Report on Ping-Pong Ball Balancing System Using ATmega3208 AVR-BLE, Ultrasonic Sensor, Servo Motor and PID Controller.

Alexandra Postolaki
CSCI5143

Introduction

The project focuses on designing and building a system to balance a ping-pong ball at a specific distance from an ultrasonic sensor. This system is crucial for understanding the principles of feedback control systems, particularly through the use of PID (Proportional, Integral, Derivative) control. The key challenge addressed is balancing a small object (the ping-pong ball) on a beam while maintaining a precise distance from the sensor. By using an ultrasonic sensor to measure distance and a servo motor to adjust the beam's tilt, the system continuously forces the ball back to its target position.

This project is related to embedded systems where hardware and software components interact closely. The embedded system includes microcontroller integration, sensor data acquisition, motor control, and real-time feedback using PID control algorithms.

Design and Implementation

The system uses the **HC-SR04 ultrasonic sensor** to measure the distance to the ping-pong ball and the **MS24 Miuzei 20 kg servo motor** to adjust the tilt of the beam. The hardware setup includes the following key components:

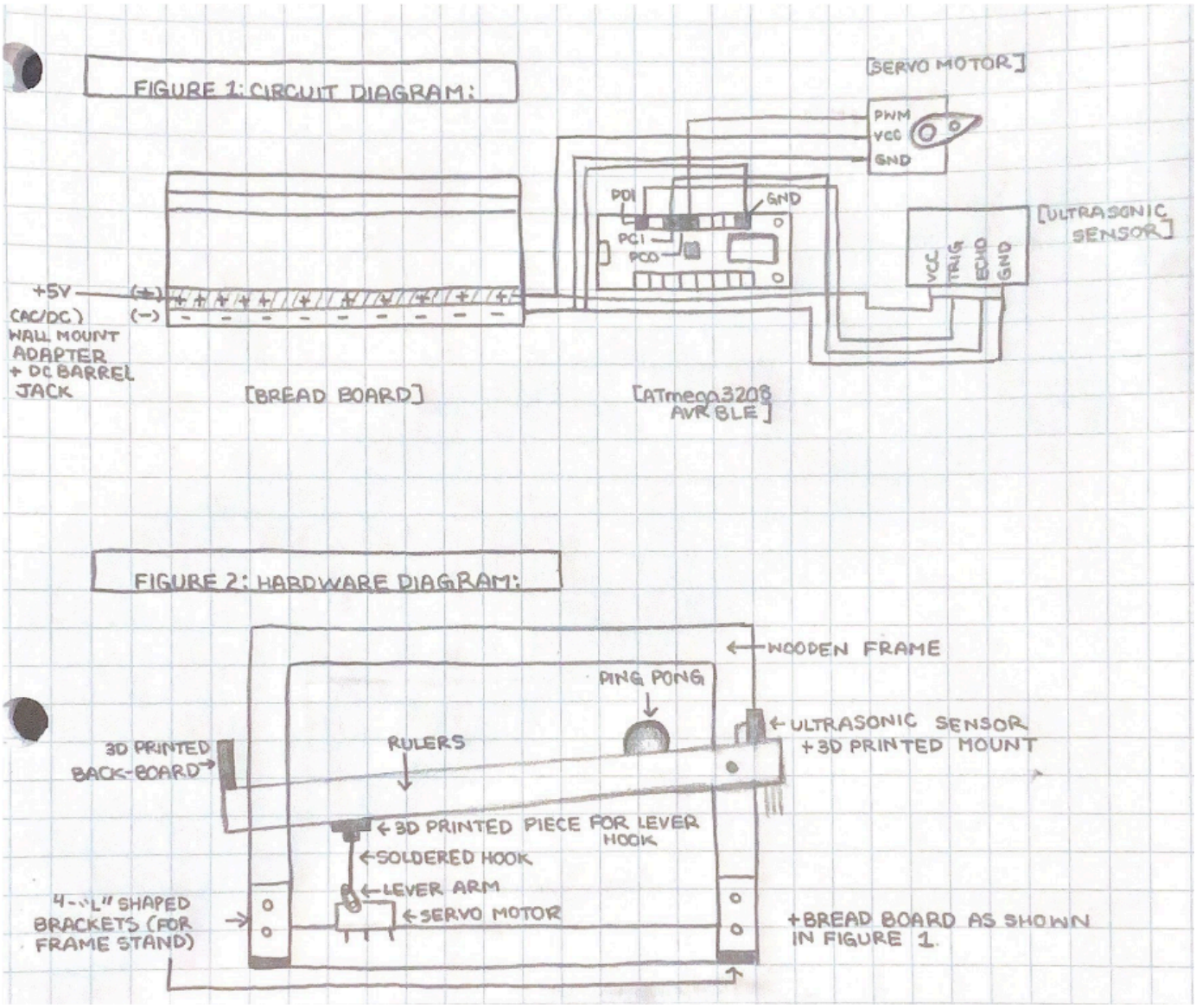
- **HC-SR04 Ultrasonic Sensor**: Measures the distance between the sensor and the ping-pong ball.
- **MS24 Miuzei 20 kg Servo Motor**: Controls the tilt of the beam to return the ball to its desired position.
- **ATmega3208 Microcontroller**: Acts as the central processing unit, running the software for sensor data processing and motor control.

The entire system is set on a wooden frame with metal legs. It is designed to adjust the beam’s tilt based on the distance data received from the ultrasonic sensor. The beam is created using two, 18-inch rulers super-glued together with some 3D-printed pieces to set them at a specific width so that the ping-pong ball could roll between each ruler. When the ball moves away from the target distance specified in the code, the PID controller calculates the required correction, and the servo motor adjusts the beam’s position accordingly through its angle of rotation. An extension to the servo motor’s arm was added using a metal rod and a paperclip soldered to the rod so that the paperclip wire could attach to the servo motor arm and the 3D printed attachment located underneath the ruler used to adjust the height of one end of the beam causing the tilt.

The initial design approach was to keep the setup as simple as possible which was why this hardware setup was chosen over others. This allowed for easier debugging, adjustments, and improvements as I went along building. For example, a **3D printed stand** was created near the end to hold the ultrasonic sensor at a fixed position relative to the beam, ensuring more consistent readings and movement. Originally, the ultrasonic sensor was fixed to the wooden frame, which

caused a lot of issues when attempting to capture the ping-pong ball's distance relative to the ultrasonic sensor. The frame supporting the beam was designed to allow height adjustments using the metal rod arm connected to the servo motor, providing flexibility for different ball positions. Additionally, a 5V AC/DC wall mount adapter and a DC Barrel jack adapter was used with a breadboard to provide the correct voltage for the ultrasonic sensor and servo motor; both the sensor and servo motor required a range of about 4-6V to work.

Circuit and Hardware Diagram



Software Design

The software implementation of this project is designed to:

1. **Trigger the ultrasonic sensor** to measure distance.

2. **Process the distance measurement** using Timer Counter B to calculate the pulse duration, which corresponds to the distance in centimeters.
3. **Use PID control** to calculate the necessary correction based on the ball's position relative to the target distance.
4. **Control the servo motor** to adjust the beam's tilt and bring the ball back to the target position.

The PID control algorithm is the core software component, tuning three variables (Kp, Ki, Kd) to adjust the beam's tilt based on the error (the difference between the target distance and the actual distance).

The software uses Timers/Counters (A0 and B0) on the ATmega3208 AVR-BLE to manage precise timing and interrupts. For measuring the distance, Timer Counter B0 is configured to detect rising edges and trigger interrupts, which allows the software to record the duration of the pulse and calculate the distance. For the servo motor, Timer Counter A0 is configured to adjust the clock's duty cycle which, in turn, adjusts the servo motor's rotational angle.

PID Control Algorithm

- **Proportional (Kp)**: Reacts to the current error (distance difference). Higher Kp results in faster response but can cause overshooting if too high.
- **Integral (Ki)**: Corrects accumulated errors over time, addressing persistent but small offsets. A high Ki can lead to overshoot if it becomes too large.
- **Derivative (Kd)**: Mitigates oscillations by reacting to the rate of change in error. Adjusting Kd helps reduce instability, especially when the system is overshooting.

The PID gains were tuned manually through trial and error, with Kp, Ki, and Kd adjusted to balance responsiveness and stability.

Additionally, code was added to toggle the red and green LEDS on the ATmega3208 microcontroller board so that if the ball was too close to the sensor, the red LED would be on. If the ball was within 2 cm of the target distance, both the red and green LEDs would be on. If the ball was too far from both the ultrasonic sensor and the target distance, only the green LED would be on. Any further than what was described as "too far" would turn both the LEDs off. This was used for debugging as I've adjusted the target distance values to see if I was correctly calculating the distance.

Challenges

- **Servo Motor Behavior**: The servo motor's rotation angle plays a critical role in the PID control's effectiveness. If the servo does not reach a sufficient tilt, the ball would not overcome friction, which caused the system to fail at times. Using a larger ball or adjusting the height of the tilt helped overcome this, but led to difficulties in overshooting.
- **Ultrasonic Sensor Accuracy**: The sensor had difficulty measuring the distance accurately when the ball was smaller (i.e., a ping-pong ball), likely due to the sensor's beam width and the small size of the ball. Again, a larger ball was easier to control. Eventually I realized that

turning the ultrasonic sensor on its side allowed for greater accuracy in measuring the distance with the ping-pong ball.
- **Pin Configuration**: The ATmega3208 microcontroller had some port configuration issues. The data sheet for ATmega3208/ATmega3209 and the MPLAB IDE port configuration diagram did not always align, causing confusion regarding the available Timer/Counter ports. With personal trial and error with the different ports, I found the configuration diagram found in MPLAB IDE to be reliable with knowing which ports I could use for the microcontroller.

Final Product

The final product is a functional system capable of balancing a ping-pong ball at a specified distance from an ultrasonic sensor. The ball is continuously repositioned to maintain the target distance through the servo motor's adjustments. The system is able to correct the ball's position in real-time using the PID control loop.

Key features of the final product include:

- **Real-Time Feedback**: The system continuously measures the distance and adjusts the beam's tilt accordingly.
- **Flexible Design**: The height of the beam can be adjusted to counteract different ball positions.
- **3D-Printed Components**: The ultrasonic sensor stand and frame provided greater stability compared to previous attempts of building the system.

Goals Met

- The ball is kept at a specified target distance from the sensor. This has been tested with adjusting the target distance (in cm) in the code and visually measuring the accuracy of the ping-pong from the ultrasonic sensor once it has been placed at the desired position.
- The PID control system was successfully implemented, with adjustments made to the servo motor angle of rotation.

Goals Not Met

- **Ping-Pong Ball Sensitivity**: The ultrasonic sensor had difficulty accurately measuring the distance with the ping-pong ball. Larger objects and flat objects caused the system to work as desired.
- **Response Time**: While the system corrected the ball's position, it did so slower than desired. This was due to suboptimal PID settings and possible friction issues with the ping-pong ball as it rolled between the rulers.
- **System Stability**: The ball would occasionally stop moving if the servo motor's angle wasn't steep enough to overcome friction on the ramp.

Reflection

With the knowledge I've gained from this project, I would consider using a larger or more consistent object for testing with the ultrasonic sensor, as the ping-pong ball caused multiple inconsistencies. Additionally, improving the PID control by making it more aggressive (by increasing Kp) could reduce response time, but it should be balanced to avoid overshooting.

The servo motor's angle could be made more adjustable to ensure that the tilt is sufficient to overcome the ball's friction with the ramp, thus preventing it from coming to a stop.

Future Improvements

If more time were available, I would focus on the following:

- **Improved Sensor Setup**: Using a more accurate sensor or experimenting with different sensor placements could resolve issues with distance measurement.
- **PID Tuning Automation**: Automating the PID tuning process would allow for a more optimized system without manual adjustments.
- **Faster Response**: Optimizing the software for faster response times and ensuring the ball returns to the target distance more quickly would improve the system's efficiency.
- **Using MPLAB Data Visualizer:** A component that I have wanted to implement for future debugging purposes was the MPLAB Data Visualization tool in order to visually display the distance (in cm) that was detected by the ultrasonic sensor to see how accurate the measurements were as well as to produce a chart to display the PID adjustments.

Additionally, because this was my first inspiration for my final project idea, I would like to try using cameras to detect the position of the ball on a pane of glass. Specifically, instead of balancing the ball in only two directions (left and right on a beam), I wanted to build a platform that the ping-pong ball would never fall off of. It would certainly be a great distraction for kids.