ME 456 Mechatronics

Workshop Assignment #2: PWM Speed Control and Ultrasonic Radar

25.03.2025

Goal:

This assignment is about familiarizing with PWM speed control of motors, and using ultrasonic sensor feedback to move the robot and control the distance to a moving object.

Prelab:

- Our goal is to adjust speed of motors and achieve bi-directional (forward-reverse) speed control. Familiarize with PWM motor speed control methods. The following tutorials can be helpful:
  - L9110 with Arduino: https://www.youtube.com/watch?v=YkfBtjs8uWgv
  - L293 with Arduino: https://www.youtube.com/watch?v=0XihwdulmDQ
  - Arduino motor shield: https://youtu.be/YIGb0Ll6Y7I

Lab Tasks [10 pts]:

1. **PWM Speed Control [+5 pts]:** Using PWM speed control, develop a code to run the motors to follow the following sequence of motion commands:

Step	Duration	Left motor	Right motor
1	1 seconds	20% full speed forward	20% full speed forward
2	1 seconds	40% full speed forward	40% full speed forward
3	1 seconds	60% full speed forward	60% full speed forward
4	1 seconds	80% full speed forward	80% full speed forward
5	1 seconds	100% full speed forward	100% full speed forward
6	1 seconds	Stop	Stop
7	1 seconds	20% full speed reverse	20% full speed reverse
8	1 seconds	40% full speed reverse	40% full speed reverse
9	1 seconds	60% full speed reverse	60% full speed reverse
10	1 seconds	80% full speed reverse	80% full speed reverse
11	1 seconds	100% full speed reverse	100% full speed reverse
12	1 seconds	Stop	Stop

2. **Proportional Position Control [+5 pts]:** Create a program loop which will move the robot at a certain distance (reference value, example 50 cm) from the object in front of it. Calculate a distance error (error = reference – measured distance) and employ “proportional position control” as follows:

$$error = ref - dist$$

$$speed = K_p * error$$

If the error is large, the robot moves with a larger speed. As it gets closer, the speed is also reduced, until it stops within a specified distance to the target. To achieve proportional speed

control your code must set speed to some proportional constant (K_p) times the distance error. Try different constants (1-25) and discuss their effects.

<u>Homework and Reporting [15 pts]:</u> Arduino-Based Mobile Robot Radar

This homework aims to use an **Arduino-based mobile robot** as a radar system utilizing an **ultrasonic sensor (HC-SR04)**. The robot will rotate, collect distance measurement data, and transmit it via serial communication to a connected computer. The received data will then be plotted to visualize the surroundings.

1. **<u>Homework Steps [+ 10 pts]:</u>** To begin, the ultrasonic sensor should be mounted on the robot. If necessary, a servo motor can be used to allow the sensor to scan independently of the robot's movement. The sensor should be connected to the Arduino, with the trigger and echo pins assigned to digital input/output pins. The motors must also be connected to the motor driver module, ensuring proper power and control wiring.

Next, the Arduino should be programmed to rotate the robot in small increments, such as 5 to 10 degrees per step, while triggering the ultrasonic sensor to measure distances. The data, consisting of angle and distance values, will be transmitted via serial communication to a connected computer. If a servo motor is used for scanning instead of rotating the entire robot, the program should control the servo accordingly to capture distance data at different angles. Position referencing while turning the robot could be challenging. To address this, one of the following methods can be used: (1) calibrating the rotational speed of the robot to estimate angles based on time, or (2) using a reference mark, such as placing a nearby object (~10 cm) as a zero-degree reference point.

On the computer side, a Python script using Matplotlib and PySerial can be used to read the incoming serial data and plot the surroundings in a polar or 2D scatter plot. The visualization should update in real-time as new data is received. As alternatives, MATLAB can be used for data processing and plotting, or Excel can be used to create static plots from logged data. Processing can also be used for an interactive graphical representation. To ensure accuracy, adjustments can be made to the scanning increments, and noisy data can be filtered using median filtering or averaging techniques.

Finally, testing should be conducted in various environments to refine the scanning and data visualization process. Enhancements can include using multiple ultrasonic sensors for faster scanning, implementing object recognition, or integrating wireless communication through Wi-Fi or Bluetooth. More advanced applications could involve incorporating SLAM algorithms for precise mapping.

2. **<u>Submission Requirements [+5 pts]:</u>** Prepare a short report with your codes and results. Include a discussion on the robot’s performance, detailing any challenges encountered and possible improvements. Additionally, submit a video of your robot in operation as part of your final submission.