Module 10 Final Report

***Abstract*—This final report presents a thorough analysis of the operation of a PiCar through mapping the duty cycles and rotational speed (RPS) through experimentation. Throughout the experiments, a sampling rate of 0.0050-second intervals (200 Hz) was used. Four objectives—control, timed movement, movement with control, and movement with guidance—were the aims of our work. To meet these objectives, we implemented a PID control to determine measurements such as rise time, overshoot, and steady-state error for a PiCar without a load and which has been propped up. We also created an algorithm, using the PiCar's camera, to detect the presence of blue to guide the car's trajectory. Then, we used threading to allow for manual control/steering over the car's direction. Finally, these processes were combined to develop an autonomous PiCar that has the capability to move towards a blue trashcan, with specific K_p, K_i, and K_d values, and stop within 2 inches of the object.**

I. Introduction

In the rapidly expanding world of autonomous vehicles, our exploration of the efficient control of DC motors is especially applicable. The following objectives aim to develop an effective understanding of control systems and the ability to use the software and hardware of Python and Raspberry Pi in order to carry out the car's movement and data collection.

II. ANALYSIS

A. *Objective 1*

a. Results of Propped-Up Car

- Sampling Rate: 0.0050 second intervals
- Rise Time: ~0.9950 secs
- Overshoot: ~90.84%
- Steady-State Error: ~0.016%

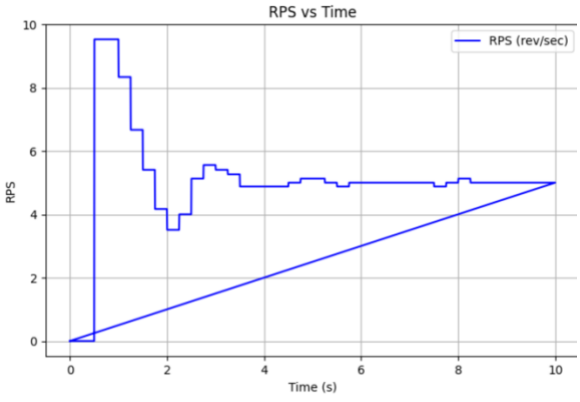

Fig. 1. RPS vs. Time on Propped Car (K_p = 3.0, K_i = 5.0, K_d = 0.0)

b. Calculations

Sampling rate was pre-determined and held constant, while rise time was estimated from the data (refer to 'car_noload_5rps.txt' in Github).

$$\text{Overshoot (\%)} = \left(\frac{9.524 - 5.000}{5.000}\right) \times 100\% = 90.84\%$$

Throughout the last two seconds of the steady-state, 49 of the 400 data points collected produced a RPS value of 5.128, while the other data points were consistent at a RPS of 5.000, making the average 5.016. As such:

$$\text{SSE (\%)} = |5.000 - 5.016| = 0.016\ \%$$

The value for K_d was kept constant at 0.0. However, the K_p and K_i values were determined after manual trial and error.

c. Methodology

Our program was dominated by a main while loop that controlled the timing interval (10 seconds). The program collected 4000 data points, although the 10 seconds timed out before all data points were used. We split the main while loop into ADC sampling, speed calculation and control, and PID control.

B. *Objective 2*

For the second objective, no data was collected, although we aimed to reach some specific goals:

- Reaching the blue trash can in less than 10 seconds.
- Starting far away from the target container.
- Having the ability for the car to steer itself back towards the target container by detecting the blue values.
- Bonus: Allowing the car to stop within 2 inches of the container.

a. Qualitative Results

We were able to achieve all our set goals with our algorithm for the second objective.

b. Notes and Observations

Our car was inclined to veer off towards the right. Several causes may be present—the presence of the blue in the room next to the hallway, difficulty in configurating the steering servo, and loose steering servo. To counter this issue, we turned the car about 30 degrees to the right before starting with the program, which allowed it to leave its trajectory just long enough before coming back.

c. Methodology

This program used the Pi camera to capture images for the Picar to detect a blue object. We utilized both the swivel and servo motos in our operation of the car towards the target. In addition, we used the standard HSV color values to determine our blue. We first converted the image, then calculated the moments to find the center of mass and aid in detecting the object, then updated the angle to determine the duty cycle of the motor.

C. *Objective 3 + Objective 4*

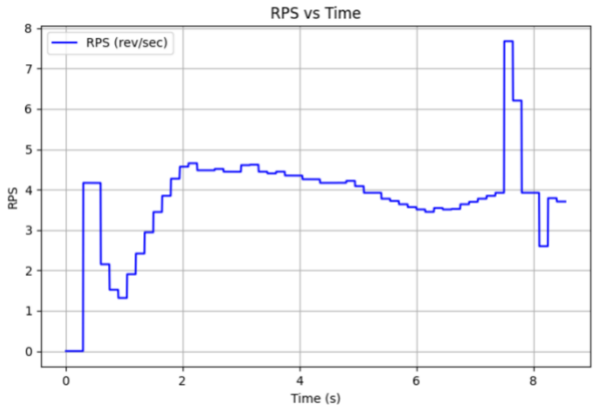

Fig. 2. RPS vs. Time Plot for Manually Controlled PiCar (Obj. 3)

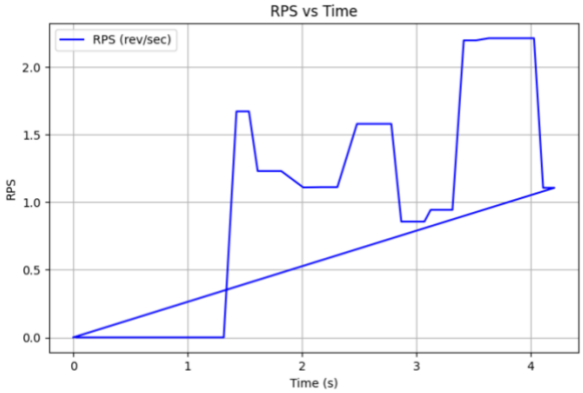

Fig. 3. RPS vs. Time for PID controlled Car directed towards blue trash can (Obj. 4).