Nathan Anderson
Indela
1221689573
Mar 15, 2024

Project 4: Maintain Speed

Description:

The goal of project 4 is to build upon the foundation laid in Project 3 by introducing encoders and a PID control system. The objective of this project is to achieve two key functionalities: precise robot positioning and consistent speed regardless of the surface (carpet, tile) the robot traverses. To demonstrate the PID control system's ability to maintain a consistent speed, the robot is programmed to wait for switch 1 to be pressed and then, after a 2 second delay, drive in a straight line for 6 seconds. This path will be repeated on different surfaces and the robot should drive the exact same distance each time.

Implementation:

Similar to Project 3, the hardware platform utilizes the FRDM KL46Z microcontroller, motors, motor driver, and other components from the provided robot kit. This project introduces encoders, one for each motor, to gather precise speed and position data. The software architecture leverages a modular approach to promote code organization, reusability, and simplify debugging. The configuration for each encoder is stored in an encoder struct which is then passed to an initialization function. This reusability makes debugging a much quicker process. Once configured, each encoder will cause interrupts to occur for its port. In the interrupt handler, each encoder struct is passed to a handleInterrupt function where if both a and b signals have been captured, the time taken for each is stored from the timer, the speed and rotation count is updated in the encoder struct, and the timer is reset. After the encoders have been updated, the two PID controller structs are passed into a PIDTick function. This module implements a PID control loop, a core component for achieving consistent speed. It receives the desired speed as a setpoint and compares it to the actual speed obtained from the encoders (feedback). The calculated error (difference between desired and actual speed) is used to adjust the PWM signal sent to the motors. By regulating the motor speed based on the error signal, the PID control system strives to maintain a consistent travel distance even on varying surfaces. The PID controller struct stores the constants required, the integral value, and the previous error for the derivative factor. The main function serves as the central control unit, initializing the system, including motor driver, encoder connections, and PID control parameters. It orchestrates the robot's behavior through the following sequence. Waits for a switch press (SW1) to initiate the movement, introduces a 2-second delay to allow for user preparation, sets the setpoint for the PID control loop, commanding the robot to move in a straight line for a predetermined duration (e.g., 6 seconds), and after the designated time elapses, stops the robot.

The pseudo code for the main functions are shown below:

```
PortAIRQ()
    Float delta
    handleEncoderInterrupt(lEncoder, &delta)
    if(lEncoder updated)
        PIDTick(lPID,rEncoder.speed, delta)
    handleEncoderInterrupt(rEncoder)
    if(rEncoder updated)
        PIDTick(rPID, rEncoder.speed, delta)

handleEncoderInterrupt(encoder, *delta)
    if(encoderAFlag)
        aTime = timerCount
    if(encoderBFlag)
        bTime = timerCount
    if(aTime && bTime)
        *delta = max(aTime,bTime) * (1/timerFrequency)
        Encoder.speed = (1/encoder.divisions)*360/delta
        resetTimer
```

```
PIDTick(pid, value, delta)
    error = pid.desired - value
    pid.intergral += error*delta
    P = pid.kp * error
    I = pid.ki * integral
    D = pid.kd * (error-pid.prevError)
    pid.prevError = error
    Return P + I + D

configureEncoder(encoder)
    configureTPM
    configureGPIO(A)
    configureGPIO(B)
    startTPM
```

Results:

The completed robot exhibits consistent speed on various surfaces due to the PID control system's ability to adjust motor power based on encoder feedback. The robot traveled very similar distance in 6 seconds even when adding additional load and when driving on other surfaces.

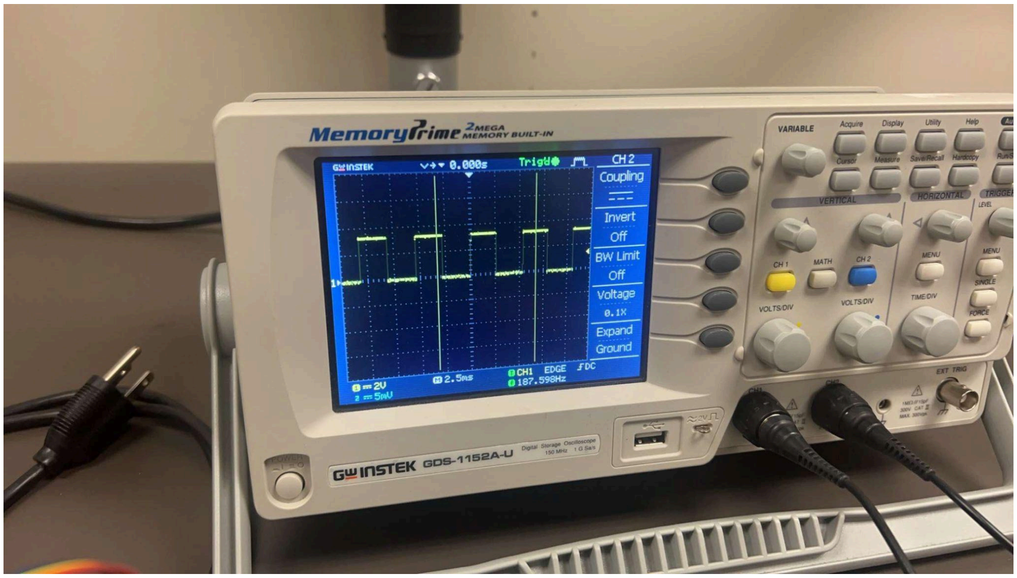

Image of encoder output on an oscilloscope

Demo Video Here:

demo.mov demo_with_load.mov

Discussion:

Project 4 represents a significant advancement over Project 3. It delves deeper into control system design principles and introduces sensor integration for feedback control. Challenges encountered included; fine-tuning the PID control parameters for optimal performance across diverse surfaces. I had to try a bunch of different values for the parameters and test the robot's response over and over again. It was especially difficult because all that I had to go off of was how it reacted to me pressing on the wheel or driving on uneven surfaces. Ensuring accurate encoder readings and proper interrupt service routines was also very challenging. Once I got the encoders firing the interrupts I still had to carefully figure out the calculations required to get the actual encoder speed. Ensuring efficient interrupt handling routines was also a struggle for me since I wanted to print out some values to debug in that function but that took too long so the interrupts would get backed up. Successful completion of Project 4 demonstrates a deeper understanding of motor control, sensor integration, and control system design principles.