Robolympics Final Competition

EGR 345 F22

Introduction

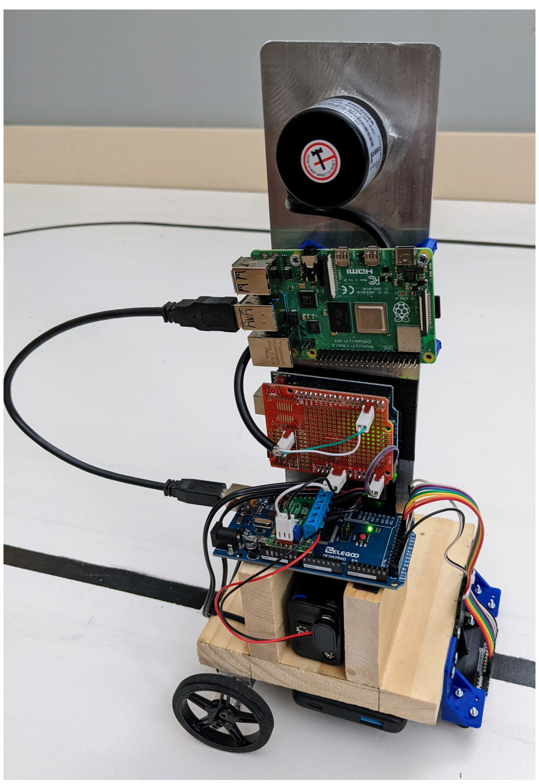
For the final project in EGR 345 you will program a pendulum/cart system to compete in a series of three events:

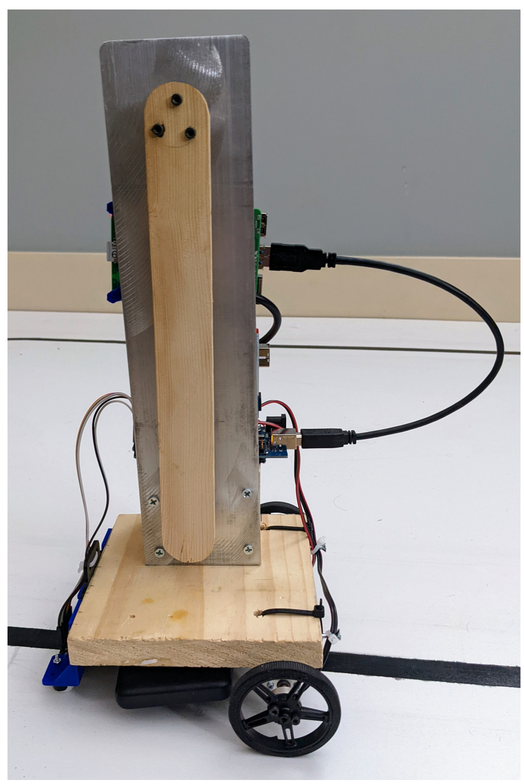
1. vibration suppression of the pendulum on the cart
2. completing three laps around the oval track while suppressing vibration of the pendulum
3. completing as many laps as possible (no time limit) while balancing the pendulum in the vertically up position

Your team's score for the final project competition will depend on how your robot performs in the three separate events.

System Description

The robot has two DC motors powered by an H-bridge. The sensors include an encoder to sense the pendulum position, and a line sensor array. An Arduino Uno is dedicated to reading the pendulum encoder. An Arduino Mega handles the H-bridge that accuates the motors and the line sensor. The main control is done by a Raspberry Pi that communicates with the Uno and Mega via i2c.





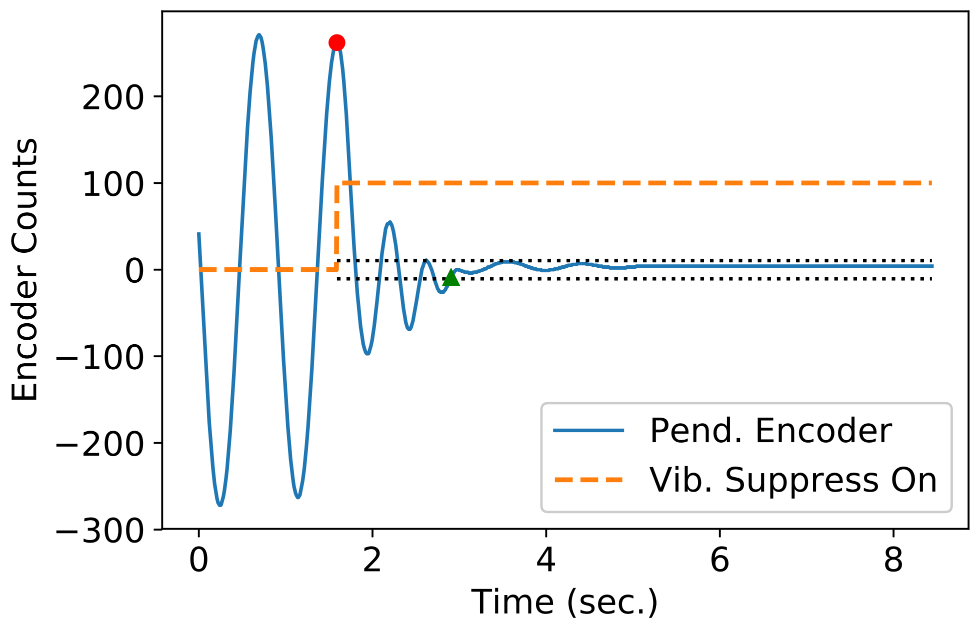
Event 1: Vibration Suppression

For the vibration suppression event, your instructor will flick the pendulum and start it swinging. Your team will then start the test by giving a command to the Raspberry Pi at the terminal. Your code must collect at least one second of data before switching on the vibration suppression control. Additionally, your vibration suppression must switch on before the encoder peaks fall below 75 counts.

Note: the cart cannot move until the vibration suppression control turns on. Your block diagram must print a variable that indicates when the vibration suppression control kicks on, in addition to the encoder data. Your score for this event will depend on the settling time that will be measured from when the vibration suppression control turns on. Settling time will be determined by finding the largest encoder value after the vibrations suppression control turns on. Your pendulum has settled when it stays within ±4% of that largest encoder value.

Note: For this event, your cart must come to rest within ±4 in of where it started the event.

Example Vibration Suppression Settling Time Graph



Recommended steps:

- Perform a swept sign test to determine the transfer function between voltage to the motors and the pendulum encoder output
- Use root locus to design the vibration suppression controller (semi-optional)

Event 2: Vibration Suppression + Line Following

For the third event, the robot will follow the line and complete three laps as quickly as possible while also suppressing pendulum vibrations. The pendulum encoder counts must remain with +/- 30 counts for the entire event (all 3 laps). If either wheel touches the line, your run is disqualified.

Event 3: Balancing the Pendulum in the Vertically Up Position While Completing as Many Laps as Possible

Complete as many laps as you can while balancing the pendulum in the vertically up position (i.e. the unstable position). There is no time limit for this event. The event ends when your pendulum falls below horizontal.

Event 3B: Balancing the Pendulum Vertically Upward without Line Following

Your team can choose to compete in event 3B instead of event 3. For 3B, the goal is simply to balance the pendulum in the up position for as long as possible. The amount of time that you balance the pendulum will be measured with a stopwatch. The primary difference between events 3 and 3B is that teams competing in 3B are not attempting to follow the line. For event 3B, the pendulum must start with a slight angle - you are not allowed to perfectly balance the pendulum and rely on friction to hold it in that position.

Word of Caution

The robots may behave somewhat differently going forward versus going backwards. The idea behind following the oval while balancing is that the robot can go one direction indefinitely.

Final Report Expectations

- For event 1: describe how you used the encoder and/or gyroscope for cornering
- For the vibration suppression control for event 2:
  - Discuss your control design approach and implementation
- For event 3:
  - describe your overall approach
  - describe your control design approach
    - if you simply used PID tuning, how do you know it is done correctly?
    - if you used root locus and/or Bode, present your work and describe your design goals
- Discuss anything noteworthy in your control design for any of the events
- Discuss any novel aspects of your software
- Simulation and modeling:
  - discuss any ways you used Python to simulate any aspect of the project
  - compare your simulation results with experiments