Development Of A Wheeled Inverted Pendulum

Nathan Smith

April 8th, 2020

Contents

1 Dynamic Model

1.1 Derivation

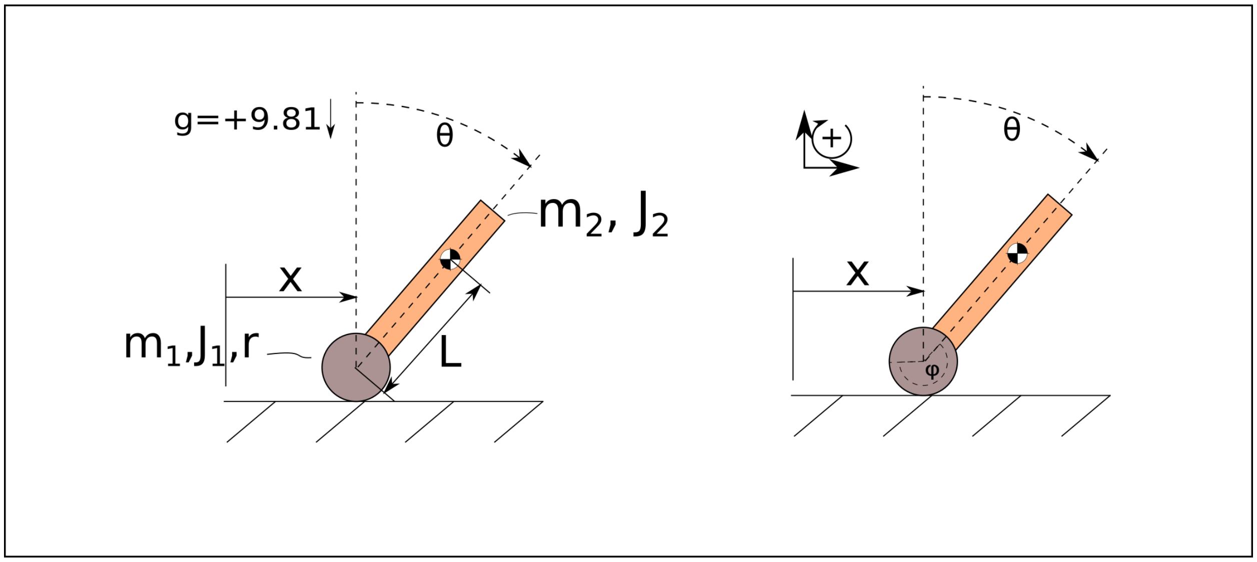

Figure 1: Overview and Definitions

As shown in 1 the configuration of the system is given by x and θ. The variable ϕ is the rotation of the wheel relative to the vertical axis and is completely determined by x:

$x = r\phi$ and $\dot{x} = r\dot{\phi}$

Positive translations are to the right and position rotations are clockwise. Also a positive torque is one which tends to make the WIP accelerate in the positive direction (and cause a negative rotation).

Since there are two bodies in this system the Euler-Lagrange equation will be used to derive the equations of motion for the WIP:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i \tag{1}$$

where $q = \{x, \theta\}$ are the generalized coordinates defining the configuration of the system and $F = \{\frac{\tau}{r}, -\tau\}$ are the generalized forces acting on each axis. The Lagrangian is also defined as the difference between the kinetic and potential energies of the system:

$$L = K - U \tag{2}$$

The Kinetic energy can be expressed as the sum of the translational and rotational kinetic energies of each body:

$$K = \frac{1}{2}m_1|\dot{\vec{x}}_1|^2 + \frac{1}{2}m_2|\dot{\vec{x}}_2|^2 + \frac{1}{2}J_1\dot{\phi}^2 + \frac{1}{2}J_2\dot{\theta}^2$$

To continue $|\dot{\vec{x}}_1|^2$ and $|\dot{\vec{x}}_2|^2$ must be evaluated. Given the definitions of these vectors in 2:

$$\vec{x}_1 = x\hat{i}$$
$$\dot{\vec{x}}_1 = \dot{x}\hat{i}$$
$$|\dot{\vec{x}}_1|^2 = \dot{x}^2$$

$$\vec{x}_2 = (x + Lsin(\theta)\hat{i} + Lcos(\theta)\hat{j}$$
$$\dot{\vec{x}}_2 = (\dot{x} + Lcos(\theta)\dot{\theta})\hat{i} - Lsin(\theta)\dot{\theta}\hat{j}$$

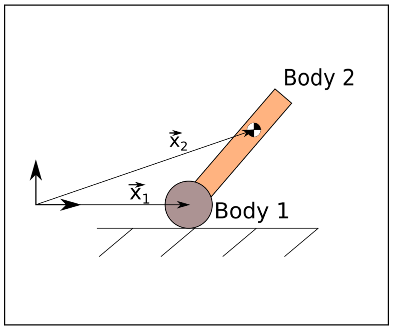

Figure 2: Definitions of $\dot{\vec{x}}_1$ and $\dot{\vec{x}}_2$

$$|\dot{\vec{x}}_2| = \sqrt{(\dot{x} + Lcos(\theta)\dot{\theta})^2 + (-Lsin(\theta)\dot{\theta})^2}$$

$$|\dot{\vec{x}}_2| = \sqrt{\dot{x}^2 + 2Lcos(\theta)\dot{x}\dot{\theta} + L^2cos^2(\theta)\dot{\theta}^2 + L^2sin^2(\theta)\dot{\theta}^2}$$

$$|\dot{\vec{x}}_2|^2 = \dot{x}^2 + 2Lcos(\theta)\dot{x}\dot{\theta} + L^2\dot{\theta}^2$$

Substituting back into the appropriate terms in the Kinetic energy along with the definition of ϕ:

$$K = \frac{1}{2}m_1\dot{x}^2 + \frac{1}{2}m_2(\dot{x}^2 + 2Lcos(\theta)\dot{x}\dot{\theta} + L^2\dot{\theta}^2) + \frac{1}{2}J_1\frac{\dot{x}^2}{r^2} + \frac{1}{2}J_2\dot{\theta}^2$$

$$K = \frac{1}{2}(m_1 + m_2 + \frac{J_1}{r^2})\dot{x}^2 + m_2Lcos(\theta)\dot{x}\dot{\theta} + \frac{1}{2}(m_2L^2 + J_2)\dot{\theta}^2$$

This equation is split into three terms, linear acceleration, angular acceleration, and an interaction term between linear and angular velocity. Therefore it seems reasonable to define an effective mass and moment of inertia as $M = m_1 + m_2 + \frac{J_1}{r^2}$ and $I = m_2L^2 + J_2$.

Substituting these into the Kinetic energy gives:

$$K = \frac{1}{2}M\dot{x}^2 + m_2Lcos(\theta)\dot{x}\dot{\theta} + \frac{1}{2}I\dot{\theta}^2 \tag{3}$$

Potential energy can be found as:

$$U = m_2gLcos(\theta) \tag{4}$$

Substituting 3 and 4 into 2 gives the full Lagrangian for the system:

$$L = \frac{1}{2}M\dot{x}^2 + m_2Lcos(\theta)\dot{x}\dot{\theta} + \frac{1}{2}I\dot{\theta}^2 - m_2gLcos(\theta) \tag{5}$$

The Lagrangian must now be substituted into 1 for each generalized coordinate. Starting with x:

$$\frac{\partial L}{\partial x} = 0$$

$$\frac{\partial L}{\partial \dot{x}} = M\dot{x} + m_2Lcos(\theta)\dot{\theta}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = M\ddot{x} + m_2Lcos(\theta)\ddot{\theta} - m_2Lsin(\theta)\dot{\theta}^2$$

$$M\ddot{x} + m_2Lcos(\theta)\ddot{\theta} - m_2Lsin(\theta)\dot{\theta}^2 = \frac{\tau}{r} \tag{6}$$

Equation 6 is the first equation of motion for the system. Repeating this for θ will give the second:

$$\frac{\partial L}{\partial \theta} = -m_2Lsin(\theta)\dot{x}\dot{\theta} + m_2gLsin(\theta)$$

$$\frac{\partial L}{\partial \dot{\theta}} = m_2Lcos(\theta)\dot{x} + I\dot{\theta}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = m_2Lcos(\theta)\ddot{x} - m_2Lsin(\theta)\dot{x}\dot{\theta} + I\ddot{\theta}$$

$$m_2Lcos(\theta)\ddot{x} - m_2Lsin(\theta)\dot{x}\dot{\theta} + I\ddot{\theta} + m_2Lsin(\theta)\dot{x}\dot{\theta} - m_2gLsin(\theta) = -\tau \tag{7}$$

Equations 6 and 7 both depend on $\ddot{x}$ and $\ddot{\theta}$ and must be solved in terms of these simultaneously. Doing this yields:

$$\ddot{x} = -\frac{m_2LIsin(\theta)}{m_2^2L^2cos(\theta) - MI}\dot{\theta}^2 + \frac{m_2^2L^2g}{m_2^2L^2cos(\theta) - MI}cos(\theta)sin(\theta) - \left[\frac{I + rm_2Lcos(\theta)}{r(m_2^2L^2cos(\theta) - MI)}\right]\tau \tag{8}$$

$$\ddot{\theta} = \frac{m_2^2L^2sin(\theta)}{m_2^2L^2cos(\theta) - MI}\dot{\theta}^2 - \frac{Mm_2gLsin(\theta)}{m_2^2L^2cos(\theta) - MI} + \left[\frac{rM + m_2L}{r(m_2^2L^2cos(\theta) - MI)}\right]\tau \tag{9}$$

These two non-linear equations describe the motion of the system in the configuration space. To make simulation easier these are converted into the state space by adding the velocities associated with each coordinate:

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{x}\\ -\frac{m_2LIsin(\theta)}{m_2^2L^2cos(\theta)-MI}\dot{\theta}^2 + \frac{m_2^2L^2g}{m_2^2L^2cos(\theta)-MI}cos(\theta)sin(\theta)\\ \dot{\theta}\\ \frac{m_2^2L^2sin(\theta)}{m_2^2L^2cos(\theta)-MI}\dot{\theta}^2 - \frac{Mm_2gLsin(\theta)}{m_2^2L^2cos(\theta)-MI}\end{bmatrix} + \begin{bmatrix}0\\ -\frac{I+rm_2Lcos(\theta)}{r(m_2^2L^2cos(\theta)-MI)}\\ 0\\ \frac{rM+m_2L}{r(m_2^2L^2cos(\theta)-MI)}\end{bmatrix}\tau \tag{10}$$

1.2 Linearization

Equation 10 can be easily simulated with tools such as Scipy's ODEint function to create a reasonable representation of the system. It is not, however, easy to develop control laws for a non-linear system and so it is necessary to linearize it. This will be done about the upwards stationary point $\vec{x}^* = [x, 0, 0, 0]^T$. There is an x in this state since there is a symmetry in x. Currently the system is of the form:

$\dot{\vec{x}} = \vec{f}(\vec{x}, u)$, where in this case $u = \tau$

Linearizing would put the system into the form:

$$\dot{\vec{x}} = A\vec{x} + Bu$$

To generate A and B the following formula must be applied: $A = \left[\frac{\partial \vec{f}(\vec{x},u)}{\partial \vec{x}}\right]_{x^*,u^*}$ and $B = \left[\frac{\partial \vec{f}(\vec{x},u)}{\partial u}\right]_{x^*,u^*}$

Doing so yields:

$$A = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \frac{m_2^2L^2g}{m_2^2L^2-MI} & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -\frac{Mm_2gL}{m_2^2L^2-MI} & 0\end{bmatrix} \quad and \quad B = \begin{bmatrix}0\\ -\frac{I+rm_2L}{r(m_2^2L^2-MI)}\\ 0\\ \frac{rM+m_2L}{r(m_2^2L^2-MI)}\end{bmatrix} \tag{11}$$

The output equation will initially return only θ and so will be:

$y = \vec{C}\vec{x} + Du$, where $\vec{C} = [0, 0, 1, 0] and D = 0$

1.3 Transfer Function

Due to some quirks in the Python-Controls library with regards to analyzing state space models, the state space representation derived will be converted into a transfer function to make analysis easier to interpret. Doing so uses:

$$\frac{\theta(s)}{\tau(s)} = \vec{C}(sI - A)^{-1}\vec{B} + D \tag{12}$$

Applying this to the current system gives:

$$\frac{\theta(s)}{\tau(s)} = \frac{a}{s^2 - b} \tag{13}$$

where $a = \frac{rM + m_2 L}{r(m_2^2 L^2 - MI)}$ and $b = -\frac{M m_2 g L}{m_2^2 L^2 - MI}$

2 Controller Design

2.1 Plant analysis

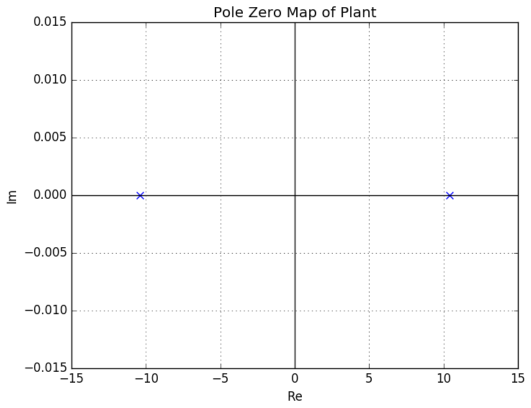

Figure 3: Pole Zero map of the transfer function. The two poles are real and symmetric, one always in the right half plane.

Figure 3 shows the pole zero map of equation 13, it can be seen that one of the poles is in the right half plane and so the system is unstable.

2.2 Direct Feedback Design with Root Locus

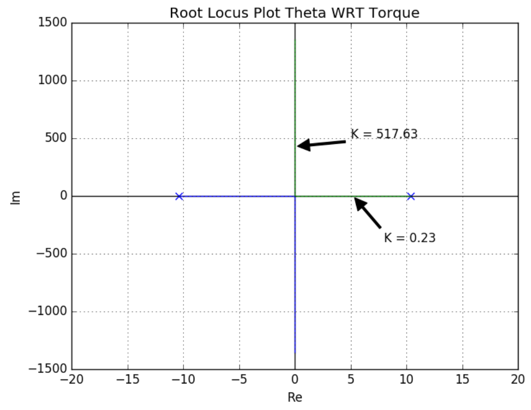

Figure 4: Root locus plot of the plant. Larger gains only yield marginal stability.

As a first attempt to stabilize the WIP a simple gain controller was implemented in the simulation. As shown in figure 6 the system is unstable for low gains and only achieves marginal stability with larger gains. This is not acceptable as a solution but does make for an interesting animation which can be seen in the readme for this project.

2.3 PD Design

While continuous time PD controllers are not realizeable, the fact that my simulation provides access to the entire state of the system I am able to sidestep this limitation though when it is implemented on hardware the controller will need to be discretized. The usual form of a PD controller is:

$$C(s) = k_d s + k_p \tag{14}$$

This equation has two parameters to vary and so makes traditional design techniques such as root locus design difficult. By factoring out k_d this becomes slightly easier since the pole can be placed by hand at the point $-\frac{k_p}{k_d}$ and k_d can be used as the root locus gain. Doing this gives the modified PD transfer function:

$$C(s) = k_d(s + \frac{k_p}{k_d}) \tag{15}$$

The PD controller effectively adds a zero to the function and in this form the ratio $\frac{k_p}{k_d}$ decides that zero's location. Since both system poles and the zero are on the real axis, the zero can either be between the open loop poles $\frac{k_p}{k_d} < b$ or to the left of them $\frac{k_p}{k_d} > b$. To get reasonably quick dynamics the latter case should be selected.

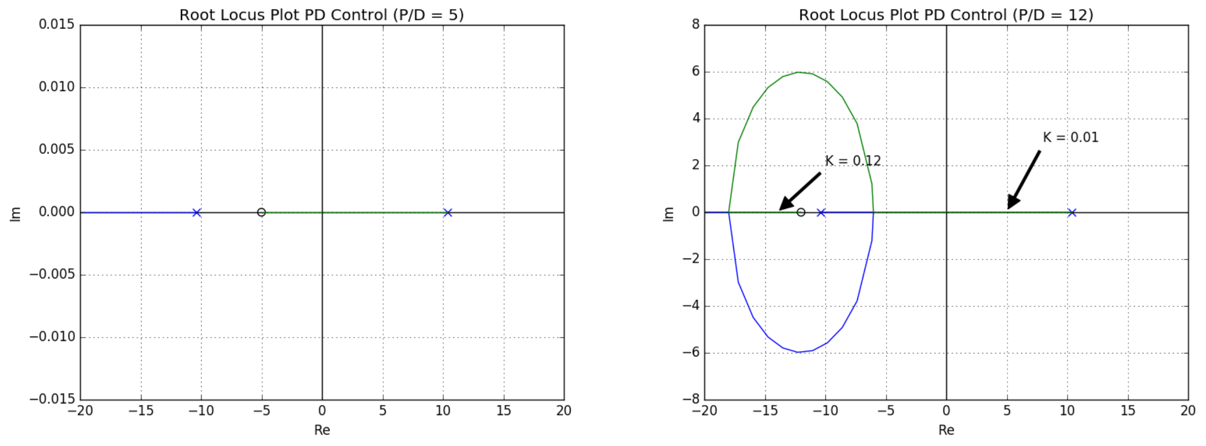

Figure 5: Root Locus plot of the plant with a PD controller. K value in image is k_d and the location of the zero is $-\frac{k_p}{k_d}$

The right plot of figure 5 shows the more desirable case. As can be seen in that plot from the annotated gains, even a gain as small as 0.12 will move the dominant pole very close to the zero. The non-dominant pole is far in the negative direction and was excluded from the figure in the interest of maintaining a reasonable scale and readability.

3 Mechanical Design

3.1 Part Sourcing

Due to the very limited budget for this project most parts were scavenged from old printers and toys. Most structural parts came from a toy erectors set which allowed for some degree of uniformity in the design due to those components having regularly spaced holes. The motor mounts were made by cutting the entire motor assembly along with the encoders out of an old printer and mounting them in the best possible way to the sides of the body. This approach led to many issues of alignment with regards to the Inertial Measurement Unit (IMU) which required extra effort to calibrate.

3.2 Custom Parts

Since the gears attached to the wheels were not designed to mount the wheels I had on hand, I had to design a custom adaptor to allow the wheels to mount.

4 Electrical Design

4.1 Component Selection

To control the robot an Arduino microcontroller was used as it was available at the start of the project. While the Arduino isn't the fastest controller available it should be more than fast enough for this project. Both motors will be driven by a single L298-n motor driver circuit. This chip also has a 5 volt regulated output and will be used to provide power to the Arduino and IMU. In order to measure the orientation of the robot an MPU-6050 Inertial Measurement Unit was used.

4.2 Motor Parameter Evaluation

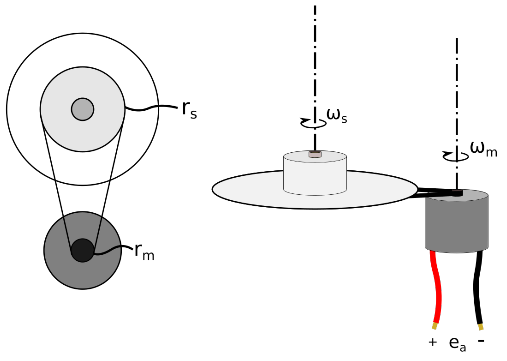

Figure 6: Overview of setup. ω_s can be measured directly and related to ω_m through geometry. The back emf, e_a, can be measured directly as well.

The following model was used to represent the motors[1]

$$\frac{\tau(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} \tag{16}$$

Where: $K_m = \frac{K}{Rb+K^2}$ and $T_m = \frac{RJ_{mot}}{Rb+K^2}$

5 Software Design

5.1 Component Testing Software

5.2 Controller Software

References

[1] Katsuhiko Ogata, *System Dynamics*, 4th edition, 2004.