Experiment C8
Pendulum with Active Damper
Procedure

Deliverables: Checked lab notebook, Technical Memo

Overview

Many mechanical systems have a tendency to oscillate or vibrate, making them unstable. Unwanted mechanical oscillations can be mitigated by incorporating an active damper. An active damper system uses an actuator to generate momentum equal and opposite to the measured oscillatory motion. Common examples of active dampers include active suspensions on cars, aircraft flight control systems, and noise cancelling headphones.

In this lab, you will use a reaction wheel to dampen the oscillations of a pendulum. Shown below in Fig. 1, the angular displacement $\boldsymbol{\theta}$ and angular speed $\boldsymbol{\omega}$ of the pendulum are measured using an encoder. Feedback from the encoder will be used to adjust the torque on the reaction wheel, causing the wheel to accelerate equal and opposite the angular momentum of the pendulum, quickly damping its oscillations.

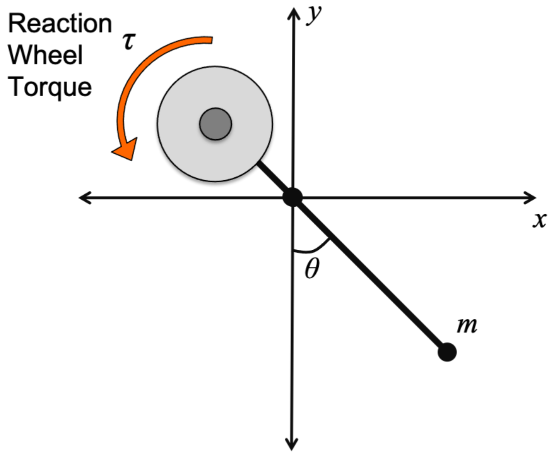


Figure 1 – A free body diagram shows the forces acting on a pendulum. A gravitational force and viscous drag force cause the pendulum naturally oscillate. A DC motor is used to apply a torque τ to the reaction wheel, thus changing the angular momentum of the pendulum.

Balancing the angular acceleration with the torque from gravity, the viscous drag force of air, and the motor yields the equation of motion for the single pendulum

$$mR^2\ddot{\theta} = -mgR\sin\theta - \gamma R^2\dot{\theta} + \tau \,, \qquad (1)$$

where m is the pendulum mass, R is the radius of gyration, γ is the viscous drag force coefficient, and τ is the motor torque applied to the reaction wheel.

The solution to Eq. (1) for an "underdamped" pendulum is

$$\theta(t) = e^{-\lambda t}\sin(\omega_d t), \tag{2}$$

where the decay constant $\lambda = \gamma/2m$ and the ringing frequency $\omega_d = \sqrt{g/R - \lambda^2}$.

Note that the pendulum is essentially a damped harmonic oscillator with a natural resonance frequency $\boldsymbol{\omega_n}$ and damping ratio $\boldsymbol{\zeta}$ which depend on the mass distribution of the system. If the damping ratio is much smaller than 1, the pendulum will oscillate for a very long time. The active damper will compute the motor torque from measured angular displacement and speed using the proportional-derivative feedback

$$\boldsymbol{\tau} = \boldsymbol{k_p}\boldsymbol{\theta} + \boldsymbol{k_d}\boldsymbol{\omega}, \tag{3}$$

where k_p and k_d are the feedback gains. Choosing appropriate values of the gains k_p and k_d will yield a new damping ratio $\boldsymbol{\zeta'}$ that is close to 1.

Subsystem A: Angle Encoder

You will begin by mounting, wiring, and calibrating the angle encoder to measure the pendulum's angular displacement and speed.

Procedure

1. Mount the encoder and the f mechanical components. Refer to the photos in Appendix B.
    a. Mount the angle encoder to the bracket using the three screws.
    b. Attached the bracket to the 2040 rail using M5 T-nuts, screws, and washers. Then, clamp the 2040 rail to the lab bench, as pictured in Appendix B.
    c. Securely fasten one of the 6mm GoBilda shaft couplings to the encoder shaft.
    d. Use M4 screws to attach the 5-hole GoBilda channel to the shaft coupling, centered on a large hole near the edge.
2. Make sure the Arduino is turned off and disconnected from any power source including the USB cable. Carefully mount the 3-axis encoder shield to the Arduino. Make sure the pins are well aligned.
3. Refer to the pin-out on the side of the encoder. Use female-female jumper wires to connect CHAN A, CHAN B, Vcc +5V DC, and ENC GND to the Arduino encoder shield. Make sure the colors match.
4. Download the "AxisEncoderShield3.ino" code from the C8 website, and save it to your C8 folder with the name "C8_encoder_calibration.ino".
5. Open the code and look at the "setup". Note that it uses pins 8, 9, and 10, so you will not be able to use these in the programs you are about to write. All other pins are available.
6. Connect the Arduino to the lab computer, select the appropriate COM port, and load the encoder shield program to the Arduino.

7. Open the "Serial Monitor" to see the output displayed as text. Rotate the encoder shaft. You should see the encoder count change for one of the three channels. Which encoder channel is the encoder connected to? X, Y, or Z?
8. In the main loop of the sketch, delete the encoder measurements and subsequent "serial.print()" functions for the two encoder channels that are not connected. Also, delete any "delay()" functions in the main loop.
9. Open the "Serial Plotter" to see the output displayed as a graph. Rotate the encoder shaft. You should see the line on the graph change.
10. Calibrate the encoder. With the code running and the Serial Monitor open, rotate the encoder shaft one full revolution (360°). How many encoder counts does this yield? Record the value in your lab notebook.
11. Modify the main loop of the code to print the time in ms and angle in degrees instead of the number of encoder counts. Save a copy of the code to your code library, and give it an intelligent file name.
    a. It should print in the format "time, angle". This will make it easier to import your data into Matlab.
    b. It should start at 0° when hanging vertically downward, increase when rotated counter-clockwise, and decrease (give negative angles) when rotated clockwise.
    c. Use the millis() function to get the time and store it as a float point variable. Convert the units from milliseconds to seconds.
    d. You may need to put a nested loop inside the main loop.
12. Test the sub-system to make sure it works. **Demonstrate it to the lab instructor or TA to receive credit on your score sheet.**

Subsystem B: DC Motor Control

You will now interface the DC motor with the Arduino using a Cytron MD10C single channel motor driver board.

Procedure

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 2 → Cytron DIR pin
3. Use thick 18 or 20 gauge wire to connect the green "POWER" screw terminals on the Cytron board to the large, regulated 12V DC power supply.

4. Use thick 18 or 20 gauge wire to connect the black "MOTOR" screw terminals on the Cytron board to the red and black wire on the DC motor. (If the motor wires have banana plugs, cut them off and strip ~5mm of insulation off the wires.)
5. Download the Cytron motor template code from the lab webpage. Upload it to the Arduino. Turn on the DC power supply, and the motor shaft should rotate slowly. Try increasing the PWM value in the "TurnForward()" function, and the shaft should spin faster.
6. Modify the Cyton motor template code to do the following.
   a. Write a third function TurnReverse(int x) that works just like TurnForward(int x), but turns the shaft in the opposite direction.
   b. Modify the main loop to turn the motor forward at 50% PWM power for 4 seconds, then stop and turn the other direction at 50% PWM power for 4 seconds. Then repeat.
7. Test the sub-system to make sure it works. **Demonstrate it to the lab instructor or TA to receive credit on your score sheet.**

Subsystem C: Mechanical Assembly

You will now mount the motor, reaction wheel, and a counter-weight to the pendulum, as pictured in Appendix B.

Procedure

1. Make sure the Arduino is turned off and disconnected from any power source including the USB cable. Make sure the DC power supply is off, as well.
2. Mount the counter-weight just below the encoder shaft, as pictured in Appendix B.
   a. Use M4 bolts to attach the 2-Post Clamping Mount (12mm Bore) to the 5 hole GoBilda U-channel.
   b. Securely clamp the 12mm aluminum tube into the 2-Post mount.
   c. Slide the large weight onto the aluminum tube. Position it about 7mm below the encoder shaft, and tighten the thumb screw.
3. Mount the DC motor to the 5 hole GoBilda U-channel using M4 screws.
4. Securely fasten the other 6mm shaft coupling to the motor shaft.
5. Use M4 bolts to attach the aluminum disc to the shaft coupling.
6. The counter-weight should hang down below the encoder shaft. If it does not, loosen the thumb screw, and lower it so it does.
7. Give the pendulum a tap, and it should oscillate with a period of about 1 second.
8. **Demonstrate it to the lab instructor or TA to receive credit on your score sheet.**

Subsystem D: Data Collection and Processing

You will now use the encoder data to compute the angular speed. The data will be recorded to determine the transient response characteristics of the pendulum.

Procedure

1. Open the encoder code from earlier, and save it with a new name. Use the measured angle and time data to compute the angular speed using a first-order finite difference formula

$$\omega \approx \frac{\theta - \theta_{prev}}{t - t_{prev}},$$

   where θ_{prev} and t_{prev} are the values from the previous iteration of the main loop. (Use degrees/second for the angular speed.)
2. Add more print statements to print the encoder data in the form "time, theta, omega".
3. Lift the pendulum to an initial angle of about 90°, and let it go.
   a. Record the oscillation of θ vs. time. You can copy and paste the data from the serial monitor into a .txt file, or you can try to directly save it using PuTTY.
   b. Import the data into Matlab and plot it with a trace of both angle and angular speed on the same plot.
   c. If the data looks good, save the workspace as a .mat file.
   d. **Show your Arduino code and plot of theta and omega to the lab instructor or TA to receive credit on the score sheet.**

Design Challenge 1 – Proportional Feedback

You will now use the measured angle θ to compute the direction and PWM power at which the motor will turn the reaction wheel.

- Combine your codes from the previous sub-systems into a code that will measure the angle θ, then us the measured value to compute the motor PWM value using the proportional feedback formula $PWM = k_P\theta$, where k_p is the proportional gain.
- Estimate a value for the proportional gain k_p to be the maximum PWM value of 255 divided by the maximum angular displacement $\Delta\theta_{\max}$.
- Use if-then statements to set the direction that the reaction wheel will turn, based on whether the calculated PWM value is positive or negative. (Think about it. Trial-and-error it. Make it work.)
- Try different values of k_p until you find one that quickly dampens the oscillations.
- When you are satisfied with the damper's performance, save the angle vs. time data from the Arduino serial monitor or PuTTY. **Demonstrate the system to the TA to receive credit on the score sheet.**

Design Challenge 2 – Proportional-derivative Feedback

You will now use the measured angle θ and angular speed ω to compute the direction and PWM power at which the motor will turn the reaction wheel.

- Compute the motor PWM value using the proportional-derivative feedback formula

  $PWM = k_P\theta - k_d\omega$, where k_p is the proportional gain and k_d is the derivative gain.
- Estimate a value for the derivative gain k_d to be the maximum PWM value of 255 divided by the estimated maximum angular speed ω_{max}.
- Use if-then statements to set the direction that the reaction wheel will turn, based on whether the calculated PWM value is positive or negative. (Think about it. Trial-and-error it. Make it work.)
- Try different values of k_p and k_d until you find values that quickly dampens the oscillations.
- When you are satisfied with the damper's performance, save the angle vs. time data from the Arduino serial monitor or PuTTY. **Demonstrate the system to the TA to receive credit on the score sheet.**

Clean-up

To receive full credit, you must return the lab bench to its initial state:

- Unplug the DC power supply. Remove the kill switch and screw terminal adapter.
- Disassemble the circuit. Disconnect the wires from the Cytron board, Encoder shield, and Arduino.
- Remove the encoder shield from the Arduino.
- Put anything that belongs in your tool kit back into the tote bin.
- Disassemble the pendulum and reaction wheel. Leave the mounting bracket on the encoder. Put it all back in the plastic bag.

Data Analysis and Deliverables

Using LaTeX or MS Word, make the following items and give them concise, intelligent captions. Make sure the axes are clearly labeled with units. Plots with multiple data sets on them should have a legend. **Additionally, write several paragraphs describing the plots/tables. Any relevant equations should go in these paragraphs.**

IMPORTANT NOTE: Add a horizontal line denoting the set-point whenever it is applicable.

1. A plot with the following angle vs. time traces (three traces on the same plot):
   a. The oscillation of the pendulum without any feedback. (Show 4 full periods of oscillation.)
   b. The oscillation of the pendulum with the active damper (either proportional or proportional-derivative feedback, whichever you like better).

Talking Points – Discuss these in your paragraphs.

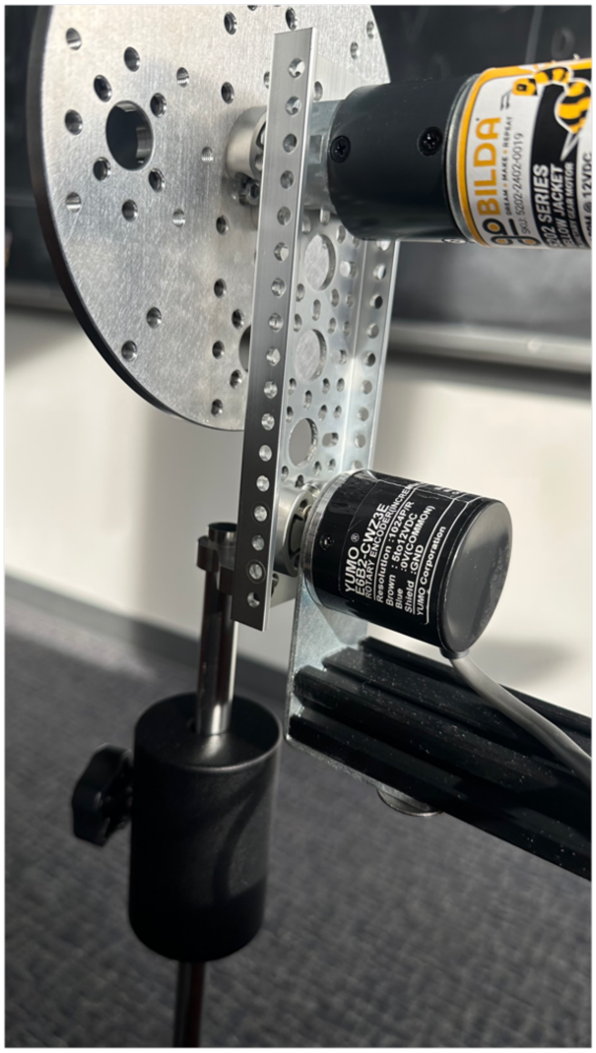
- Include at least one relevant equation.
- Comment on the effectiveness of the active damper.

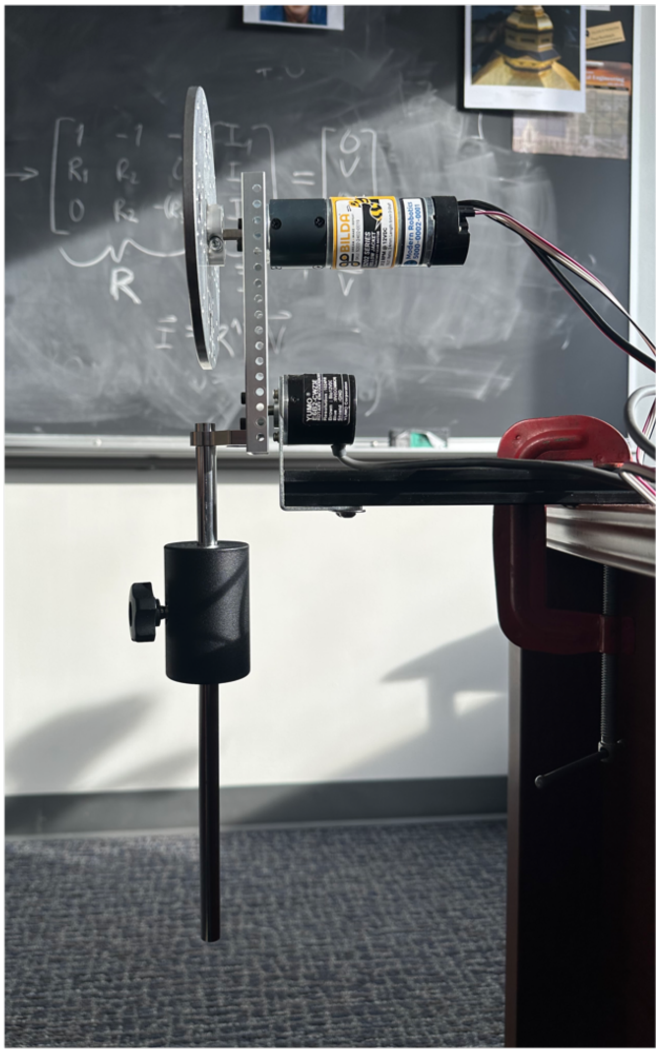
Appendix A

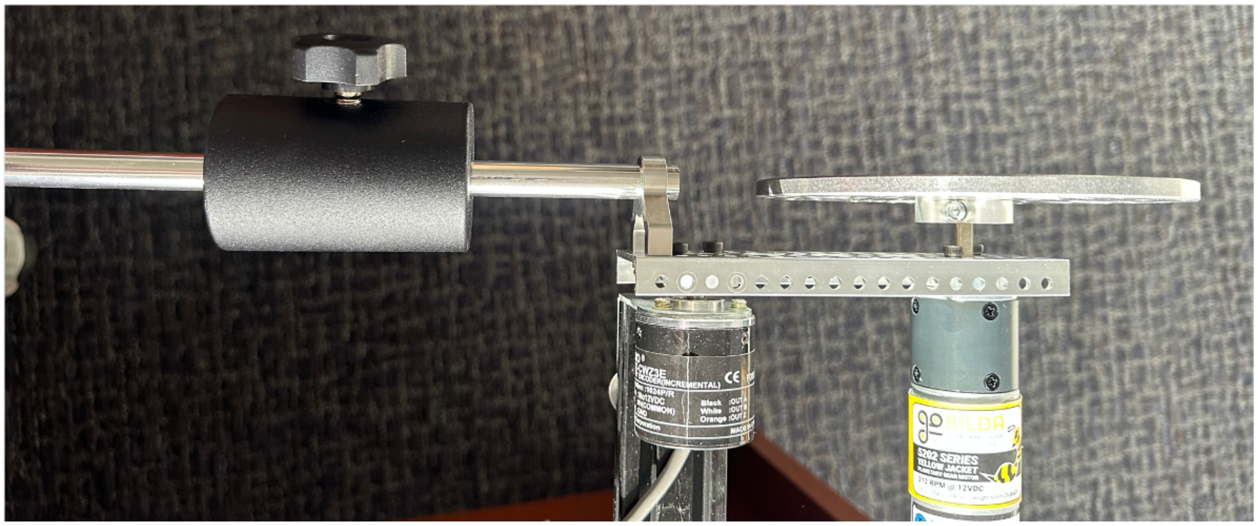
Equipment

- Standard Mechatronics lab kit in small tote (above lab bench)
- 40-20 Rails
- C-clamp
- Regulated 12V DC Power Supply
- 6mm Shaft Rotary Encoder 1024 P/R w/ mounting bracketSKU: RB-Spa-1042
- RoboGaia 3 Axis Encoder Counter Arduino Shield (SKU 00021)
- Cytron MD10C motor driver board
- 5202 Series Yellow Jacket Planetary Gear Motor, 19.2:1 Ratio, 24mm Length 6mm D-Shaft, 312 RPM
- GoBilda 1121 Series Low-Side U-Channel (5 Hole, 144mm Length)
- 1309 Series Sonic Hub (6mm Bore) – SKU 1309-0016-0006
- 1140 Series Aluminum Baseplate (6mm Thickness, 144mm Diameter)
- 1400 Series 1-Side, 2-Post Clamping Mount (12mm Bore)
- 4100 Series Aluminum Tube (10mm ID x 12mm OD, 300mm Length)
- Sky-Watcher S20540 Star Adventurer Counter Weight Kit

Appendix B


goBILDA
5202 SERIES
YELLOW JACKET
YUMO®
E6B2-CWZ3E
ROTARY ENCODER(INCREMENTAL)
Resolution :1024P/R
Brown :5to12VDC
Blue :0V(COMMON)
Shield :GND
YUMO Corporation




E6B2-CWZ3E
ENCODER(INCREMENTAL)
goBILDA
5202 SERIES
YELLOW JACKET
312 RPM @ 12VDC