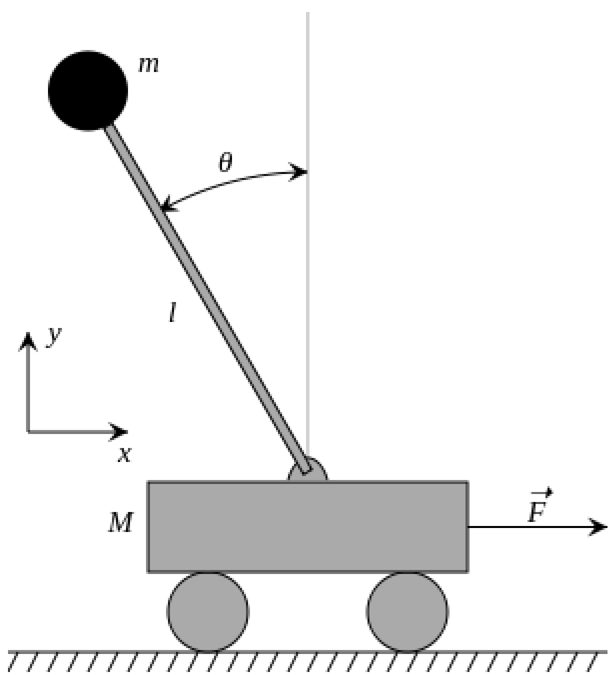
Simulating an Inverted Pendulum on a Moving Platform



Submitted by: Adrian Newby

Submission on: April 11, 2014

Submitted to: Dr. Faisal Quereshi

Introduction

The purpose of this simulation is to model the intricate physics involved in the motion that an inverted pendulum would produce under the effects of gravity and a force applied by a moving platform as its base. This simulation will involve linear and rotational kinematics equations and Lagrangian equations. This simulation will be programmed using java and the java open source physics library. With further development this simulation could be used in the application of modelling a human walking.

Problem Statement

Many people think that an inverted pendulum is a simple construct. However they are unaware of the level of physics that is involved in the kinematics and linear and angular motion in its application. An inverted pendulum on a moving plane also involves effects of more than one mass into the system that must be taken into account. Without a visual representation, it is difficult to show how these things affect the motion of an inverted pendulum.

Model

The model that was used for this simulation is one that uses 11 state variables. This means that 10 first order differential equations were used with one state variable being used for time. Since this was a 2-dimensional simulation, 2 equations were needed to represent each differential equation for linear motion. This means one set for movement along the x-axis for position, velocity, and one set for movement along the y-axis for position and velocity, of the platform and the point mass and a set of differential equations for the angle and angular velocity and one for time. The basic equations that were linear and rotational motion as well as the conservation of energy are as follows:

$$x(t) = x_0 + (v_x * t) \qquad y(t) = y_0 + (v_y * t)$$

$$\frac{dx}{dt} = v_{x_0} + a_x t \qquad \frac{dy}{dt} = v_{y_0} + a_y t$$

$$\frac{dv_x}{dt} = \frac{(v_{x_f} - v_{x_0})}{t} \qquad \frac{dv_y}{dt} = \frac{(v_{y_f} - v_{y_0})}{t}$$

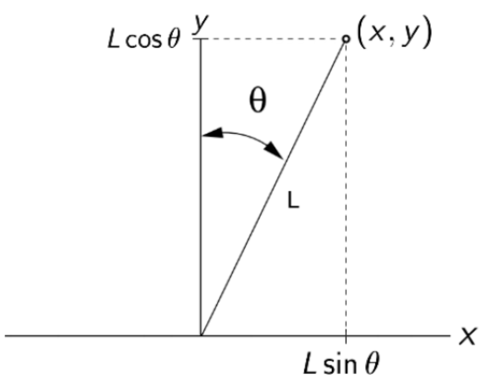
$$\theta(t) = \frac{s}{t}$$

$$\frac{d\theta}{dt} = \frac{|v|\sin(\theta)}{|r|}$$

$$\frac{d\omega}{dt} = \frac{a}{r}$$

$$L = \frac{1}{2}mv^2 + mgh$$

For this model some variations on these formulas were used. The position of the point mass must be changed depending on the angle, so the following equations are used for position and linear velocity of the point mass:



$$x = L\sin\theta \qquad \dot{x} = L\cos(\theta)\dot{\theta}$$

$$y = L\cos\theta \qquad \dot{y} = -L\sin(\theta)\dot{\theta}$$

For the linear velocity of the point mass, the following equation can be used to model it where x refers to position of the pivot point and v_2 is the velocity of the point mass:

$$v_2^2 = \dot{x}^2 - 2\ell\dot{x}\dot{\theta}\cos\theta + \ell^2\dot{\theta}^2$$

From this equation the x and y directional velocities were obtained. The energy of this system can be modelled by the Lagrangian equation:

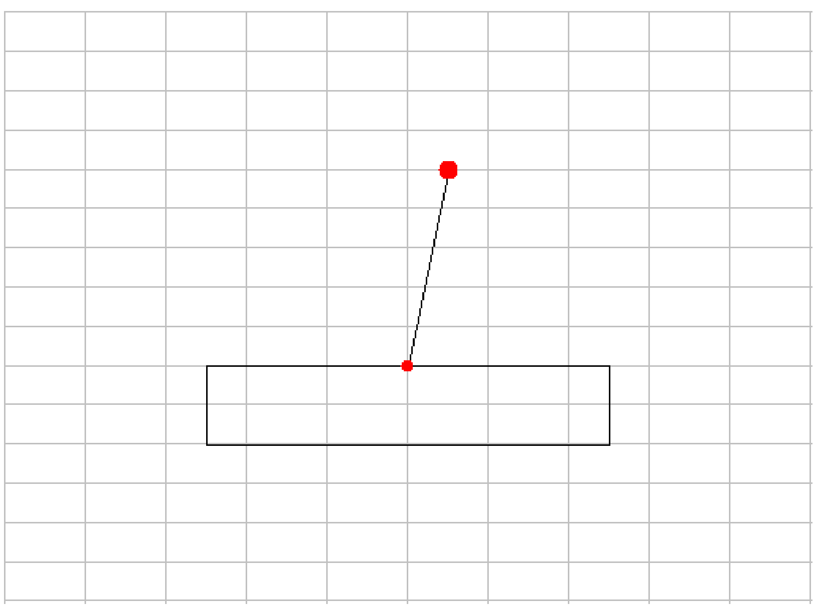
$$\mathcal{L} = \frac{1}{2}mL^2\dot{\theta}^2 - mgL\cos\theta$$

This in turn can be used to determine the angular acceleration in the system with the following differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = \frac{\partial \mathcal{L}}{\partial \theta}$$
$$mL^2\ddot{\theta} = mgL\sin\theta$$
$$\ddot{\theta} = \frac{g}{L}\sin\theta$$

Implementation

The model started with initializing x and y coordinates for the point mass and separate coordinates for the pivot point. The pivot point would be initialized first and the location of the mass would be dependent on where the pivot point was by adding to it the length of the pendulums rod. Because the pendulum should be balanced at $\theta = \frac{\pi}{2}$, the angle was modified to allow the point mass to fall under the influences of gravity by subtracting 0.1 from the angle. Then when applying this to adjust the coordinates for the point mass, the x-coordinate was multiplied by $cos\theta$ and the y-coordinate by $sin\theta$ to get the relative positions. Gravity was set to -9.8 m/s^2 and was applied to the point mass by the equation $\alpha = \frac{g}{r}sin\theta$ to give the angular acceleration. Because of directional components, when the angle is greater than $\frac{\pi}{2}$ the direction is reversed by multiplying the gravity by -1.



The linear velocity of the point mass in the x-axis is modelled by the equation $\frac{dx}{dt} = -lsin\theta$ and the velocity in the y-axis is modelled by the equation $\frac{dy}{dt} = lcos\theta * \omega$. When the angle is less than $\frac{\pi}{2}$ the equation for x velocity changes to $\frac{dx}{dt} = lsin\theta$. By linearizing the velocity for the point mass, the relative position of the mass can be updated. The initial angular velocity is set to 0, and is updated by the angular acceleration. The linear velocity of the platform is initialized at 0 and can only change in the x-direction. This change is only applied when the user presses either the left or right arrows. When an arrow is pressed, the an instantaneous velocity of 3 m/s is applied to the platform in the respective directions to prevent the need for the platform to have to slow down before moving in the desired direction. A constant force of 15N was applied by the platform which is then translated into angular acceleration and is applied in the opposite direction of the current angular acceleration of the point mass to reverse its direction in an attempt to balance it.

In order to make the base and rod connected to the mass, the Arrow class of osp was copied and renamed to Line and then modified to allow use of a constructor with no arguments to be used. This constructor would set the size of the arrow head to 0 resulting in a line. A method called setAB, which took 2 parameters, was also created under the Line class because the current implementation of the Line class does not allow for change of the arrow heads location which causes a problem with the changing location of the point mass and the platform since the Line class was used to draw the moving platform.

When the angle was at $\theta=\pi/2$, the mass would begin to fall straight down. This was fixed by setting the gravity to 0 when $\theta=\pi/2$, and then restoring it back to $-9.8m/s^2$. Another issue encountered was that at times the distance between the point mass and the pivot point would exceed or in some cases become less than the specified length of the rod. This was fixed by implementing a check to see if the Euclidean distance was equal to the rod length. If it was not, then the correct location of the mass was calculated using the current angle value.

A function to square two values and perform sin and cos operations were made in order to improve the readability of the equations. A setBox method was also created to update the position of the platform and is called in the doStep.

Results

Successful in simulating an inverted pendulum on a moving platform to an extent. The simulation does not allow variations in weights of the point mass and platform, or variations in the amount of force applied by the platform which would help to demonstrate the effects of a moving platform to balance the pendulum with varying forces. The simulation does not include any friction, air drag, or impulse when the pendulum hits the platform as it could potentially bounce back up a little in an inelastic collision. There is also no collision detection which would have made the pendulum stop when once it hits the platform. In this simulation if the pendulum gains energy after a few rotations past the platform which could have been fixed by implementing collision detection and setting the acceleration to 0.

Analysis

This simulation shows that in order to balance an inverted pendulum a force grater that the result force on the mass due to gravity must be applied in the opposite direction. If the force applied is too small then the mass will continue to fall. However, if the force is too great, then it may be impossible to balance by moving the platform in the opposite direction because the force applied initially will get added to the force due to gravity and the mass will fall and reach the platform before enough force is applied to reverse the direction of the point mass.

Conclusion

Inverted pendulums are believed to be a simple construct by many. But that is only because they do not understand the underlying physics involved. This simulation shows the interactions between an inverted pendulum and a moving platform and the calculations performed dhow how complicated it can be. This simulation is very basic and can be extended to allow users to modify parameters such as the amount of force, change the amount of gravity, change the masses, or alternatively instead of changing the force directly, the simulation could incorporate the calculation of the force with the user modifying accelerations and mass and the trying to balance the pendulum. Another way to extend this simulation could be to modify it to simulate a person walking and the act of walking has the legs as inverted pendulums.

References

http://www.cas.mcmaster.ca/~qiao/courses/cs4xo3/presentations/InvPend.pdf

http://francesa.phy.cmich.edu/people/andy/physics110/book/chapters/chapter6.htm

http://en.wikipedia.org/wiki/Inverted_pendulum

http://www.engr.usask.ca/classes/EE/480/Inverted%20Pendulum.pdf

http://physics.stackexchange.com/questions/15098/linear-acceleration-vs-angular-acceleration-equation