5-2012

Analysis Modeling and Optimization of a Smart Sulky

Nicholas C. Noble

ANALYSIS MODELING AND OPTIMIZATION OF A SMART SULKY

by

Nicholas C. Noble

A Thesis Submitted in Partial Fulfillment
of the Requirements for a Degree with Honors
(Mechanical Engineering)

The Honors College

University of Maine

May 2012

Advisory Committee:
Michael Peterson, Professor of Mechanical Engineering, Advisor
James E. Gallagher, Associate Professor of Sociology Emeritus, Honors College
Zhihe Jin, Associate Professor of Mechanical Engineering
Alireza Sarvestani, Assistant Professor of Mechanical Engineering
Senthil Vel, Professor of Mechanical Engineering

Abstract

For several hundred years, the Standardbred harness racing industry has remained relatively unchanged. The sulkies pulled by the horses have seen only minor changes. New construction techniques and lighter materials have allowed sulkies to become lighter and generally faster. However, there is currently no sulky that uses a modern control system to adapt to changing racetracks, horses, and drivers. A Smart Sulky would be able to accomplish these things. The Smart Sulky would be capable of measuring changes in the movements of the horse and compensate for rotational movement and vibration, which the horse transfers to a conventional sulky. The Mechanical Engineering Smart Sulky capstone group has taken up the challenge of designing such a sulky. As part of that work it was necessary to develop the equations of motion that govern the movements of the Smart Sulky, as well as to perform analyses of testing done on conventional sulkies to determine the general behavior of a sulky. Data analysis was done to determine the constants in a system representing the Smart Sulky. The formulation of the equations of motion governing the Smart Sulky movement is the focus of this work. These will be implemented as a part of the Smart Sulky Mechanical Engineering capstone project.

Table of Contents

Table of Figures

The Problem

The original problem for this work began as part of the Smart Sulky Mechanical Engineering Capstone group. The purpose of the Smart Sulky is to develop a harness racing sulky, for use in Standardbred harness racing, which will go faster than a conventional sulky. This idea comes from the cousin to harness racing, Thoroughbred racing.

> Horse racing has existed in its current form for over 200 years. The apparently uncomfortable modern race riding posture was developed in the United States in the late 19th century… This change in riding style…corresponds to a dramatic improvement of 5 to 7% in race times in the United States between 1890 and 1900…strongly suggesting that the adopted posture benefits racing performance. This improvement is greater than that observed over the following century. (Pfau et al. 2009)

In the late 19^{th} century, Thoroughbred racing jockeys changed from a sitting position to the current position. In the current position, the jockey uses his legs to lift himself off the horse. By lifting himself off the horse, the jockey is able to compensate for the jerking motion that occurs when the horse's hooves hit the ground. Using his legs, the jockey is able to constantly keep his momentum moving forward, and thus reduce the work the horse must do. This work reduction allows for the 5 to 7 % improvement in race times mentioned by Pfau. The change in Thoroughbred racing position, and the resulting improvement in race times, is the basis for the idea of a Smart Sulky. Instead of using the jockey's legs to compensate for the horse's motion, a mechanical system would be required in the sulky arms to compensate for the horse's movements.

To begin with, several approaches were proposed for this project. The alternatives considered included installing linear actuators in each arm of the sulky, using adjustable shock absorbers, a hydraulic accumulator system, or a pneumatic system.

During initial work on the project, it became evident that equations of motion would allow the proposed design to be optimized. The modeling was ideal as an Honor's thesis because it is independent from the remaining work on the capstone. The focus of this thesis is to develop the equations of motion for the Smart Sulky in order to try to optimize the proposed design and determine starting values for constants in the system.

Initial Work

To develop the equations of motion, a method must be selected for deriving the equations. The approach that is conventionally taught in modern engineering courses is Newtonian. With this method, the forces, moments, masses and mass moments of inertia acting on a body are summed in each axis to determine the equations governing the actions of the body.

Newtonian Mechanics

A simple example system, which will be used to illustrate the Newtonian approach for deriving equations of motion, is shown in figure 1.

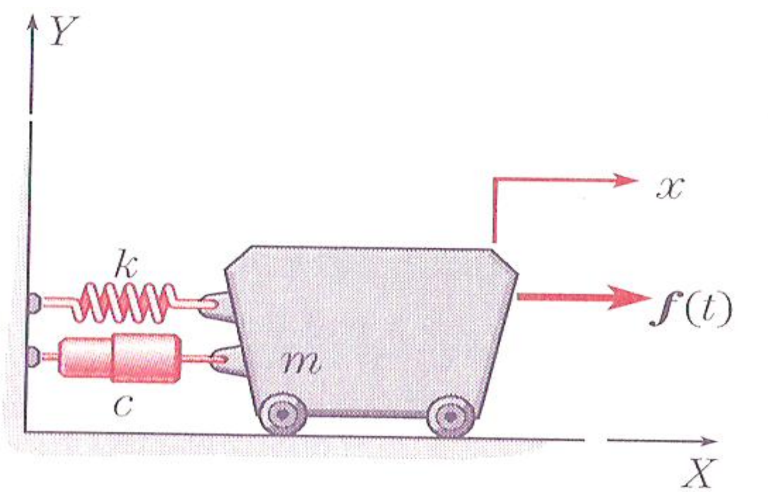

Figure 1. Example of a single mass with spring, damper, and applied force (Williams 1996).

The system in figure 1 is rather simple, forces only act in one direction, the x direction. In addition to the applied force, f(t), the forces related to the spring and the damper also act on the mass. The sum of the forces in the direction of interest, the x direction is equal to the mass of the body times its acceleration because this is a dynamic system (a system which can change position over time).

$$\sum F_x = m\ddot{x} \qquad \text{(Eq.1)}$$

Where F_x are forces in the x-direction and $\ddot{x}$ is the second time derivative of position. Eq. 2 shows the forces applied by the damper, the spring, and the applied force on the right hand side of the equation.

$$\sum F_x = m\ddot{x} = -c\dot{x} - kx + f(t) \qquad \text{(Eq. 2)}$$

Rearranging these equations so that only the applied force is on the right hand side of the equation; is one common way to display the equation of motion.

$$m\ddot{x} + c\dot{x} + kx = f(t) \qquad \text{(Eq. 3)}$$

Newtonian mechanics is an approach to developing the equations of motion of a body that works well for simple systems. However, for systems that have more than two degrees of freedom (systems that move in more than two independent directions) Newtonian mechanics become somewhat more difficult. For systems where multiple motions are coupled, more sophisticated methods are preferred. The initial concept for the Smart Sulky was that some kind of device would be put in each arm of the sulky. This means that the sulky would become at least a four degree of freedom system. The front part of each arm would move independently of the back part of each arm for a total of four degrees. In addition, the rotation of the sulky around the center of mass would need to be considered. With the prospect of a system this complex, the idea of using Newtonian mechanics was never even considered. As an alternative, Lagrangian mechanics could be used to derive the equations of motion for the Smart Sulky.

Lagrangian Mechanics

The difference between Lagrangian and Newtonian mechanics is that in Newtonian mechanics the relationships between rigid bodies are defined in terms of force and acceleration vectors. In Lagrangian mechanics, the parts of the system are characterized in terms of energy, a scalar quantity. By following the rules that govern the actions of the various parts of the system, new components can easily be added to the equation.

In order to determine the equations of motion of a system using Lagrangian mechanics, a prescribed system can be used.

1. Determine the generalized coordinates of the system, i.e. x,y,z.
2. Determine the generalized forces. These are the nonconservative forces acting on a system, typically associated with dampers and applied forces.
3. Determine the Lagrangian, this consists of the inertia and conservative forces in the system, examples include moving bodies and springs.
4. Create Lagrange’s equation; this is the equation of motion for the system, created using the generalized forces and the Lagrangian (Williams 1996).

Below is the same example solved before using Newtonian mechanics. However, it is now solved with Lagrangian mechanics.

The first step in determining the equations of motion of this system is to determine the generalized coordinates. For this basic example, the only coordinate is: x. The next step is to determine the generalized forces in the system, the equation for this is:

$$\delta W^{nc} = \sum_{i=1}^{N} \boldsymbol{f}_i^{nc} \cdot \delta \boldsymbol{R}_i = \sum_{j=1}^{n} \Xi_j \delta \xi_j \qquad \text{(Eq. 4)}$$

While this equation appears intimidating, what it actually means is quite simple. The work done in the system by the nonconservative forces is equal to all the nonconservative forces times the generalized coordinate over which the force acts. This is also equal to the generalized forces times the displacement associated with the forces. Nonconservative forces are the forces that dissipate or introduce energy, in this case the damper and the applied force:

$$\delta W^{nc} = f(t)\delta x - c\dot{x}\delta x = \Xi_x \delta x \qquad \text{(Eq. 5)}$$

From this, the generalized force Ξ_x is

$$\Xi_x = f(t) - c\dot{x} \qquad \text{(Eq. 6)}$$

The next step is to determine the Lagrangian of this system. In this system, the only forces in the Lagrangian are the kinetic energy of the mass and the potential energy of the spring.

$$\mathcal{L} = T^* - V = \frac{1}{2}m\dot{x}^2 - \frac{1}{2}kx^2 \qquad \text{(Eq. 7)}$$

Next, using the generalized forces and the Lagrangian, Lagrange's equation can be formed.

$$\frac{d}{dt}\left(\frac{\delta \mathcal{L}}{\delta \dot{x}}\right) - \frac{\delta \mathcal{L}}{\delta x} = \Xi_x \qquad \text{(Eq. 8)}$$

Substituting in the generalized forces and computing the partial derivative of the Lagrangian:

$$\frac{d}{dt}(m\dot{x}) + kx = f(t) - c\dot{x} \qquad \text{(Eq. 9)}$$

Finally, taking the derivative with respect to time and rearranging the equation results in the equation of motion for the simple system. Note that Eq. 10 below is the same as Eq. 3 (Williams 1996).

$$m\ddot{x} + c\dot{x} + kx = f(t) \qquad \text{(Eq. 10)}$$

Equations of Motion

Before any work could be done on deriving equations of motion for the Smart Sulky, it was first necessary to determine the proposed design of the Smart Sulky. Figure 2 shows the Smart Sulky design proposed at the end of the fall semester.

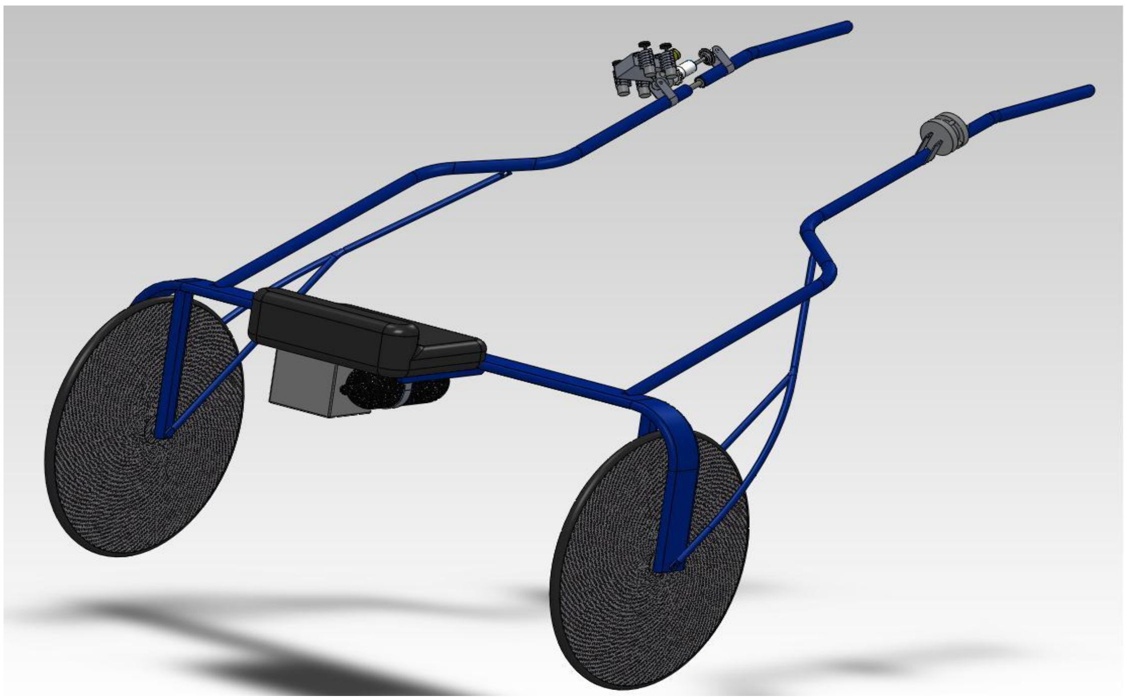

Figure 2. Image of Smart Sulky Design.

Once the proposed design was finalized, the equations of motion for the sulky could be derived. On the left arm of the sulky is an adjustable shock absorber consisting of a spring and a damper. On the right arm is a pneumatic cylinder that will apply a force, which will be out of phase with the force applied by the horse. The equations in Appendix A represent the initial work to determine the equations of motion for the Smart Sulky.

Initially, the problem was approached by developing equations of motion for the Smart Sulky represented in its physical form, see figure 12 in Appendix A. The equations of motion for the entire sulky were developed following the Lagrangian mechanics approach until an eight degree of freedom system had been developed; see Appendix A for the entire original equations. Once the equations had been developed, they were programmed in MATLAB (MathWorks, Waltham MA) and solved using the MATLAB function ode45. The ode45 function is a built-in function in MATLAB that can be used to solve differential equations, such as the equations of motion. Appendix B gives the complete MATLAB code created for the equations of motion. The plots in Appendix C were created using the complete equations of motion and the MATLAB code in Appendix B.

The plots of the position and velocity of masses 3 and 4 appeared to be reasonable. The pneumatic cylinder attached to mass 3 applies a force that is out of phase with the force applied from the horse, shown in peaks on the plot of mass 4. However, the plots for masses 1 and 2 were not reasonable. Both plots start at zero and then after a period of approximately 200 seconds go to either negative or positive infinity. Because of the apparent instability of the equations, the model used for the sulky was simplified. Figure 3 below is a sketch of the simplification used for determining the equations of motion of the Smart Sulky.

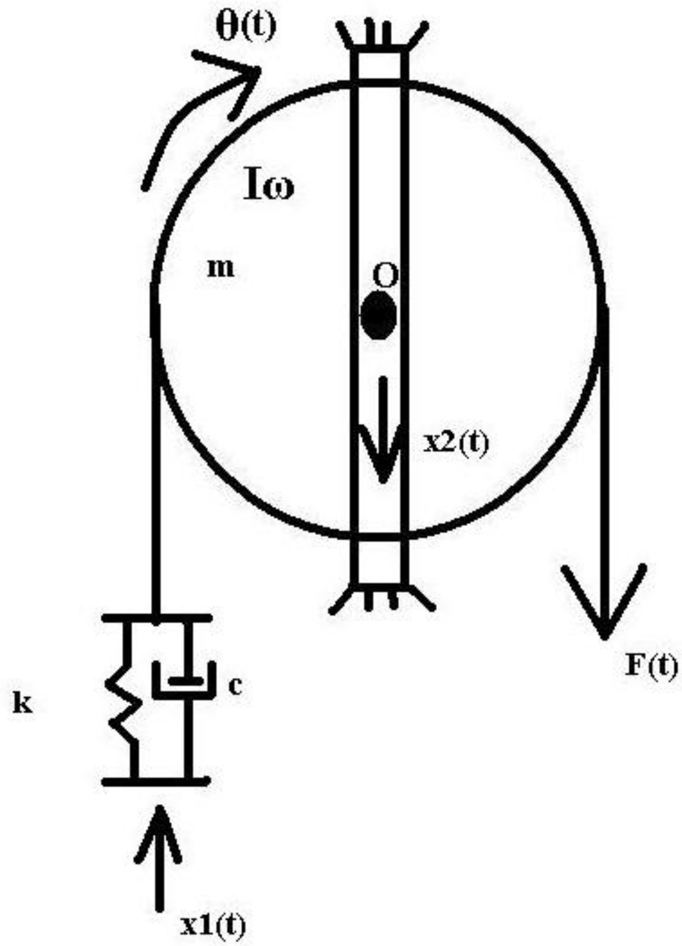

Figure 3. Final Smart Sulky representation.

The change from using the equations of motion for the Smart Sulky in Appendix A, to the representation in figure 3, was made for several reasons. First, attempts to determine the sulky's movements from the total model of the sulky were unsuccessful. Secondly, as the project progressed, the overall objectives of the group began to change. Originally, the outcome of the project would be a completed racing sulky ready to go race. Over the course of the year, this objective changed to creating a sulky that could be used for prototyping methods of limiting the vibrations from being transferred between horse and sulky. Because the objectives changed, it was no longer necessary to have detailed equations of motion. Instead of modeling how the sulky would react under all conditions, the equations of motion would now be used to tune the force required to damp out the vibrations. In addition, starting values for the spring constant and damping coefficient can also be determined using these equations. A simpler representation of the sulky, allows Newtonian instead of Lagrangian mechanics to be used to determine the equations of motion.

The first step is to establish that the sum of all moments about point O are equal to the rotational moment of inertia times the rotational acceleration. This dynamic system is then:

$$\sum M_o = I_\omega * \ddot{\theta} \qquad \text{(Eq. 11)}$$

The next step is to sum moments about point O. A moment, is a force times the distance from point O, summing these moments:

$$\sum M_o = -c * \left(R * \dot{\theta} - \dot{x}_1\right) * R - k * \left(R * \theta - x_1\right) * R + R * F(t) \qquad \text{(Eq. 12)}$$

Eq. 11 and 12 are then combined to obtain Eq. 13. Equation 13 is the first equation of motion of the Smart Sulky; it represents the rotational movement of the sulky.

$$I_\omega * \ddot{\theta} = -c * \left(R * \dot{\theta} - \dot{x}_1\right) * R - k * \left(R * \theta - x_1\right) * R + R * F(t) \qquad \text{(Eq. 13)}$$

The next equation of motion for the sulky is for the x-direction. In figure 3, the sulky is modeled as a body constrained to move only in the x-direction. All the forces in the x-direction are equal to the mass of the sulky times the acceleration in the x direction. Because the sulky is a dynamic system, the result is:

$$\sum F_x = m * \ddot{x}_2 \qquad \text{(Eq. 14)}$$

With the forces in the x-direction, as shown in figure 3, giving:

$$\sum F_x = c * \left(R * \dot{\theta} - \dot{x}_1\right) + k * \left(R * \theta - x_1\right) + F(t) \qquad \text{(Eq. 15)}$$

Eq. 14 and 15 can now be combined to show the sulky's equation of motion in the x direction.

$$m * \ddot{x}_2 = c * \left(R * \dot{\theta} - \dot{x}_1\right) + k * \left(R * \theta - x_1\right) + F(t) \qquad \text{(Eq. 16)}$$

Eq. 13 and 16 represent the required equations to characterize the sulky at a given moment. Eq. 17 is a rearrangement of Eq. 13. The form of Eq. 17 is suitable for equations in a mathematical modeling program.

$$\ddot{\theta} = \frac{d\theta}{dt} * \left(\frac{-c*R^2}{I_\omega}\right) + \theta * \left(\frac{-k*R^2}{I_\omega}\right) + \dot{x}_1 * \left(\frac{c*R}{I_\omega}\right) + x_1 * \left(\frac{k*R}{I_\omega}\right) + \frac{F(t)*R}{I_\omega} \qquad \text{(Eq. 17)}$$

Eq. 18 rearranges Eq. 16 in the same way Eq. 17 rearranged Eq. 13. Eq. 17 depends on the rotational position and velocity, so that Eq. 17 must be solved first and then the results from that used to solve Eq. 18.

$$\ddot{x}_2 = \dot{\theta} * \frac{c*R}{m} + \theta * \frac{k*R}{m} - \dot{x}_1 * \frac{c}{m} - x_1 * \frac{k}{m} + \frac{F(t)}{m} \qquad \text{(Eq. 18)}$$

Once the equations of motion have been determined, the known values of the system should be determined. The mass of the sulky was determined using a scale, and the rotational moment of inertia was calculated by the Smart Sulky group using a torsional pendulum. An initial spring constant has been based on the spring that was selected by the Smart Sulky group. The equations of motion will make it possible to tune the damping coefficient and the applied force to minimize motion of the sulky. The determination of x_1 is described below .

Data analysis

In an attempt to design an algorithm to determine the kinematics of at least one arm of the sulky, the capstone group collected acceleration data, during training, from a conventional sulky. The data was taken using an Arduino microcontroller, Analog Devices ADXL345 accelerometer, and a SD card datalogger. Code for the accelerometer was written by the Smart Sulky team and was only tested in use. This acceleration can be used to determine the position, which would be the basis for the kinematics of each arm of the sulky. This analysis is part of the modeling effort as required input. One approach was to use the cumtrapz function in MATLAB. The purpose of cumtrapz is to perform "cumulative trapezoidal numerical integration" (Anonymous 2012). Double numerical integration was performed on the test data in order to determine an equation governing the position of the sulky. Figure 4 shows the results of analysis with the cumtrapz function.

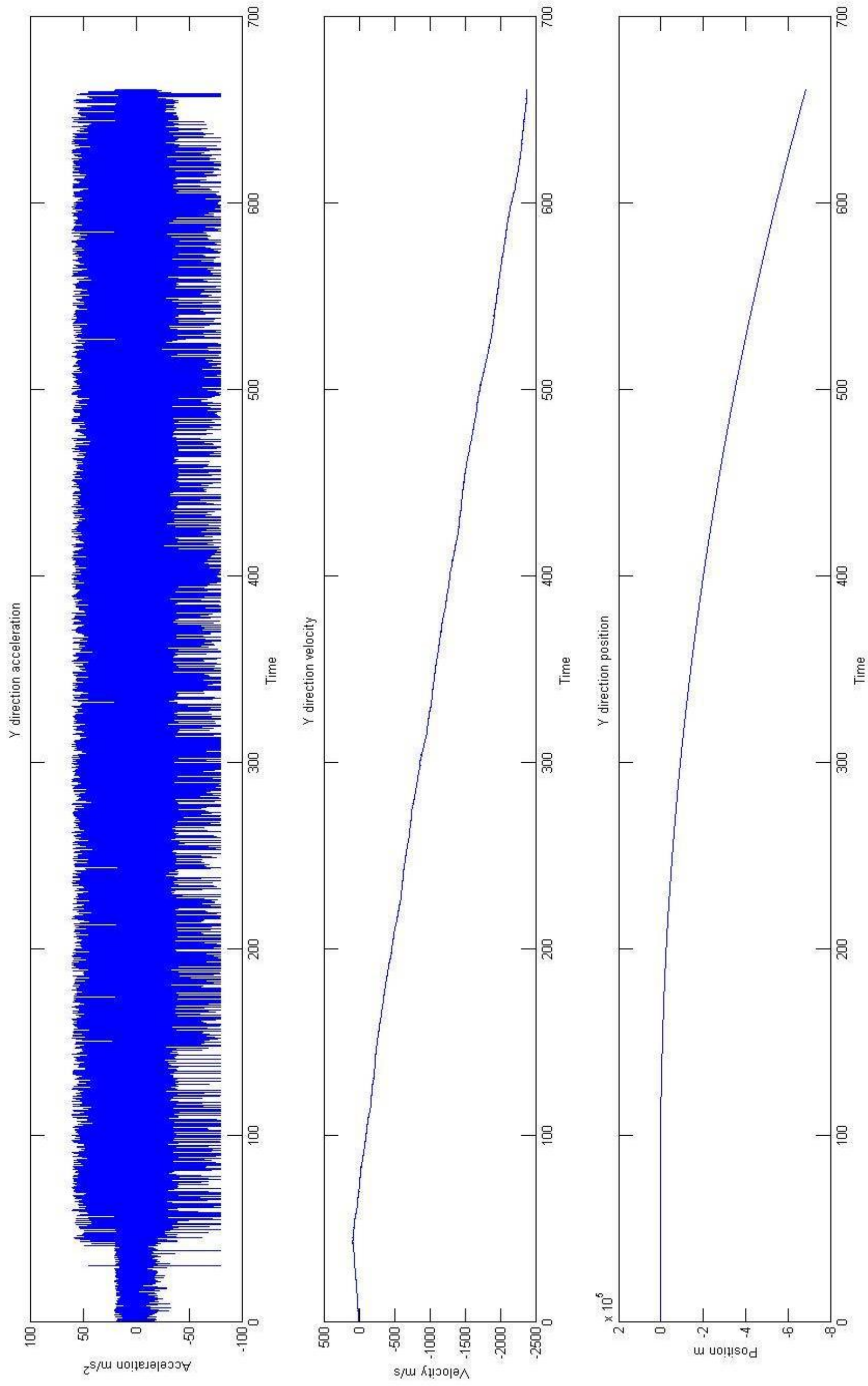

Figure 4. Attempt to determine velocity and position from raw data using cumtrapz.

The top graph in figure 4 shows the acceleration recorded during the testing, the middle plot is the calculated velocity, and the bottom plot is the position calculated from the acceleration. The acceleration graph is the data taken during testing, and the velocity and position graphs were created using the cumtrapz function. Alternative numerical integration codes were explored that would give reasonable plots for the position of the sulky over time. Problems with the cumtrapz function not working were caused by the test data's low signal to noise ratio. The sampling frequency was not constant, ranging from 90 and 102 Hz. The erratic nature of the test data is probably why the cumtrapz function could not correctly integrate the data.

In order to characterize the noise, a time frequency spectrogram and Discrete Fourier Transform (DFT) were performed on the data. Spectrograms and DFT analyses are methods of analyzing various types of signals to determine if any structure occurs in the frequencies present in the data. Figures 5, 6, and 7, below, are of the DFT analysis and spectrogram that were performed on the test data.

The DFT was performed separately on each arm of the sulky. For each arm the DFT was performed using four different approaches. Because of the way the DFT algorithm works, the number of samples must equal two raised to an integer power. For example, 2^4 samples or 2^{15} samples, etc. If the number of samples is not equal to 2 raised to a power, then the program automatically inserts zeros in the data set until there are a number of samples equal to 2 raised to a power. Alternatively, the code can be modified to force the analysis to finish when it runs out of samples and not add the zeros until it gets to 2 to a power. Sometimes, doing this can bring out patterns in the data that are not evident when using the padding zeros. The other two graphs in figures 5 and 6 were created by eliminating the offset in the data. This was also done simply to see if any new patterns would arise. Figure 5 shows the DFT analysis for the left arm, and figure 6 shows the DFT for the right arm.

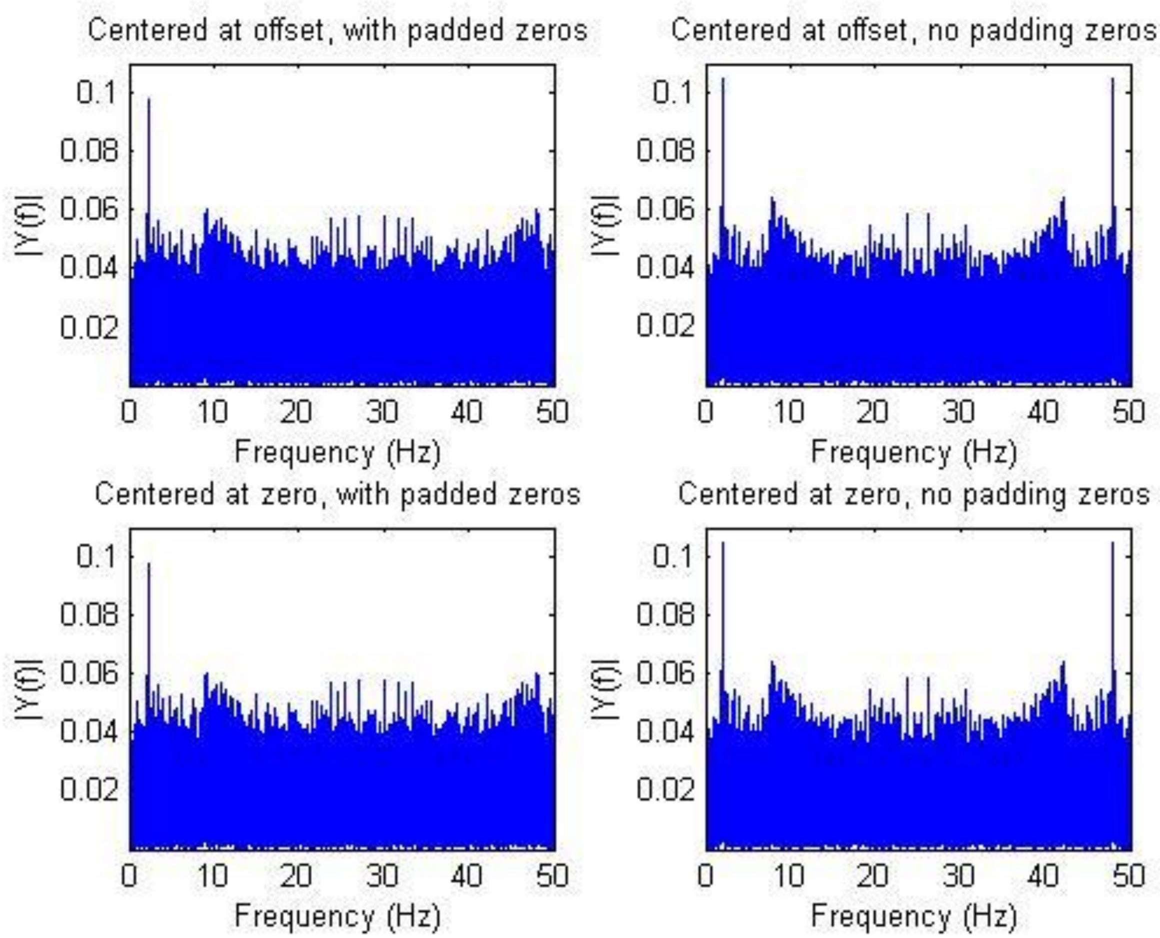

Figure 5. DFT analysis of acceleration data from left arm of sulky.

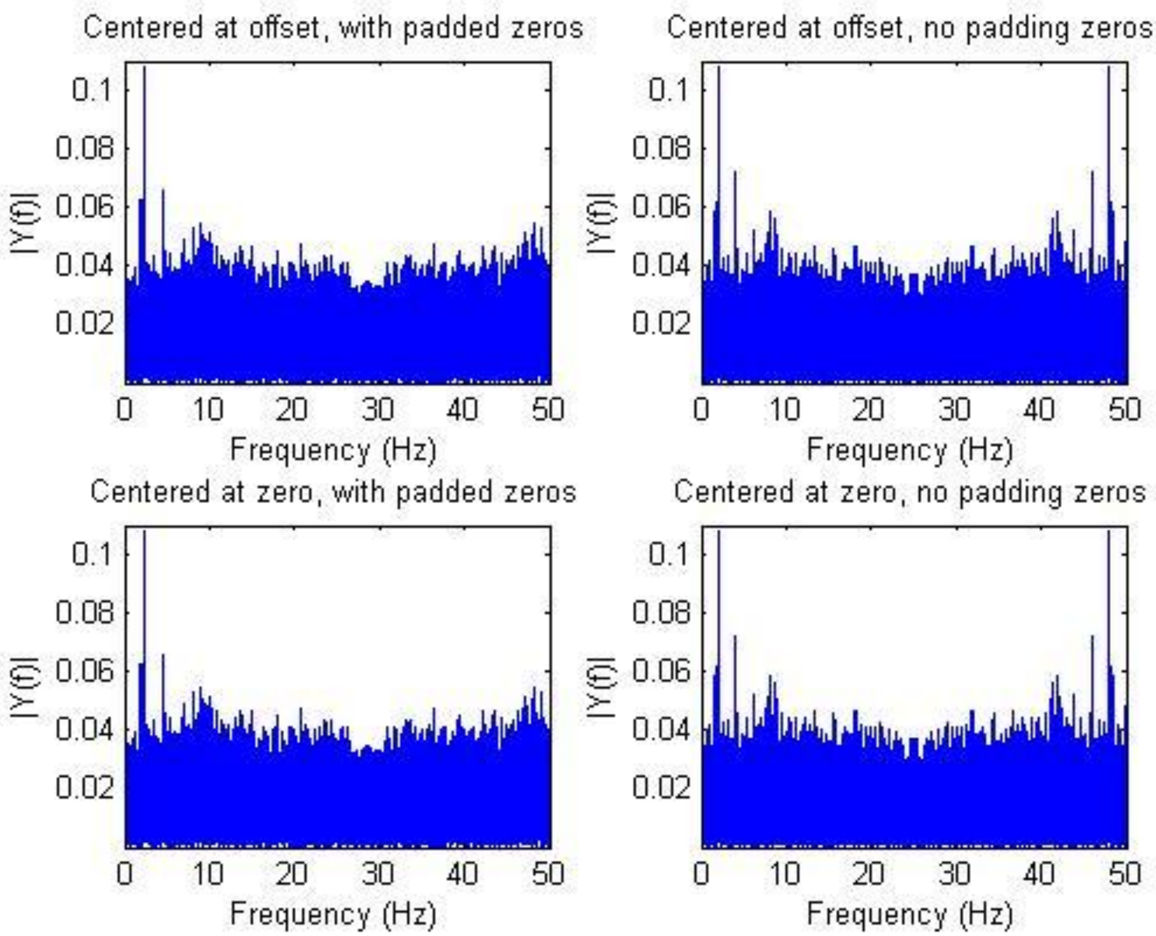

Figure 6. DFT analysis of acceleration data from right arm of sulky.

In both frequency analyses, the two graphs that were created without padding zeros are mirrored at about 25 Hz. The graphs created with the padding zeros also have a mirror point, but in those graphs, it is closer to 30 Hz. The higher value of the mirror point for the graphs with zeros is because there are more samples included in the analysis. Therefore, there is no difference between the analysis using padding zeros or not. The primary peak in each graph at near 3 Hz suggests that the gait related oscillation frequency of the horse and sulky is around 3 Hz. This information corresponds to research done by the Smart Sulky group, which suggests the average stride frequency of a

Standardbred racing horse is approximately 3 Hz (Leleu 2005). A spectrogram was also performed to validate the 3 Hz peak from the DFT. In figure 7 below, a line appears at around 3 Hz, this serves as further evidence that approximately 3 Hz is the oscillation frequency of the sulky.

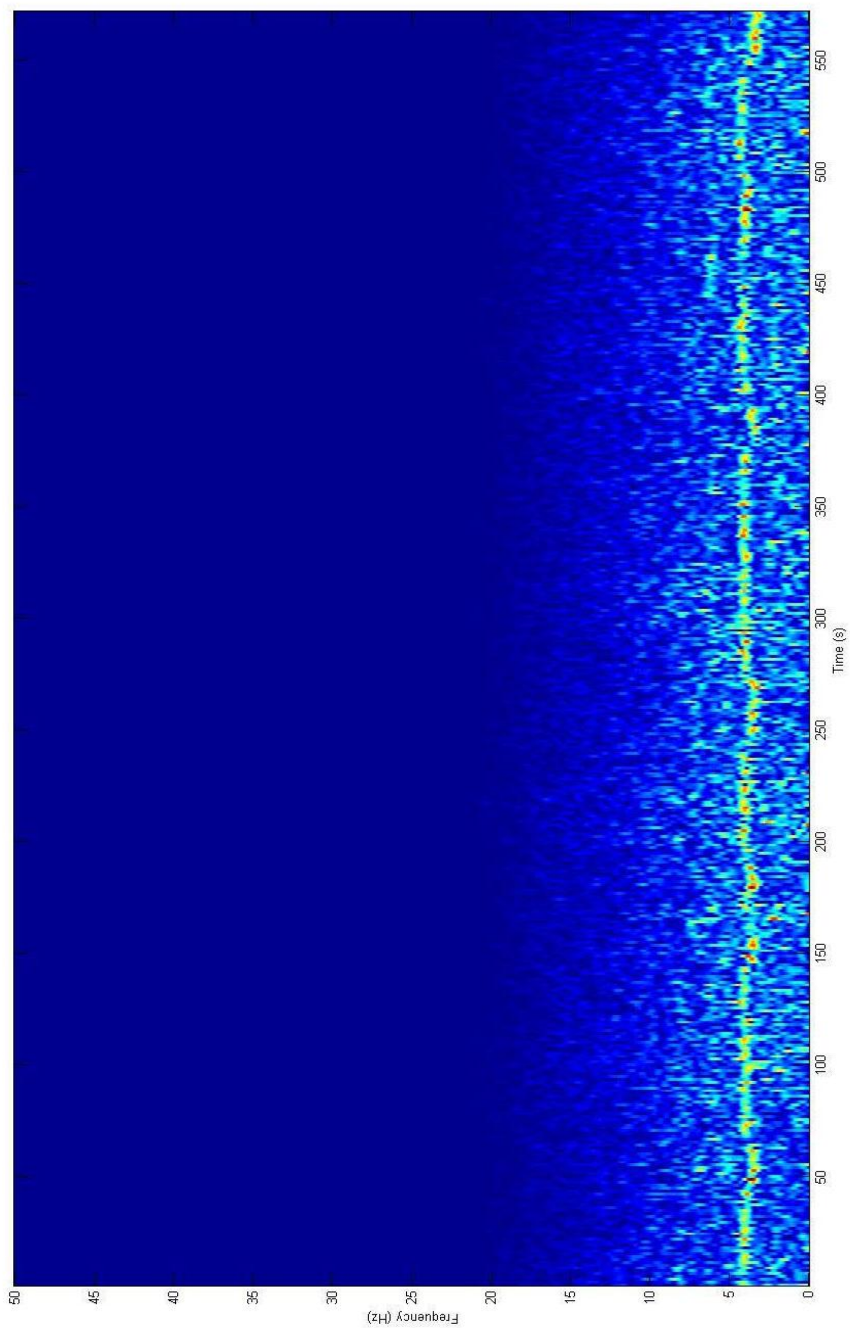

Figure 7. Spectrogram of acceleration data from left arm.

The test sulky data suggests that the 3 Hz frequency is the dominant motion of the sulky that will require a response from the Smart Sulky control system. By approximating the acceleration with a sine wave, the acceleration and force information required for the equations of motion were simplified.

Smart Sulky Motion Predictions

With the equations of motion complete, and the values of the input displacement simplified, the next step is to create a mathematical model of the Smart Sulky. The easiest way to do this is with a computer-modeling program, in this case, Simulink (MathWorks, Waltham, MA). Figure 8 is the Simulink block diagram created for solving the equations of motion, Eq. 17 and 18 above. The red blocks solve Eq. 17, the equation for the rotational acceleration of the sulky, theta double dot. The values of rotational velocity and position are then used to solve for the translational acceleration (blue blocks), the acceleration of x2. The large oval blocks in the center of each equation are summation blocks; these blocks combine the effects of all the elements of the equations of motion. Each input to the summation block is labeled, allowing for easy identification and modification of the inputs.

The objective of all this work has been to determine the equations of motion of a Smart Sulky. The equations of motion will then be used to reduce the vibrations that are transferred from the horse to the sulky. With a set of equations, and a completed block diagram, the next step is to adjust parts to the system in order to reduce the vibrations as much as possible.

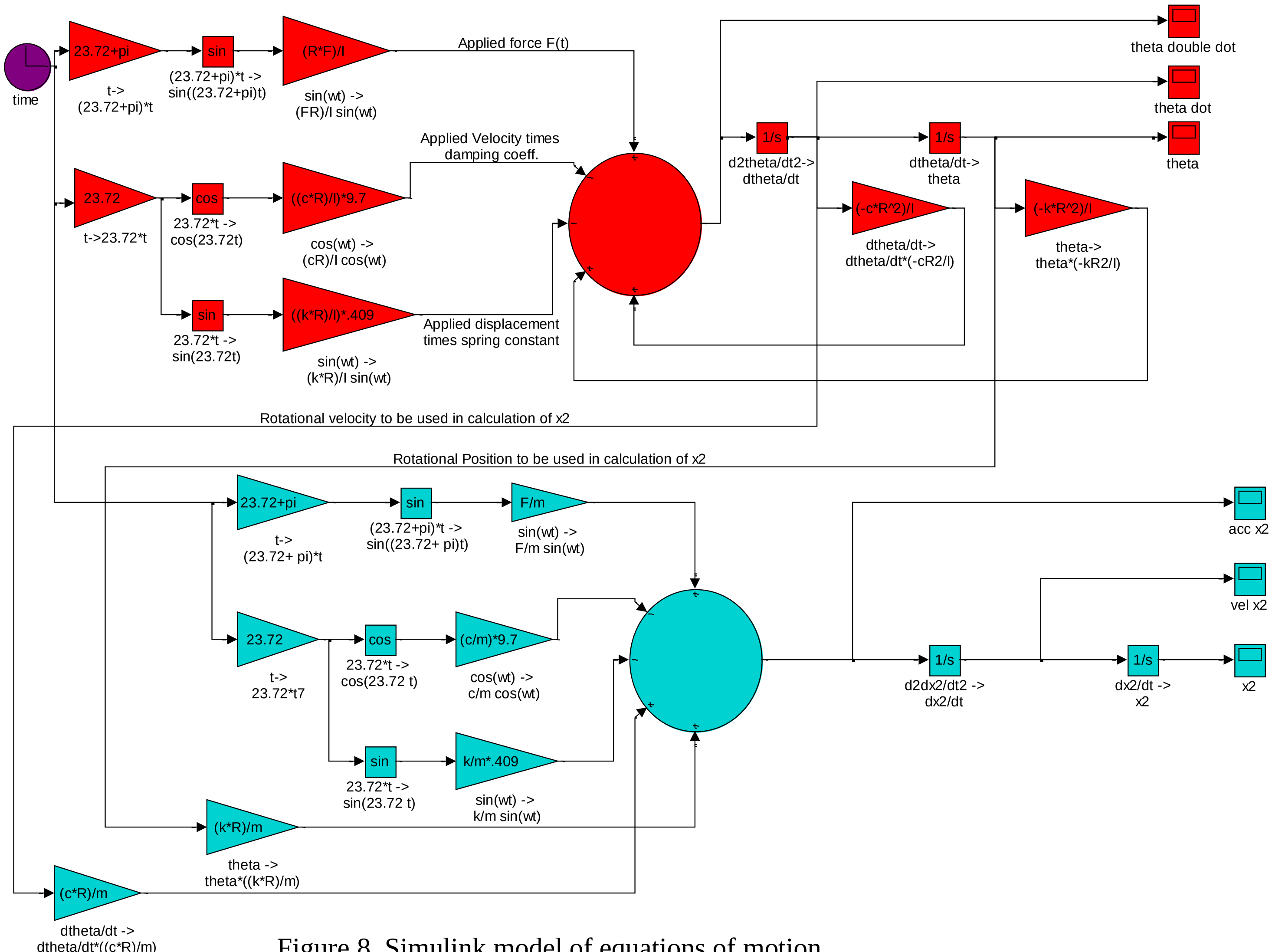

Figure 8. Simulink model of equations of motion.

The input displacement is prescribed as -0.409*sin(23.72*t). This value was determined by calculating the amplitude of the acceleration during testing. The amplitude was measured to be 230.23 ft/sec^2, and the frequency of the left arm was 3.776 Hz. Using these values creates a sine wave approximation of the acceleration of 230.23*sin(3.776*2*pi*t). This sine wave was then integrated to obtain the input velocity, and integrated again to obtain position. In addition, the mass and rotational moment of inertia are characteristic of the sulky and cannot be modified. The only values that can be modified are the spring constant, the damping coefficient, and the amplitude of the applied force.

In order to determine the optimum values for the spring constant and damping coefficient a mesh plot was created using the Simulink diagram in figure 8. A simple MATLAB script was written which would vary the spring constant from 10 to 2000 lb/ft in steps of 5 lb/ft. The value of 2000 lb/ft as a maximum spring constant was not the first one used. Initially, the maximum value used was the value of the modulus of elasticity for aluminum, divided by the area of the sulky arm. This approximated the arm as a solid spring. With a solid spring approximation, the mesh plot that was created did not show any patterns. The maximum spring constant was then gradually reduced to 2000 lb/ft until a plot was created which showed patterns in the rotational amplitude. For each 5 lb/ft step, the damping coefficient for critical damping was calculated, and the damping coefficient from zero to the value for critical damping was divided into 3000 steps. Figure 9 is the mesh plot that was created from this analysis.

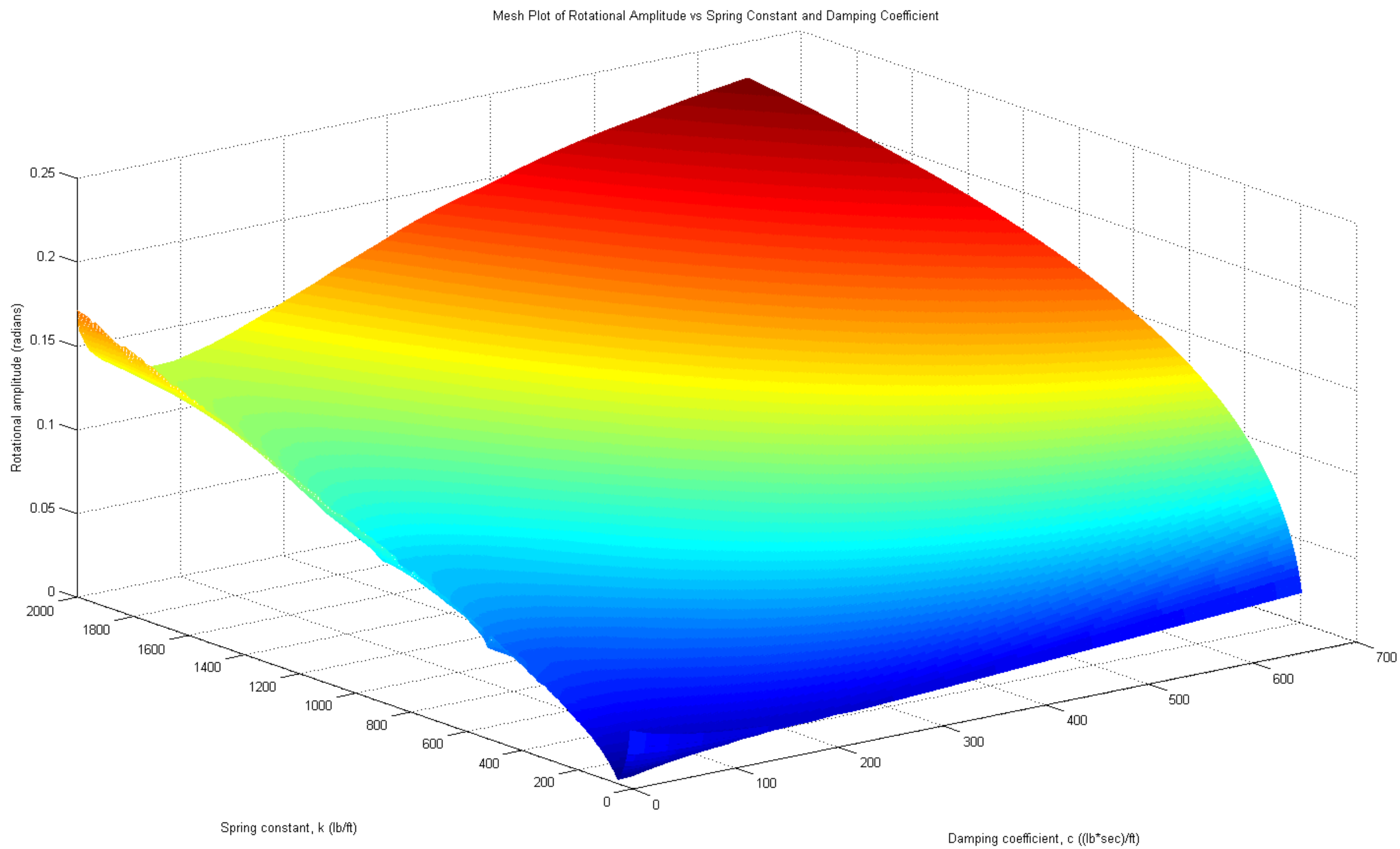

Figure 9. Mesh plot to determine optimum spring constant and damping coefficient.

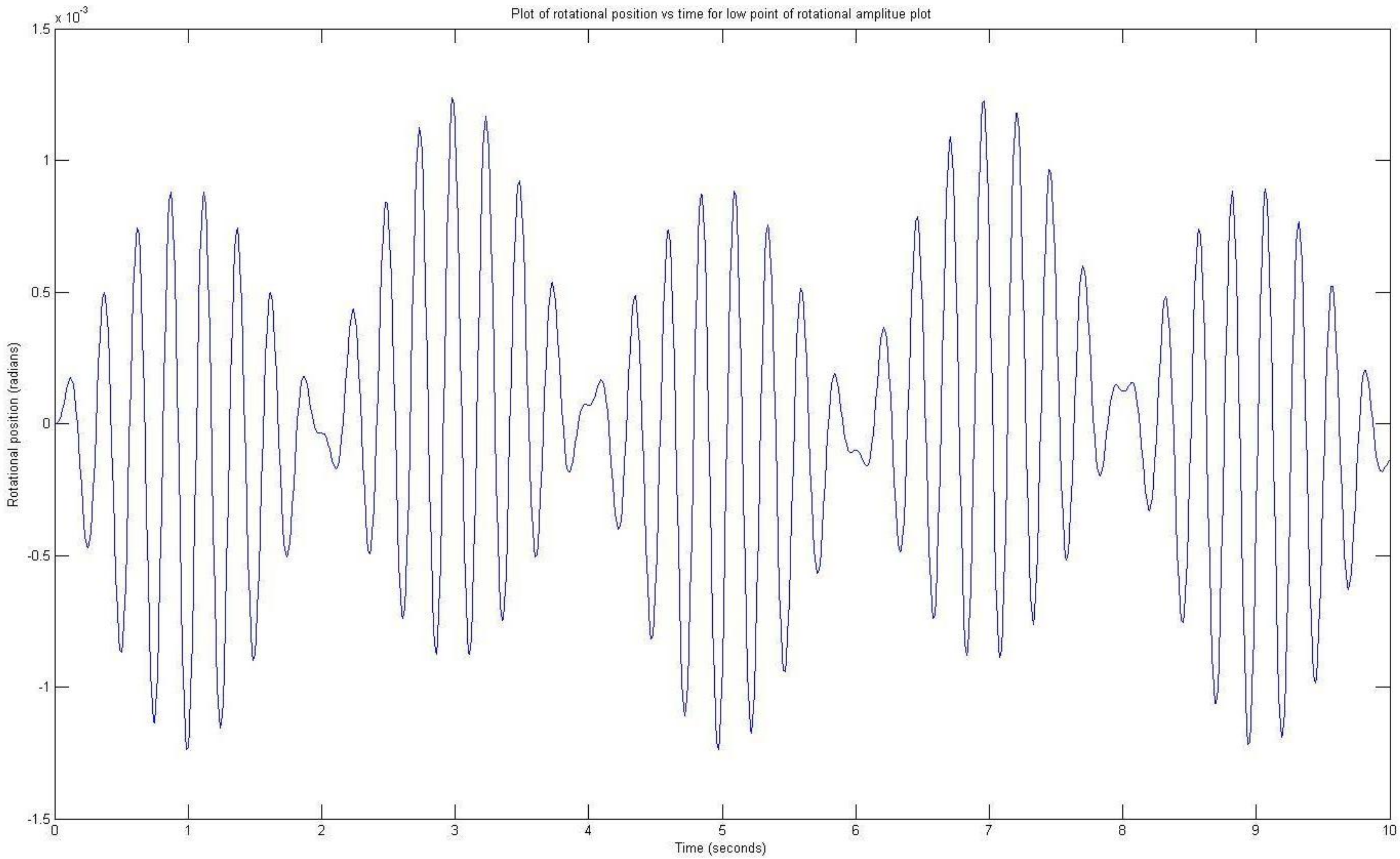

Figure 10. Plot of rotational position vs. time, using low spring constant and damping coefficient

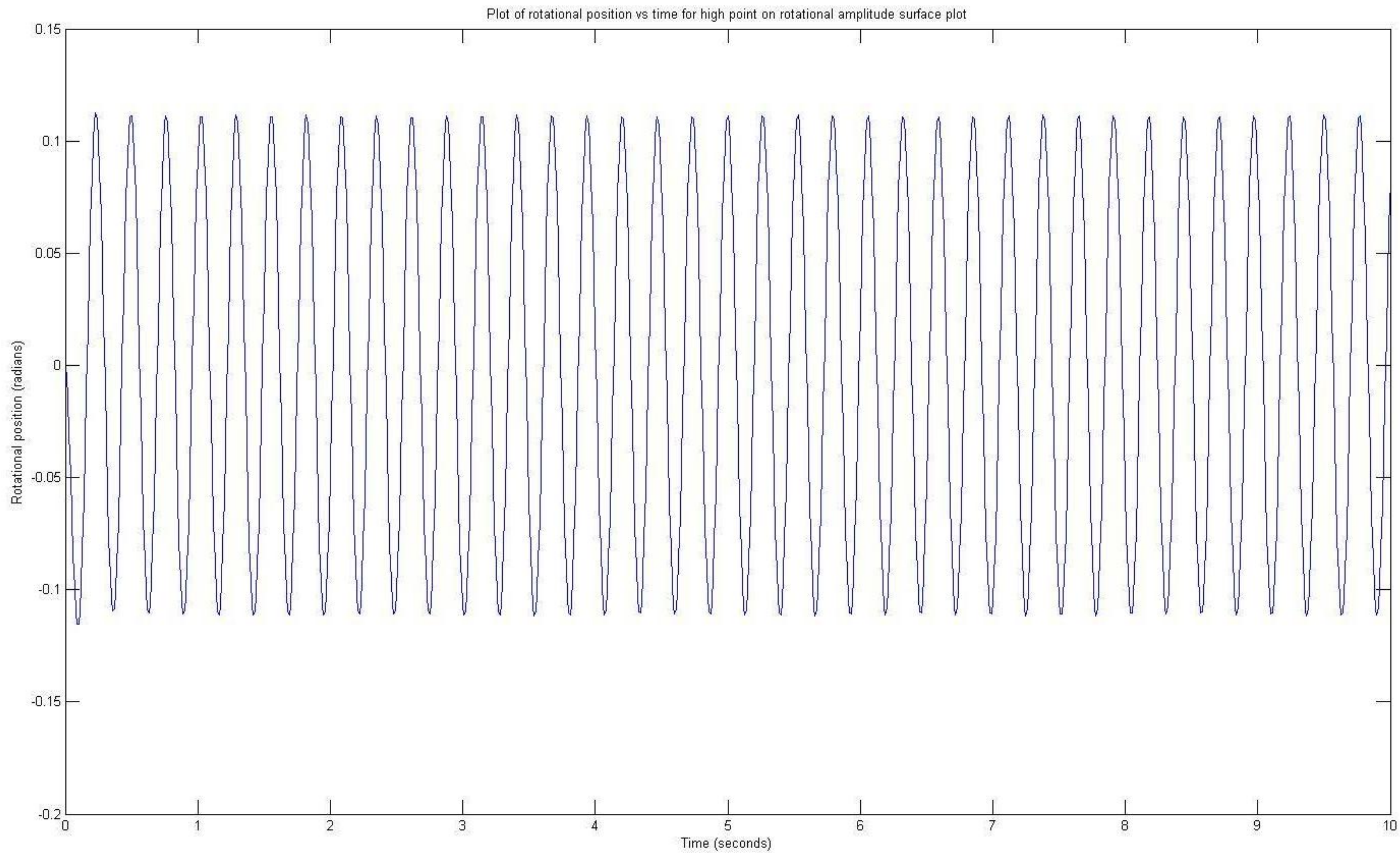

Figure 11. Plot of rotational position vs. time, using high spring constant and damping coefficient

The mesh plot in figure 9 shows that the lowest amplitude of rotational position is at a damping coefficient of zero, and a spring constant of 55 lb/ft. In addition, the maximum amplitude occurs at a spring constant of 2000 lb/ft and a damping coefficient of 648 $^{lb * sec}/_{ft}$. Figures 10 and 11 above are plots of the rotational position using the spring constant from the lowest and highest points, respectively.

The plot of rotational position created with a spring constant of 55 lb/ft, and a damping coefficient of zero has maximum amplitude of 0.0025 radians. At the other end of the possible range, the plot created with a spring constant of 2000 lb/ft and a damping coefficient of 648 $^{lb * sec}/_{ft}$ has maximum amplitude of 0.23 radians, two orders of magnitude higher than with the minimum spring constant and damping coefficient. Therefore, the initial values for use in tuning the Smart Sulky prototype are 55 lb/ft spring constant, and no damping.

Conclusion

The equations of motion for the Smart Sulky when it is represented as a rotating disk constrained to translate in one direction, are:

$$\ddot{\theta} = \frac{d\theta}{dt} * \left(\frac{-c * R^2}{I_\omega}\right) + \theta * \left(\frac{-k * R^2}{I_\omega}\right) + \dot{x}_1 * \left(\frac{c * R}{I_\omega}\right) + x_1 * \left(\frac{k * R}{I_\omega}\right) + \frac{F(t) * R}{I_\omega} \qquad \text{(Eq. 17)}$$

$$\ddot{x}_2 = \dot{\theta} * \frac{c * R}{m} + \theta * \frac{k * R}{m} - \dot{x}_1 * \frac{c}{m} - x_1 * \frac{k}{m} + \frac{F(t)}{m} \qquad \text{(Eq. 18)}$$

It was experimentally determined that the ideal representation of the input displacement to the system is a sine wave with an amplitude equal to that of a test sulky, 0.409 ft, and a frequency equal to the natural frequency of accelerations of a test sulky, 23.72 radians/sec. With this information, the initial values for spring constant and damping coefficient for use in tuning the Smart Sulky are 55 lb/ft and 0 $^{lb*sec}/_{ft}$, respectively. In the future, it would be desirable to again try to develop the equations of motion of the complete sulky, likely using Lagrangian mechanics. With a better understanding of how to derive equations with Lagrangian mechanics, and a more methodical approach, it would likely be possible to derive the equations of motion of the complete sulky. For a more methodical approach, the equations of motion would be derived using both Newtonian and Lagrangian mechanics of the simple Smart Sulky approximation in figure 3. Then the equations of motion of a series of increasingly more complex approximations would be derived using both methods. Once the equations of motion could successfully be derived for a sufficiently complicated approximation using both methods, the next step would be to derive the equations of motion for the entire sulky using Lagrangian mechanics alone. If the complete equations of motion for the Smart Sulky could be derived, these would allow a more complex, and accurate model of the exact movements of the Smart Sulky over time.

References

Anonymous. "cumtrapz." *MathWorks*. The MathWorks Inc. 2012, 11 March 2012. http://www.mathworks.com/help/techdoc/ref/cumtrapz.html

Leleu, C, et al. "Relationships between biomechanical variables and race performance in French Standardbred trotters." *Livestock Production Science* 92, (2005): 39-46.

Pfau, Thilo, et al. "Modern Riding Style Improves Horse Racing Times." *Science* 325, no. 5938 (2009): 289. Accessed September 10, 2011. http://www.sciencemag.org. prxy4.ursus.maine.edu/content/325/5938/289.full

Williams, James H. Jr. *Fundamentals of Applied Dynamics.* New York: John Wiley & Sons, Inc., 1996.

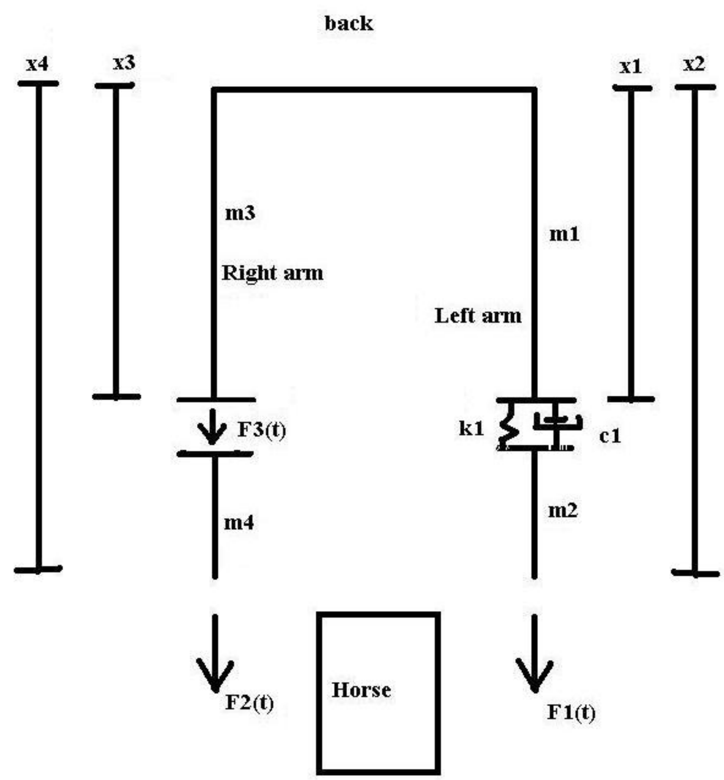

Figure 12. Representation of sulky, x-direction.

Appendix A Original Equations of Motion

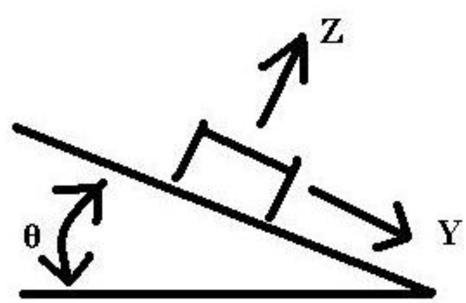

Figure 13. Sulky on tilted track surface.

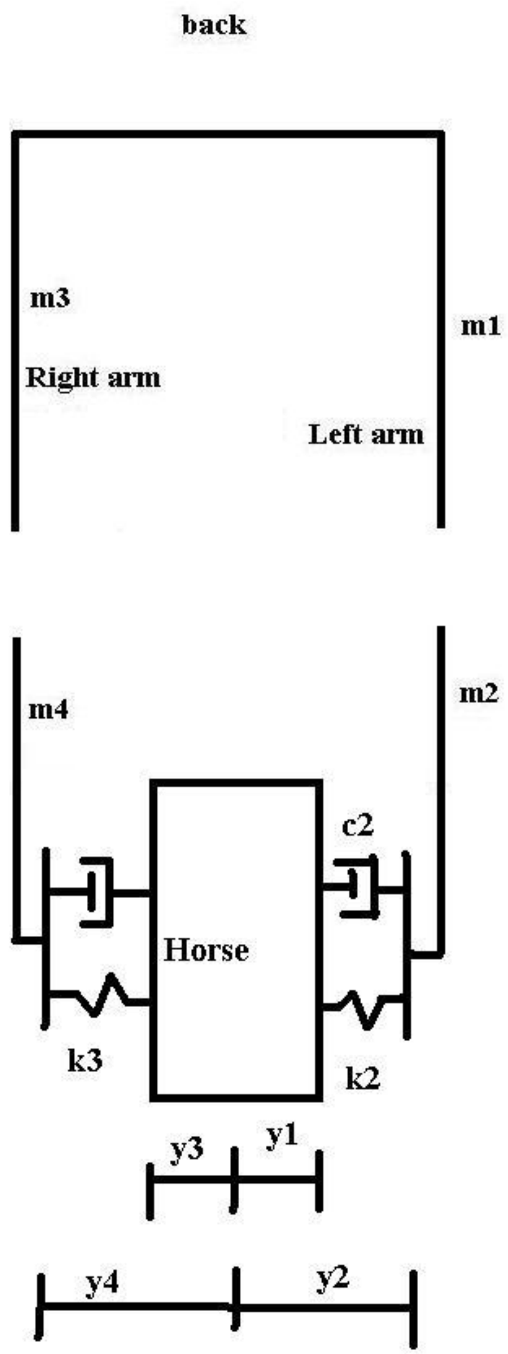

Figure 14. Y-Direction representation of sulky.

Equation of motion left side only, single force pulling sulky with shock absorber in arm

Flat surface

Generalized Forces

$$-c\cdot\left|x'_2 - x'_1\right|\cdot\left|\delta x_2 - \delta x_1\right| + F(t)\cdot\delta x_2 = \Xi_{x1}\cdot\delta x_1 + \Xi_{x2}\cdot\delta x_2$$

$$\Xi_{x1} = -c\cdot\left|x'_1 - x'_2\right|$$

$$\Xi_{x2} = F(t) - c\cdot\left|x'_2 - x'_1\right|$$

Lagrangian

$$L = T^* - V$$

$$T^* = \frac{1}{2}\cdot m_1\cdot {x'_1}^2 + \frac{1}{2}\cdot m_2\cdot {x'_2}^2 \qquad V = \frac{1}{2}\cdot k\cdot\left|x_2 - x_1\right|^2$$

$$L = \frac{1}{2}\cdot m_1\cdot {x'_1}^2 + \frac{1}{2}\cdot m_2\cdot {x'_2}^2 - \frac{1}{2}\cdot k\cdot\left|x_2 - x_1\right|^2$$

Lagrange's Equation

$$\frac{d}{dt}\left(\frac{d}{dx'_1}L\right) - \frac{d}{dx_1}L = \Xi_{x1} \qquad \frac{d}{dt}\left(\frac{d}{dx'_2}L\right) - \frac{d}{dx_2}L = \Xi_{x2}$$

$$m_1\cdot x''_1 - k\cdot x_1 + k\cdot x_2 = -c\cdot\left|x'_1 - x'_2\right| \qquad m_2\cdot x''_2 + k\cdot x_1 - k\cdot x_2 = F(t) - c\cdot\left|x'_1 - x'_2\right|$$

Equation of motion

$$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix}\cdot\begin{pmatrix} x''_1 \\ x''_2 \end{pmatrix} + \begin{pmatrix} c & -c \\ -c & c \end{pmatrix}\cdot\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} + \begin{pmatrix} -k & k \\ k & -k \end{pmatrix}\cdot\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ F(t) \end{pmatrix}$$

Equation of motion left side only, single force pulling sulky with shock absorber in arm

Tilted surface

X-Direction Calculations: same as above

$$T^*_x = \frac{1}{2}\cdot m_1\cdot x'_1{}^2 + \frac{1}{2}\cdot m_2\cdot x'_2{}^2$$

$$V_x = \frac{1}{2}\cdot k\cdot\left(x_2 - x_1\right)^2$$

$$-c\cdot\left(x'_2 - x'_1\right)\cdot\left(\delta x_2 - \delta x_1\right) + F(t)\cdot\delta x_2 = \Xi_{x1}\cdot\delta x_1 + \Xi_{x2}\cdot\delta x_2$$

$$\Xi_{x1} = -c\cdot\left(x'_1 - x'_2\right)$$

$$\Xi_{x2} = F(t) - c\cdot\left(x'_2 - x'_1\right)$$

$$L = T^* - V$$

$$L = \frac{1}{2}\cdot m_1\cdot x'_1{}^2 + \frac{1}{2}\cdot m_2\cdot x'_2{}^2 - \frac{1}{2}\cdot k\cdot\left(x_2 - x_1\right)^2$$

$$\frac{d}{dt}\left(\frac{d}{dx'_1}L\right) - \frac{d}{dx_1}L = \Xi_{x1} \qquad \frac{d}{dt}\left(\frac{d}{dx'_2}L\right) - \frac{d}{dx_2}L = \Xi_{x2}$$

$$m_1\cdot x''_1 - k\cdot x_1 + k\cdot x_2 = -c\cdot\left(x'_1 - x'_2\right) \qquad m_2\cdot x''_2 + k\cdot x_1 - k\cdot x_2 = F(t) - c\cdot\left(x'_1 - x'_2\right)$$

$$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix}\cdot\begin{pmatrix} x''_1 \\ x''_2 \end{pmatrix} + \begin{pmatrix} c & -c \\ -c & c \end{pmatrix}\cdot\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} + \begin{pmatrix} -k & k \\ k & -k \end{pmatrix}\cdot\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ F(t) \end{pmatrix}$$

Y-Direction calculations:

$$T^*_y = \frac{1}{2}\cdot m_1\cdot y'^2_1 + \frac{1}{2}\cdot m_2\cdot y'^2_2$$

$$V_y = m_1\cdot g\cdot y_1\cdot\cos(\theta(t)) + m_2\cdot g\cdot y_2\cdot\cos(\theta(t))$$

$$0\cdot\left|\delta y_2 - \delta y_1\right| = \Xi_{y1}\cdot\delta y_1 + \Xi_{y2}\cdot\delta y_2$$

$$\Xi_{y1} = 0$$

$$\Xi_{y2} = 0$$

$$L = T^* - V$$

$$L = \frac{1}{2}\cdot m_1\cdot y'^2_1 + \frac{1}{2}\cdot m_2\cdot y'^2_2 - m_1\cdot g\cdot y_1\cdot\cos(\theta(t)) - m_2\cdot g\cdot y_2\cdot\cos(\theta(t))$$

$$\frac{d}{dt}\left(\frac{d}{dy'_1}L\right) - \frac{d}{dy_1}L = \Xi_{y1} \qquad \frac{d}{dt}\left(\frac{d}{dy'_2}L\right) - \frac{d}{dy_2}L = \Xi_{y2}$$

$$m_1\cdot y''_1 - m_1\cdot g\cdot\cos(\theta(t)) = 0 \qquad m_2\cdot y''_2 - m_2\cdot g\cdot\cos(\theta(t)) = 0$$

$$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix}\cdot\begin{pmatrix} y''_1 \\ y''_2 \end{pmatrix} = \begin{pmatrix} m_1\cdot g\cdot\cos(\theta(t)) \\ m_2\cdot g\cdot\cos(\theta(t)) \end{pmatrix}$$

$$\begin{pmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & m_2 \end{pmatrix}\cdot\begin{pmatrix} x''_1 \\ x''_2 \\ y''_1 \\ y''_2 \end{pmatrix} + \begin{pmatrix} c & -c & 0 & 0 \\ -c & c & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\cdot\begin{pmatrix} x'_1 \\ x'_2 \\ y'_1 \\ y'_2 \end{pmatrix} \cdots = \begin{pmatrix} 0 \\ F(t) \\ m_1\cdot g\cdot\cos(\theta(t)) \\ m_2\cdot g\cdot\cos(\theta(t)) \end{pmatrix}$$

$$+ \begin{pmatrix} -k & k & 0 & 0 \\ k & -k & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\cdot\begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix}$$

Equation of motion right side only, single force pulling sulky with pneumatic cylinder in arm

Flat surface

$$T^* = \frac{1}{2} \cdot m_3 \cdot x'_3{}^2 + \frac{1}{2} \cdot m_4 \cdot x'_4{}^2$$

$$V = 0$$

$$F_2(t) \cdot \delta x_4 + F_3(t) \cdot \delta x_3 = \Xi_{x3} \cdot \delta x_3 + \Xi_{x4} \cdot \delta x_4$$

$$\Xi_{x3} = F_3(t)$$

$$\Xi_{x4} = F_2(t)$$

$$L = T^* - V$$

$$L = \frac{1}{2} \cdot m_3 \cdot x'_3{}^2 + \frac{1}{2} \cdot m_4 \cdot x'_4{}^2$$

$$\frac{d}{dt}\left(\frac{d}{dx'_3} L\right) - \frac{d}{dx_3} L = \Xi_{x3}$$

$$\frac{d}{dt}\left(\frac{d}{dx'_4} L\right) - \frac{d}{dx_4} L = \Xi_{x4}$$

$$m_3 \cdot x''_3 = F_3(t)$$

$$m_4 \cdot x''_4 = F_2(t)$$

$$\begin{pmatrix} m_3 & 0 \\ 0 & m_4 \end{pmatrix} \cdot \begin{pmatrix} x''_3 \\ x''_4 \end{pmatrix} = \begin{pmatrix} F_3(t) \\ F_2(t) \end{pmatrix}$$

Equation of motion for sulky without wheels, with shock absorber in left arm, pneumatic cylinder on right arm, separate forces on each arm, on a tilted surface

$$\begin{pmatrix} m_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_4 \end{pmatrix} \cdot \begin{pmatrix} x''_1 \\ x''_2 \\ x''_3 \\ x''_4 \\ y''_1 \\ y''_2 \\ y''_3 \\ y''_4 \end{pmatrix} + \begin{pmatrix} c & -c & 0 & 0 & 0 & 0 & 0 & 0 \\ -c & c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{pmatrix} \ldots$$

$$+ \begin{pmatrix} -k & k & 0 & 0 & 0 & 0 & 0 & 0 \\ k & -k & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 0 \\ F_1(t) \\ F_3(t) \\ F_2(t) \\ m_1 \cdot g \cdot \cos(\theta(t)) \\ m_2 \cdot g \cdot \cos(\theta(t)) \\ m_3 \cdot g \cdot \cos(\theta(t)) \\ m_4 \cdot g \cdot \cos(\theta(t)) \end{pmatrix}$$

Approximation of attachment to horse as spring damper

$$T^* = \frac{1}{2}\cdot m_2\cdot y'_1{}^2 + \frac{1}{2}\cdot m_4\cdot y'_2{}^2$$

$$V = \frac{1}{2}\cdot k_2\cdot\left(y_1 - y_3\right)^2 + \frac{1}{2}\cdot k_3\cdot\left(y_4 - y_2\right)^2$$

$$-c_2\cdot\left(y'_1 - y'_3\right)\cdot\left(\delta y_1 - \delta y_3\right) - c_3\cdot\left(y'_4 - y'_2\right)\left(\delta y_4 - \delta y_2\right) = \Xi_{y1}\cdot\delta y_1 + \Xi_{y2}\cdot\delta y_2 \ldots + \Xi_{y3}\cdot\delta y_3 + \Xi_{y4}\cdot\delta y_4$$

$$\Xi_{y1} = -c_2\cdot\left(y'_1 - y'_3\right) \qquad \Xi_{y3} = -c_2\cdot\left(y'_3 - y'_1\right)$$

$$\Xi_{y2} = -c_3\cdot\left(y'_2 - y'_4\right) \qquad \Xi_{y4} = -c_3\cdot\left(y'_4 - y'_2\right)$$

$$L = T^* - V$$

$$L = \frac{1}{2}\cdot m_2\cdot y'_1{}^2 + \frac{1}{2}\cdot m_4\cdot y'_2{}^2 - \frac{1}{2}\cdot k_2\cdot\left(y_1 - y_3\right)^2 - \frac{1}{2}\cdot k_3\cdot\left(y_4 - y_2\right)^2$$

$$\frac{d}{dt}\left(\frac{d}{dy'_1}L\right) - \frac{d}{dy_1}L = \Xi_{y1} \qquad \frac{d}{dt}\left(\frac{d}{dy'_2}L\right) - \frac{d}{dy_2}L = \Xi_{y2}$$

$$\frac{d}{dt}\left(\frac{d}{dy'_3}L\right) - \frac{d}{dy_3}L = \Xi_{y3} \qquad \frac{d}{dt}\left(\frac{d}{dy'_4}L\right) - \frac{d}{dy_4}L = \Xi_{y4}$$

$$m_2\cdot y''_1 + k_2\cdot y_1 - k_2\cdot y_3 = -c_2\cdot\left(y'_1 - y'_3\right)$$

$$m_4\cdot y''_2 + k_3\cdot y_2 - k_3\cdot y_4 = -c_3\cdot\left(y'_2 - y'_4\right)$$

$$-k_2 \cdot y_1 + k_2 \cdot y_3 = -c_2 \cdot (y'_3 - y'_1)$$

$$-k_3 \cdot y_2 + k_3 \cdot y_4 = -c_3 \cdot (y'_4 - y'_2)$$

$$\begin{pmatrix} m_2 & 0 & 0 & 0 \\ 0 & m_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} y''_1 \\ y''_2 \\ y''_3 \\ y''_4 \end{pmatrix} + \begin{pmatrix} c_2 & 0 & -c_2 & 0 \\ 0 & c_3 & 0 & -c_3 \\ -c_2 & 0 & c_2 & 0 \\ 0 & -c_3 & 0 & c_3 \end{pmatrix} \cdot \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{pmatrix} \ldots = 0$$

$$+ \begin{pmatrix} k_2 & 0 & -k_2 & 0 \\ 0 & k_3 & 0 & -k_3 \\ -k_2 & 0 & k_2 & 0 \\ 0 & -k_3 & 0 & k_3 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Equation of motion for sulky without wheels, with shock absorber in left arm, pneumatic cylinder on right arm, separate forces on each arm, approximation of attachment to horse as a spring damper, on a tilted surface

$$\begin{pmatrix} m_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_1 + m_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 + m_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_4 \end{pmatrix} \cdot \begin{pmatrix} x''_1 \\ x''_2 \\ x''_3 \\ x''_4 \\ y''_1 \\ y''_2 \\ y''_3 \\ y''_4 \end{pmatrix} \ldots = \begin{pmatrix} 0 \\ F_1(t) \\ F_3(t) \\ F_2(t) \\ m_1 \cdot g \cdot \cos(\theta(t)) \\ m_2 \cdot g \cdot \cos(\theta(t)) \\ m_3 \cdot g \cdot \cos(\theta(t)) \\ m_4 \cdot g \cdot \cos(\theta(t)) \end{pmatrix}$$

$$+ \begin{pmatrix} c_1 & -c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -c.1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_2 & 0 & -c_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_3 & 0 & -c_3 \\ 0 & 0 & 0 & 0 & -c_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -c_3 & 0 & c_3 \end{pmatrix} \cdot \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{pmatrix} \ldots$$

$$+ \begin{pmatrix} -k_1 & k_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_1 & -k.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_2 & 0 & -k_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_3 & 0 & -k_3 \\ 0 & 0 & 0 & 0 & -k_2 & 0 & k_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_3 & 0 & k_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Equations of motion for sulky wheels flat surface

$$T^* = \frac{1}{2} \cdot I_\omega \cdot \left(\frac{x'_i}{R}\right)^2$$

$$V = 0$$

$$L = T^* - V$$

$$L = \frac{1}{2} \cdot I_\omega \cdot \left(\frac{x'_1}{R}\right)^2 + \frac{1}{2} \cdot I_\omega \cdot \left(\frac{x'_3}{R}\right)^2$$

$$0 = \Xi_{x1} \cdot \delta x_1 + \Xi_{x3} \cdot \delta x_3$$

$$\Xi_{x1} = 0$$

$$\Xi_{x3} = 0$$

$$\frac{d}{dt}\left(\frac{d}{dx'_1} L\right) - \frac{d}{dx_1} L = \Xi_{x1} \qquad \frac{d}{dt}\left(\frac{d}{dx'_3} L\right) - \frac{d}{dx_3} L = \Xi_{x3}$$

$$\frac{I_\omega \cdot x''_1}{R^2} = 0 \qquad \frac{I_\omega \cdot x''_3}{R^2} = 0$$

$$\begin{pmatrix} \frac{I_\omega}{R^2} & 0 \\ 0 & \frac{I_\omega}{R^2} \end{pmatrix} \cdot \begin{pmatrix} x''_1 \\ x''_3 \end{pmatrix} = 0$$

Equation of motion for sulky with wheels, with shock absorber in left arm, pneumatic cylinder on right arm, separate forces on each arm, approximation of attachment to horse as a spring damper, on a tilted surface

$$\begin{pmatrix} m_1 + \frac{I_\omega}{R^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 + \frac{I_\omega}{R^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_1 + m_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 + m_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_4 \end{pmatrix} \cdot \begin{pmatrix} x''_1 \\ x''_2 \\ x''_3 \\ x''_4 \\ y''_1 \\ y''_2 \\ y''_3 \\ y''_4 \end{pmatrix} \ldots = \begin{pmatrix} 0 \\ F_1(t) \\ F_3(t) \\ F_2(t) \\ m_1 \cdot g \cdot \cos(\theta(t)) \\ m_2 \cdot g \cdot \cos(\theta(t)) \\ m_3 \cdot g \cdot \cos(\theta(t)) \\ m_4 \cdot g \cdot \cos(\theta(t)) \end{pmatrix}$$

$$+ \begin{pmatrix} c_1 & -c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -c.1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_2 & 0 & -c_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_3 & 0 & -c_3 \\ 0 & 0 & 0 & 0 & -c_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -c_3 & 0 & c_3 \end{pmatrix} \cdot \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{pmatrix} \ldots$$

$$+ \begin{pmatrix} -k_1 & k_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_1 & -k.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_2 & 0 & -k_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_3 & 0 & -k_3 \\ 0 & 0 & 0 & 0 & -k_2 & 0 & k_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_3 & 0 & k_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Appendix B Equation of Motion MATLAB Code

```
function dx = xdirequations(t,x)


% Known Constants

Aleft= 230.23;  %(ft/sec^2)
Fleft= 3.776;

Aright=230.55;  %(ft/sec^2)
Fright=2.323;

AForce3=230.55;

%k1=350; %(lb/in)
k1=4200; %(lb/ft)

c1=1200; %(lb*sec)/ft

m1=17.25;  %lb
m2=3;  %lb
m3=25.25;  %lb
m4=3;  %lb
I=2.966; %ft^2 lb

R=14/12; %ft

% Forces applied by horse and actuator
F1=Aleft*sin(2*pi*Fleft*t)*(m1+m2);
F2=Aright*sin(2*pi*Fright*t)*(m3+m4);

% Assume actuator needs to apply an opposite force of F2, this would be
an
% out of phase sine wave.
F3=AForce3*sin(2*pi*Fright*t+pi)*m3;

% Equation to be solved

dx = zeros(8,1);

dx(1) = x(2);
dx(2) = (k1/(m1+I/R^2))*x(1) -(c1/(m1+I/R^2))*x(2) -
(k1/(m1+I/R^2))*x(3) + (c1/(m1+I/R^2))*x(4);
dx(3) = x(4);
dx(4) = -(k1/m2)*x(1) +(c1/m2)*x(2) +(k1/m2)*x(3) -(c1/m2)*x(4) +
F1/m2;
dx(5) = x(6);
dx(6) = F3/(m3+I/R^2);
dx(7) = x(8);
dx(8) = F2/m4;
```

```
close all

%%%% Equation Plotting Script

Aleft= 230.23;  %(ft/sec^2)
Fleft= 3.776;

Aright=230.55;  %(ft/sec^2)
Fright=2.323;

k1=4200; %(lb/ft)

c1=1200;

m1=17.25;  %lb
m2=3;  %lb
m3=25.25;  %lb
m4=3;  %lb
I=2.966; %ft^2 lb

R=14/12; %ft

AForce3=230.55;

%Time to be solved over
tspan = (0:60);     %(seconds)

%Initial conditions
x0 = [0; 0; 57/12; 0; 0; 0; 7; 57/12];  %Positions (feet)


%Solve
[t,x] = ode45(@xdirequations, tspan ,x0);

F3=AForce3*sin(2*pi*Fright*t+pi)*m3;
acc(:,1)=(k1/(m1+I/R^2))*x(:,1) -(c1/m1)*x(:,2) -(k1/m1)*x(:,3) +
(c1/m1)*x(:,4);
acc(:,2)=Aleft*sin(2*pi*Fleft*t);
acc(:,3)=F3/(m3+I/R^2);
acc(:,4)=Aright*sin(2*pi*Fright*t);

%Plot figures for x direction
k=1;
for i=1:2:7

    figure
    subplot(3,1,1), plot(t,x(:,i))
    title(['Position of Mass ',int2str(k)])
    xlabel('Time')
    ylabel('Position')
```

```
    subplot(3,1,2), plot(t,x(:,i+1))
    title(['Velocity of Mass ',int2str(k)])
    xlabel('Time')
    ylabel('Velocity')

    subplot(3,1,3), plot(t,acc(:,k))
    title(['Acceleration of Mass ',int2str(k)])
    xlabel('Time')
    ylabel('Acceleration')

    k=k+1;

end

%hold on




[t,y] = ode45(@ydirequations, tspan ,x0);

%figure
%plot(t,y(:,1)), title('Position vs. Time')
```

Appendix C Equations of Motion Plots

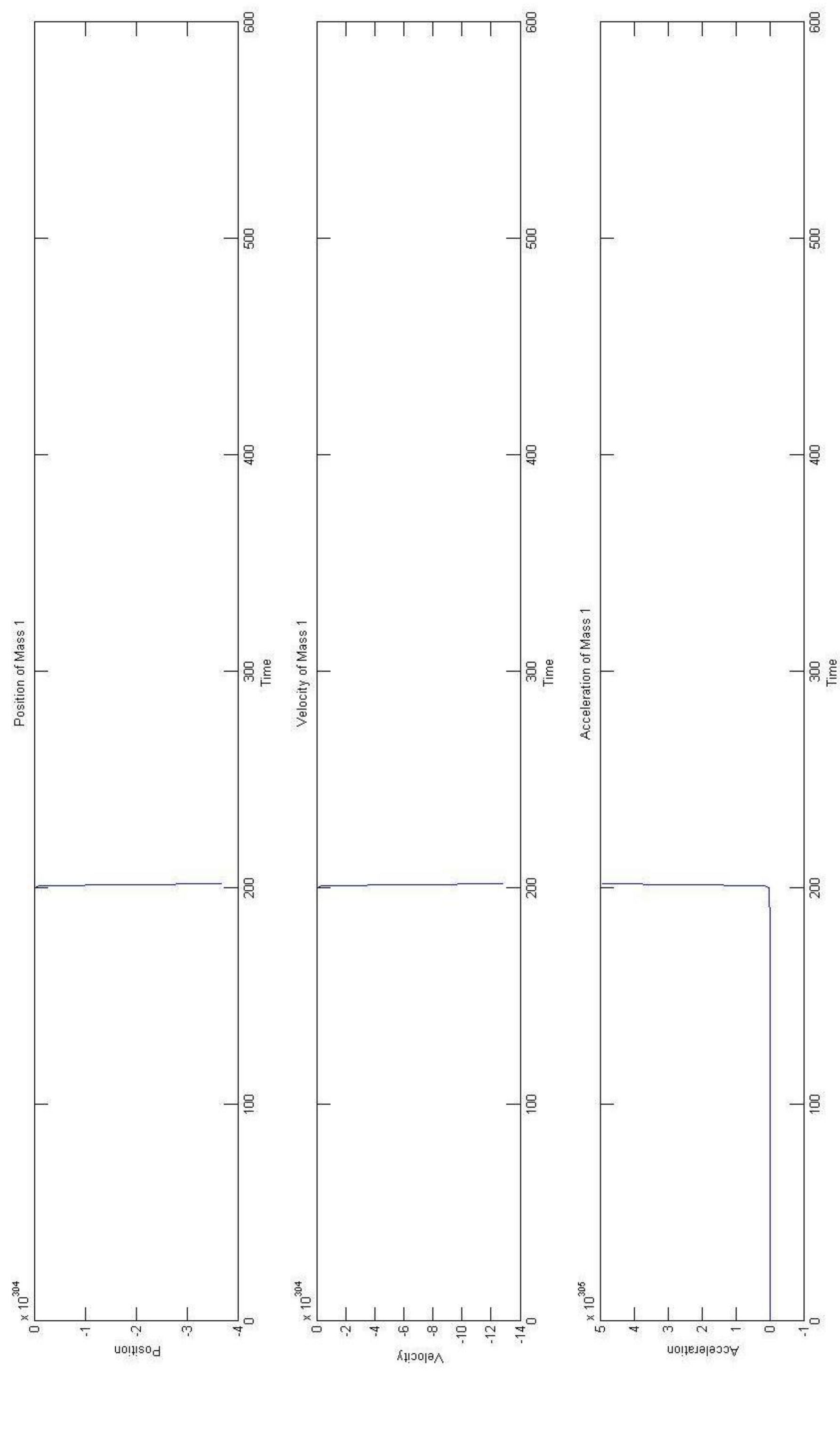

Figure 15. Position, velocity, and acceleration of mass 1.

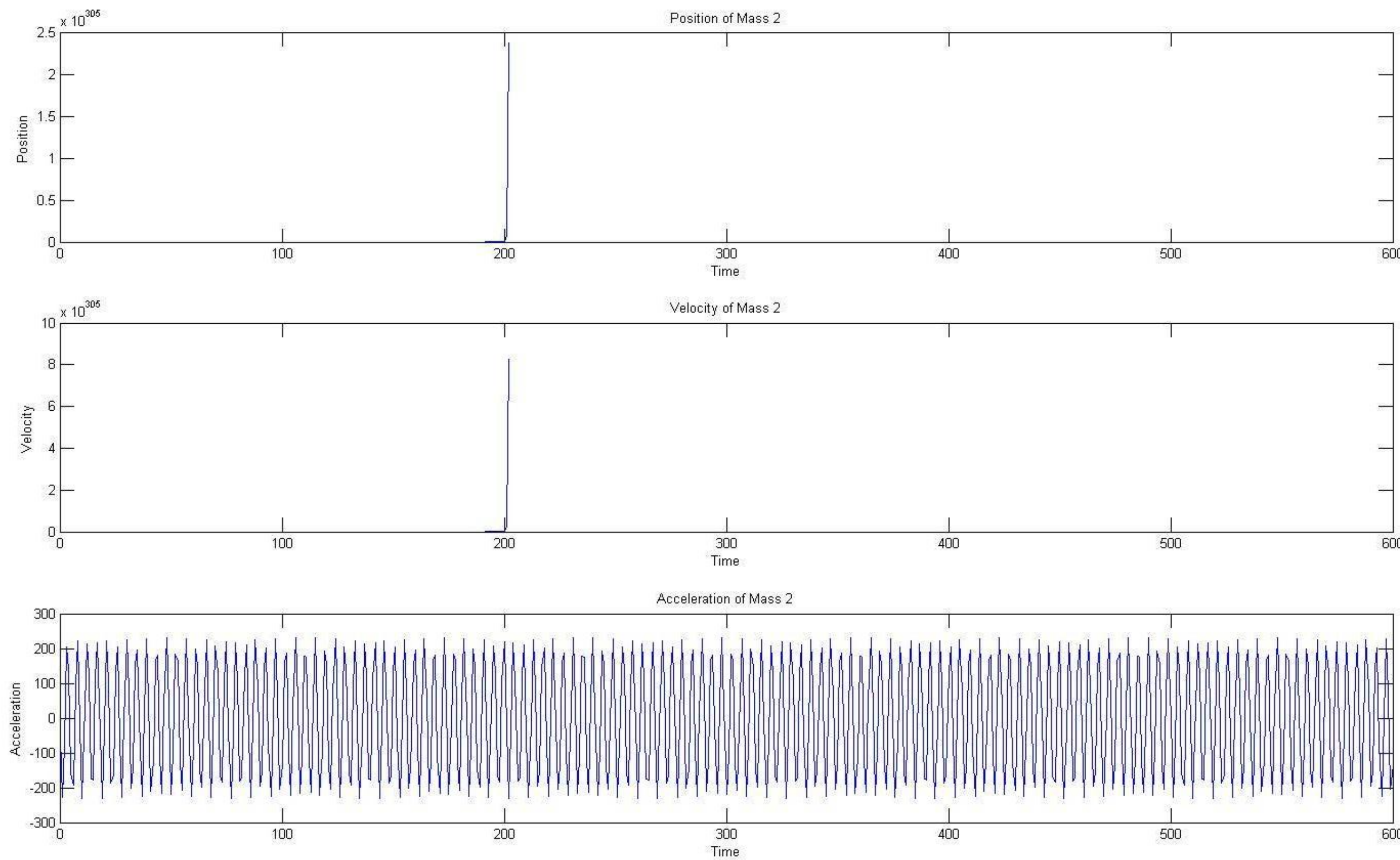

Figure 16. Position, velocity, and acceleration of mass 2.

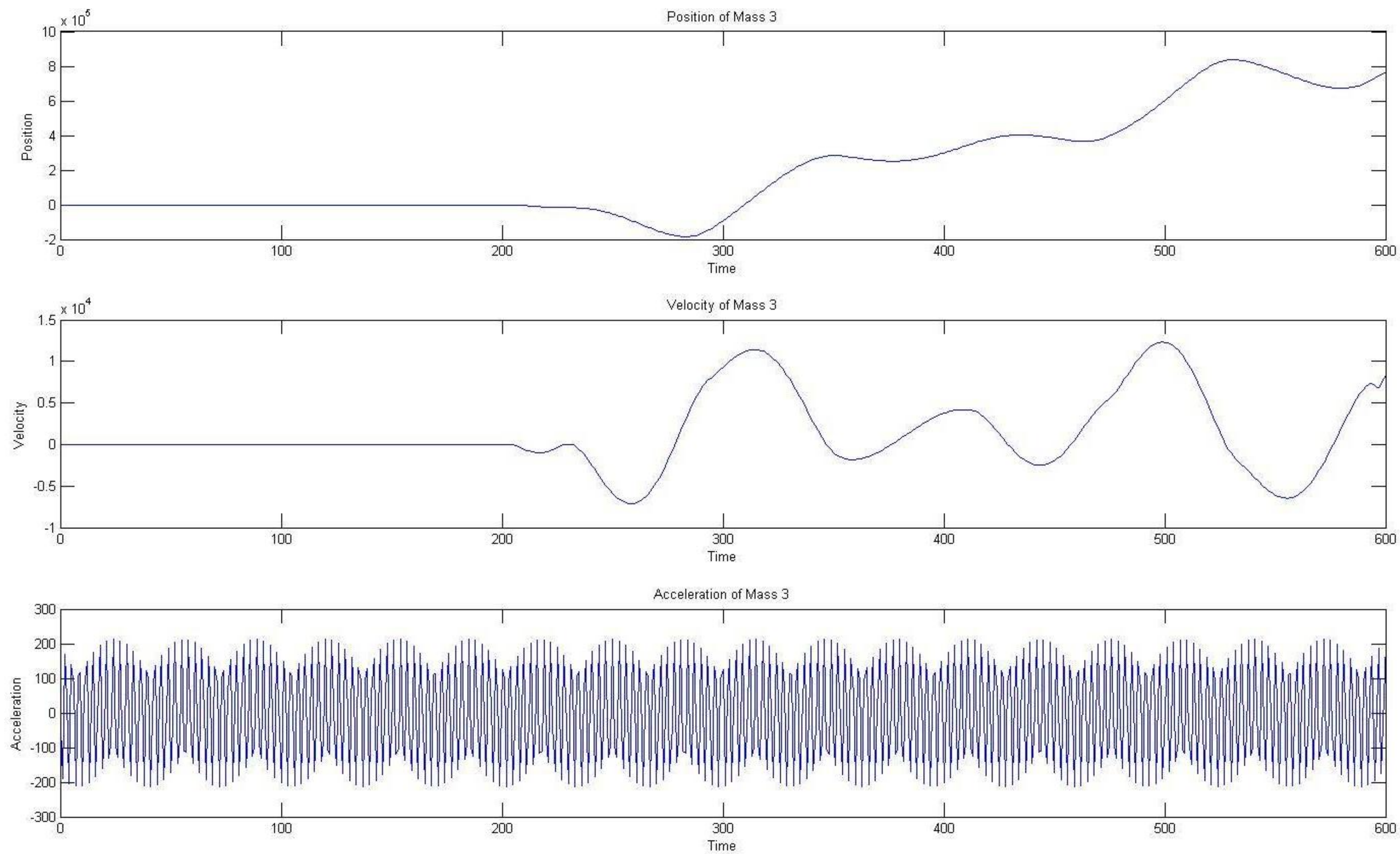

Figure 17. Position, velocity, and acceleration of mass 3.

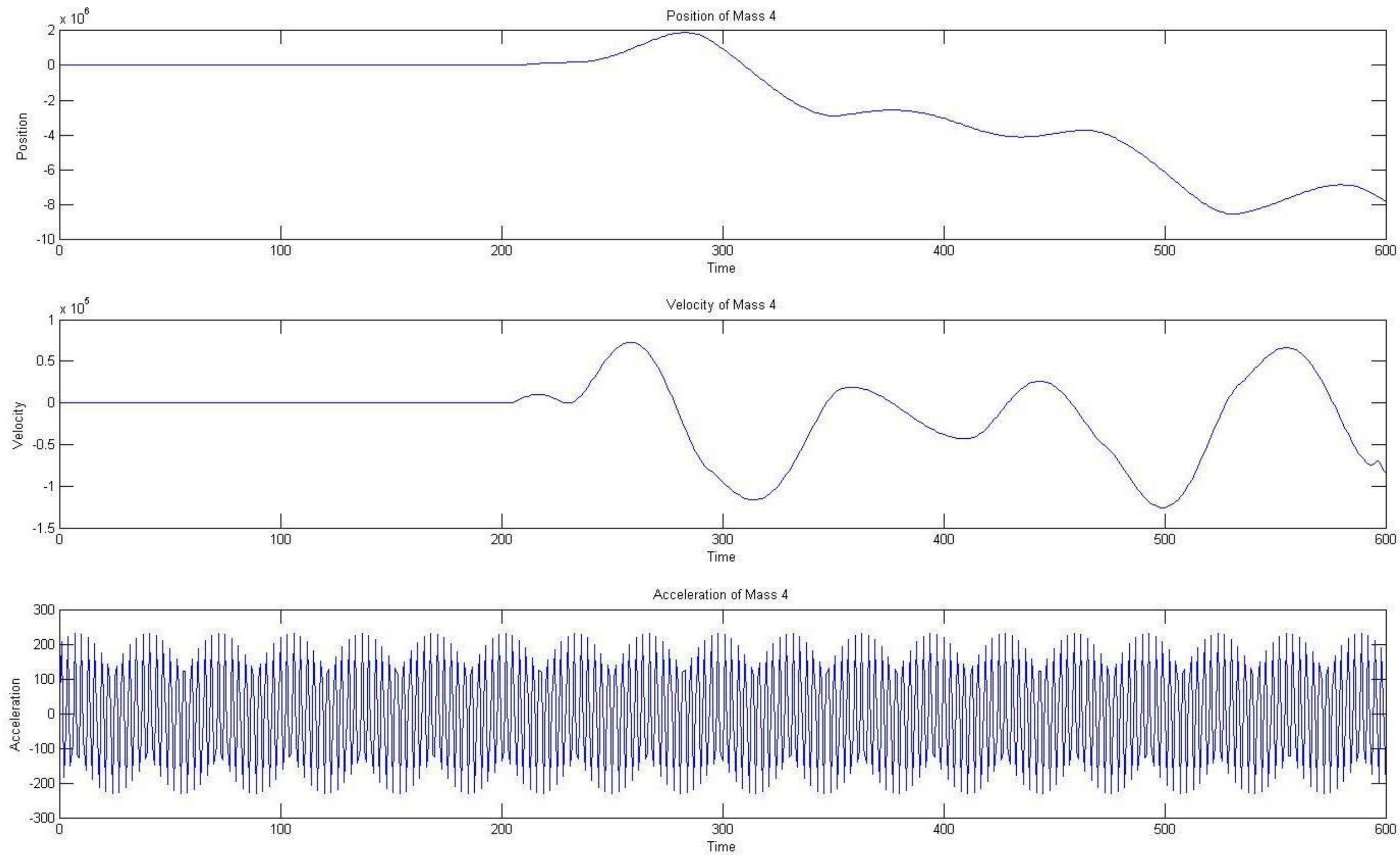

Figure 18. Position, velocity, and acceleration of mass 4.

Appendix D Cumtrapz MATLAB Code

```
%Left Hand side
%close all

%Check to see whether or not data has already been imported from the
excel
%file, if not them import it.

if exist('leftside')==1

else
    leftside=importdata('LeftsideTEST.xlsx');
end

%Define stuff about the data


z=leftside.data.Sheet1(:,4);                     %Positive Up
x=-leftside.data.Sheet1(:,3);                    %Positive Forward
y=leftside.data.Sheet1(:,2);                     %Positive Left


g = 9.81;                                        %m/s^2

z = g*z;                                         %Vertical accelerations
y = g*y;                                         %Acc. in dir of travel
x = g*x;                                         %Acc. perp. to dir of
travel
t = leftside.data.Sheet1(:,1);                   %Time vector
dt=0.01;                                         %Sampling rate approx
%trange=length(leftside.data.Sheet1);


zvel=cumtrapz(t,z);
xvel=cumtrapz(t,x);
yvel=cumtrapz(t,y);

%zvel=
%xvel=
%yvel=

xpos=cumtrapz(t,xvel);
ypos=cumtrapz(t,yvel);
zpos=cumtrapz(t,zvel);

figure
subplot(3,1,1), plot(t,z)
title('Z direction acceleration')
xlabel('Time')
ylabel('Acceleration m/s^2')
```

```
subplot(3,1,2), plot(t,zvel)
title('Z direction velocity')
xlabel('Time')
ylabel('Velocity m/s')

subplot(3,1,3), plot(t,zpos)
title('Z direction position')
xlabel('Time')
ylabel('Position m')

figure
subplot(3,1,1), plot(t,x)
title('X direction acceleration')
xlabel('Time')
ylabel('Acceleration m/s^2')

subplot(3,1,2), plot(t,xvel)
title('X direction velocity')
xlabel('Time')
ylabel('Velocity m/s')

subplot(3,1,3), plot(t,xpos)
title('X direction position')
xlabel('Time')
ylabel('Position m')

figure
subplot(3,1,1), plot(t,y)
title('Y direction acceleration')
xlabel('Time')
ylabel('Acceleration m/s^2')

subplot(3,1,2), plot(t,yvel)
title('Y direction velocity')
xlabel('Time')
ylabel('Velocity m/s')

subplot(3,1,3), plot(t,ypos)
title('Y direction position')
xlabel('Time')
ylabel('Position m')
```

Appendix E Fast Fourier Transform MATLAB Code

```
%Left Hand side
close all

%Check to see whether or not data has already been imported from the
excel
%file, if not them import it.

if exist('backforth')==1

else
    backforth=importdata('backforth.xlsx');
end

%Define stuff about the data
Fs = 100;                                    % Sampling frequency
T = 1/Fs;                                    % Sample time
L = length(backforth.data.Sheet1);           % Length of signal
t = (0:L-1)*T;                               % Time vector

%Take average of data and subtract in order to have zero offset

offset = mean(backforth.data.Sheet1(:,9));

leftdatacorrected= backforth.data.Sheet1(:,9)-offset;


NFFT = 2^nextpow2(L); % Next power of 2 from length of data
% This is used because fourier transform takes place over a length of
2^N
% adding zeros after data has run out. fft will do this on its own but
we
% do this to make it easier for Y and the frequency to have the same
size
% vector.

%Fourier transform without subtracting offset using padding zeros
Y1 = fft(backforth.data.Sheet1(:,9),NFFT)/L;

%Fourier transform without subtracting offset no padded zeros, data
back to
%back
Y2 = fft(backforth.data.Sheet1(:,9),L)/L;

%Fourier transform centered on zero using padding zeros
Y3 = fft(leftdatacorrected,NFFT)/L;

%Fourier transform centered on zero, without using padding zeros
Y4 = fft(leftdatacorrected,L)/L;

f = Fs/2*linspace(0,1,L);
```

```
subplot(2,2,1), plot(f,2*abs(Y1(1:L)))
title('Centered at offset, with padded zeros')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
axis([0 50 -Inf .11])
subplot(2,2,2), plot(f,2*abs(Y2(1:L)))
title('Centered at offset, no padding zeros')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
axis([0 50 -Inf .11])
subplot(2,2,3), plot(f,2*abs(Y3(1:L)))
title('Centered at zero, with padded zeros')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
axis([0 50 -Inf .11])
subplot(2,2,4), plot(f,2*abs(Y4(1:L)))
title('Centered at zero, no padding zeros')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
axis([0 50 -Inf .11])

%Y = fft(leftdata)/L;
%f = Fs/2*linspace(0,1,L);


% Plot single-sided amplitude spectrum.
%plot(f,2*abs(Y(1:L)))%NFFT/2+1)))
%title('Single-Sided Amplitude Spectrum of y(t)')
%xlabel('Frequency (Hz)')
%ylabel('|Y(f)|')
```

Appendix F Spectrogram MATLAB Code

```
%Left Hand side
close all



if exist('backforth')==1

else
    backforth=importdata('backforth.xlsx');

end

%Define stuff about the data
Fs = 100;                                   % Sampling frequency
T = 1/Fs;                                   % Sample time
L = length(backforth.data.Sheet1);          % Length of signal
t = (0:L-1)*T;                              % Time vector


%NFFT = 2^nextpow2(L); % Next power of 2 from length of y
%Y = fft(backforth.data.Sheet1(:,9),NFFT)/L;

%Take average of data and subtract in order to have zero offset

offset = mean(backforth.data.Sheet1(:,9));

leftdata= backforth.data.Sheet1(:,9)-offset;

leftdata=filter(Hd1,leftdata);

%window=blackmanharris(256);
window=hamming(256);
%overlap=window/2;
% calculate the table of amplitudes

[B,f,t]=spectrogram(leftdata,window,128,10000,Fs);


imagesc(t,f,(abs(B)));

%

% label plot

axis xy;

xlabel('Time (s)');
```

```
ylabel('Frequency (Hz)');

%

% build and use a grey scale
%{
lgrays=zeros(100,3);

for i=1:100

    lgrays(i,:) = 1-i/100;

end

colormap(lgrays);
%}
```

Author's Biography

Nicholas Noble was born in, New Hampshire, in 1988. He grew up in Amherst, New Hampshire, and moved to Richmond, Maine, at age 10. He was the valedictorian of the Richmond High School class of 2007. After graduating from UMaine in 2012, with a B.S. in Mechanical Engineering, and minors in Mathematics and Education, Nicholas plans to work on his family's dairy farm for several years, as well as work part-time for Biologically Applied Engineering, LLC, in Orono. His goals include working towards becoming a licensed Professional Engineer while making significant technological advancements on his family's farm.