Experiment A10
Second Order Transient Response
Procedure

Deliverables: Checked lab notebook, Brief Tech Memo

Recommended Reading: Sections 7.3 and 9.1, Chapters 18, 19, 20, and 22

Background

Structures used in aerospace and mechanical engineering often vibrate and oscillate. In fact, any system—mechanical, chemical, thermal, or electrical—with dynamic behavior governed by a second order differential equation of the form

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = f(t) \quad (1)$$

will oscillate or vibrate at a natural resonance frequency ω_n. The damping ratio ζ determines how long oscillations will persist. The term on the right-hand side $f(t)$ represents external forces or inputs to the system. In this lab, we will exam such a system subject to two well-known external forces $f(t)$ using a pendulum:

- **Step Input or "Impulsive" Force** – The pendulum will be held up at an initial angle using the force of your hand, then abruptly released. The pendulum will swing back and forth at the ringing frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$, and the oscillations will decay at a rate related to the damping ratio.

- **Oscillating Force Input** – The pendulum will be driven by an electric motor with a torque that oscillates sinusoidally at a driving frequency ω.

Part I: Pendulum – Response to an Impulse

Theoretical Background

Illustrated in Fig. 1a, a pendulum can be modeled as a point mass *m* located a fixed distance *R* away from the axis of rotation. (The distance *R* is often referred to as the "radius of gyration".) Balancing the angular acceleration with the torque from gravity and a viscous drag force yields the equation of motion for the pendulum

$$mR^2\ddot{\theta} = -mgR\sin\theta - \gamma R^2\dot{\theta}\,, \quad (2)$$

where γ is the drag force coefficient, which can be due to viscous friction or electromagnetic forces. If we assume the θ is small, then $\sin\theta \approx \theta$, and Eq. (1) can be re-written as

$$\ddot{\theta} + \frac{\gamma}{m}\dot{\theta} + \frac{g}{R}\theta = 0\,. \quad (3)$$

Comparing Eq.(3) to Eq. (1), we see that the pendulum will oscillate with some measurable natural resonance frequency and damping ratio.

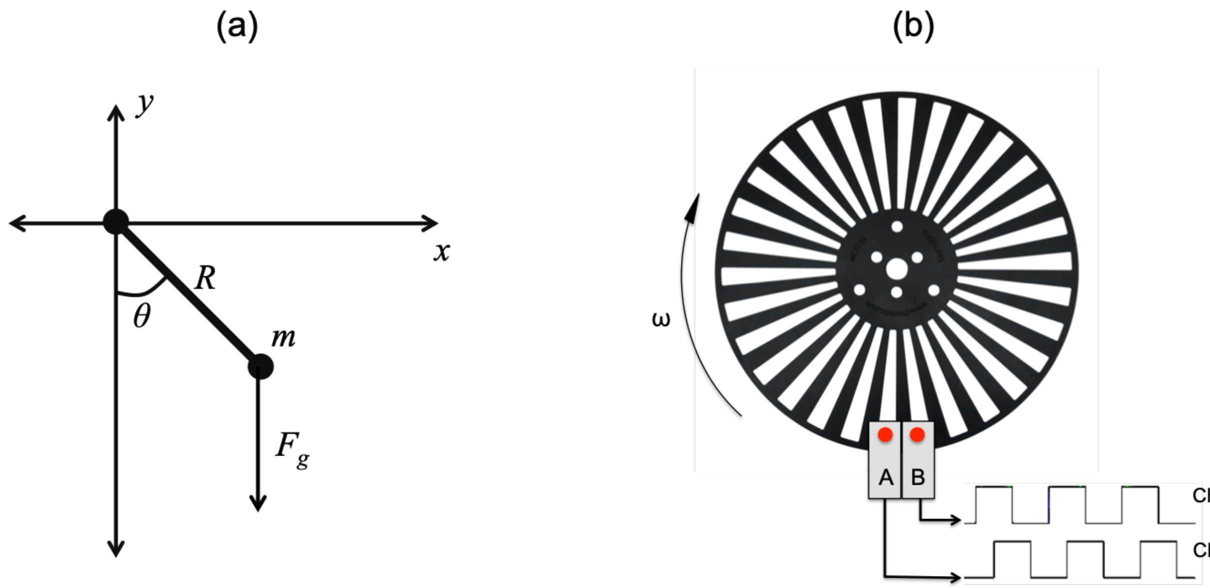

Figure 1 – (a) A free-body diagram for the pendulum can be used to derive the rotational equation of motion. (b) An encoder will be used to measure the pendulum angle. An optical encoder consists of a slotted wheel with two photogates. Alternatively, a magnetic encoder uses magnetic strips on the wheel with two Hall effect magnetic sensors.

Experimental Procedure

Measuring and controlling the angular position and velocity of rotating mechanical parts is often critical in many applications, especially robotics. You will use a digital sensor called a quadrature encoder and an Arduino UNO microcontroller to measure the angular position of the pendulum. An optical encoder contains a slotted wheel, which is mounted to the rotating shaft. Illustrated in Fig. 1b, the slotted wheel rotates through two photogates (similar to the Photogates in the inclined plane lab), and an electronic pulse train is produced. Alternatively, the wheel may have magnetized strips that rotate past two Hall effect magnetic sensors.

The Arduino microcontroller is used to count the number of voltage pulses, and the number of counts is proportional to the angle of rotation. The encoder you will use in this lab has 466 pulse counts per 360° of rotation.

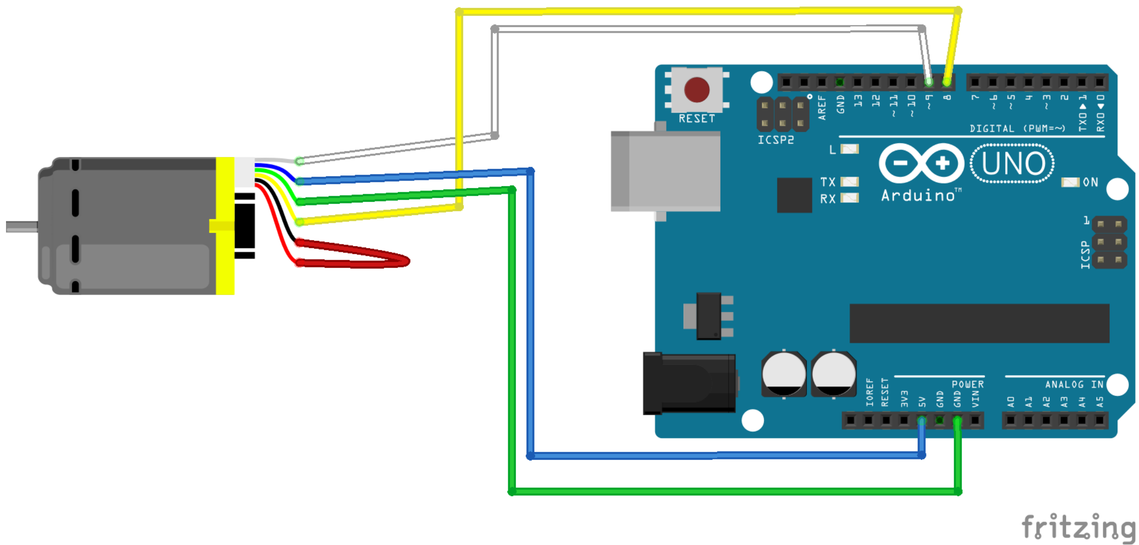

Figure 2 – A wiring diagram illustrating how to connect the encoder to the Arduino.

1. Make sure the pendulum mount is securely clamped to the edge of the lab bench.
2. Measure the distance from the shaft to the center of the brass weight at the bottom. Record the value in your lab notebook. Also, note that the pendulum has a mass m = 0.147 kg.
3. Connect the Arduino to the lab bench PC with the USB cable. A green light should appear and remain on the Arduino. If it does not, please ask the lab instructor for assistance.
4. Use jumper wires to connect the motor encoder wires on the back of the pendulum to the Arduino. Use the correct colored jumper wires. Ask the lab instructor for a wire if you are missing a color.
   a. GREEN → GND on Arduino
   b. BLUE → 5V on Arduino
   c. YELLOW → Digital pin 8 on Arduino
   d. WHITE → Digital pin 9 on Arduino
5. Use a single jumper wire to short circuit the red and black motor power wires, as illustrated in Fig. 2.
6. Download the Part 1 Encoder code template from the lab webpage. Right-click the link > "Save link as…", and save the code with an intelligent file name (i.e. "Encoder_yourName.ino"). Open it with the Arduino IDE software. Take a moment to admire the code and try to understand what it will do.
7. In the Arduino IDE software, check that the Quadrature Encoder and Enable Interrupt libraries have been added. Go to "Tools" > "Manage Libraries…" and search for "QuadratureEncoder". If it has not been installed, then click "Install". Do the same for "EnableInterrupt".
8. In the Arduino IDE software, go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.

9. Click the check mark at the top of the Arduino program to check the code for errors.
10. Press the arrow button to compile the program and send it to the Arduino.
11. Go to "Tools" > "Serial Monitor" (or press "Ctrl + Shift + M") to view the output from the "Serial.print()" commands, and set the baud rate to 9600.
12. You should see the time (s), encoder count, and angle (degrees) printed in the serial monitor in this order. The code will run for several seconds, then stop.
13. Let the pendulum hang at rest in its equilibrium position of $\theta = 0$. Clear the data in the serial monitor and press the reset button on the Arduino to restart the data collection. Move the pendulum and check to make sure the values are reasonable.
14. If you are confident the system is measuring angle correctly, clear the serial monitor, press the reset button on the Arduino, lift the pendulum up to about 45 degrees, and release it. This will give you data for the pendulum's response to setup input.
15. If you are confident you have good data, save the data.
    a. Highlight all of the data in the serial monitor up to time $t = 0$. (Make sure you are only selecting data from the most recent drop.)
    b. Press "Ctrl + c" to copy the data.
    c. Paste the data in a blank .txt file using notepad.
    d. Save the data as a .txt file with an intelligent file name.
16. Remove the jumper wire that is short circuiting the red and black motor power wires. Repeat the measurement for the response to a step input with the motor no longer short-circuited. Save the data with an intelligent file name.

Data Processing – System Identification and Characterization

The pendulum's radius of gyration R and the drag force coefficient γ are often difficult to calculate theoretically. It is much easier to extrapolate them from measured data. To do this, the pendulum is subject to an impulse (which you just did). For a single pendulum subject to an impulse, the theoretical solution to Eq. (3) is

$$\theta(t) = \theta_0 e^{-\lambda t} \cos(\omega_d t) \qquad (4)$$

where θ_0 is the initial displacement, $\lambda = \gamma/2m$ is the decay constant, and $\omega_d = \sqrt{\frac{g}{R} - \lambda^2}$ is the ringing frequency. The parameters λ and ω_d can be extracted from the measured angle θ (degrees) vs. time t (s) data. The viscous drag force coefficient γ and radius of gyration R can then be determined using the mass of the pendulum $m = 0.147$ kg. The natural resonance frequency can then be calculated using

$$\omega_n = \sqrt{\frac{g}{R}}, \qquad (5)$$

and the damping ratio can be calculated using

$$\zeta = \frac{\lambda}{\omega_n} \ . \tag{6}$$

Part II: Response to an Oscillating Force

You will now use the brushed DC motor to drive the pendulum with a torque that oscillates sinusoidally with a driving frequency ω. The governing equation for the angular displacement as a function of time becomes

$$\ddot{\theta} + \frac{\gamma}{m}\dot{\theta} + \frac{g}{R}\theta = \frac{|\tau_m|}{mR^2}\sin(\omega t). \tag{7}$$

where $|\tau_m|$ = 0.112 Nm. is the maximum torque (peak amplitude) applied by the motor. After several cycles of oscillations, the pendulum will oscillate at the driving frequency with some amplitude and phase shift, as predicted by the solution to (7)

$$\theta(\mathrm{t}) \ = \ \frac{|\tau_m|\sin(\omega \mathrm{t}+\phi)}{mR^2\sqrt{\left(\omega_n^2-\omega^2\right)^2-(2\zeta\omega)^2}}, \tag{8}$$

with units of radians. The oscillations will be phase shifted from the sinusoidal torque input by an angle in radians

$$\phi = \begin{cases} arctan\left(\frac{2\zeta\omega\omega_n}{\omega_n^2-\omega^2}\right) & \text{if } \omega < \omega_n \\ -\pi - arctan\left(\frac{2\zeta\omega\omega_n}{\omega_n^2-\omega^2}\right) & \text{if } \omega > \omega_n \end{cases}. \tag{9}$$

Experimental Procedure

You will now implement an H-bridge motor driver board that will modulate the electric power and direction of current flowing through the motor. Similar to the heater power modulation circuit, this system will use a PWM signal from the Arduino UNO with four transistors connected in an H-bridge configuration to amplify the signal.

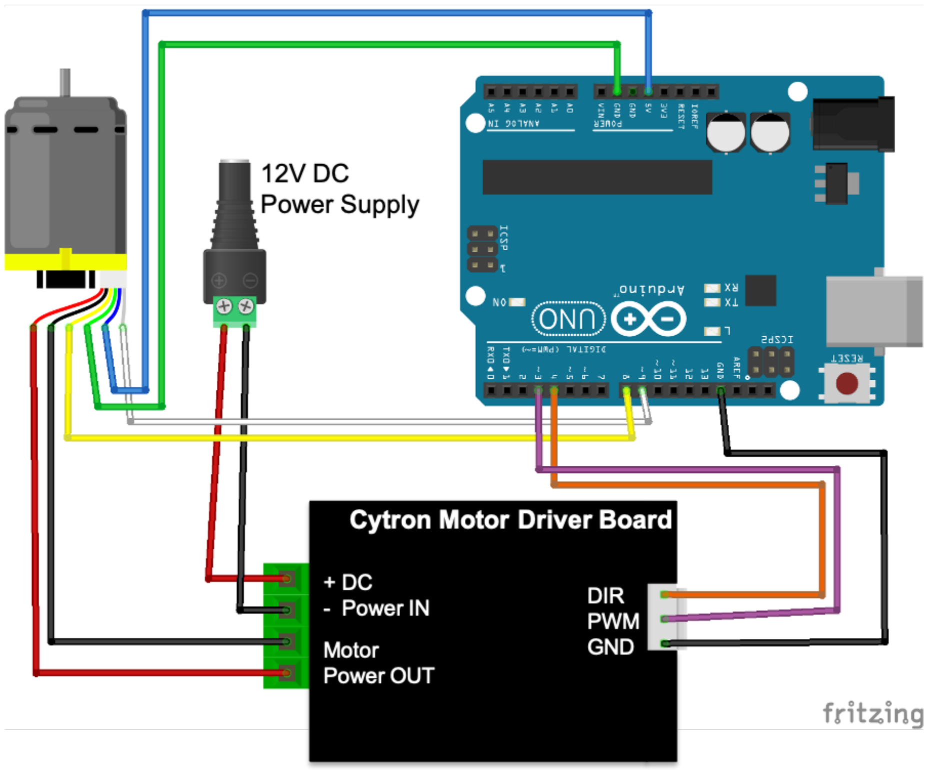

Figure 3 – A wiring diagram illustrating how to connect the Cytron MD10C H-bridge motor driver board to the Arduino and electric motor.

1. Make sure the Arduino is turned off and disconnected from any power source including the USB cable.
2. Make the following connections with the Arduino and Cytron motor driver:
    a. Arduino GND → Cytron GND pin
    b. Arduino Digital Pin 3 → Cytron PWM pin
    c. Arduino Digital Pin 4 → Cytron DIR pin
3. Use thick 20 gauge wire to connect the green "POWER" screw terminals on the Cytron board to the 12V DC power supply. Make sure you use the kill switch. Twist and straighten the ends of the wires to ensure there are no stray "whiskers" poking out.

4. Connect the black “MOTOR” screw terminals on the Cytron board to the red and black wire on the DC motor.
5. Download the Part 2 Power Modulation code from the lab webpage. Right-click the link > “Save link as…”, and save the code with an intelligent file name (i.e. “Modulation_yourName.ino”).
6. Upload it to the Arduino. Turn on the DC power supply, and the pendulum should oscillate under the impetus of oscillating motor torque. The code will run for several seconds, then stop. Clear the data in the serial monitor and press the reset button on the Arduino to restart the data collection.
7. Open the serial monitor. You should see the time (s), angle (degrees), input torque (Nm), and PWM strength printed in the serial monitor in this order. Compare the values of the angle in the serial monitor to the motion of the pendulum and check that they seem reasonable.
8. If you are confident the system is measuring angle correctly, clear the serial monitor, press the reset button on the Arduino. This will give you data for the pendulum’s response to an oscillating force input.
9. When you are confident you have good data, save the data.
   a. Highlight all of the data in the serial monitor up to time $t = 0$. (Make sure you are only selecting data from the most recent drop.)
   b. Press “Ctrl + c” to copy the data.
   c. Paste the data in a blank .txt file using notepad or some other text editor.
   d. Save the data as a .txt file.
10. Email the data files to your lab partner.

Data Analysis and Deliverables

Create plots and other deliverables listed below, and include them in a tech memo.

1. For the response to an impulse, plot of the measured angle θ (degrees) vs. time t (s) for the pendulum with the motor short circuited and with the motor in an open circuit configuration (both on the same plot).
2. Make a table with the following parameters for the *short-circuited* pendulum's response to a step input. These parameters should be extrapolated from the measured angle θ (degrees) vs. time t (s) data from Part I.
   a. The half-life of the oscillations τ_{half} (s).
   b. The decay constant λ (1/s).
   c. The ringing frequency ω_d (rad/s).
   d. The radius of gyration R (m).
   e. The damping ratio of the pendulum ζ.
3. For the pendulum driven by an oscillating torque, make a single plot of *both* the motor torque (Nm) and measured angle (degrees) vs. time (seconds), similar to the "oscillating heat source" in the 1st order response lab.
   a. Show 2 or 3 periods of steady-state oscillation toward the end of the data set. (Do not show the initial transient.)
   b. Use the "yyaxis left" command to plot the applied torque on the left axis.
   c. Use the "yyaxis right" command to plot measured angle on the right axis.
   d. Make sure it is clear to the grader which time trace corresponds to which y-axis.
   e. Make sure both signals are vertically centered, such that they both oscillate about the middle of the plot. (Similar to how you centered the sine waves on the oscilloscope, which made it easier to determine the phase.)
4. Make a table containing the following parameters for the driven pendulum.
   a. The theoretical amplitude of the oscillations $|\theta|$. See Eq. (8), and think about it.
   b. The measured amplitude of the oscillations determined from your plot.
   c. The theoretical phase given by Eq. (9).
   d. The measured phase determined from the time lag in your plot $\phi = -\omega\Delta t$.

Talking Points (continued on next page) …

Talking Points – Please address the following questions in your paragraphs.

- Does the radius of gyration R seem reasonable for the single pendulum?
- Does the pendulum behave differently when the motor is short-circuited?
- Based on the measured damping ratio, was the pendulum underdamped or overdamped?

Appendix A

Equipment

Part I (set up on inner lab benches)

- Blue pendulum apparatus
- Bag containing:
    - 20 gauge black and red wires
    - Dupont pin jumper wires
    - Cytron MD10C motor driver board
- C-clamp
- Ruler
- 12V DC power supply w/ kill switch and screw terminal adapters
- Arduino UNO microcontroller
- USB cable to connect Arduino to workstation PC