Inverted Pendulum

by Devin Johnson, Zach Kirch

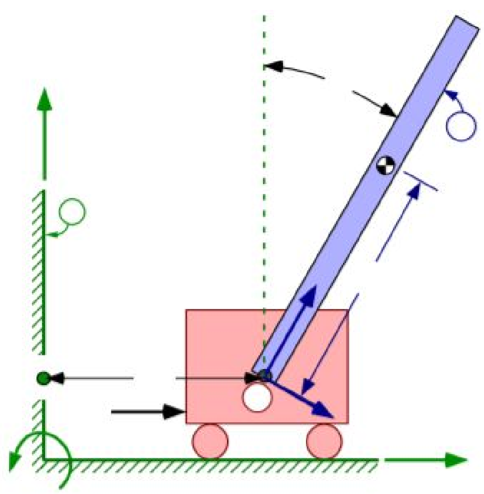
Mentor: Kevin Luna

Introduction:

For this project it is our goal to examine the notion of the inverted pendulum and develop a code that helps us understand the motion and stability of the system. An inverted pendulum is the idea that you can create a pendulum which has points of stability at the normal downward position (declared 180 degrees for this problem) and at the vertical position at 0 degrees). This phenomenon is created via a vibrating base, which is moved either vertically or horizontally to achieve stability at the top of the pendulum. Through the midway point of this project we have thoroughly examined previous iterations of the inverted pendulum, as well as the methods of solving the differential equations that govern the pendulum and its motion, such as using an Runge-Katta solving method and looking at pre-made code examples to help us code our own later if necessary. The eventual hope is to run tests on the pendulum (most likely from a far in the current world conditions) and take this data to understand the specific stability of our pendulum with hope to apply the findings more generally to other pendulums or to create a modified theoretical example of the inverted pendulum problem.

History:

First we want to start with a little history and background of the inverted pendulum, which saw its roots in the development of long range missiles and spacecraft. Beginning with the development of long range and further intercontinental ballistic missiles, it was important that the missile had proper tracking systems, specifically, self orientation properties to maintain proper angle and elevation. However an issue that presented itself was that at lower speeds, aerodynamic stability would not be present so it was necessary to develop a feedback loop system that would allow for control of fins and rutters to help stabilize the rocket at lower speeds, which required the use of an inverted pendulum.



Another example of the inverted pendulum, is the inverted pendulum in a cart example.
This example is in practice very similar to our base example. In this problem a pendulum is placed in a cart with the desire of tracking an object above it, say in the sky while the cart moves back and forth (See Fig 1.1). The cart acts as the vibrating base, while the pendulum is only acted on by the forces created by the cart's motion, which can be controlled by a user or remotely.

One of the first proposed solutions to the inverted pendulum was in the work of Roberge (1960) and was titled "The Mechanical Seal". This is an interesting title because it is an example of the concept of the inverted pendulum in nature. This of course references circus seals and zoo seals being able to be taught to balance objects, mostly balls, on their noses, which they accomplish by rocking their body back and forth which simulates the action of vibrating motion in the base, while their head, the shaft equivalent, moves similarly to the acting motion but with the tip remaining relatively still. As such this natural phenomenon is what we are looking to essentially study and understand.

An interesting problem that has come up in the inception of the inverted pendulum however, is the slight variations in motion and controllability of the vibrating base. This occurs due to the vibrating base having its own moment of inertia, along with having existing factors such as friction in between moving parts, causing the system to have very slight decay in oscillatory motion. This will be something that we look into and discuss further the more we are able to view data from our pendulum to determine how the actual mass of the system affects the results.

Theory:

In this section we will go over the theory of the inverted pendulum and how we look to solve the problem and be able to hopefully compare our results to actual test data of an inverted pendulum, as well as possibly derive an alternative functionality of an inverted pendulum system.

First let us consider the Lagrangian equation for the inverted pendulum. This is setup by looking at the difference of Kinetic and Potential energy in the system defined by Fig 2.1:

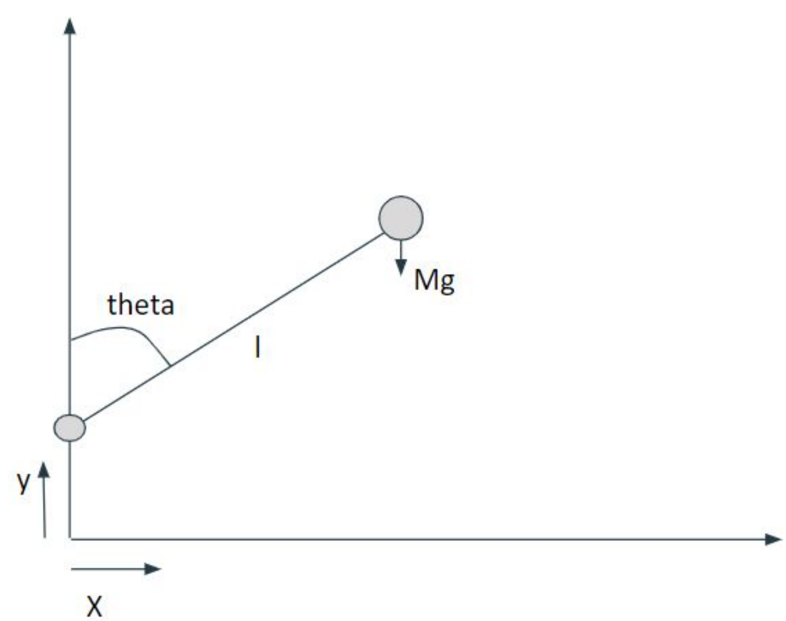


Fig 2.1

In this figure we see the setup for a traditional inverted pendulum, where a mass, represented by a circle furthest from the axis, which we will call point P, and the base which possesses the ability to move vertically which we will call point B, and the origin will be noted as point O. As seen above, the pendulum, if void of vertically moving base, will eventually come to rest on the negative y axis, as the only forces acting on it will be from gravity on the mass at the end of the pendulum, so to determine the effects of the moving base we need to look at the Lagrangian of the system.

The Lagrangian of the system is defined as being the difference of the Kinetic Energy and the Potential Energy, with the goal being that we will be examining where this equation is zero to obtain a point or points of stability within the system. The Lagragian for the system is defined by:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \theta'}\right) - \frac{\partial T}{\partial \theta} + \frac{\partial V}{\partial \theta} = 0$$

Where T is the the Kinetic energy of the system and V is the Potential Energy which are found by finding each in both the x and y directions, resulting in:

$$T = \frac{1}{2}m(x')^2 + \frac{1}{2}m(y')^2 + \frac{1}{2}I_0(\theta')^2$$

$$V = mgy$$

Where I is the moment of inertia of the shaft mass system.

We then need a way to formulate a differential equation, or multiple ones, that can be used to solve the system's function of motion. To do this we need to decompose each of the inertia vectors into easier to handle functions. This is done by stating that the y position of mass is the cosine of theta time the length of the shaft l and the additional distance from the origin due to the location of the base, which we will assume to follow an oscillating function cosine and have amplitude A with frequency ω. The x position of the mass will then simply be defined by the sine of theta times l. This gives us:

$$x = l sin(\theta)$$

$$y = l cos(\theta) + A cos(\omega t)$$

Using these equations we can substitute into the Lagrangian, resulting in:

$$\theta'' + \frac{[Alm\cdot\omega^2 cos(\omega t) - mgl]}{I} \cdot sin(\theta) = 0$$

This will be the equation we use for the ODE solver using an initial problem solving technique known as Runge-Katta, which will be the main focus of the project. As well this result was as shown to be the findings by others in Shew Woody's paper on inverted pendulums.

Effective Potential of System

The effective potential of the system is indeed important to analyze, and this is because it can give a great deal of insight into the stable points of the system. In a simple model without this added oscillation of the base of the pendulum, such as a standard spring system, we can obtain an important equation of motion in the form of $\theta'' = f(\theta) = -\frac{dU(\theta)}{d\theta}$. This equation is important as you can integrate explicitly to find $U(\theta)$, as long as $f(\theta)$ does not depend on t. For a more complex system, such as our oscillating base pendulum, $f(\theta)$ does depend on t, which indicates that the system does not have a known potential $U(\theta)$.

However, as the derivation will show, separating the 'fast' and 'slow' oscillating terms and performing a time average on the equation serves to remove the 'fast' oscillations and leave us with the 'slow' oscillations, which give us the effective motion of the pendulum, the motion we care about. In doing so, we can treat the system as though it does not have this time dependence integrated in the equations, and use the same effective procedure as in the simpler cases.

Given our initial equation,

$$\theta'' + \frac{[Alm\omega^2 cos(\omega t) - mgl]}{I} \cdot sin(\theta) = 0$$

We can find our initial potential energy, $U(\theta)$, by assuming no vibrating base and eliminating the ω term from the above equation, yielding,

$$\theta'' - \frac{(mgl)}{I} \cdot sin(\theta) = 0$$

Which tells us that,

$$-\frac{dU(\theta)}{d\theta} = \theta'' = \frac{(mgl)}{I} \cdot sin(\theta)$$

$$U(\theta) = \frac{(mgl)}{I} \cdot cos(\theta)$$

Giving us our initial potential energy of the system with no vibrating base assumed.

Our equation now has the following form:

$$\theta'' = -\frac{dU(\theta)}{d\theta} + f(\theta, t)$$

$$f(\theta, t) = f_1(\theta) \cdot cos(\omega t)$$

$$f_1(\theta) = -\frac{[Alm \cdot \omega^2]}{I} sin(\theta)$$

Of course, the variable θ is a time dependent variable within the system. However, there are two different governing factors describing $\theta(t)$: a slow dependency based on the oscillation of the pendulum, and a rapid dependency based on the oscillation of the vibrating base. Therefore, we decompose this motion and express this function $\theta(t)$ as:

$$\theta(t) = X(t) + \varepsilon(t)$$

Where $X(t)$ is the slow oscillating term and $\varepsilon(t)$ is the fast oscillating term.

This is an important step during the derivation as it allows us to introduce time averaging of the terms over a period of oscillation (for simplicity, we can choose the time period to be over a period of the 'fast' oscillation). This looks as following for a given function $\overline{g(t)}$, which is the time averaged function of the original function $g(s)$ over the time period $T_f = \frac{2\pi}{\omega}$:

$$\overline{g(t)} = \frac{1}{T_f} \int_t^{t+T_f} g(s)\, ds$$

Using this time average property, one can see that:

$$\overline{\theta(t)} = \overline{X(t)} + \overline{\varepsilon(t)} = \overline{X(t)}$$

The above equation holds true because we are performing the time average over the time period of the “fast” oscillating term. Therefore, when integrating over the time period T_f, the $\varepsilon(t)$ time averaged function will tend to zero. This can be seen in the expressions below:

$$\overline{\theta(t)} = \frac{1}{T_f} \int_t^{t+T_f} \theta(s)\, ds = \theta(t)$$

$$\overline{\varepsilon(t)} = \frac{1}{T_f} \int_t^{t+T_f} \varepsilon(s)\, ds = 0$$

Substituting the “fast”/“slow” decomposition for theta, the following equation is obtained:

$$X''(t) + \varepsilon''(t) = -\frac{dU[X(t)+\varepsilon(t)]}{d\theta} + f(X(t) + \varepsilon(t), t)$$

Now, appropriate Taylor Series expansions are applied to the $\frac{dU[X(t)+\varepsilon(t)]}{d\theta}$ term and the $f(X(t) + \varepsilon(t), t)$ term. Since we assume that $\varepsilon(t)$ is small, the terms from the Taylor Expansion that have an $\varepsilon(t)$ term of the 2nd order or higher can be disregarded, as they are relatively insignificant. The resulting equation becomes:

$$X''(t) + \varepsilon''(t) = -\frac{dU(X)}{dX} - \frac{d^2U(X)}{dX^2}\varepsilon + f(X,t) + \frac{\partial f(X,t)}{\partial X}\varepsilon$$

Upon applying the fact that ε depends on ω as well, or in other words, that we have been omitting the fact that it should be expressed as $\varepsilon(\omega t)$. As well as the fact that $X''(t)$ as well as $-\frac{dU(X)}{dX}$ don’t directly depend on t or ε, so they can be omitted from either side of the equation, we are left with the following equation. Additionally, a substitution for $\tau = t\omega$ is made, as it tends to simplify the later steps of the derivation:

$$\omega^2\varepsilon''(\tau) = -\frac{d^2U(X)}{dX^2}\varepsilon + f(X,\tau) + \frac{\partial f(X,\tau)}{\partial X}\varepsilon$$

Now, we can realize that $\omega^2 >> \varepsilon$, so the terms that have ε in them can be dropped off the overall equation.

$$\omega^2\varepsilon''(\tau) = f(X,\tau)$$

Dividing through by ω^2 gives,

$$\varepsilon''(\tau) = \frac{f(X,\tau)}{\omega^2}$$

And solving explicitly for $\varepsilon(\tau)$:

$$\varepsilon(\tau) = -\frac{f(X,\tau)}{\omega^2} = \varepsilon''(\tau)$$

Now, perform this resulting substitution in our original approximation equation to obtain:

$$X''(t) + \varepsilon(t) = -\frac{dU(X)}{dX} - \frac{d^2U(X)}{dX^2}\varepsilon + f(X,t) + \frac{\partial f(X,t)}{\partial X}\varepsilon$$

We can now take the time average of both sides of the equation to obtain:

$$\overline{X''(t)} + \overline{\varepsilon(t)} = \overline{-\frac{dU(X)}{dX}} - \overline{\frac{d^2U(X)}{dX^2}\varepsilon} + \overline{f(X,t)} + \overline{\frac{\partial f(X,t)}{\partial X}\varepsilon}$$

Now, because we are averaging over a full time period relative to the ε terms, the time average across the interval will go to zero, and they can be dropped:

$$\overline{X''(t)} = \overline{-\frac{dU(X)}{dX}} + \overline{f(X,t)} + \overline{\frac{\partial f(X,t)}{\partial X}\varepsilon}$$

Additionally, because $X(t)$ oscillates at a slow rate, the time average of the $X''(t)$ and $-\frac{dU(X)}{dX}$ terms over the "fast oscillation" period will just equal the original terms. Additionally, since we are averaging over a full period, the $f(X,t)$ goes away to produce the following equation:

$$X''(t) = -\frac{dU(X)}{dX} + \overline{\frac{\partial f(X,t)}{\partial X}\varepsilon}$$

Recalling the equation obtained earlier containing ε, we can write this as:

$$X''(t) = -\frac{dU(X)}{dX} - \overline{\frac{\partial f(X,t)}{\partial X} \cdot \frac{f(X,t)}{\omega^2}}$$

This can be simplified to,

$$X''(t) \ = \ -\frac{dU(X)}{dX} + \frac{1}{\omega^2}\frac{\partial}{\partial X}(\overline{f^2})$$

If the differentiation with respect to X is pulled out on the right hand side of the equation, this becomes,

$$X''(t) \ = \ -\frac{\partial}{\partial X}(U \ + \frac{1}{\omega^2}\overline{f^2} \ )$$

Now, the effective potential just becomes the resulting expression on the right side, or:

$$U_{eff} = \ U \ - \frac{1}{2\omega^2}\overline{f^2}$$

The final step is substituting for the given expressions on the right side. The two equation of interested, derived earlier in the section, are:

$$U(\theta) = \frac{(mgl)}{I} \cdot cos(\theta)$$

$$f(\theta, t) \ = f_1(\theta) \cdot cos(\omega t) \ = \ -\frac{[Alm \cdot \omega^2]}{I} sin(\theta) \cdot cos(\omega t)$$

Combining all of these results in a final effective potential energy equation of:

$$U_{eff} = \frac{(mgl)}{I} \cdot cos(\theta) \ + \frac{1}{\omega^2} \cdot [(-\frac{[Alm \cdot \omega^2]}{2I})sin(\theta)]^2$$

$$U_{eff} = \frac{(mgl)}{I} \cdot cos(\theta) \ + \frac{1}{\omega^2} \cdot [\frac{[A^2 l^2 m^2 \cdot \omega^4]}{4I^2} sin^2(\theta)]$$

$$U_{eff} = \frac{(mgl)}{I} \cdot cos(\theta) \ + \frac{1}{4}\frac{A^2 l^2 m^2 \omega^2}{I^2} sin^2(\theta)$$

Initial Equation Solver:

Initially, our team analyzed the swing path that a pendulum arm would take on a vibrating base, and as explain in the earlier sections, this lead us to the following equation:

$$\theta'' + \frac{[Alm\omega^2 cos(\omega t) - mgl]}{I} \cdot sin(\theta) = 0$$

This equation was utilizing polar coordinates to reduce the complexity of the overall analysis of the system, and the term θ represents the angle the pendulum arm makes with the defined Z-axis, or the line through the arm at rest.

Based on this equation, we needed to write a program that could solve for θ at different time periods, giving us data that would enable us to observe how the pendulum would behave with different parameters m (mass), w (frequency of oscillation), A (amplitude of oscillation), l (length of arm), and I (moment of inertia).

Because this was a second order Ordinary Differential Equation (ODE), the easiest way to approach the coded solver was to define two different variables, u and v, to essentially turn this single 2nd order ODE to two 1st order ODEs. To do this, we set:

$$u = \theta$$
$$v = \theta' = u'$$

After doing this, we defined the function to represent the initial equation. This was done by passing the two dimensional vector of [θ, θ'] that was called y in our function call. The function then set u = θ and v = θ ', and finally returned the vector [θ' , θ''], which both could be expressed in terms of u, v, and the parameters set in the initial 2nd-Order ODE:

$$\begin{bmatrix} u \\ v \end{bmatrix}' = f\left(\begin{bmatrix} u \\ v \end{bmatrix}\right) = \begin{bmatrix} v \\ -\frac{Alm\omega^2 \cos(\omega t) - mgl}{I} * \sin(u) \end{bmatrix}$$

So, the function would return the derivative of the function at this point, which was important in the implementation of the RK4 ODE solver that would initially use the scipy functionality to solve the system of ODEs, but later this solver was personally coded by our team to better understand the underlying details of the solver.

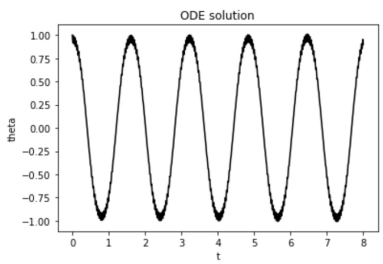
Of course, all of the initial variables in the equation had to be defined, which was done in the following lines of our program:

- m = mass of end of arm = 1 g (Did this to simplify overall equation)
- l = length of arm = 1 meter (Did this to simplify overall equation)
- A = amplitude of oscillation of base = 0.05 meters
- ω = frequency of oscillation = 177 Hz

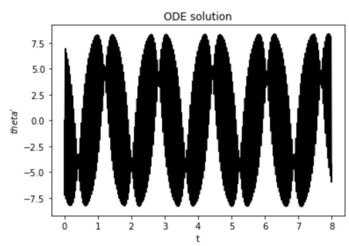
- I = moment of inertia of system = 1 (Did this to simplify overall equation)
- g = gravitational acceleration = 9.1 m/s^2

The amplitude and frequency of oscillation were chosen carefully as such because they allowed for the upright position of the pendulum to be a stable point, depending on the initial configuration of the system set later in the code.

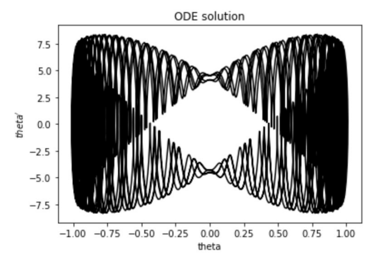
After this, the rest of the program essentially boiled down to calling the 'solve_ivp' function that was provided in the scipy library, and then using those results to obtain a series of plots. These plots looked at θ vs. t, θ ' vs. t, and θ ' vs. θ.



Graphs (1) θ vs. t



Graphs (2) θ' vs. t



Graphs (3) θ' vs. θ

These graphs gave us an initial understanding of what the pendulum arm movement would be, and would be explored further with the later programs that we wrote.

Initial Equation Solver implementing our own RK4 ODE solver:

The next step in the coding process was to write our own RK4 ODE solver, to gain a better understanding of what the ‘solve_ivp’ function was doing under the hood, and validate that our group could write a similar program to obtain the same output data.

The approach we used to solve this ODE was, as stated before, the Fourth-Order Runge Kutta method. To give a brief summary on what this looks like, it is most beneficial to display a small example. Say we have:

$$y'(t) = f(y(t),\ t)$$
$$y(t_0) = y_0$$

Based on these equations, there are four approximations made for the slope of the function at the initial point. These are as follows, with the first entry approximating slope at the beginning of the time step, the second and third approximating the slope halfway through the time step, and the last approximating it at the end of the time step (note: $y^*(t_0)$ is an approximation to $y(t_0)$):

$$k_1 = f(y^*(t_0), t_0)$$
$$k_2 = f(y^*(t_0) + k_1 \cdot \tfrac{h}{2}, t_0 + \tfrac{h}{2})$$
$$k_3 = f(y^*(t_0) + k_2 \cdot \tfrac{h}{2}, t_0 + \tfrac{h}{2})$$
$$k_4 = f(y^*(t_0) + k_3 \cdot h, t_0 + h)$$

Finally, to approximate the value of $y^*(t_0 + h)$:

$$y^*(t_0 + h) = y^*(t_0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} h$$

A function was then written up to implement this RK4 method, as shown below:

```
def Rk4(f,y0,t0,tf, dt):
    t=np.arange(t0,tf,dt) #create array starting at t0 to tf in even steps of dt
    nt=t.size
    ny=y0.size
    y=np.zeros((ny,nt))

    y[:,0] = y0[:,0]



    for k in range(nt - 1):

        k1 = dt*f(t[k], y[:,k])
        k2 = dt*f(t[k] + dt/2, y[:,k] + k1/2)
        k3 = dt*f(t[k] + dt/2, y[:,k] + k2/2)
        k4 = dt*f(t[k] + dt, y[:,k] + k3)

        dy = (k1 + 2*k2 + 2*k3 + k4)/6

        y[:,k+1] = y[:,k] + dy

    return y,t
```

Now, if this function was called in the Initial Equation solver code, instead of the 'solve_ivp' function, it would produce identical graphs, and we gained a better understanding of the RK4 method of solving ODE.

Pendulum Motion Simulator:

Finally, the last step we wanted to take in terms of code implementation was to translate the information that was output in these plots, and put them into more tangible results that better displayed the motion of the pendulum arm with the given parameters.

Unfortunately, given the circumstances that we placed upon us this semester, and the more complex nature of formulating such a program, we did not code this part ourselfs. Rather, we used code given to us by our mentor for this task. However, we analyzed each line of code, and spent time understanding how the program turned the graphical outputs from the Equation solver code into an animated set of pictures showing the trajectory of the pendulum.

So after the Equation solver code is implemented, and the θ vs. t and θ' vs. t data is exported, the Pendulum simulation code then takes this data and goes from Polar Coordinates back into Cartesian Coordinates. This is because the animation software code uses a Cartesian setup, and in order to use it properly, it must be fed data in this format.

The code implemented this in the following way:

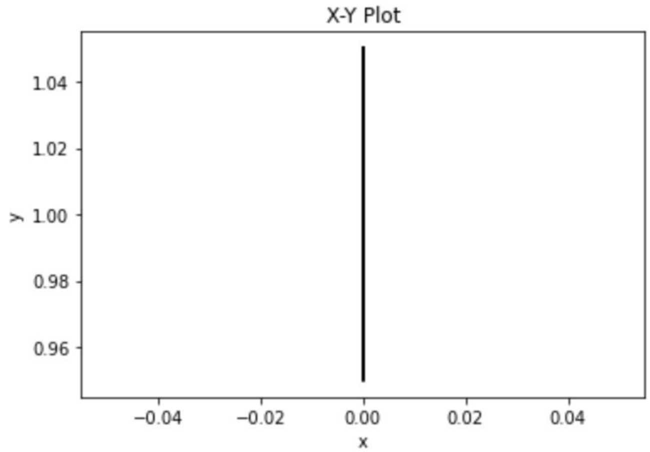
$x = l \cdot sin(\theta)$ ← These theta values were the set of values obtained from the first part of the Equation Solver

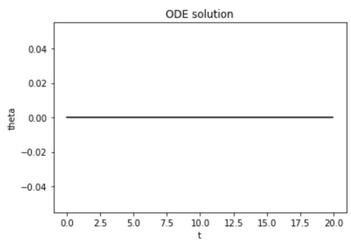
$y = l \cdot cos(\theta) + A \cdot cos(\omega t)$ ← These theta values were the set of values obtained from the first part of the Equation Solver, and the second term is added in the y equation to account for the vibrating base

The remainder of this simulator code was all centered around creating a GIF from this data, and while we read through this and understood this code, it's not entirely necessary to get into the semantics of it.

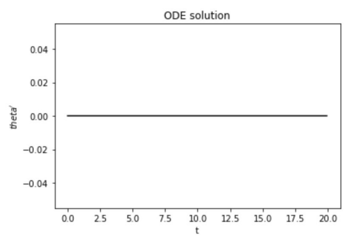
Here are the resulting pendulum arm trajectories given a set of initial y_0 conditions, using the same initial parameters as outlined in the Initial Equation Solver section:

Initial Condition y_0 = [0,0], or pendulum arm starting at a perfectly upright position

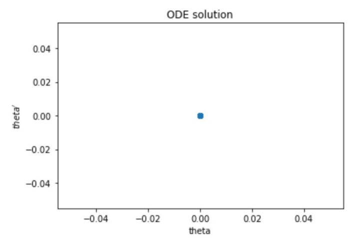




Graph (0.1) θ vs. t

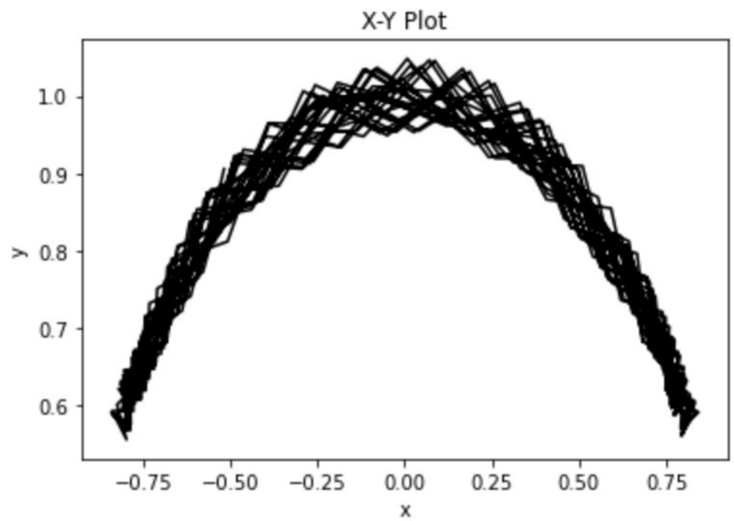


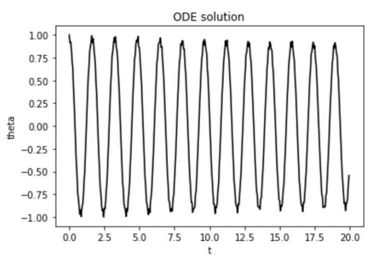
Graph (0.2) θ' vs. t



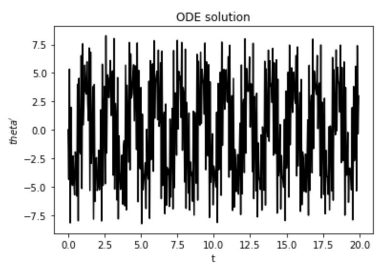
Graph (0.3) θ' vs. θ

Initial Condition y_0 = [1,0], or pendulum arm starting at a +1 radian (around 57.3 degrees) from upright position

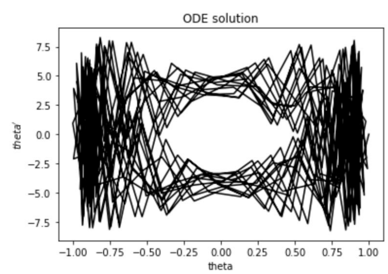




Graph (1.1) θ vs. t

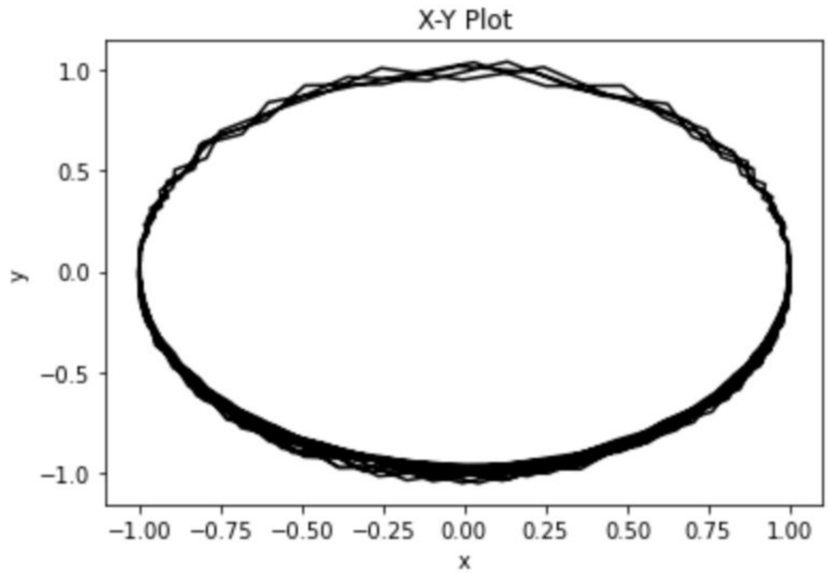


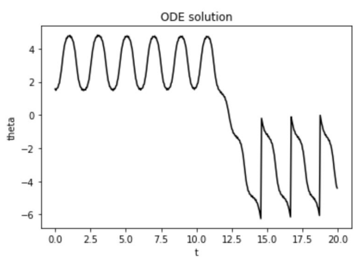
Graph (1.2) θ' vs. t



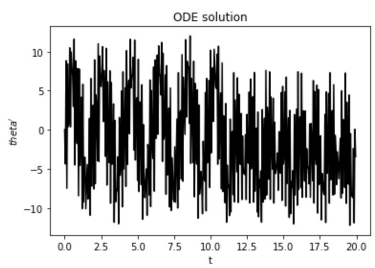
Graph (1.3) θ' vs. θ

Initial Condition y = [π/2, 0], or pendulum arm starting at a +π/2 radian (90 degrees) from upright position.

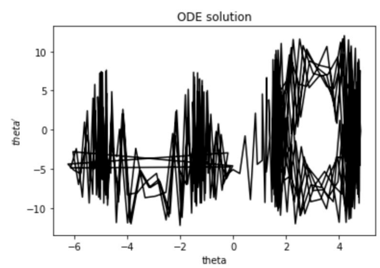




Graph (2.1) θ vs. t

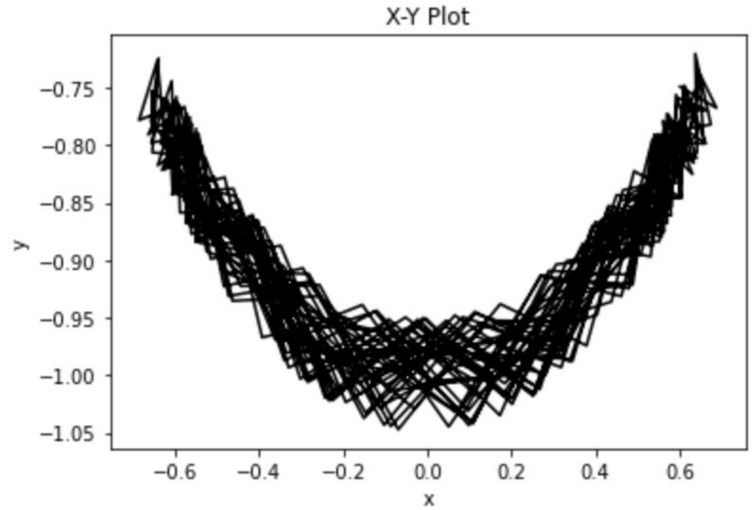


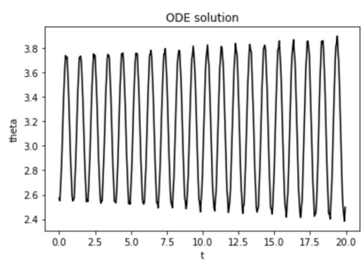
Graph (2.2) θ' vs. t



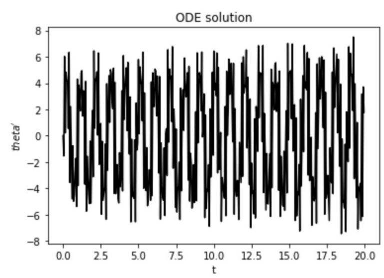
Graph (2.3) θ' vs. θ

Initial Condition y = [π/2 + 1, 0], or pendulum arm starting at a π/2 + 1 radian (around 147.3 degrees) from upright position.

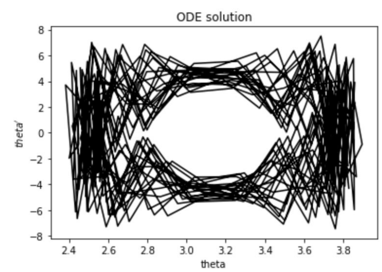




Graph (3.1) θ vs. t



Graph (3.2) θ' vs. t



Graph (3.3) θ' vs. θ

Based on these outputted GIF files, there is quite a bit of insight that can be made about the inverted pendulum system, and the effect that the initial condition has on the system itself. For instance, from the initial condition y_0 = [0,0], we can see that if the arm starts at the perfectly upright position, it will stay in that configuration and simply oscillate along with the vibrating base. Of course, in reality it would be difficult to replicate this condition based on several real-world limitations, but it's important to observe this.

At y_0 = [1 radian,0], we see a perfect example of the inverted state acting as an equilibrium point for the system. Without the vibrating base, the end of the arm would fall to the bottom stable condition, and then be propelled up to the other side to an almost opposite angle, but due to air friction it would be slightly less. This would continue until friction ceased all motion and it settled at a stable state. A similar process would happen with the vibrating base attached, but as seen in the GIF it oscillates around the upright position, and eventually would stabilize in this configuration.

The y = [π/2 radians , 0] is another interesting example that reveals another interesting feature of this system. When watching the animation, for the first 12 seconds, the pendulum behaves similarly to how it would without a vibrating base, oscillating back and forth across the downright stable position. But after this time, the oscillating base gives the arm enough momentum to surge past the initial π/2 position, and swing a full 360 degrees around the pendulum's fulcrum. This reveals that over time, the oscillating base is transferring momentum to the arm, and can alter the far most right and far most left positions that the arm oscillates across.

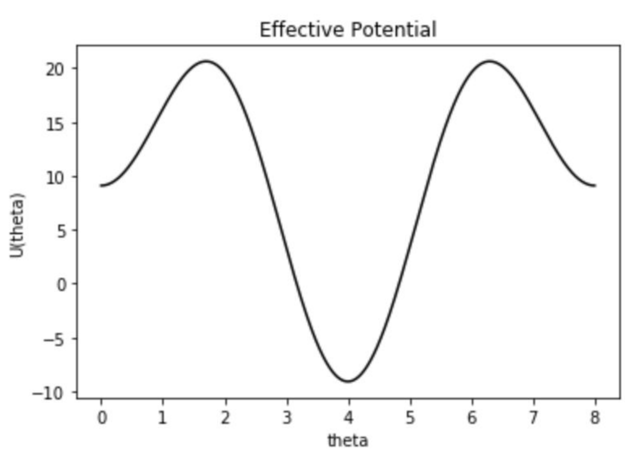
Finally, the y = [π/2 + 1 radians, 0] displays a fairly typical pendulum motion. But, if one observes closely, it can be shown that the far most left and far most right positions of the pendulum do increase over this 20 second time lapse, giving another example to verify what had been observed in the other simulations.

Stability of Inverted Pendulum System

Given the effective potential energy of the pendulum, we can analyze the stability of the pendulum at different initial angles. As derived in the Effective Potential section, we obtained:

$$U_{eff} = \frac{(mgl)}{I} \cdot cos(\theta) + \frac{1}{4}\frac{A^2 l^2 m^2 \omega^2}{I^2} sin^2(\theta)$$

Below is a plot of the effective potential versus the angle θ:



If we find the angles for which $\frac{dU_{eff}}{d\theta} = 0$, this gives us a set of the predicted equilibrium angles θ_{eq}:

$$\frac{dU_{eff}}{d\theta} = 0 = -\frac{(mgl)}{I} \cdot sin(\theta) + \frac{1}{2}\frac{A^2 l^2 m^2 \omega^2}{I^2} sin(\theta)cos(\theta)$$

$$0 = sin(\theta) \cdot [-\frac{(mgl)}{I} + \frac{1}{2}\frac{A^2 l^2 m^2 \omega^2}{I^2} cos(\theta)]$$

From this equation, it is quite clear that there are four total equilibrium points of the system, and that they occur when either $sin(\theta) = 0$ or when $\frac{(mgl)}{I} + \frac{1}{4}\frac{A^2 l^2 m^2 \omega^2}{I^2} cos(\theta) = 0$. Therefore, for the $sin(\theta) = 0$ case, we have $\theta_{eq} = 0$ and $\theta_{eq} = \pi$ within the $[0, 2\pi]$ range.

The other equations solutions for the equilibrium angles is a bit more involved, but works out such that:

$$0 = -\frac{(mgl)}{I} + \frac{1}{2}\frac{A^2 l^2 m^2 \omega^2}{I^2} cos(\theta)$$

$$\frac{(mgl)}{I} = \frac{1}{2}\frac{A^2 l^2 m^2 \omega^2}{I^2} cos(\theta)$$

$$\frac{(mgl)}{I} = \frac{1}{2}\frac{A^2 l^2 m^2 \omega^2}{I^2} cos(\theta)$$

$$\frac{2gI}{A^2 \omega^2 ml} = cos(\theta)$$

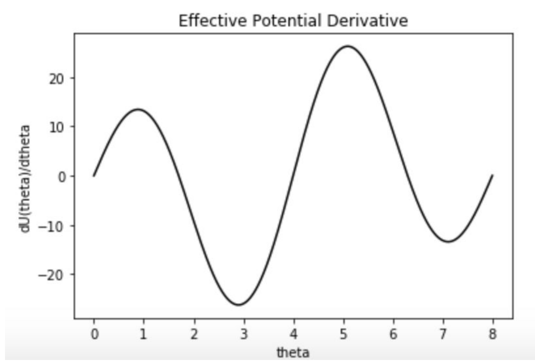
$$\theta_{eq} = \pm arccos(\frac{2gI}{A^2 \omega^2 ml})$$

To determine the stability of these equilibrium points, the second derivative of the effective potential, $\frac{d^2 U_{eff}}{d\theta^2}$ is analyzed at these particular equilibrium points. Positive $\frac{d^2 U_{eff}}{d\theta^2}$ indicate a stable equilibrium point, while negative $\frac{d^2 U_{eff}}{d\theta^2}$ values indicate an unstable equilibrium point.

$$\frac{d^2U_{eff}}{d\theta^2}(\theta) = -\frac{(mgl)}{I}\cdot cos(\theta) + \frac{1}{2}\frac{A^2l^2m^2\omega^2}{I^2}cos^2(\theta) - \frac{1}{2}\frac{A^2l^2m^2\omega^2}{I^2}sin^2(\theta)$$

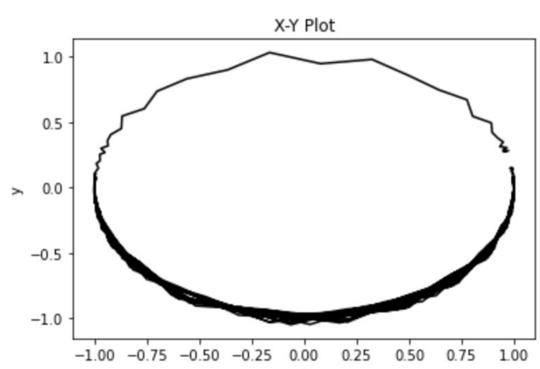
$$\frac{d^2U_{eff}}{d\theta^2}(\theta) = -\frac{(mgl)}{I}\cdot cos(\theta) + \frac{1}{2}\frac{A^2l^2m^2\omega^2}{I^2}\cdot cos(2\theta)$$

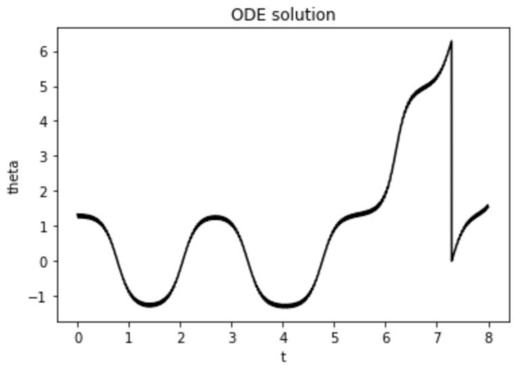
It's also quite helpful to plot $\frac{dU_{eff}}{d\theta}$ vs θ, as it can show the equilibrium points quite clearly as well as display the stability of these points. This plot uses the same parameters as outlined in the sections above:



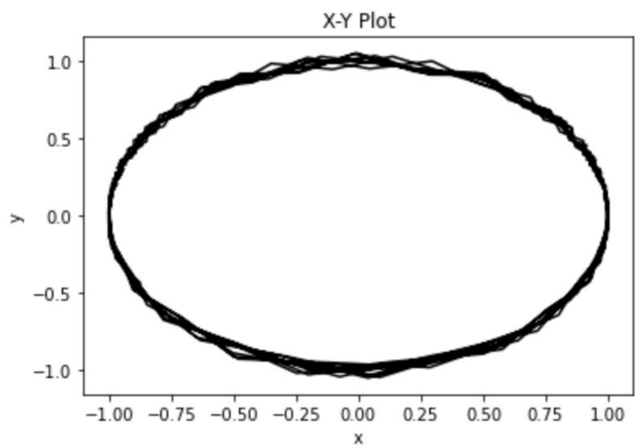
Below are the corresponding graphs at the θ_{eq} points that are non-trivial for our parameters:

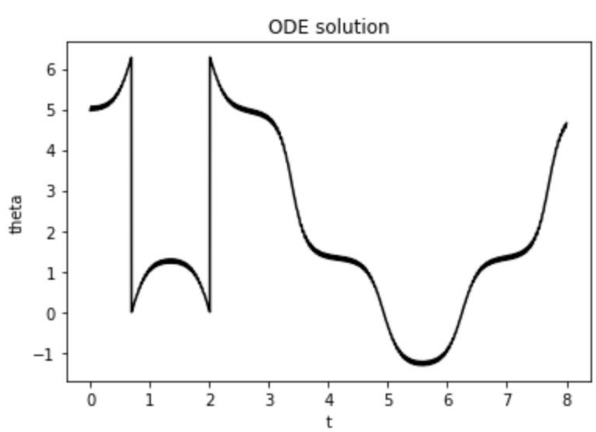
Case 1: $\theta_{eq} = +arccos(\frac{2gI}{A^2\omega^2ml}) = 1.336\ radians$





Case 2: $\theta_{eq} = -arccos(\frac{2gI}{A^2\omega^2 ml}) = 4.974\ radians$





While these graphs initially might look peculiar, keep in mind that both of these equilibrium points are unstable, so even if we start the pendulum at these particular points, they will tend to move away from this equilibrium point since it is unstable.

For the parameters that we chose to work with, this is the stability table

θ_{eq}	Stability
0	Stable
π	Stable
$+ arccos(\frac{2gI}{A^2\omega^2ml})$	Unstable
$- arccos(\frac{2gI}{A^2\omega^2ml})$	Unstable

It will also be important to look at creating a feedback loop and controller that can help maintain the stable point at the top of the pendulum. This is done by first obtaining a transfer function that relates the s time plane which can be used to understand how the system is acted. This is accomplished by forming the same relations of kinetic and potential energy as we did before but this time taking the Laplace transform and solving for the ratio between the input and output functions. According to Roberge the solution to the transfer function is:

$$\frac{\theta}{X}(s) = \frac{s^2/g}{(L/g)\cdot s^2 - 1}$$

This ratio can then be used to create a controller that is also in the s time plane that can help automate the process of maintaining stability. This controller however would have to be derived from test data and then tested itself, which we are unsure at this time if we will have the ability to do.

We also can look at the Bode and Nyquist plots of the data to better understand the stability of the system, which is a way to relate the gain of the system at a particular point, which can be done easily using MATLAB if we have the proper test data.

Conclusion

So far in the project we have found a way to solve the differential equation and to simulate the motion of the inverted pendulum which we can test against a physical one and its resulting data, which will help us to better understand and analyze the minute internal system properties such as inertia and friction loss. As well we hope to be able to control the system with a feedback loop or at least to theorize a way that could be used to properly control and stabilize the system. Additionally we aim to try and derive a separate application for the inverted pendulum and look at how it differs from the standard version but has various new uses with similar solving methods.

Next Steps

The Next part of our project will be to look at a modified pendulum system., and in this case, we have decided to look at the circular motion base. This means that the pendulum will swing as before with the oscillating base but instead of being restricted to one direction of motion, it will be constrained to a path that can be defined by a length in the y axis and a length in the x axis, creating either circular or elliptical motions. To understand the motion of the pendulum from a dynamics perspective we will do this by adding an addition component in the x direction, which now looks like:

$$x = lcos(\theta) + Bsin(\omega t)$$

Where B is the amplitude of the x axis displacement, and it can be equal to the displacement in y (A) or greater or less than A. This equation would then be substituted in as previously done and would be carried through each step as outlined before, which we will discuss further in the final report.