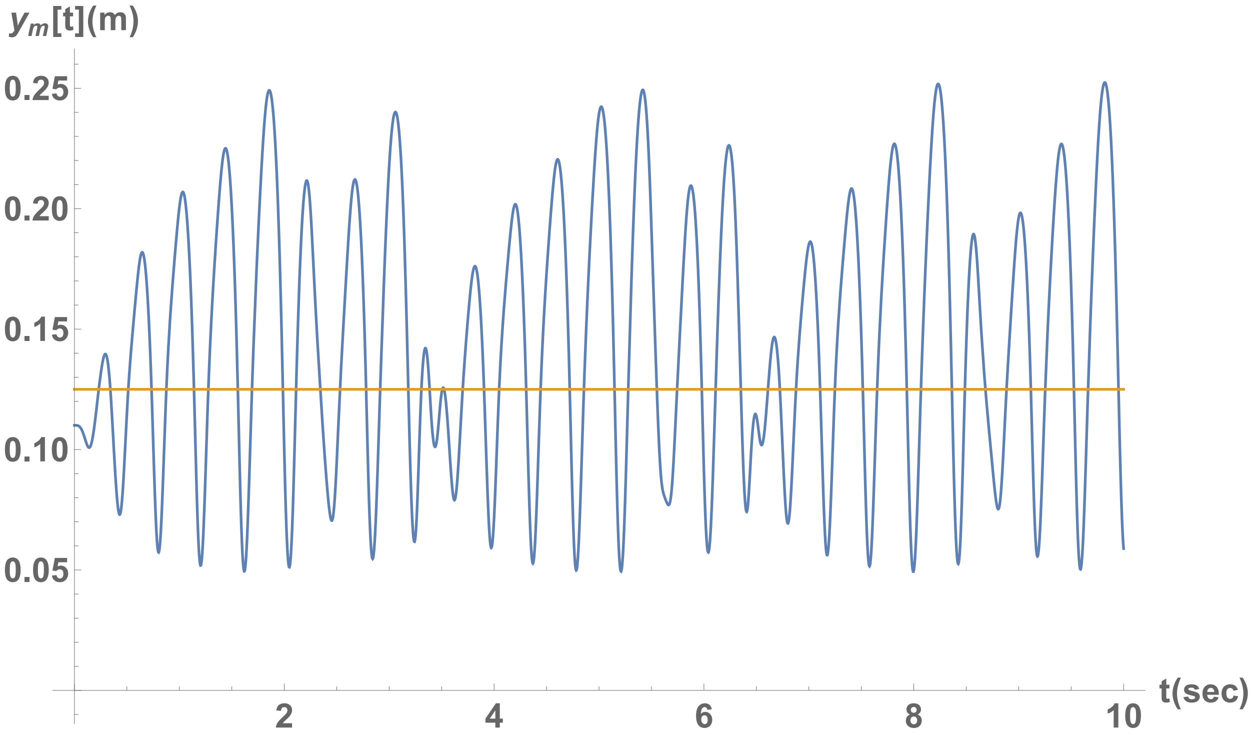

Figure 4.11. Basketball Jump Response with No Virtual Spring Controller

In figure 4.11, $\lambda = 10$ and ω_d is the resonant frequency calculated from equation 2.30. There appears to be an oscillation of jumps.

One can analyze this oscillation by the phase difference in the input force and the output $mHeight[t]$ response. There is a change in the phase shift with each jump. The change in phase shift can be more clearly seen in the figures below:

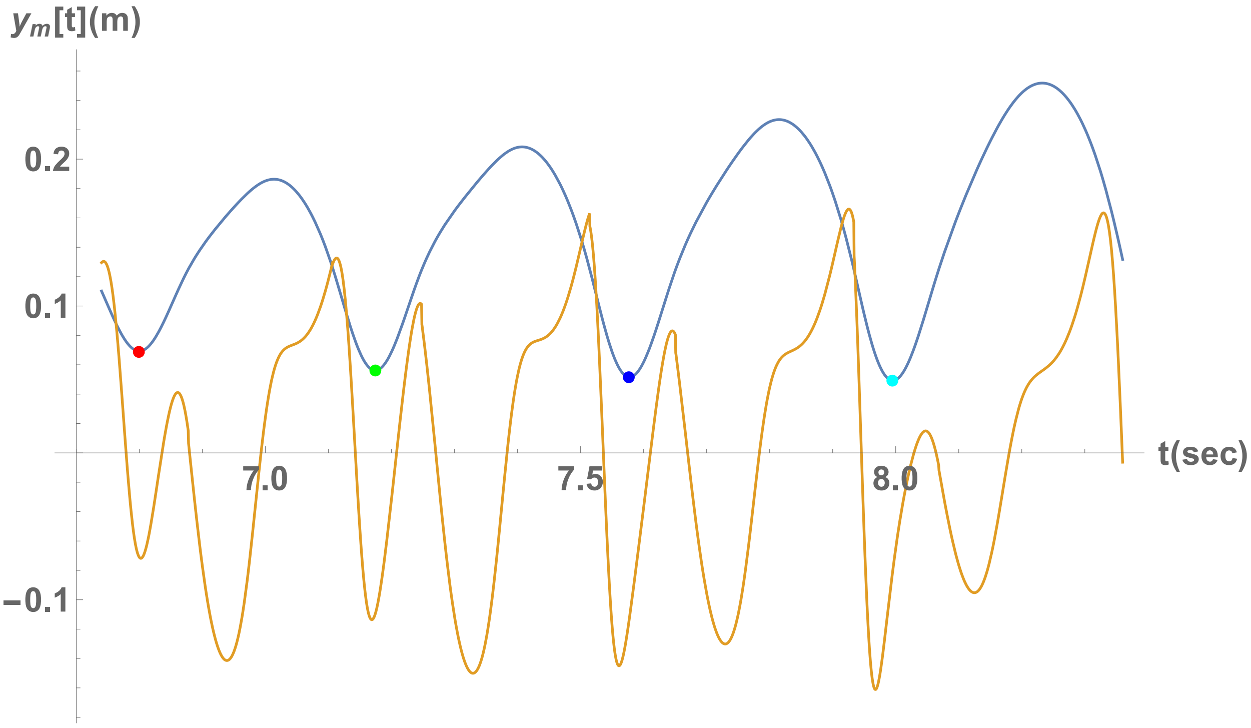

Figure 4.12. Basketball Jump Response with No Virtual Spring Controller Zoomed In

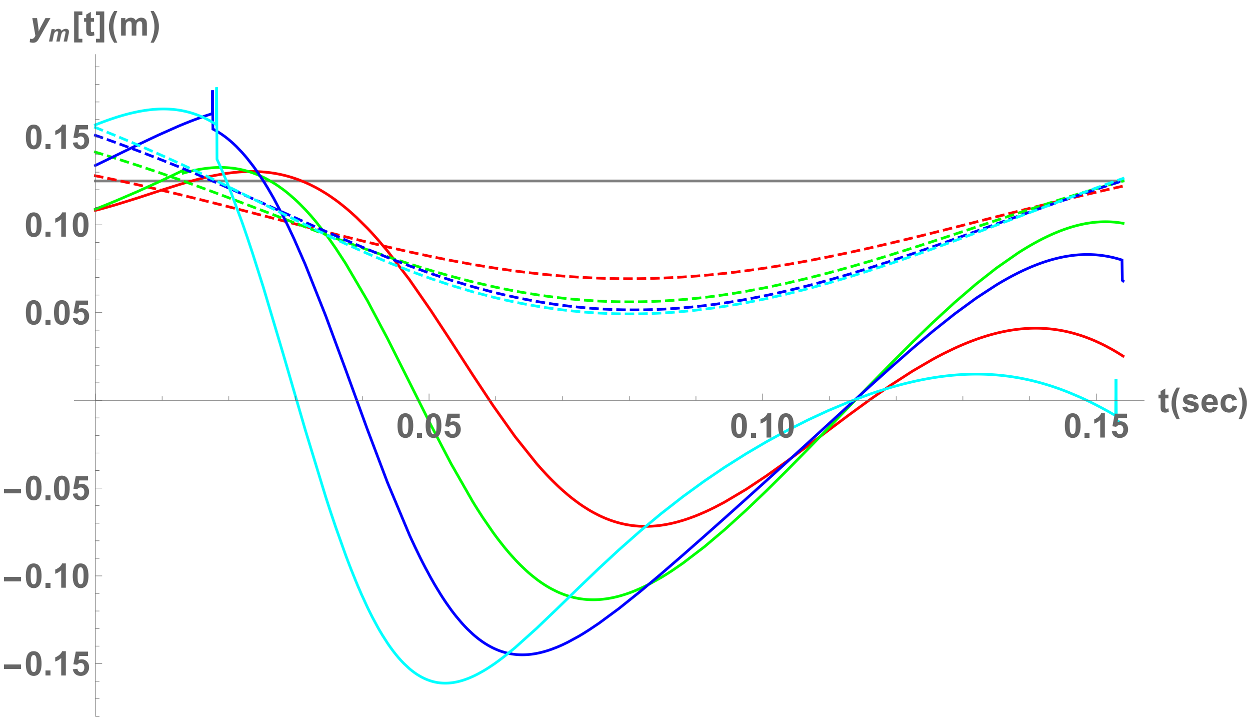

Figure 4.13. Basketball Jump Response with No Virtual Spring Controller Time Normalized

In figures 4.12 and 4.13, The color of the dots correspond to the color of each "take-off". A "take-off" is the part of the response when the robot is in contact with the ground. In figure 4.12, the yellow curve is the input force of the spinners. In figure 4.13, the solid curves are this same curve, corresponding to each color. One can see that with each "Take-off" the phase difference between the input force of the spinners and the output response increases. Eventually, the phase difference returns to the original first phase difference and a oscillation is created in the response.

Now, let's analyse the total mechanical energy. According to the controller design, when the robot is jumping, the robot diverges from the invariant set. The following plot shows the total mechanical power of the robot, when there is no spring compensation:

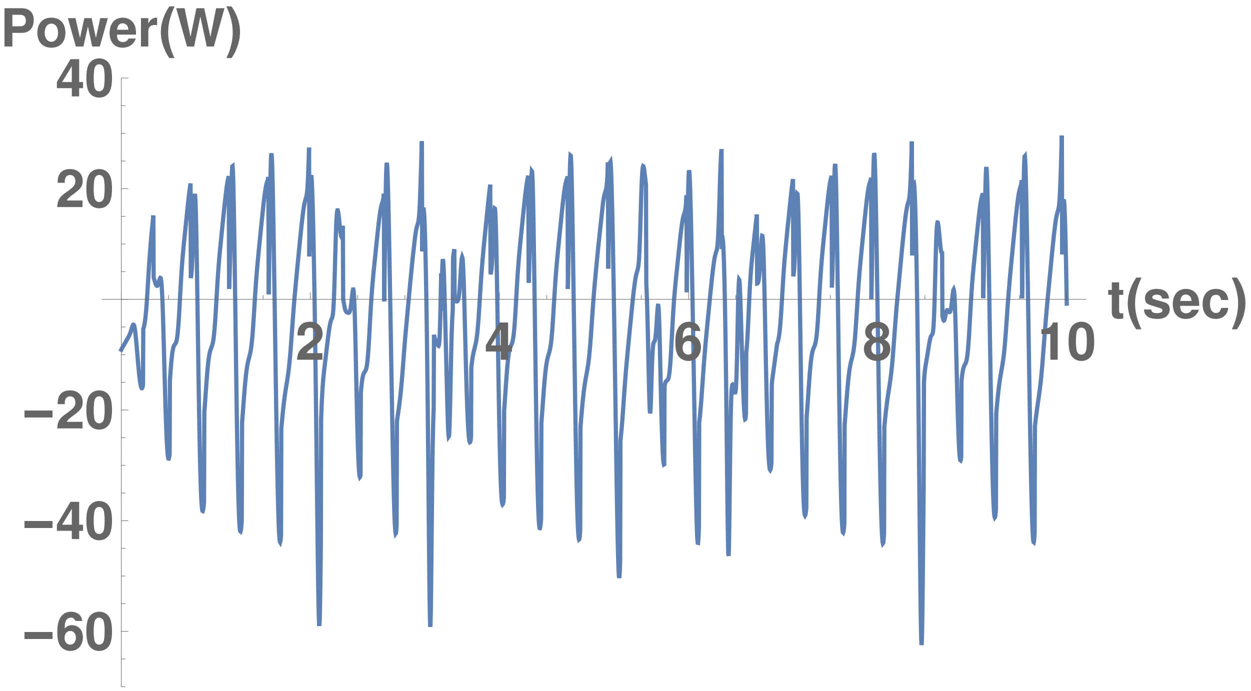

Figure 4.14. Total Mechanical Power Response with No Spring Compensator

The response in figure 4.14 shows how the oscillations in the system response correspond to mechanical power. For, as the robot diverges from the invariant set, the total mechanical power increases, until it reaches a critical value. At the critical value, when in contact with the spring, the significant magnitude in convergence of the system to the invariant set forces the system to the invariant set. This creates an overshoot. After the overshoot, the build up of mechanical power starts over. This response is not desireable. We want to minimize the mechanical power. By minimizing the mechanical power, we will have more power left over for control. Also, we want the oscillations to be predictable, constant, and stable.

4.2.3. Full Controller Response

Let us simulate the basketball robot with the same parameters as the simulation in section 4.2.2, but now with the spring compensator:

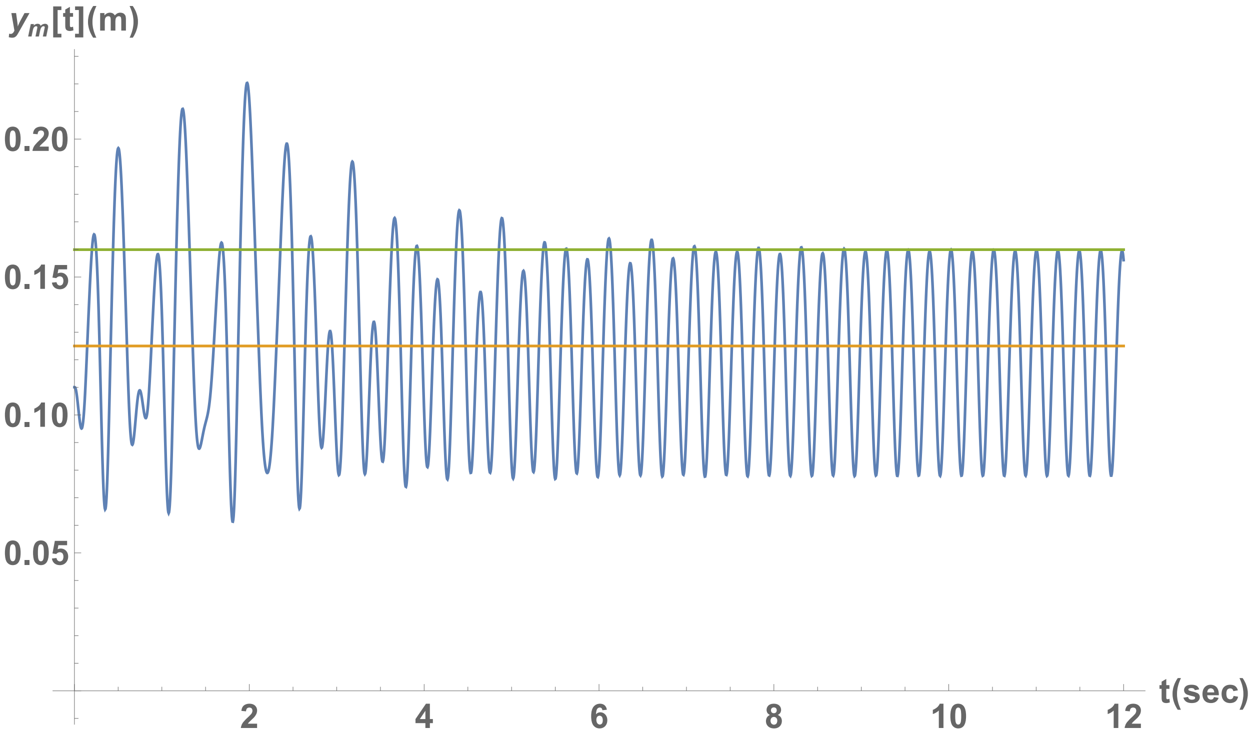

Figure 4.15. Basketball Jump Response with Virtual Spring Controller, $\lambda = 10$

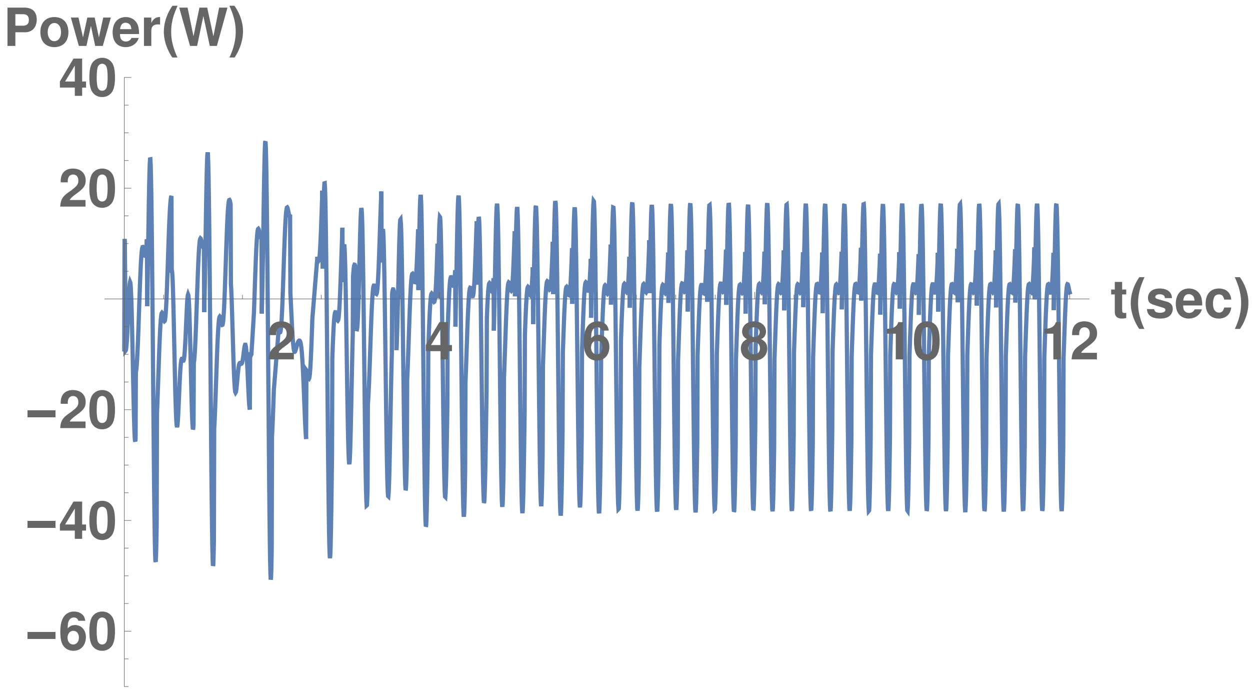

Figure 4.16. Total Mechanical Power Response

The simulation in figure 4.15 shows the oscillations, in steady state, are stablized!

Figure 4.16 shows that the mechanical power is stabilized as well. If the invariant set was properly identified, Figure 4.16 would show the system diverging from the invariant set when the robot is flying and converging to the invariant set when in contact with the ground.

The maximum spring deflection, at steady state, has an amplitude of 0.047 meters. The simulation in figure 4.10, using the resonant frequency has maximum spring deflection, at steady state, of 0.133 meters. This means that the spring should be able to store more energy at the resonant frequency!

We can try adjusting λ. Let us set λ to 0.1. The plot below shows the response:

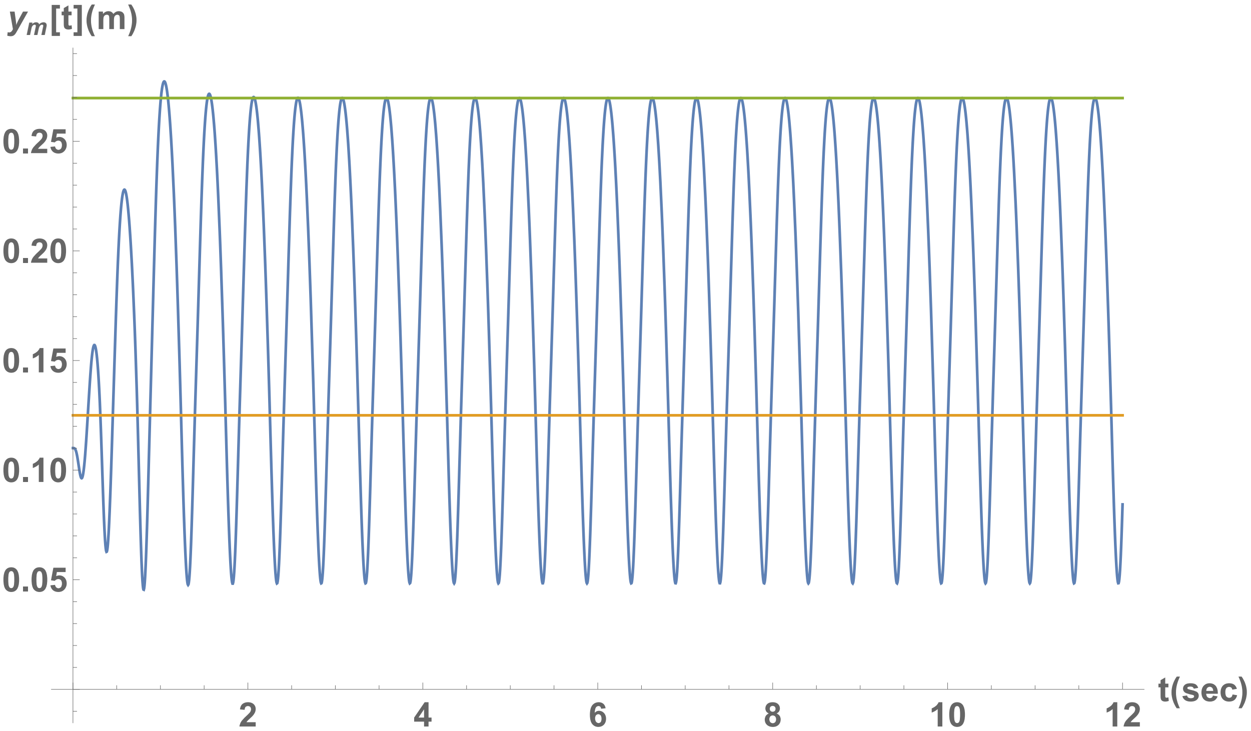

Figure 4.17. Basketball Jump Response with Virtual Spring Controller, $\lambda = 0.1$

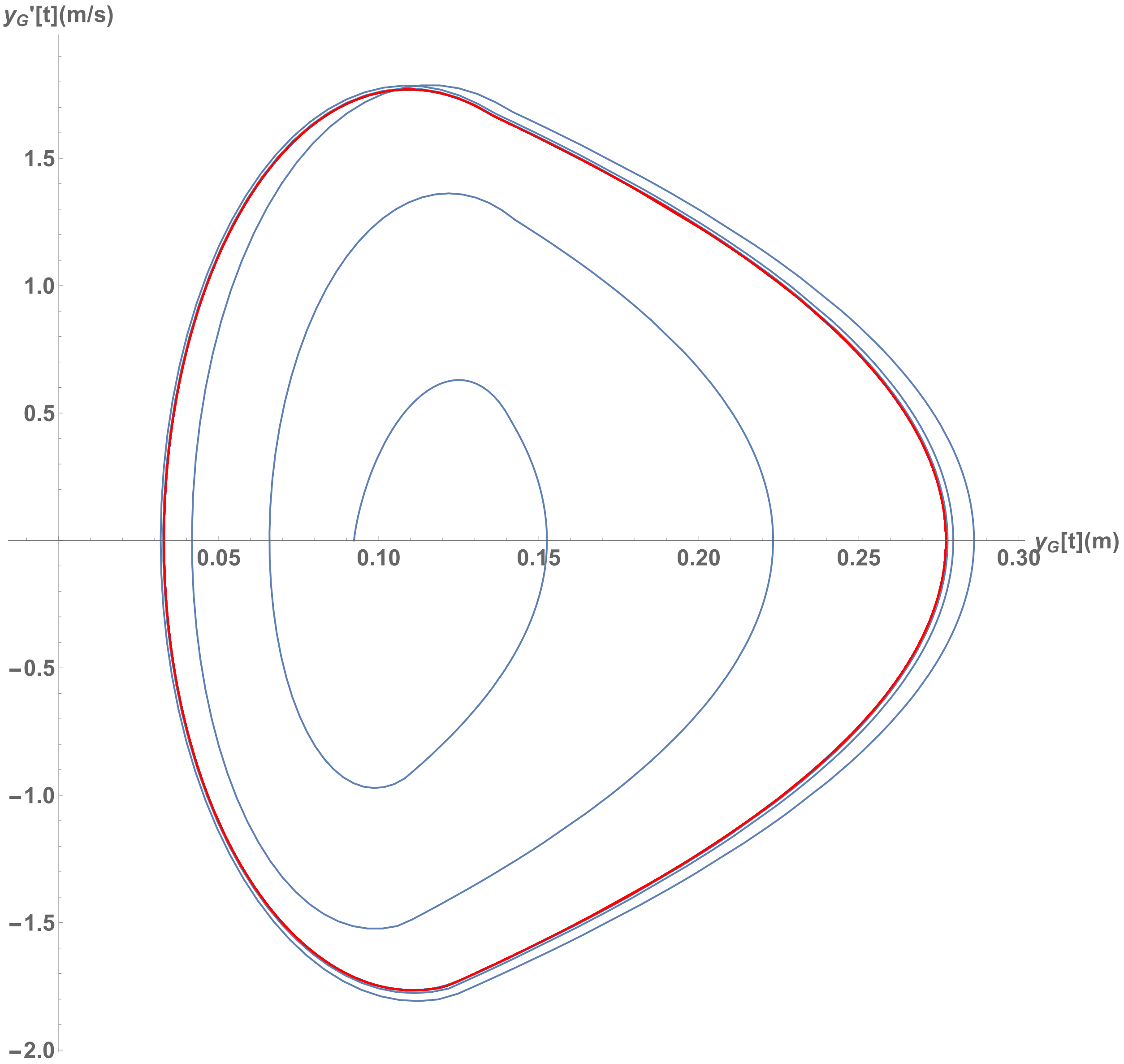

Figure 4.18. Basketball Robot Limit Cycle, $\lambda = 0.1$

Figure 4.18, shows the stable periodic oscilations achieved by this controller. Notice that the variables in Figure 4.18 are in terms of the center of mass. Lees [10] used the center of mass as a measurement of jump height instead of a physical point on the jumper.

By changing λ, we can change the performance.

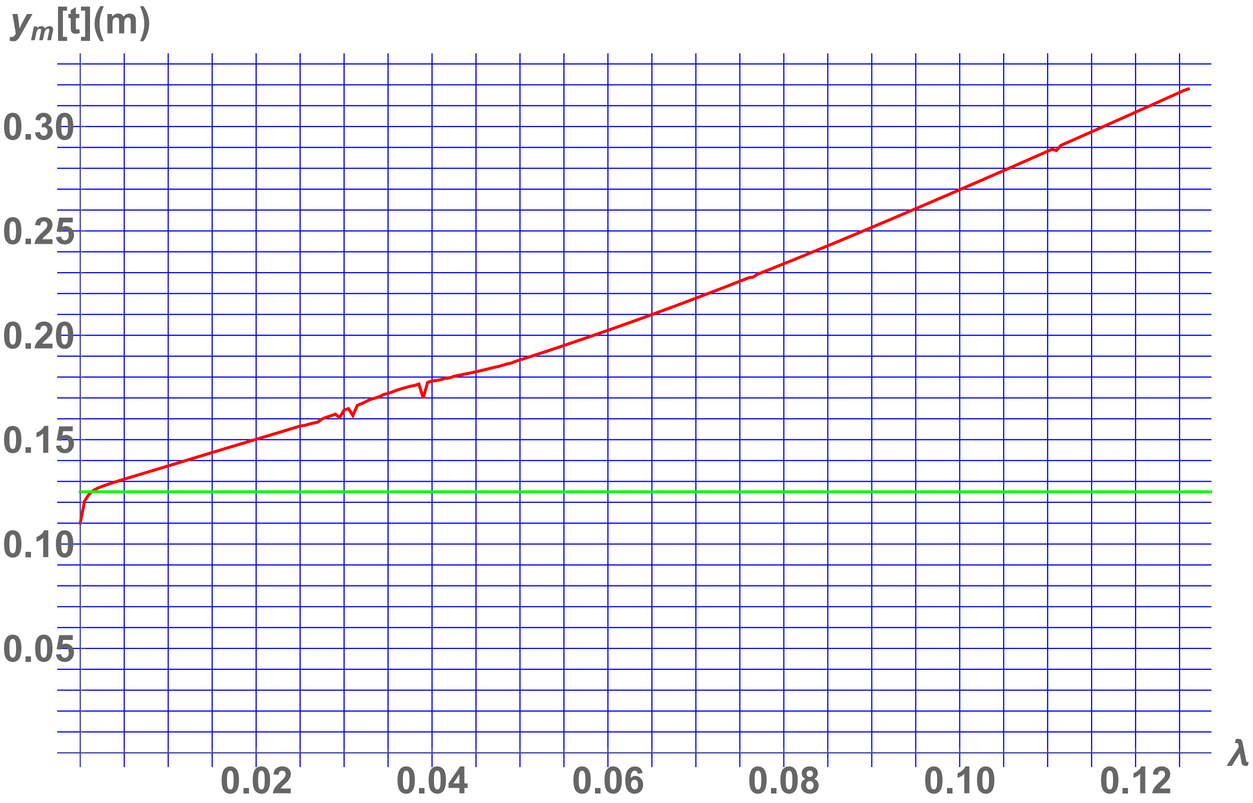

Figure 4.19. Basketball Robot Response Curve, $\omega_d = 25.7$

Figure 4.19 was obtained by simulating the system at its resonance frequency and varying λ between 0 and 0.126, and calculating the maximum response in the last two seconds of the simulation. The last two seconds were arbitrarly chosen to be in steady state for the simulations. There is a monotonically increasing relationship from the point the robot begins to jump and its maximum at $\lambda = 0.126$.

As a side note, one can adjust both λ and ω_d to get an even bigger jump:

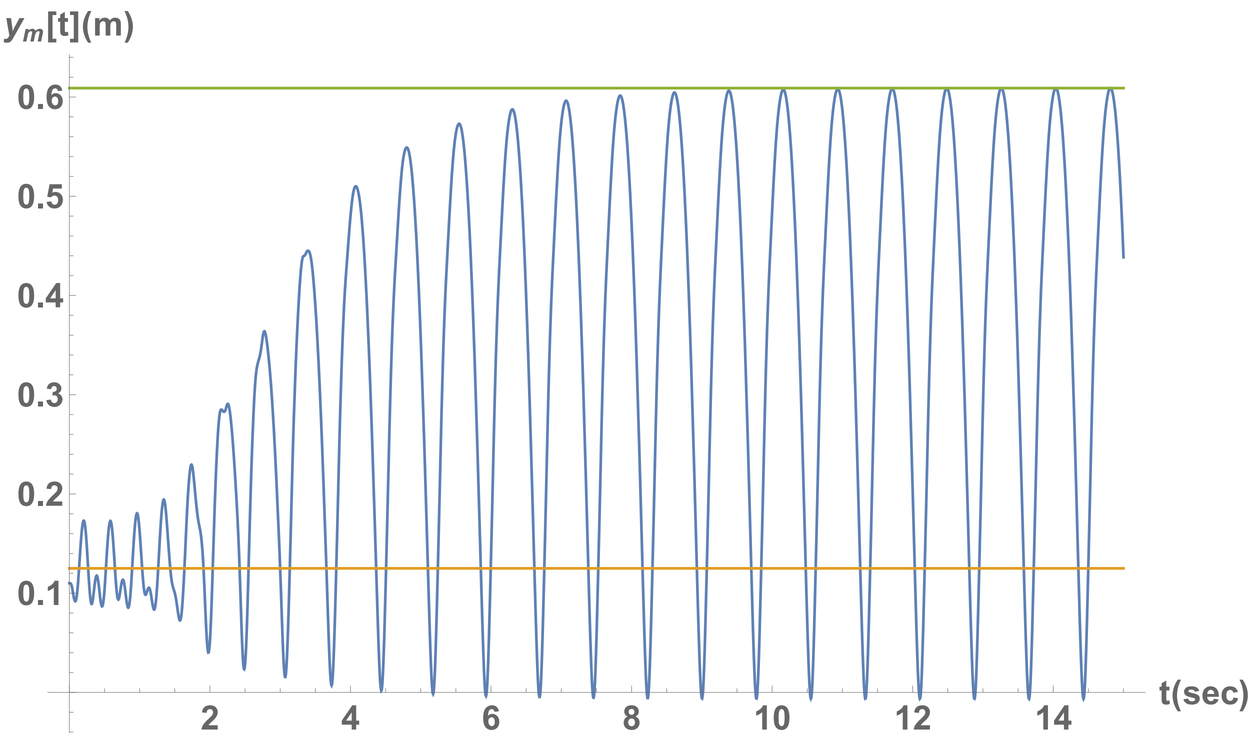

Figure 4.20. Basketball Robot Super Jump, $\omega_d = 45$ and $\lambda = 0.08$

The steady state jump height in figure 4.20 is 0.6 meters! Also, the spring deflection is greater than what was calculated as the limit from the spring contact analysis. Pushing the controller this far is not advisable because it is very sensitive to changes in λ and ω_d. Also, the physical system would not be able to sustain this performance. Limiting factors for the response include:

1. Maximum Battery Current
2. Maximum Motor Torque
3. Structural Strength of Mechanical Assembly
4. Available Spring Travel

It is recommended to keep ω_d at or below the resonant frequency. The reason pushing the controller this far works, is that when ω_d is set at the resonant frequency, it actually mostly stays below the resonant frequency because of the dynamics. Increasing λ will push it more towards the resonant frequency, but never beyond it. If ω_d is greater than the resonant frequency and λ is too large, then the system will spin above resonance. However, if λ is not too large, beyond the resonant frequency, then it will lift the actual frequency closer to the resonant frequency. And, there will be more resonance.

An analogy can be drawn to better explain this controller. Consider a human jumping on a trampoline. If the reader has jumped on a trampoline, the he will know that jump height is controlled by how hard one jumps, not how fast. Therefore, by shifting the control paradigm from input frequency to input force, the natural linear control one experiences on a trampoline can be extended to this system.

4.3. Tapping Robot Simulation Results

The following are the results of the dynamics and controller provided in this paper for the tapping robot. Some simulations failed in Mathematica, so future work needs to be done in find a more stable simulation method. Also, because of the instability, the results given in this section are not conclusive, but do provide an estimation to the performance in experimentation.

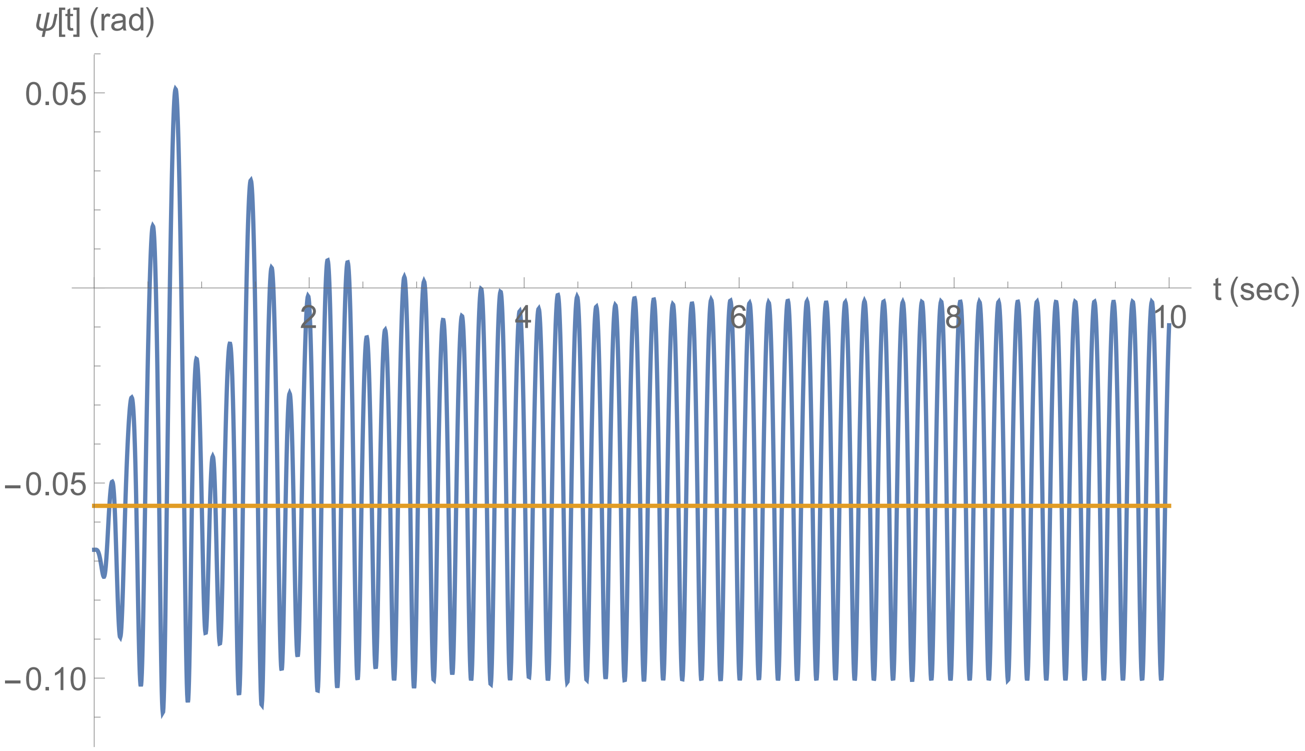

Figure 4.21. Tapping Robot Modeled Transition Response, $\omega_d = 35.3$ and $\lambda = 0.5625$

The simulation in figure 4.21 is the response of the tapping robot, when the robot is run at the transition frequency defined in section 2.3 and the spring compensator is ignored. Expected from the constant angular velocity assumption, the results of the basketball robot and the linearization of the dynamics, the robot should have a steady state response at S_f. This does not happen.

In the simulation, the tapping robot jumps higher than the S_f. The yellow line is the S_f.

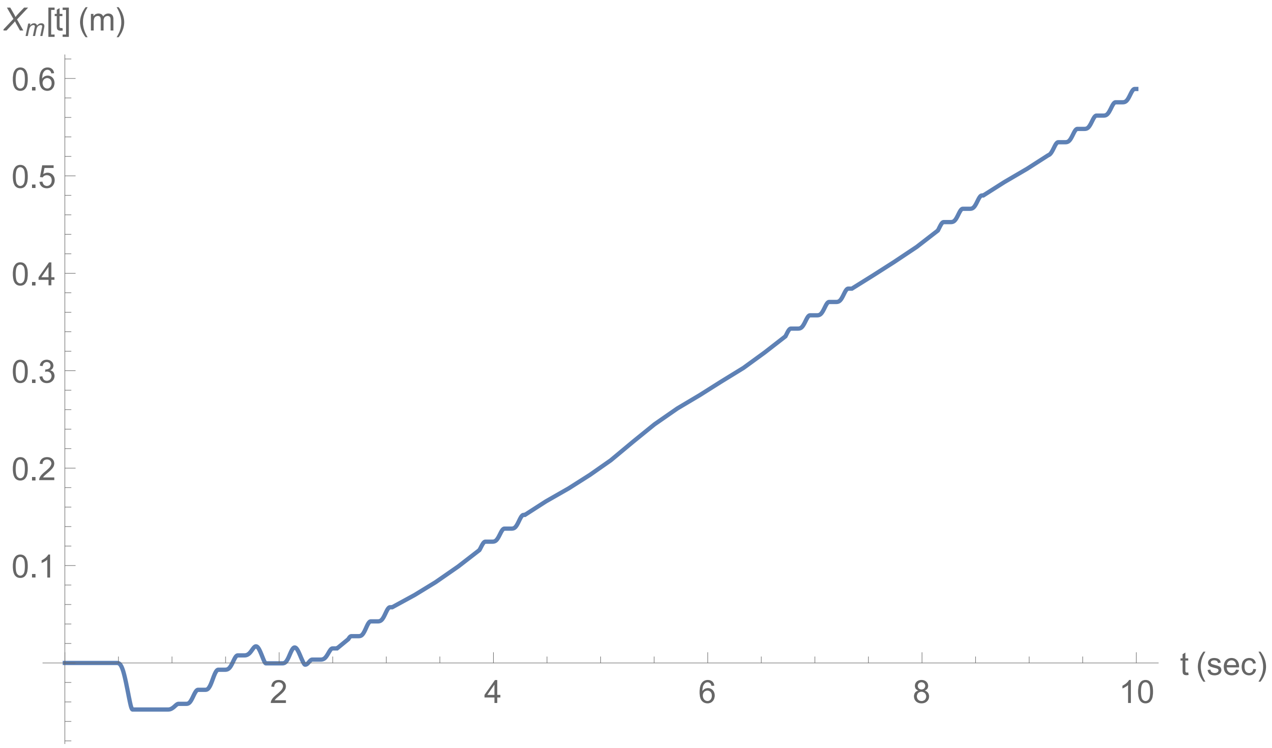

Figure 4.22. Tapping Robot Modeled Transition Response Displacement, $\omega_d = 35.3$ and $\lambda = 0.5625$

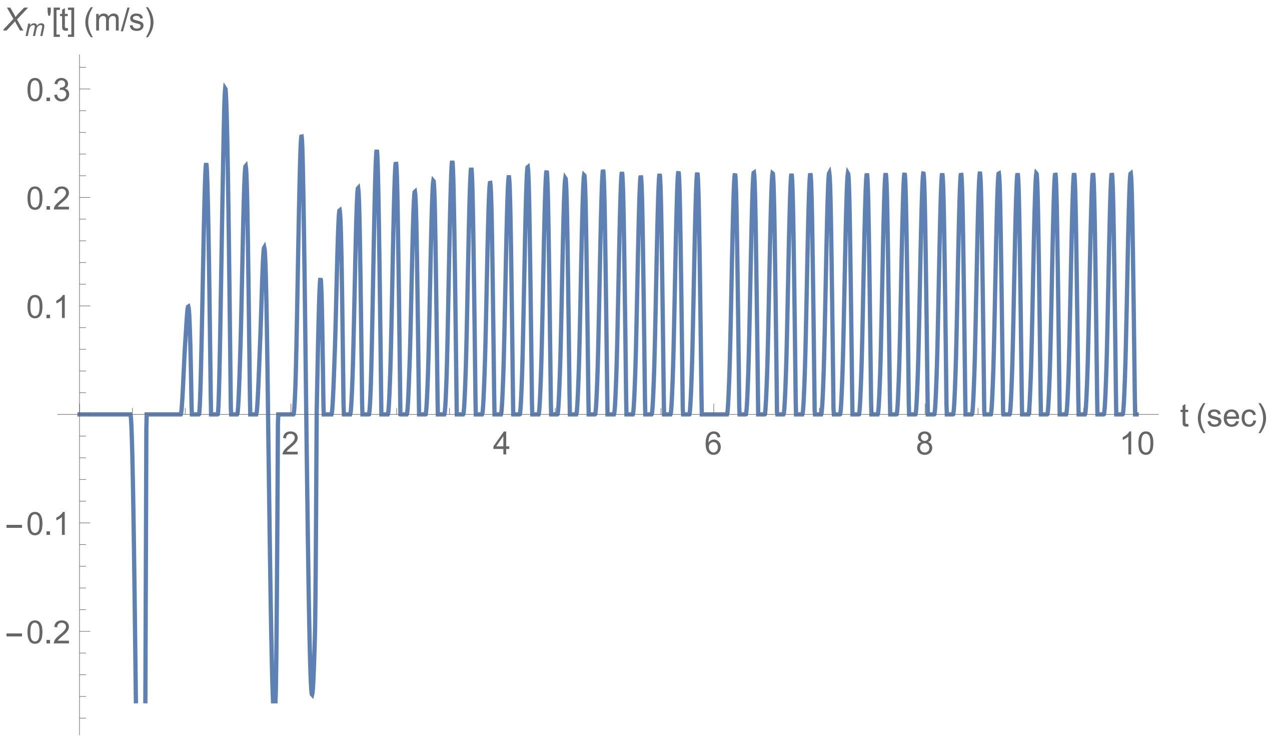

Figure 4.23. Tapping Robot Modeled Transition Response Displacement, $\omega_d = 35.3$ and $\lambda = 0.5625$

Figures 4.24 and 4.25 are the results of the tapping robot simulation run at the resonance frequency estimated in 2.3. The proportional gain was tuned to provide a jump, but not become unstable. No stable oscillations were found that this frequency. One possible explanation for this is that the dynamics are too fast for the simulator. Another possible explanation is that the system diverges too fast from the invariant set for the spring compensator to stabilize.

Figure 4.25 shows the unstable oscillations.

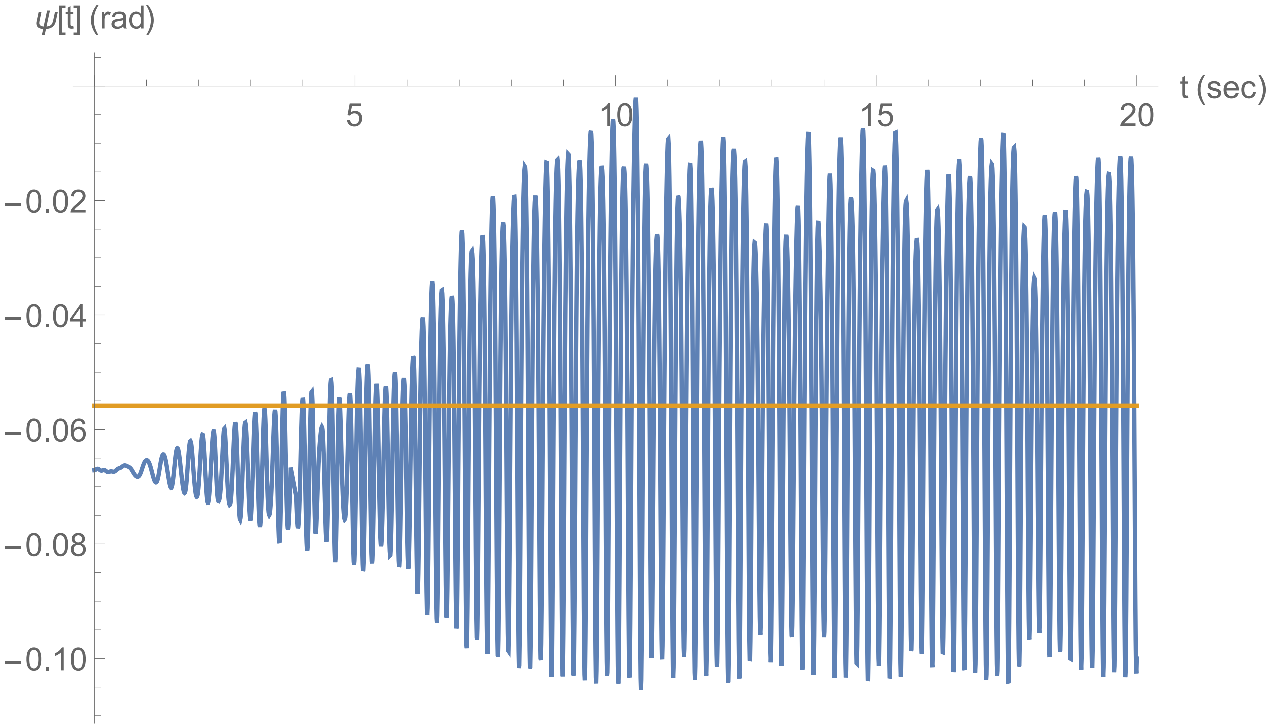

Figure 4.24. Tapping Robot Modeled Resonance Response, $\omega_d = 49$ and $\lambda = 0.00032$

Figure 4.23 shows the forward progression of the robot.

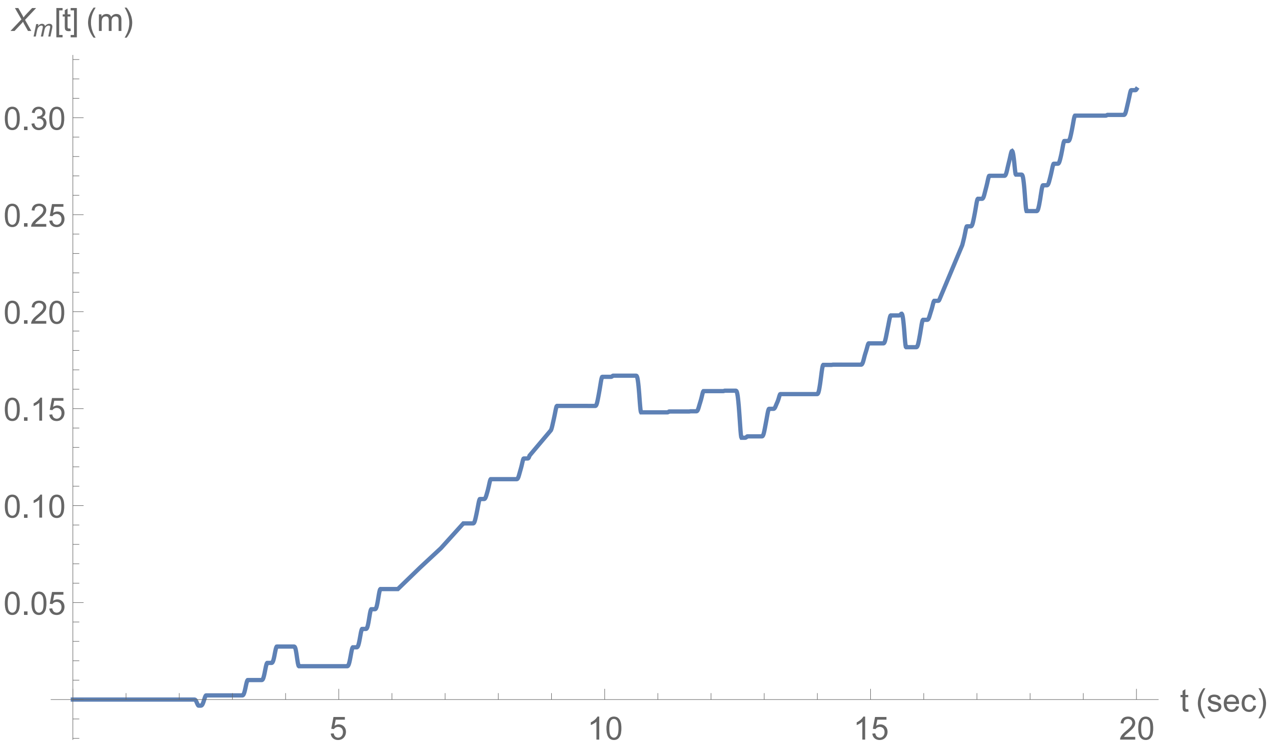

Figure 4.25. Tapping Robot Modeled Resonance Response Displacement, $\omega_d = 49$ and $\lambda = 0.00032$

The dynamics can become more controllable if ω_d is reduced to 39 rad/s.

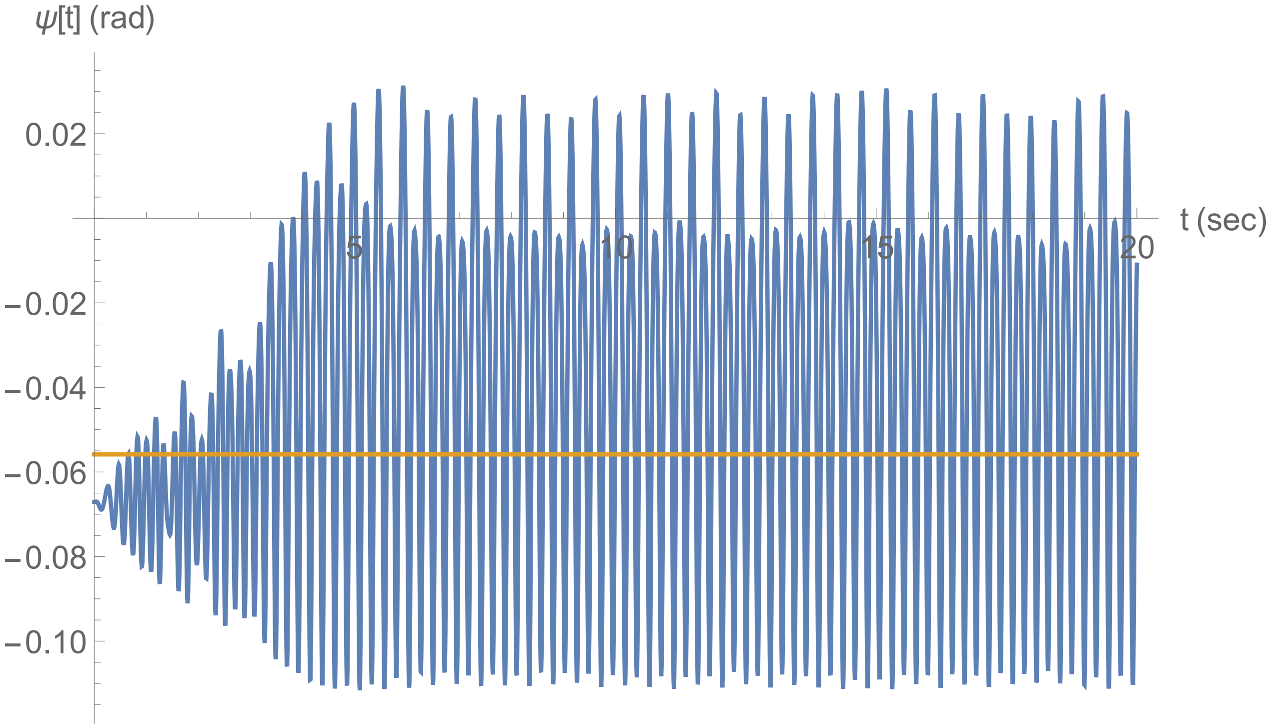

Figure 4.26. Tapping Robot Modeled Pseudostable Response, $\omega_d = 39$ and $\lambda = 0.003$

Figure 4.26 shows the response reaching steady state at 6.5 seconds.

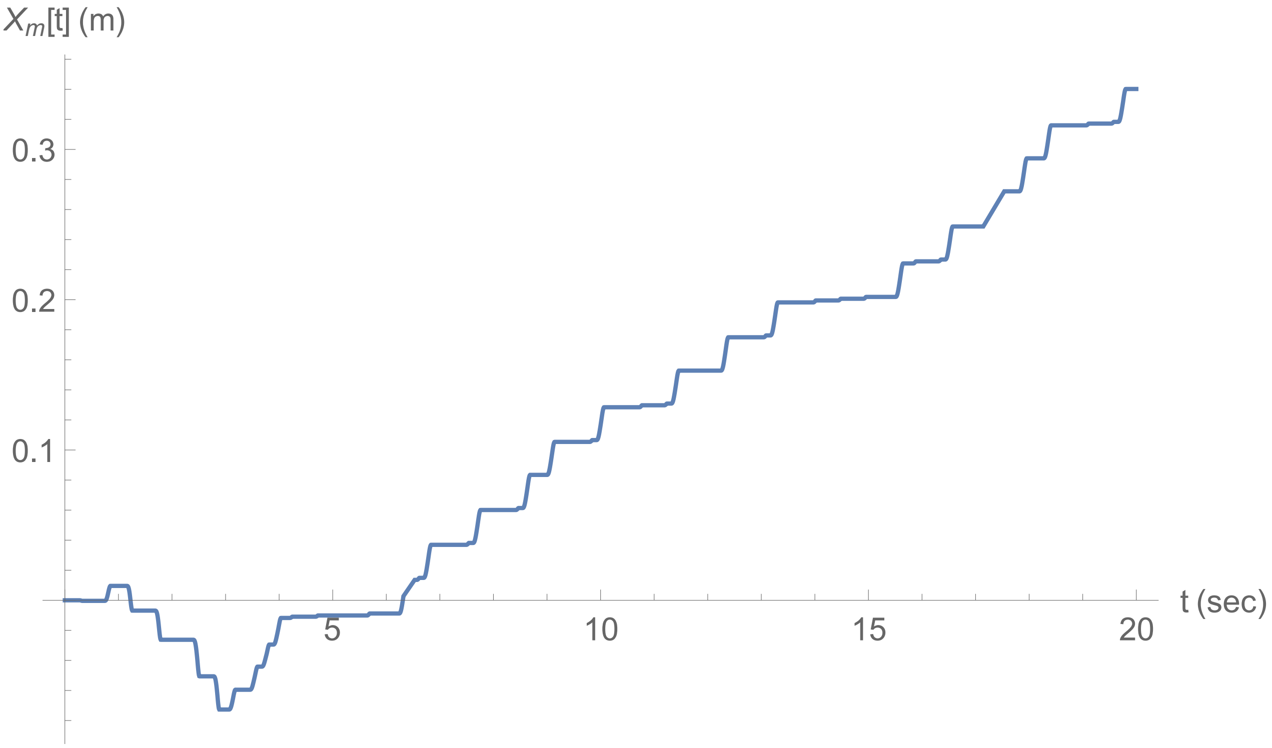

Figure 4.27. Tapping Robot Modeled Pseudostable Response Displacement, $\omega_d = 39$ and $\lambda = 0.003$

The effect of the spring compensator can be seen when it is removed.

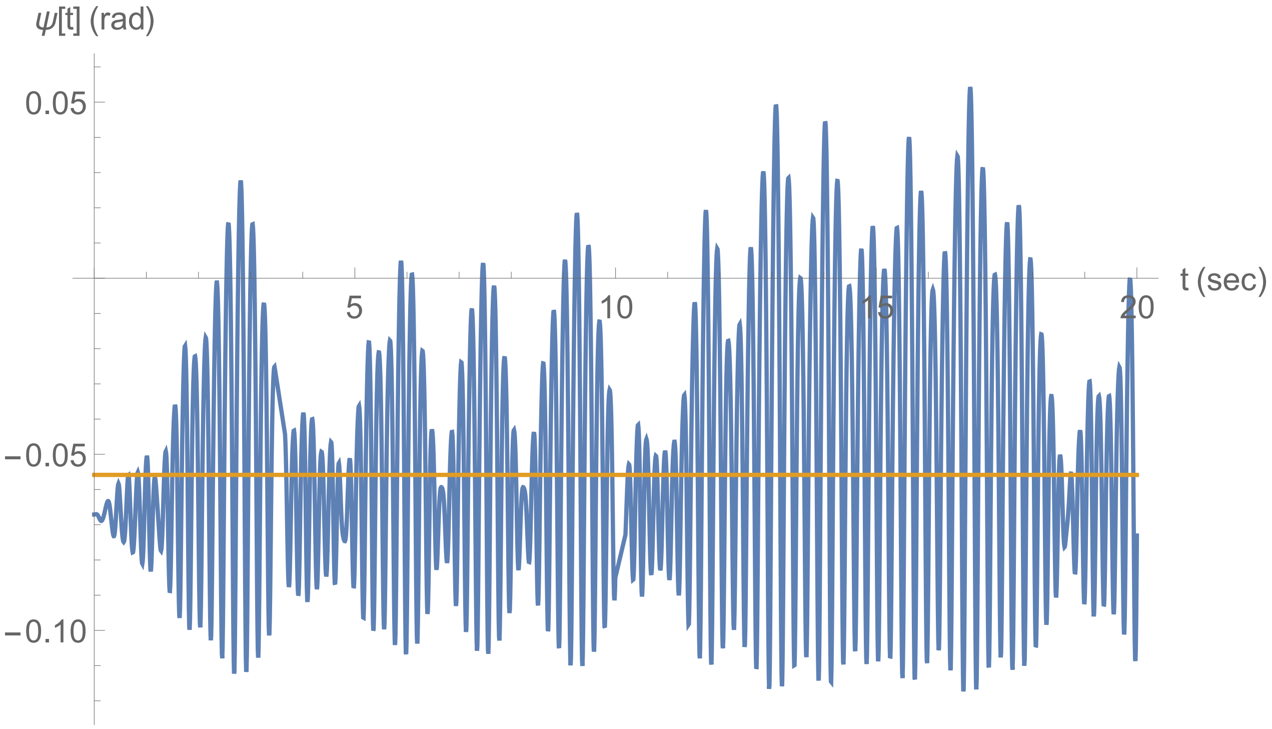

Figure 4.28. Tapping Robot Modeled Pseudostable Response, with No Spring Compensator, $\omega_d = 39$ and $\lambda = 0.003$

The response in figure 4.28 uses the same parameters at figure 4.26. Clearly the response has been more stabilized with the spring compensator. Without the spring compensator, the response in figure 4.26 would not be feasible.

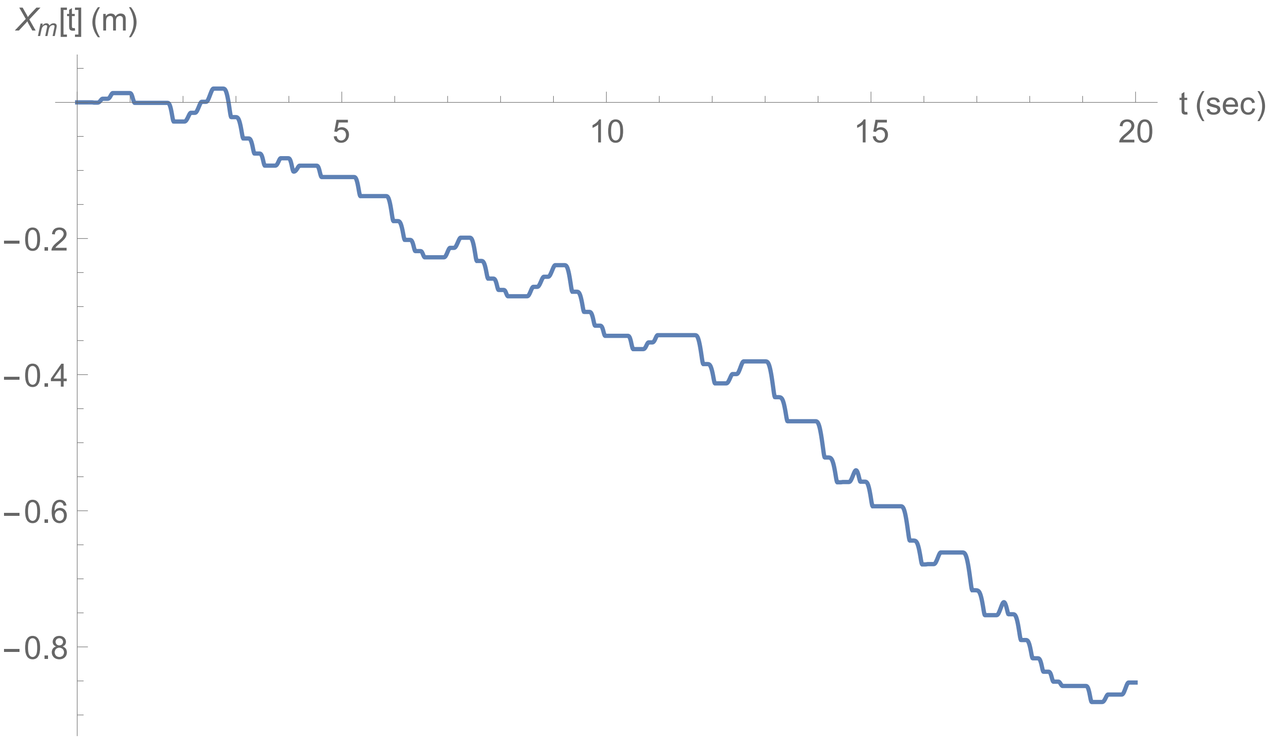

Figure 4.29. Tapping Robot Modeled Pseudostable Response Displacement, with No Spring Compensator, $\omega_d = 39$ and $\lambda = 0.003$

The response in figure 4.29 shows that when the spring compensator is turned off, then the robot actually progresses backwards.

Chapter 5

EXPERIMENTAL SETUP

Kashki's Basketball Robot experimental setup no longer exists. To extend the work of inertially actuated robotics, the tapping robot was built.

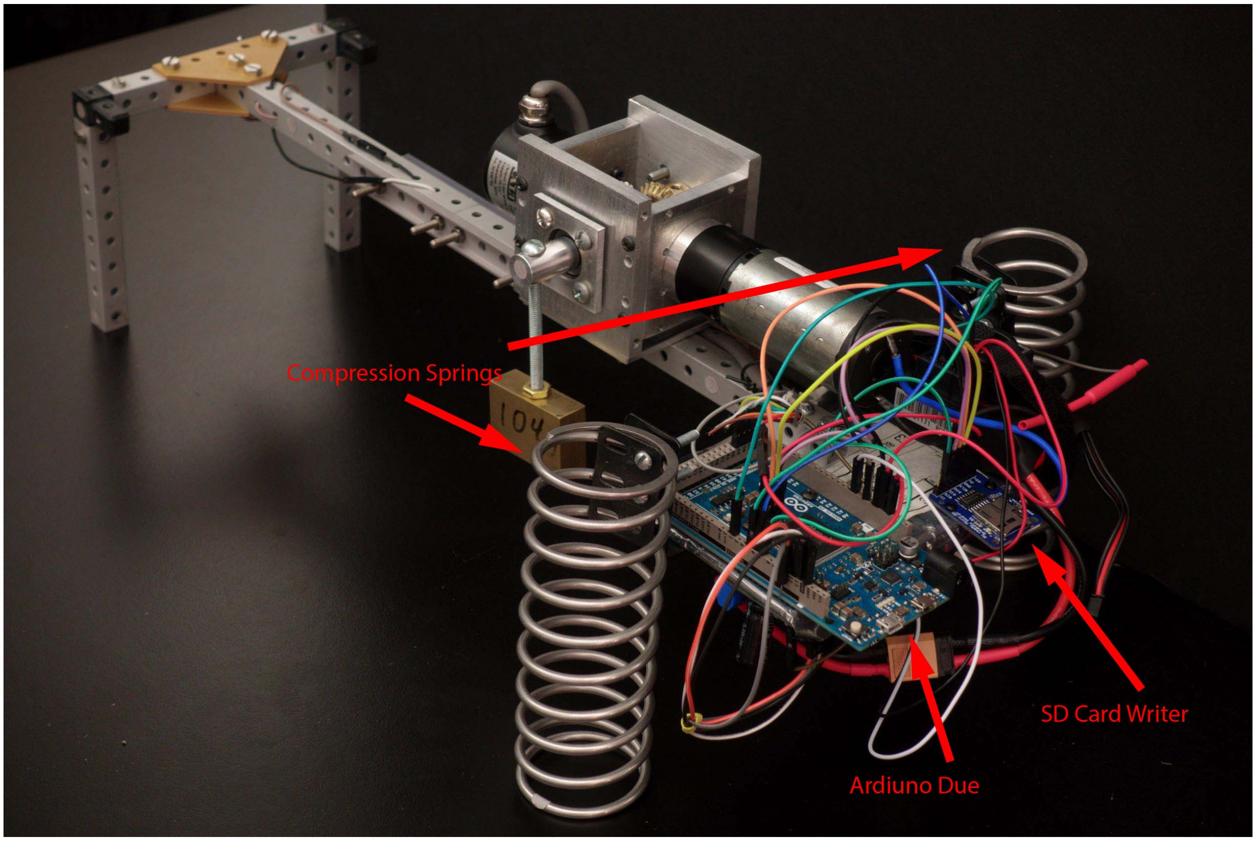

Figure 5.1. Top Right Front View

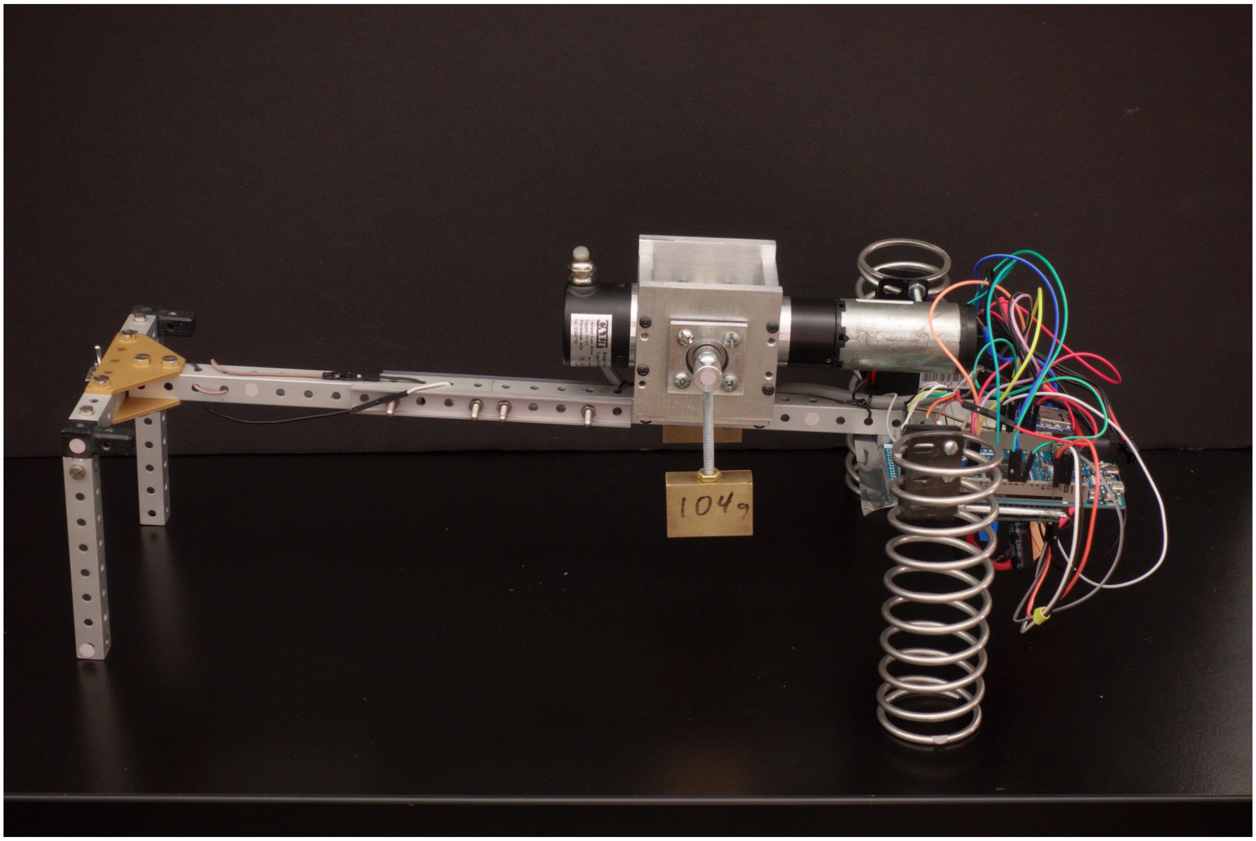

Figure 5.2. Top Right View

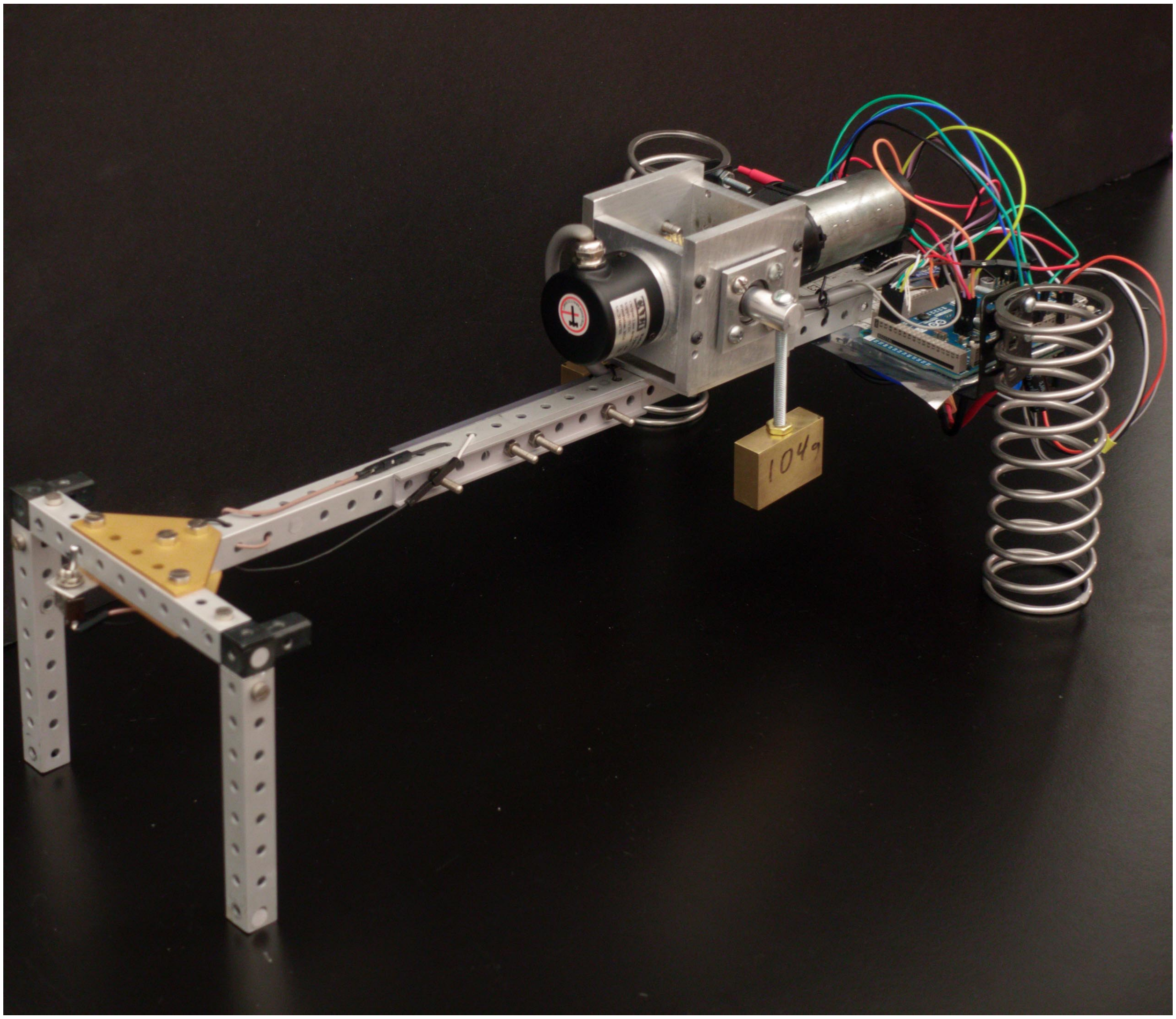

Figure 5.3. Top Right Back View

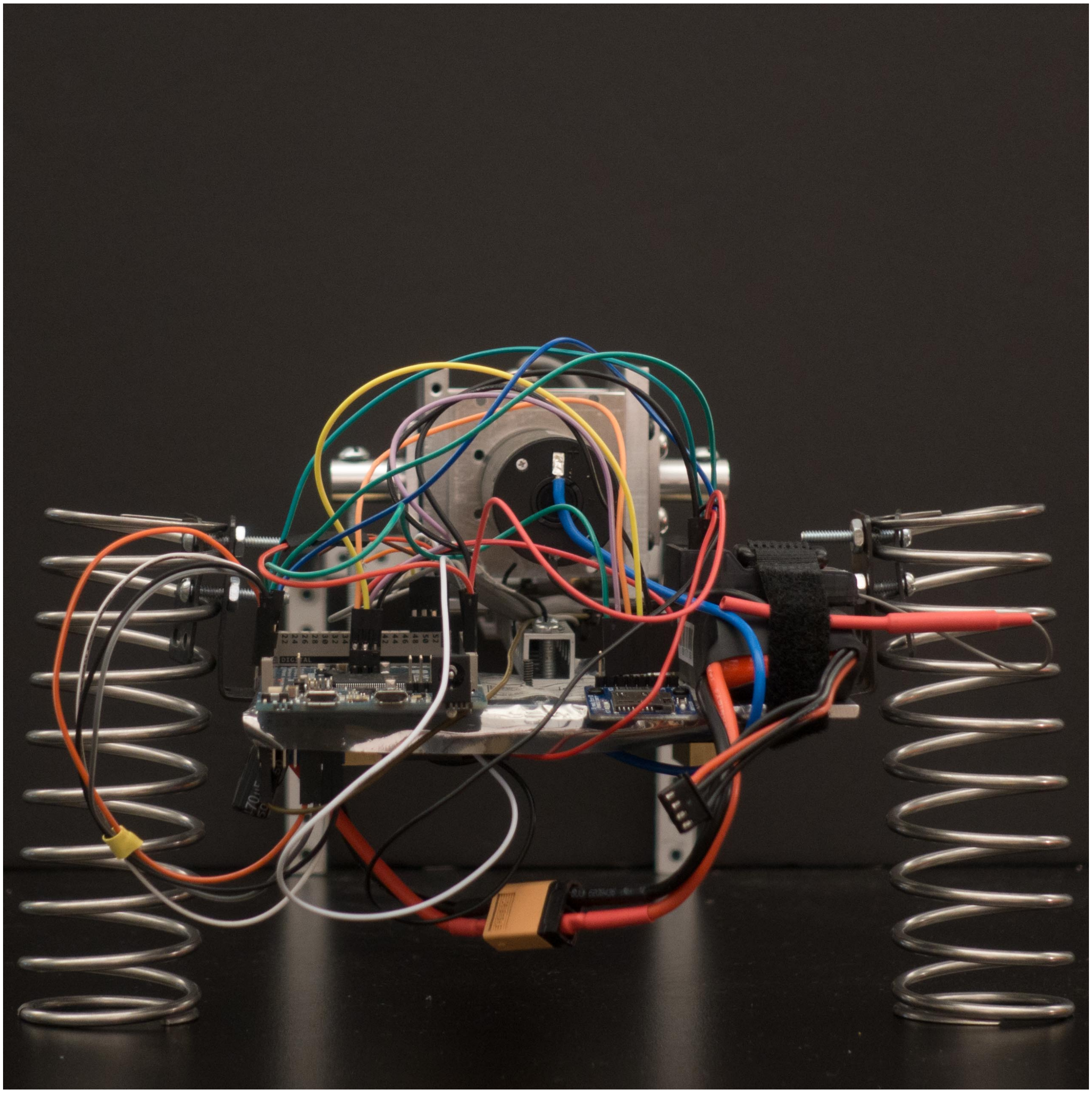

Figure 5.4. Front View

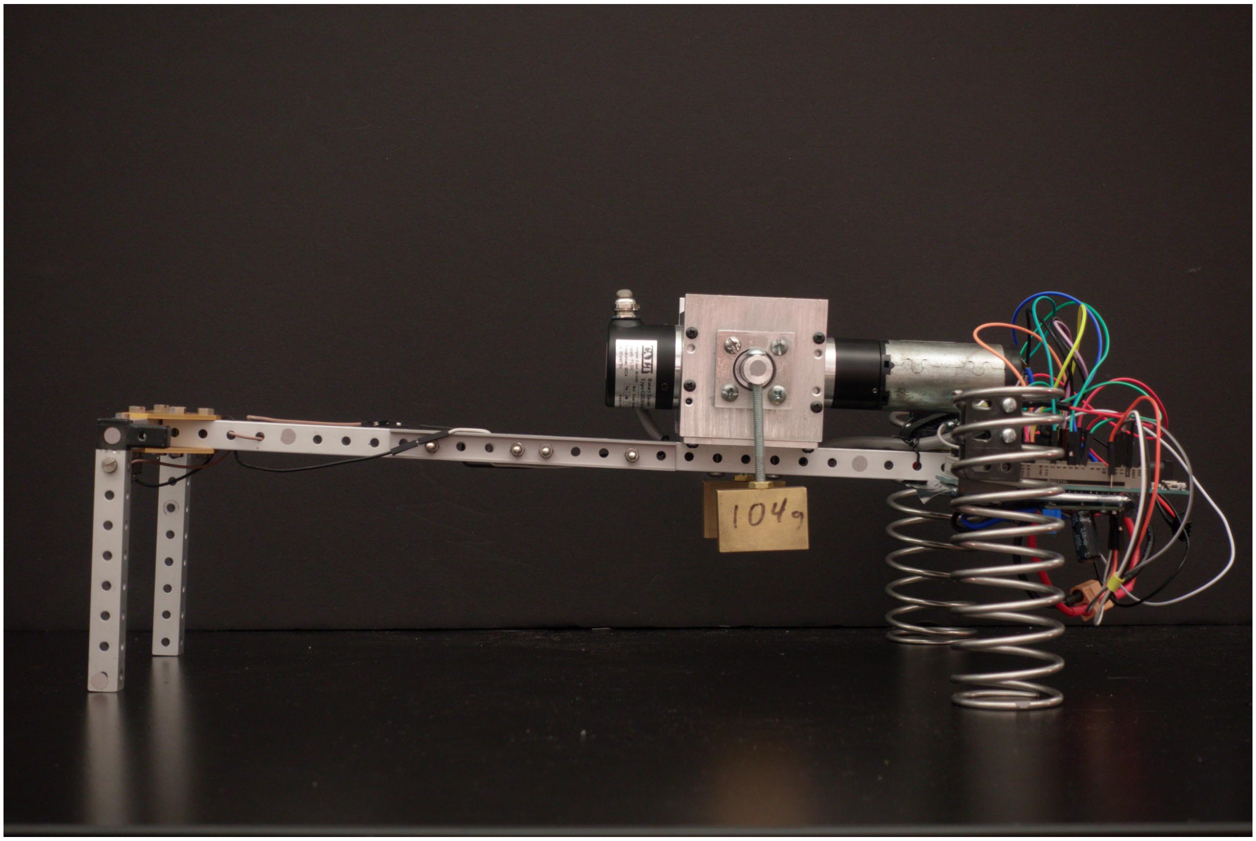

Figure 5.5. Right View

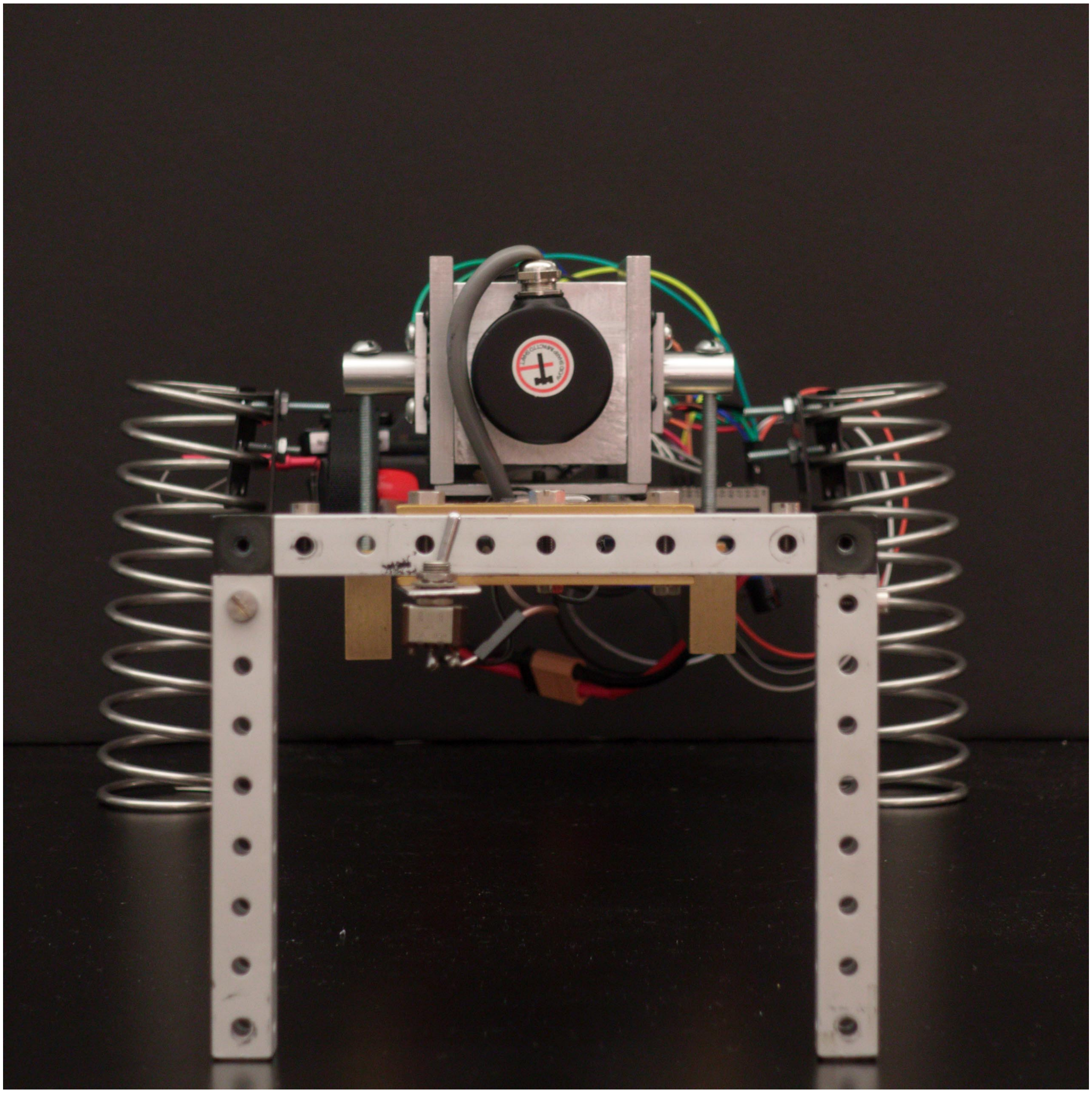

Figure 5.6. Back View

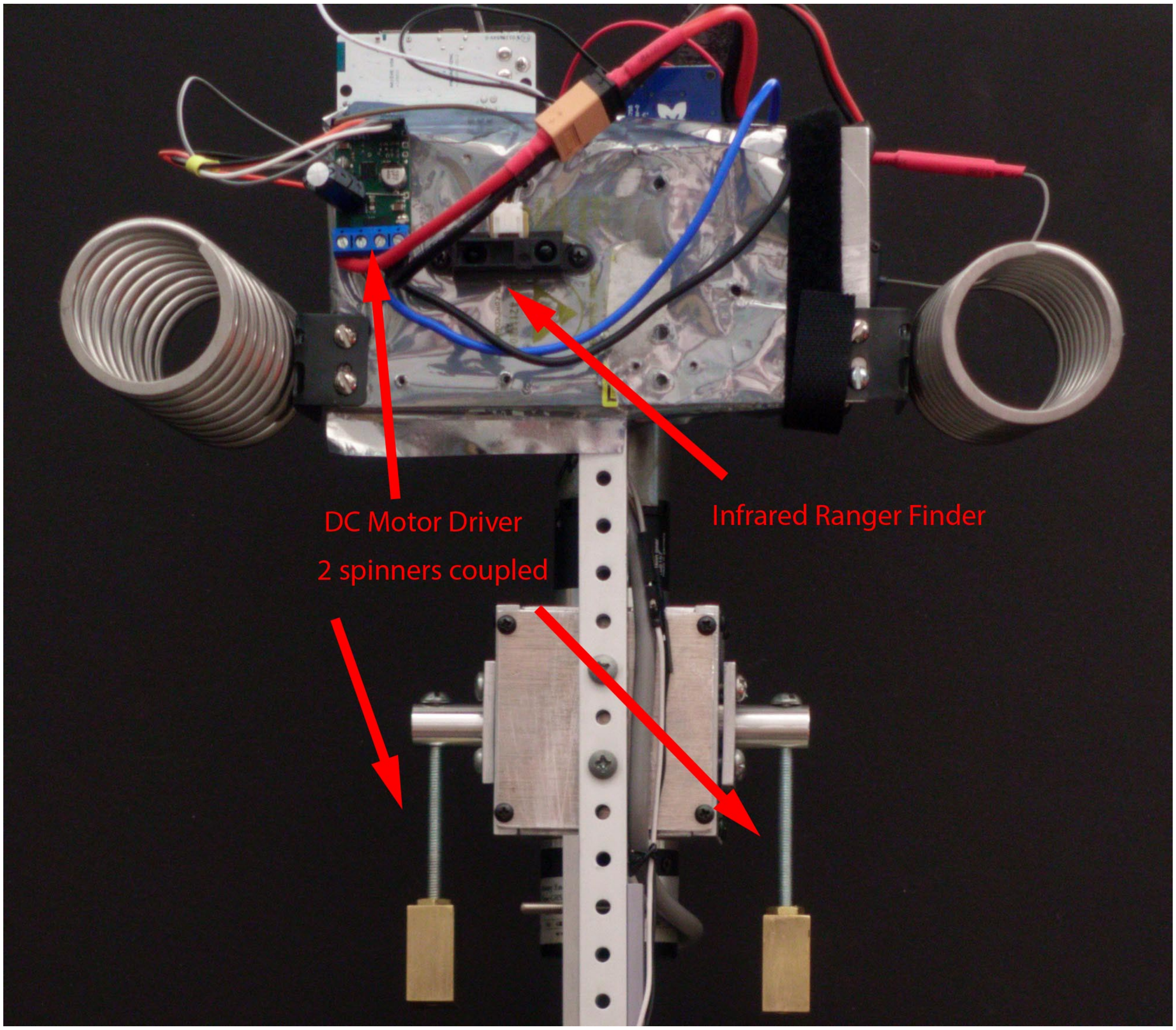

Figure 5.7. Bottom View

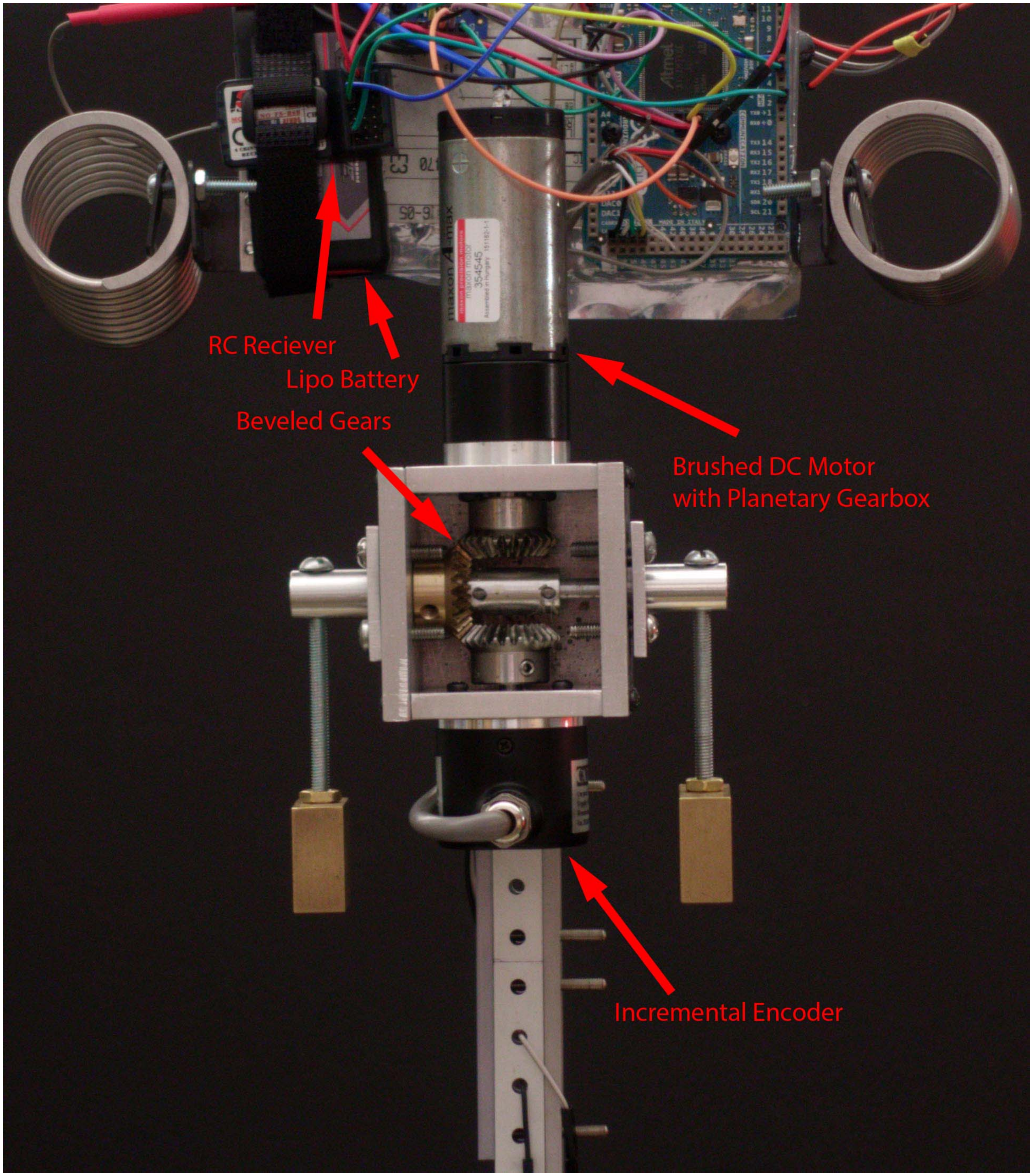

Figure 5.8. Top View

5.1. Mechanical Components

The main components of the tapping robot are the spinners, the frame, and the springs. The spinners were custom machined at SMU's machine shop to be interchangeable with different masses. In the experiment, two 104 gram brass weights were locked onto two 3" 10-32 machine screws. The two spinners were coupled together inside the gearbox, such that they spin together, essentially creating a single spinner. Inside the gearbox, there are miter gears linked together. The back miter gear connects to an incremental encoder with an index. The middle miter gear connects to the spinner. The front miter gear connects to the motor. The frame came from an aluminum erector set. The springs are custom made oppositely wound springs. The two springs have opposite windings to cancel out any moments created by their compression.

The rotary encoder is a CALT model GHS38-6G1024BML5. The encoder has an 6mm shaft. 1024 Pulse/Revolution resolution. This creates a 0.0879° resolution using a quadrature decoder function on the Arduino Due. The encoder has an index output triggered every revolution. This prevents a drift in the position reading. The encoder is linked to the system through the miter gear.

The miter gears are SDP\SI 25 Teeth, 1 Module, ISO 8 /Brass Miter Gears. They are greased to reduce vibrations. The middle miter gear is secured to the shaft with a setting screw coated in blue lock tight.

Due to the high torques required in the robot, the Maxon A-max 32 32 mm diameter, Graphite Brushes, 15 Watt, with terminals was selected. To increase the available torque, the motor was attached to a Planetary Gearhead GP 32 A 32 mm, 0.75 - 4.5 Nm, Metal Version. The motor has a nominal voltage of 12 volts; a no-load speed of 4680 rpm; stall torque of 101 mNm; and a maximum efficiency of 77%. The gearbox has a gear ratio of 5.8:1; a maximum continuous torque of 0.75 Nm; and a maximum efficiency of 80%.

The springs were custom made and donated by Hanson Springs, in Dallas, Texas. The spring parameters:

OD	1.937 in	49.2 mm
ID	1.641 in	41.7 mm
Free Length (FSL)	5.000 in	127. mm
Rate	16.000 lbs/in	2800. N/m
Sugg. Max. Defl.	3.000 in	76.2 mm
Sugg. Max. Load	47.000 lbs	209. N
Solid Length	1.260 in	32. mm
Wire Dia.	0.148 in	3.76 mm
Total Coils	8.500	
Material	Stainless Steel	
Ends	Closed and Ground	
Finish	None	

5.2. Electrical Components

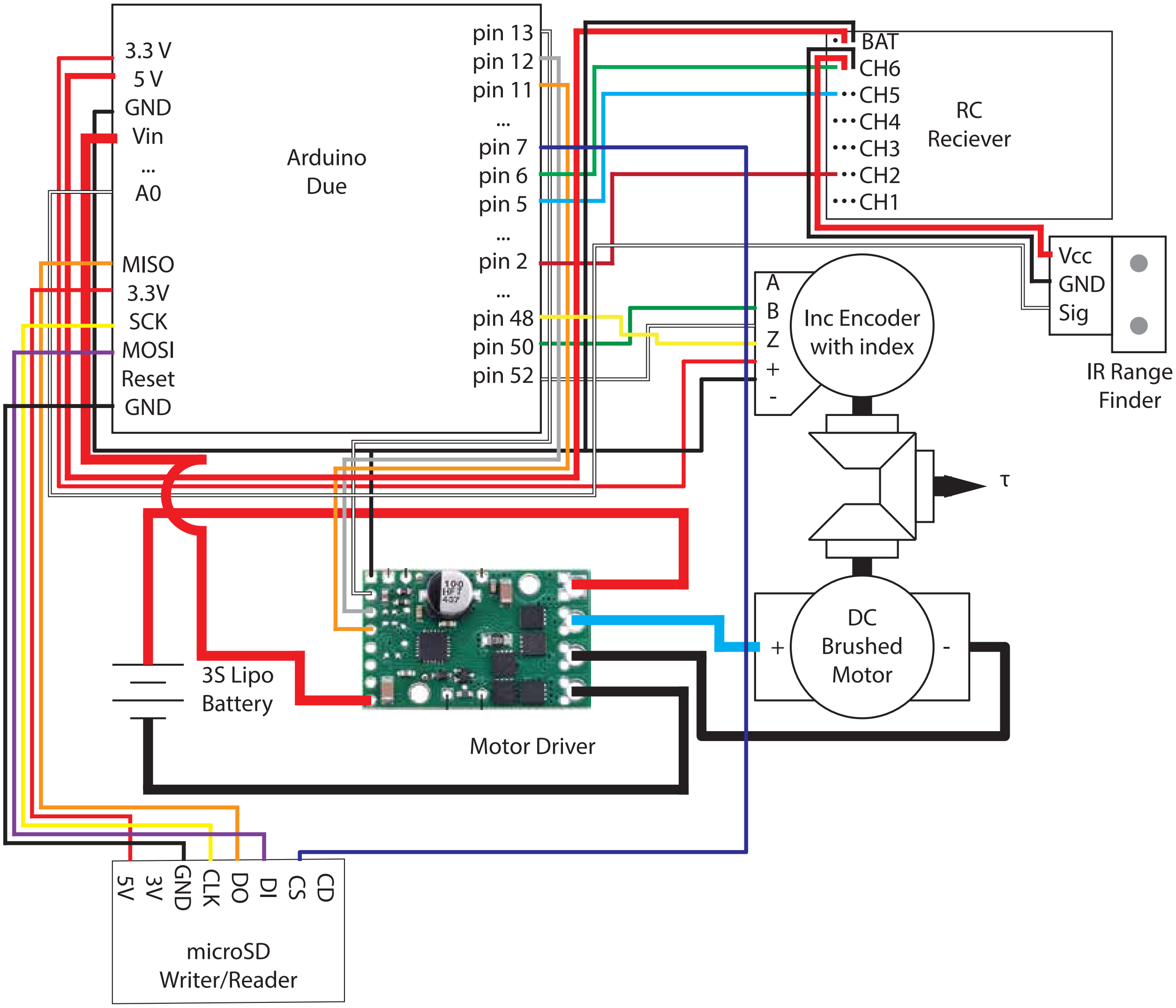

Figure 5.9. Electrical Schematic

The entire controller is run on an Arduino Due. The Arduino Due runs on 84MHz with a 3.3V operating voltage. The Due has 54 digital input/output pins. 12 pins are PWM. All digital pins have interrupts for encoders. The Due also has a SPI header. There are 12 analog input pins for analog sensors.

To power the robot a 3-cell lithuim polymer battery is used. The lithium polymer has an operating voltage of 12.6 to 9 volts. The specific brand and model of the battery is a Turnigy 1000 mAh 3S 45C Discharge Graphene battery.

The power is sent from the batteries to the motor driver. The motor driver used is a Pololu G2 High-Power Motor Driver 18v17. The motor driver can provide a continuous output current of 17 Amps.

A RC Reciever is added to provide direct control of the robot. Navigation is controlled by the user through the RC transmitter/reciever. The RC Reciever is a FlySky FS-R6B receiver. The RC Transmitter is a FlySky FS-T6 transmitter.

A IR Range Finder from Sharp (0A41SK F 6Y) is used. The range finder is specified to work 4-30cm. The sensor sends out an analog signal inversely proportional to the height of the robot. Though the analog signal is continuous, the refresh rate, internally, is 60Hz. The sensor is calibrated offline. There are two parameters that must be calibrated: m and b. The equation for the sensor is as follows:

$$distance = b + \frac{m}{signal} \tag{5.1}$$

The signal is the analog signal read by the digital to analog converter on the Arduino. The distance is measured in meters. The units of the signal are canceled out by empirical experimentation. The experiment is as follows:

1. Place the robot on a stable and flat surface.

2. Through a serial connection to the computer, find the average value of the sensor for a given period of time. Assign this measurement to s_1.

3. Using a caliper, measure the actual height of the robot. Assign this measurement to x_1.

4. Raise the robot from the stable and flat surface by about 10cm.

5. Through a serial connection to the computer, find the average value of the sensor for a given period of time. Assign this measurement to s_2.

6. Using a caliper, measure the actual height of the robot. Assign this measurement to x_2.

7. Using equation 5.2, find b and m.

$$\begin{aligned} b &= \tfrac{s_1 x_1 - s_2 x_2}{s_1 - s_2} \\ m &= \tfrac{s_1 s_2 (x_2 - x_1)}{s_1 - s_2} \end{aligned} \tag{5.2}$$

Finally, a microSD card writer/reader is connected to gather data online. The microSD card writer/reader is the Adafruit MicroSD card breakout board+. The board communicates with the Arduino over SPI.

Chapter 6

EXPERIMENTAL RESULTS

6.1. Spinner System Identification

The experimental results for the Off-Line Spinner Parameter Identification are as follows. One assumption of the model for the spinner is that the spinner is fixed at its pivot point. Obviously, the pivot is not actually fixed in the actual tapping robot. So, in performing the Off-Line Spinner Parameter Identification, the frame of robot was fixed to a table. Also, another assumption of the fixed spinner model is that the available voltage is constant. However, the voltage of a battery is not constant. Therefore, a B&K Precision Quad Display Triple Output DC Power Supply Model 1672 was used to maintain a constant voltage. The voltage was set at 12 volts. The robot was connected to a computer through a serial connection. The Arduino IDE Serial Monitor was used to read the output of the robot. The output was exported to excel. The system identification was ended when the change in any parameter in one step is less than 0.0013. Each step was 10 milliseconds.

Three different experiments were run with different initial conditions. The curves corresponding to each set of initial conditions is shown below.

The results follow:

Curve Color	Blue	Orange	green
$\hat{a}_2[0]$	0	0	2.3
$\hat{a}_3[0]$	0	10	30
$\hat{a}_4[0]$	0	10	30

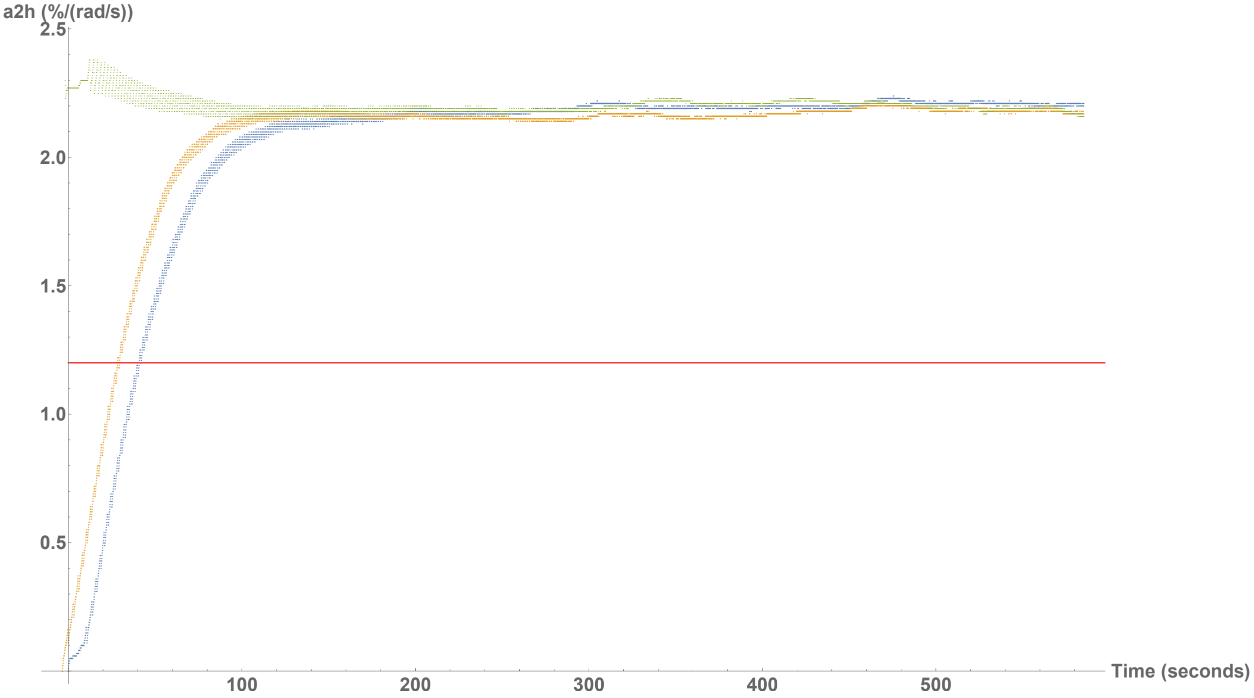

Figure 6.1. $\theta'[t]$ parameter convergence

Figure 6.1 shows the convergence of the $\hat{a2}[t]$ term in equation 3.10. According to the specifications of the maxon motor, the speed constant is 1.165%/(rad/s). However, as shown by the plot, the controller levels off at 2%/(rad/s). Other experimentation showed that there exists a friction force that is constant and not related to the angular velocity. This results in an inverse relationship between the estimated speed constant and the input desired angular velocity. This results in an unstable system at higher speeds. Another experiment, where the desired angular velocity was 10 rad/s, the final estimated speed constant was 1.55%/(rad/s). To solve this, 1.2%/(rad/s) was chosen as a safe guess because stability is more important than performance.

One will notice there is a oscillation in the parameter. This oscillation is due to errors in the filter design and electromechanical sensor of the system.

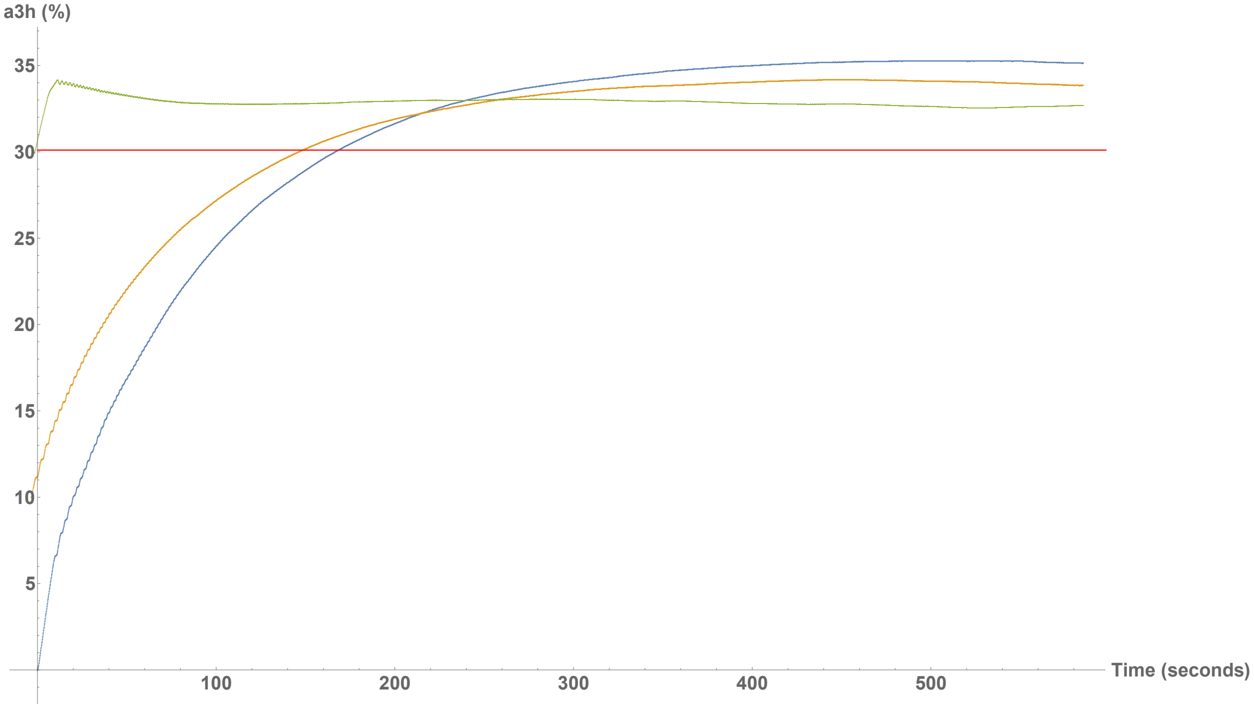

Figure 6.2. $Sin[\theta[t]]$ parameter convergence

Figure 6.2 shows the convergence of the $\hat{a}3[t]$ term in equation 3.10. One way to empirically verify that the parameter has truly converged is that the oscillations in the convergence disappear. The final estimated value is 35.15%, 33.85%, and 32.71%.

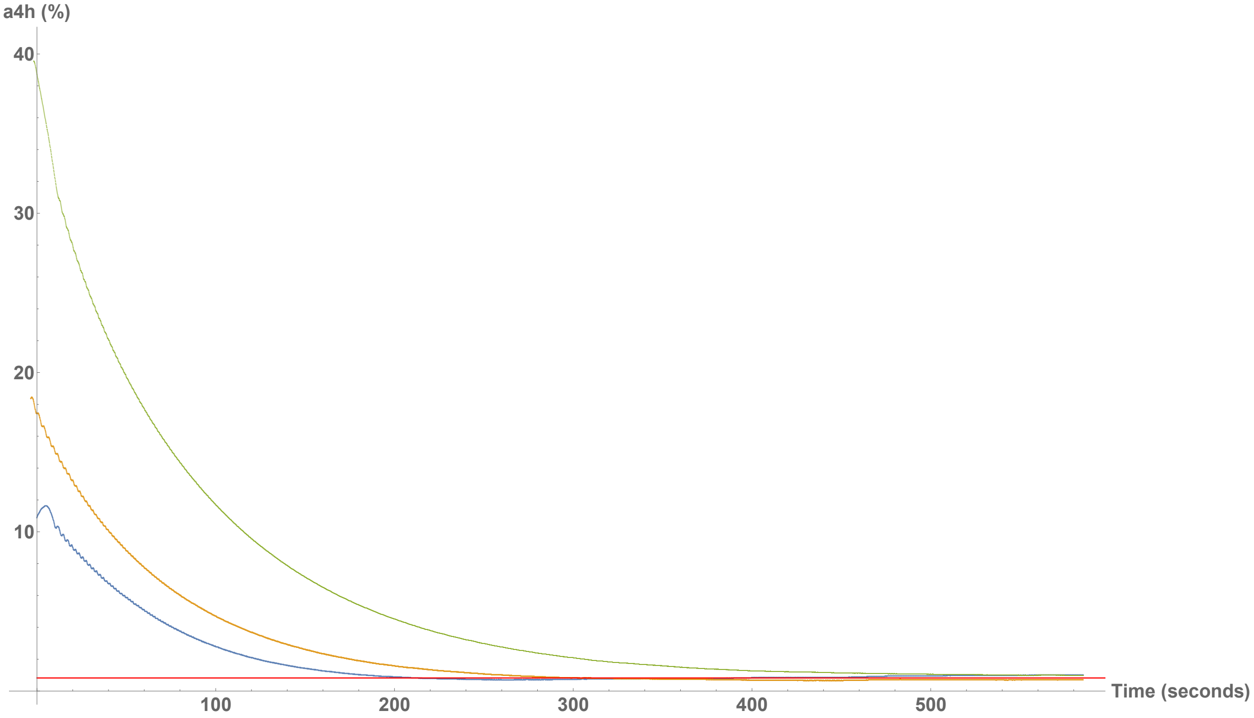

Figure 6.3. $Cos[\theta[t]]$ parameter convergence

Figure 6.3 shows the convergence of the $\hat{a}4[t]$ term in equation 3.10. One way to empirically verify that the parameter has truly converged is that the oscillations in the convergence disappear. The final estimated value is 1.01%, 0.7%, and 0.99%.

Using the trigonometric identity used in equation 3.2, the magnitude of the Sin and Cos parameters in the experiment is 35.2%, 33.9%, and 32.7%. Using standard measuring tools (Calipers, Scales, 3D Modeling, etc...), the magnitude is 29.90%. This shows the two measuring methods yield similar results.

6.2. Tapping Robot

The tapping robot was run at $\omega_d = 36$ rad/s and $\lambda = 0.5625$. Experimentally, through trial-and-error, these parameters were found to correspond to the transition response. Unfortunately, the spring compensator was not engaged in the controller. Therefore, the tapping robot was not run significantly more than the transition frequency. The performance in the experiment can be best compared to figure 4.23.

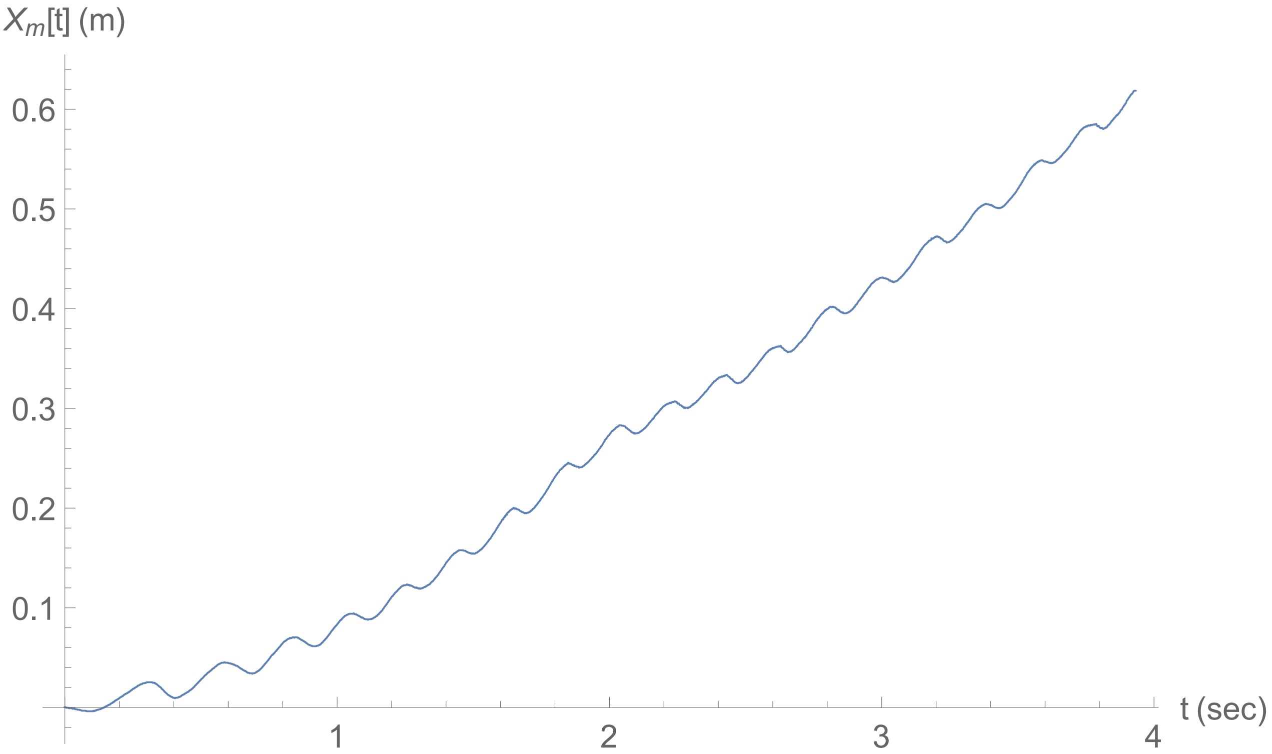

Figure 6.4. Experimental Horizontal Progression of Tapping Robot

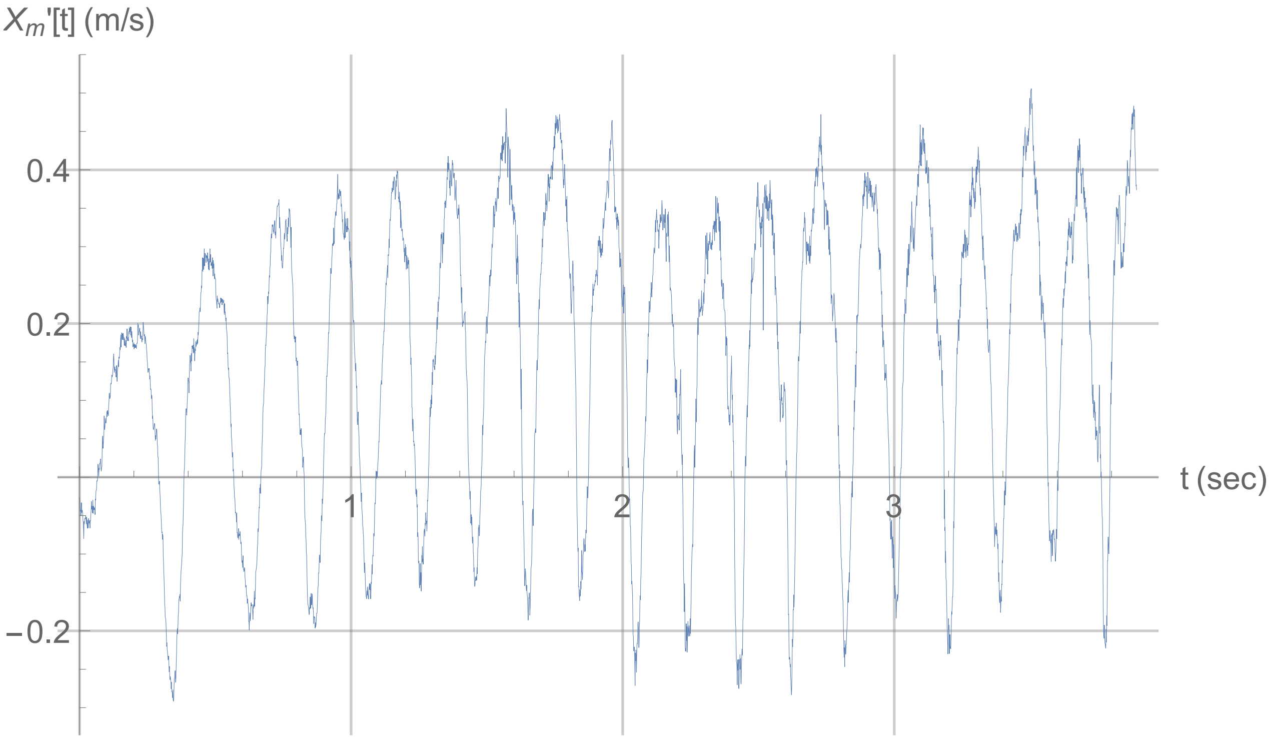

Figure 6.5. Experimental Horizontal Velocity of Tapping Robot

The experiment in figure 6.4 shows a steady forward progression of the tapping robot. In comparison to the simulation in figure 4.23, the robot has a stronger periodic oscillation in its progression. This can be seen in the velocity plot in figure 6.5. In the simulation, the tapping robot's velocity is never negative, where as in the experiment, the robot slides forwards and backwards. The experiment shows the feasibilty of nonlinear feedback linearization of IAJR.

The motion in the experiment is considered the sliding mode.

Chapter 7

DISCUSSION

In this paper we designed a nonlinear feedback linearization controller for inertially actuated jumping robots. We showed that the dynamics of an inertially actuated robot rapidly increase in complexity when changing the design from one dimension (Spinner Angle on a Fixed Pivot) to two dimensions (Basketball Robot Height and Spinner Angle) to three dimensions (Tapping Robot horizontal displacement, Tapping Robot tilt, and Spinner Angle). In order to deal with the increasing complexity, a controller paradigm was created around a invariant set. The invariant set is defined as the total mechanical energy in the system, when in steady state and never jumping. By perserving this invariant set when jumping, through a feedback linearization, the resonance can be used to create a maximum response. However, in order to implement a feedback linearization, the robot's parameters must be known. To solve this problem, a system identification controller was designed to find the parameters of the spinners alone. Next, tuning was used to find the parameters of a spring compensator. The spring compensator is an extra feedback linearization term introduced to help the robot converge to the invariant set more while jumping. Simulation results showed the effectiveness of the controller. A robot was built. Finally, experimentation showed the feasibility of the controller design.

Chapter 8
CONCLUSION

While inertially actuated jumping robots promise speed and manueverability, the work in this paper shows the complexity for controlling them. There is more future work to be done.

The future of IAJR is in drones. The automation in the controller could be combined with unmanned guidance because the quick response of the robots would be best paired with automation and not the slow reflexes of a human operator.

Another future work lies in better analysis of the tapping robot motion. The data gathered in this paper was captured by a camera for 2 dimensional data. This created error in the results that significantly hindered analysis.

The model of the tapping robot requires improvement as well. Experimental results showed that the back end of the robot does not stay in contact with the ground. However in the model, it is assumed to. This could explain a miss-match of the data from the experimental results.

There is also a need for a better understanding of the dynamics of friction between the robot and the ground. First, the static and kinetic friction coefficients were guessed and not experimentally found. However, this is assuming that coulomb friction is even applicable.

Also, the springs are assumed to be fixed to a line perpendicular to the frame of the tapping robot. Experimental results show that there is a longitudinal deflection of the spring to the frame of the tapping robot.

The electronics of the tapping robot were very limiting. Due to a slow range finder, the spring compensator was not able to be implemented in the experiment.

Also, different springs should be tested on the tapping robot to further test the model.

Once the tapping robot's dynamics are perfectly understood and controlled, then IAJR can be extended to greater and more complex designs, such as the designs introduced by Zoghzoghy [30].

The controller designed in this paper shows there is much untapped power in IAJR.

Appendix A

Simulation Parameters

A.1. Basketball Robot Parameters

g	9.81	m/s^2
m_p	0.2*2	kg
l_p	0.08	m
I_p	0	kg*m^2
m_m	1.4	kg
k	1182	N/m
b	5	(N*s)/m
S_f	0.125	m

A.1.1. Basketball Robot Motor Parameters

η	0.625	
Kv	93.2	rad/(sec*V)
R_g	35	
T_s	2104.9	rad/(m*N*sec)
V_b	12.6	volts

A.2. Tapping Robot Parameters

$\delta\theta$	0.448571429	rad
g	9.81	m/s/s
m_p	0.165934	kg
l_p	0.042	m
I_p	0.000461	kg*m
m_m	1.62	kg
I_m	0.015	kg*m
l_s	0.0635	m
l_m	0.3048	m
h_m	0.00296	m
h_p	0.10759	m
k	3600	N/m
b	5	(N*s)/m
S_f	0.087	m
μ_s	0.3	
μ_k	0.25	

A.2.1. Tapping Robot Motor Parameters

η	0.8	
Kv	41.46902303	(rad/s)volts
R_g	5.8	
T_s	4921.828491	rad/(m*N*sec)
V_b	11.6	volts

BIBLIOGRAPHY

[1] Chen, X., quan Wang, L., fen Ye, X., Wang, G., and long Wang, H. Prototype development and gait planning of biologically inspired multi-legged crablike robot. *Mechatronics 23* (2013), 429–444. 1

[2] Hale, E., Schara, N., Burdick, J., and Fiorini, P. A minimally actuated hopping rover for exploration of celestial bodies. *Robotics and Automation* (2000). 1

[3] Hayashi, R., and Tsujio, S. High-performance jumping movements by pendulum-type jumping machines. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)* (2001), vol. 2, pp. 722–727 vol.2. 6

[4] Hurmuzlu, Y. Dynamics of bipedal gait: Part iistability analysis of a planar five-link biped. *Journal of Applied Mechanics 60* (1993), 337. 1

[5] Hurmuzlu, Y., Gnot, F., and Brogliato, B. Modeling, stability and control of biped robotsa general framework. *Automatica 40* (2004), 1647–1664. 1

[6] Hurmuzlu, Y., and Moskowitz, G. D. Bipedal locomotion stabilized by impact and switching: ii. structural stability analysis of a four-element bipedal locomotion model. *Dynamics and Stability of Systems 2* (1987), 97–112. 1

[7] Iida, F., Dravid, R., and Paul, C. Design and control of a pendulum driven hopping robot. *Intelligent Robots and Systems* (2002).

[8] Kashki, M., Ercan, S., and Hurmuzlu, Y. Pivot walking of an inertially actuated robot. *IEEE Transactions Robotics 32* (2016), 1152–1162. 6

[9] Kashki, M., Zoghzoghy, J., and Hurmuzlu, Y. Adaptive control of inertially actuated bouncing robot. *IEEE/ASME Transactions on Mechatronics 22*, 5 (Oct 2017), 2196–2207. 3

[10] Lees, A., Vanrenterghem, J., and Clercq, D. D. Understanding how an arm swing enhances performance in the vertical jump. *Journal of Biomechanics 37* (2004), 1929–1940. 52

[11] Li, B., Deng, Q., and Liu, Z. A spherical hopping robot for exploration in complex environments. *Robotics and Biomimetics (ROBIO)* (2009). 1

[12] Miyashita, K., Ok, S., and Hase, K. Evolutionary generation of human-like bipedal locomotion. *Mechatronics 13* (2003), 791–807. 1

[13] Murphy, M. P., Saunders, A., Moreira, C., Rizzi, A. A., and Raibert, M. The littledog robot. *Int J Rob Res 30* (2011), 145–149. 1

[14] Murthy, S. S., and Raibert, M. H. Hopping in legged systems-modeling and simulation for the two-dimensional one-legged case. *IEEE Transactions on Systems, Man, and Cybernetics 18* (1984), 27–27. 1, 6

[15] Narendra, K. S., and Annaswamy, A. M. *Stable Adaptive Systems*. Prentice Hall, 1989. 26

[16] Okubo, O. H., Nakano, E., and Handa, M. Design of a jumping machine using self-energizing spring. *Intellegent Robots and Systems* (1996). 6

[17] Prosser, J., and Kam, M. Control of hopping height for a one-legged hopping machine. *Control Applications* (1992). 6

[18] Raibert, M., Blankespoor, K., Nelson, G., and Playter, R. Bigdog, the rough-terrain quadruped robot. *The international federation of automatic control 41* (2008), 10822–10825. 1

[19] Raibert, M. H. *Legged robots that balance*. The MIT Press, 1968. 1

[20] Slotine, J.-J. E., and Li, W. *Applied Nonlinear Control*. Prentice Hall, 1991. 28

[21] Song, G., Yin, K., Zhou, Y., and Cheng, X. A surveillance robot with hopping capabilities for home security. *IEEE Transactions on Consumer Electronics 55* (2009), 2034–2039. 2

[22] Tavakoli, A., and Hurmuzlu, Y. Gravity powered locomotion and active control of two simple systems. *Proceedings of ASME dynamics and control conference* (2009). 6

[23] Verrelst, B., Vanderborght, B., Vermeulen, J., Ham, R. V., Naudet, J., and Lefeber, D. Control architecture for the pneumatically actuated dynamic walking biped lucy. *Mechatronics 15* (2005), 703–729. 1

[24] Westervelt, E., Chevallereau, C., Morris, B., Grizzle, J., and Choi, J. H. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007. 1

[25] WU, Q., and CHEN, J. Effects of ramp angle and mass distributions on passive dynamic gait an experimental study. *International Journal of Humanoid Robotics 07* (2010), 55–72. 1

[26] Wu, Q., and Swain, R. On lyapunov's stability control of two-link base-excited inverted pendulums with applications to human locomotion. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on* (2000), vol. 5, pp. 3330–3335 vol.5. 29

[27] Yi, S. Reliable gait planning and control for miniaturized quadruped robot pet. *Mechatronics 20* (2010), 485–495. 1

[28] Zhang, J., Song, G., Li, Y., Qiao, G., Song, A., and Wang, A. A bio-inspired jumping robot: Modeling, simulation, design, and experimental results. *Mechatronics 23* (2013), 1123–1140. 1

[29] Zoghzoghy, J., Alshorman, A., and Hurmuzlu, Y. Inertially actuated baton locomotor. *Proceeding of ASME dynamics and control conference* (2013). 6

[30] Zoghzoghy, J., Zhao, J., and Hurmuzlu, Y. Modeling, design, and implementation of a baton robot with double-action inertial actuation. *Mechatronics 29* (2015), 1–12. 5, 86