6.115 Final Project Proposal: Swinging Up the Inertial Wheel Pendulum

Albert Wu (wualbert@mit.edu)

Introduction

The inertial wheel pendulum is a classic under-actuated system. The mechanical structure has 2 degrees of freedom: a free-swinging base joint and an actuated inertial wheel. Figure 1 provides an illustration of a canonical inertial wheel pendulum. The system is mechanically simple with well-understood dynamics, but it is nontrivial to control. Innovative control systems for this system is still an active field of research. In fact, the inertial wheel pendulum often serves as a testbed for new control algorithms because of its simplicity. That being said, there exists many established controllers that can reliably control the system. In this project, the problem of inverting an inertial wheel pendulum will be tackled with a hybrid control system. The objectives of the project are to build the system, stabilize the system at the upright position, and drive the system to the upright position from any initial condition.

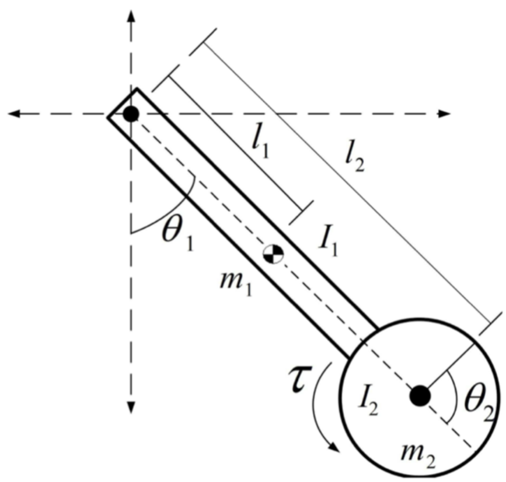

Figure 1. Illustration of a canonical inertial wheel pendulum. Adapted from [1]

Hardware Description

Mechanical Hardware

As illustrated in Figure 1., the pendulum has two joints, which will be referred to as the "base joint" (θ_1) and the "wheel joint" (θ_2) in this project. The base joint is a free-spinning pin joint. The wheel joint is a pin joint directly driven by the motor. Both joints will have an encoder. The base joint will have a magnetic encoder (AS5047P). The wheel joint encoder is built into the motor (Pololu 50:1 Metal Gearmotor 37Dx70L mm with 64 CPR Encoder). Both the pendulum and the wheel will be made out of laser-cut acrylic. Figure 2 and figure 3 shows the CAD of the link and the wheel.

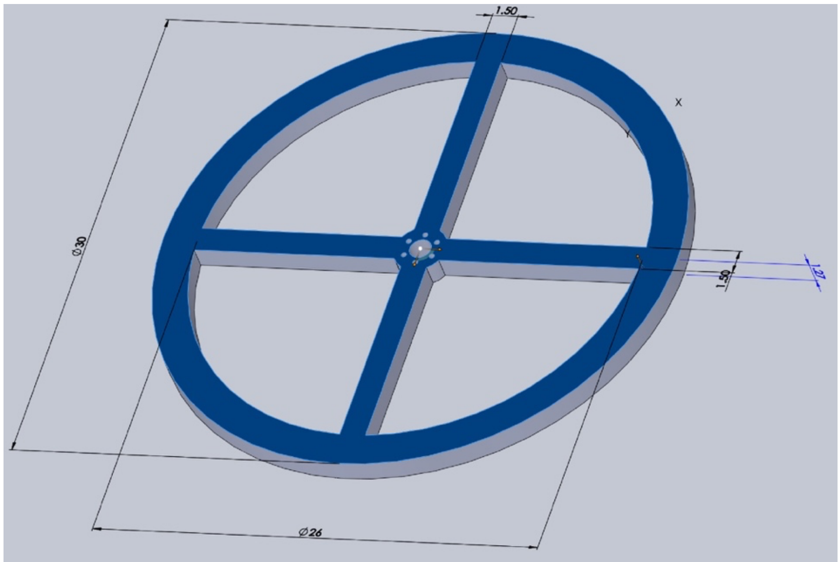

Figure 2. Inertial wheel CAD

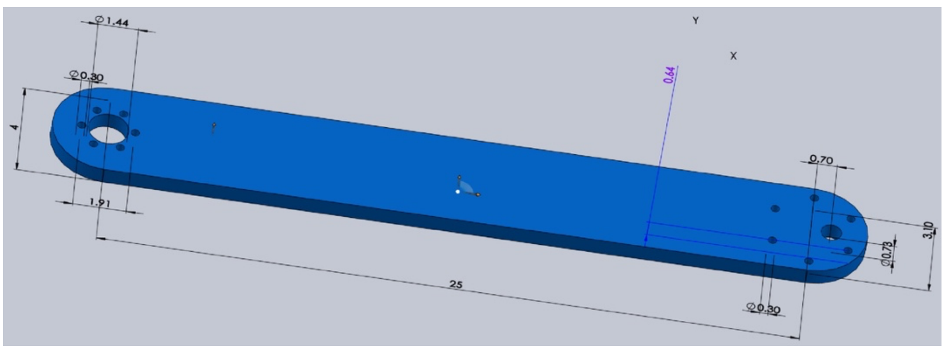

Figure 3. Pendulum link CAD

Electrical Hardware
The AS5047P and motor encoders supports ABI outputs, which can be directly interpreted by both the PSoC and 8051. A commercial MC10C motor controller board will be used for driving the motor. A current feedback system will be built for closed-loop motor current control. The feedback system consists of differential amplifiers and a maximum value clamp circuit. The schematic of the system is shown in figure 4.

Safety Features
The entire pendulum will be enclosed in a box for safety reasons.

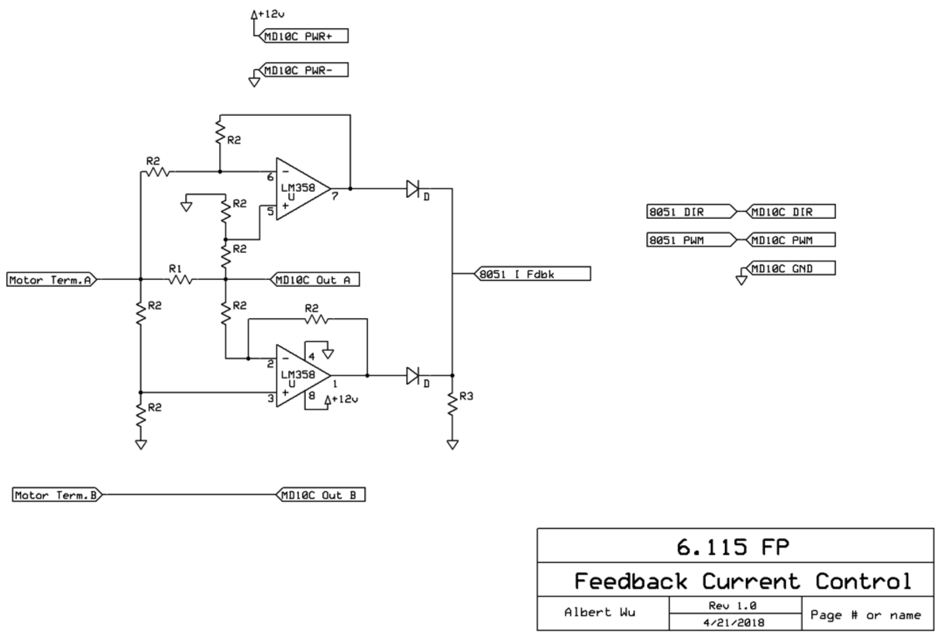

Figure 4. Current Feedback Circuit

Software Description

The software, primarily for controlling the pendulum, is split between the 8051 and PSoC. 8051 will contain the closed-loop motor controller code. The PSoC will contain the high-level controller code.

8051

The main function of the 8051 code is to provide closed-loop torque control on the motor. It takes in inputs from the PSoC in the form of target motor torque and achieve so with a closed-loop controller. Since DC motor torque is governed by the equation

$$\tau_m = K_t i$$

torque control can be achieved with current control. The 8051 closes the loop on current and sends a PWM signal to the motor driver boards, which outputs the motor armature voltage. The current dynamics are characterized with the standard DC motor equations.

PSoC

The PSoC generates high level control output, which is sent to the 8051 to control the motor. The software is just an implementation of the control laws, described in more detail in the next section and appendix. The PSoC also takes in readings from the 2 encoders and form a state vector

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

for controlling the pendulum.

Figure 5 is a block diagram of the system.

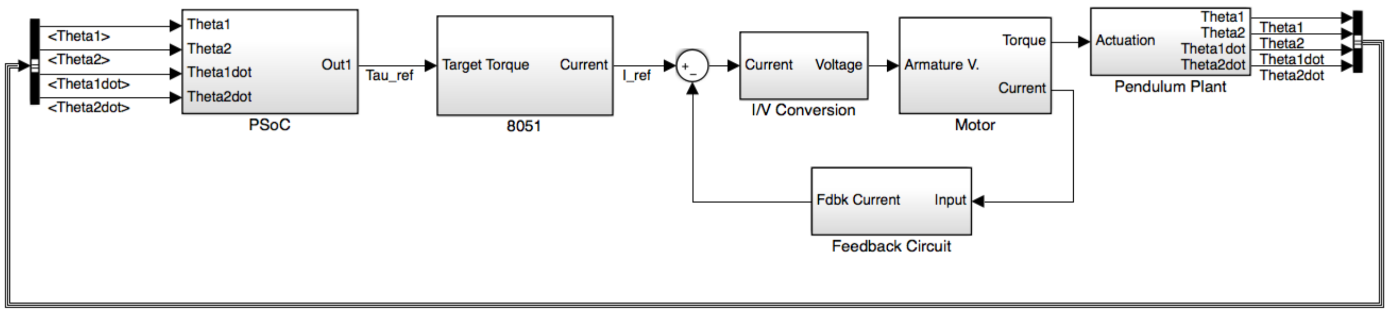

Figure 5. Inertial wheel pendulum system block diagram

The software flow is shown in figure 6.

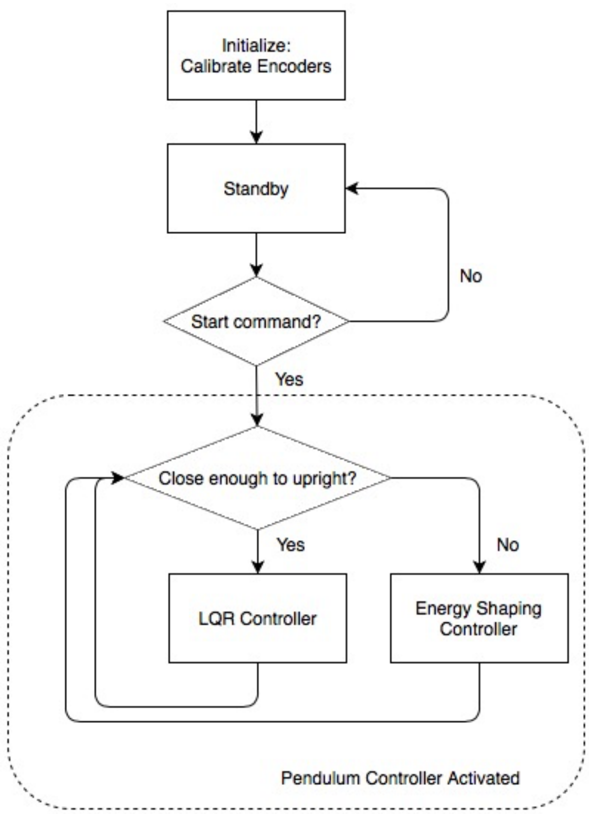

Figure 6. Software flow diagram

Controlling the Pendulum

There are 2 modes of operations in this system: stabilization and swing-up. The former uses an LQR controller and the latter uses an energy-shaping controller. The PSoC will decide which one to use depending on the state vector. The controller matrices for LQR will be precomputed and hard-coded onto the PSoC. Control laws for the energy-shaping controller will also be derived in advance. The appendix provides the derivation of the controllers.

Project Scope and Management

The project can be broken down into the following parts, listed in completion priority:

1. Building the mechanical hardware of the system
    a. Design the parts required for the system
    b. Manufacture the parts
    c. Assemble the inertial wheel pendulum
2. Building the basic software of the system
    a. Implement a closed-loop torque controller with 8051
    b. Read encoder values with the PSoC
    c. Achieve communication between the PSoC and 8051
    d. Have all sensor inputs and actuator outputs available on the PSoC
3. Achieving stabilization at upright position
    a. Calculate and linearize the dynamics of the inertial wheel pendulum about the upright position
    b. Use MATLAB to find the optimal A,B,C,D matrices
    c. Implement the controller on the PSoC
    d. Test the system and tune the matrices
4. Converging to upright from any initial condition
    a. Design a controller that can move the inertial wheel pendulum to the admissible range of the LQR controller
    b. Implement the controller on the PSoC
    c. Design and implement a decision rule for switching controllers
    d. Test and tune the controller

1~3 are basic project goals. 4 is more advanced and is relatively disjoint from the others.

Special Components

The following parts are required for this project. All of them have be obtained:

1. 1* DC motors with built in encoder
2. 1* Rotary encoder (AS-5047P)
3. Motor driver board (Cytron MD 10C)
4. Stock acrylic to cut into parts
5. Pillow block bearing for mounting the pendulum
6. Screws, hubs, shafts, and other miscellaneous items for assembly

Timetable

Week of 4/16: Build the pendulum
Week of 4/23: Enable communication between PSoC, 8051, actuator, and sensors
Week of 4/30: Complete full sensor reading and actuator control
Week of 5/7: Implement and tune the control algorithms
Week of 5/14: Buffer time

References

1. Ramirez-Neria, Mario & Sira-Ramirez, Hebertt & Garrido, Ruben & Luviano-Juárez, Alberto. (2015). On the linear Active Disturbance Rejection Control of the inertia wheel pendulum. 3398-3403. 10.1109/ACC.2015.7171857.

Appendix: Derivation of the Control Laws

In the appendix, the variable definitions from Figure A1 will be used for all derivations.

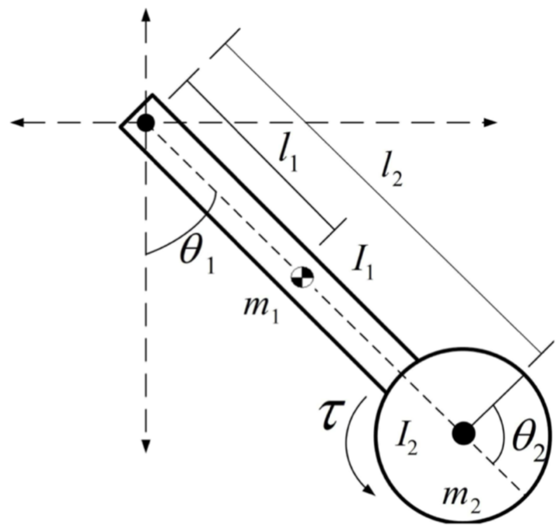

Figure A1. Illustration of a canonical inertial wheel pendulum. Adapted from [1]

System Dynamics

Equation of Motion

Define the generalized coordinates q and the system state x as follows:

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

Assuming no damping, the system equation of motion is

$$M\ddot{q} = \tau_g(q) + Bu$$

with

$$M = \begin{bmatrix} m_1 l_1^2 + m_2 l_2^2 + I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix}, \tau_g = \begin{bmatrix} -(m_1 l_1 + m_2 l_2) g \sin\theta_1 \\ 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and $u = \tau$, the control input. [1][

LQR Controller

Linearization

The system can be linearized around the fix point objective $\theta_1 = \pi$. Writing out the state space representation of the system gives

$$\begin{bmatrix} I_2 & 0 \\ 0 & M \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ -(m_1 l_1 + m_2 l_2) g \sin\theta_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

We are interested in the upright fixed point $\theta_1 = \pi$. Around fixed points, $\dot{x} = 0$. Let $\tau_g = \tau_f + \bar{\tau}, u = u_f + \bar{u}$. τ_f and u_f cancel out each other. It can be verified that at this fixed point, $\tau_f = 0$ and $u_f = 0$. Linearize τ_g to get

$$\tau_g(x) = \tau_f + \dot{\tau}_g \bar{x}$$

where, at the fixed point,

$$\dot{\tau_g} = \begin{bmatrix} 0 \\ 0 \\ -(m_1 l_1 + m_2 l_2) g \cos(\theta_{1f}) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ (m_1 l_1 + m_2 l_2) g \\ 0 \end{bmatrix}$$

The EOM becomes

$$\dot{\bar{x}} = \begin{bmatrix} I_2 & 0 \\ 0 & M \end{bmatrix}^{-1} \begin{bmatrix} 0_2 & & I_2 \\ (m_1 l_1 + m_2 l_2) g & 0 & \\ 0 & 0 & 0_2 \end{bmatrix} \bar{x} + \begin{bmatrix} I_2 & 0 \\ 0 & M \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

Notice that the inertia matrix $\begin{bmatrix} I_2 & 0 \\ 0 & M \end{bmatrix}$ is positive definite, so it is always invertible.

LQR Design

Upon close inspection, θ_2 does not affect the evolution of other state variables. Since θ_2 is irrelevant in this system (the wheel position does not matter), it should be eliminated from the LQR state space. Define the reduced state

$$x_r = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

The new state evolution equation becomes

$$\dot{\bar{x}}_r = \begin{bmatrix} 1 & 0_{12} \\ 0_{21} & M \end{bmatrix}^{-1} \begin{bmatrix} 0 & I_{12} \\ (m_1 l_1 + m_2 l_2) g & \\ 0 & 0_2 \end{bmatrix} \bar{x}_r + \begin{bmatrix} 1 & 0_{12} \\ 0_{21} & M \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

The matrices Q, R can be chosen arbitrarily. Putting this into a solver yields the optimal control law, in the form of

$$u = -K x_r$$

The controller can be switched on and will guarantee convergence to the upright position as long as the state is within the region of attraction.

Energy-Shaping Controller

Motivation

One can see that the pendulum will always "pass" the target state at some point as long as the total mechanical energy is correct. Therefore, instead of driving the pendulum to a specific point, it is driven to a specific energy level. However, under this condition, the system can "swing around" if a small perturbation is applied. To resolve this problem, once the pendulum gets close enough to the upright position, the LQR controller switches on to ensure local stability.

Non-collocated Feedback Linearization

Since we are interested in controlling θ_1 but only have actuation on θ_2, a non-collocated feedback linearization must be used to achieve the desired energy level. Recall the EOM of the system:

$$(m_1 l_1^2 + m_2 l_2^2 + I_1 + I_2)\ddot{\theta}_1 + I_2 \ddot{\theta}_2 = -(m_1 l_1 + m_2 l_2) g \sin\theta_1$$

$$I_2 \ddot{\theta}_1 + I_2 \ddot{\theta}_2 = u$$

For non-collocated feedback linearization on θ_1, we eliminate θ_2. Define the following:

$$a = m_1 l_1^2 + m_2 l_2^2 + I_1 + I_2$$
$$b = (m_1 l_1 + m_2 l_2) g \sin\theta_1$$
$$c = m_1 l_1 + m_2 l_2$$

Then the EOM in terms of θ_1 becomes

$$(a - I_2)\ddot{\theta}_1 = -b - u$$

Select the control input u as

$$u = u_l + u_d, u_l = -b, u_d = (I_2 - a)\ddot{\theta}_{1d}$$

$\ddot{\theta}_{1d}$ is the desired acceleration. This way, the system dynamics can be canceled, and

$$\ddot{\theta}_1 = \ddot{\theta}_{1d}$$

Energy Shaping

For convergence to the upright position, rotational kinetic energy about θ_2 is irrelevant. Therefore, the wheel rotational velocity is disregarded, and we have

$$E = \frac{1}{2} a \dot{\theta}_1^{\,2} - c\cos\theta_1$$

The desired energy is

$$E_d = c$$

Define the energy error $\tilde{E}$ as

$$\tilde{E} = E - E_d$$

Evaluate the derivative of $\tilde{E}$

$$\dot{\tilde{E}} = a\dot{\theta}_1\ddot{\theta}_1 + c\dot{\theta}_1 \sin\theta_1$$
$$= \dot{\theta}_1(a\ddot{\theta}_1 + b)$$

Select the target acceleration

$$\ddot{\theta}_{1d} = -\frac{b}{a} - \frac{k\dot{\theta}_1\tilde{E}}{2}$$

The energy derivative becomes

$$\dot{\tilde{E}} = -k\dot{\theta}_1^{\,2}\tilde{E} \le 0$$

The system will converge to the desired energy level except when $\dot{\theta}_1 = 0$. This is a transient state except when $\dot{\theta}_1 = \theta_1 = 0$. The issue is easily compensated through adding a small perturbation when necessary. The full control law is therefore

$$u = u_l + u_d = -b + \frac{(I_2 - a)bk\dot{\theta}_1\tilde{E}}{2a}$$

References

1. Russ Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on April 8, 2018 from http://underactuated.mit.edu/
2. Assignment 3 of MIT 6.832. Downloaded on March 17, 2018 from http://underactuated.mit.edu/