Robotic Car to Navigate Natural Landscapes

EECS 452: Digital Signal Processing Design Lab – Winter 2021

Katie Bertcher, Elisabeth Jahnke, Amir Naser, Maya Pandya, Adithya Subbiah, Jack Winkelried

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY OF MICHIGAN

JOHN WOLVEREENE

Introduction / Motivation

This project demonstrated a precursor to an alternative smart harvesting technology by creating a vehicle capable of autonomously navigating a row of stakes. Sporadic, continuous growth prevents one big harvest, so it's important to avoid damage during harvest. Around 75% of production costs come from labor, so the use of smart harvesting technology will make growing asparagus more cost efficient.

Challenges / Innovation / Differentiation

Current Asparagus Harvesters include rail and trench systems. However there are no autonomously navigating vehicle systems yet. Our project attempts to solve this problem by using camera vision to detect stakes and autonomously navigate the stakes. But, one issue that was persistent during the project development was camera sensitivity. The camera was very sensitive to light changes, so we will need to implement some form of adaptive thresholding moving forward

System Architecture / Specification

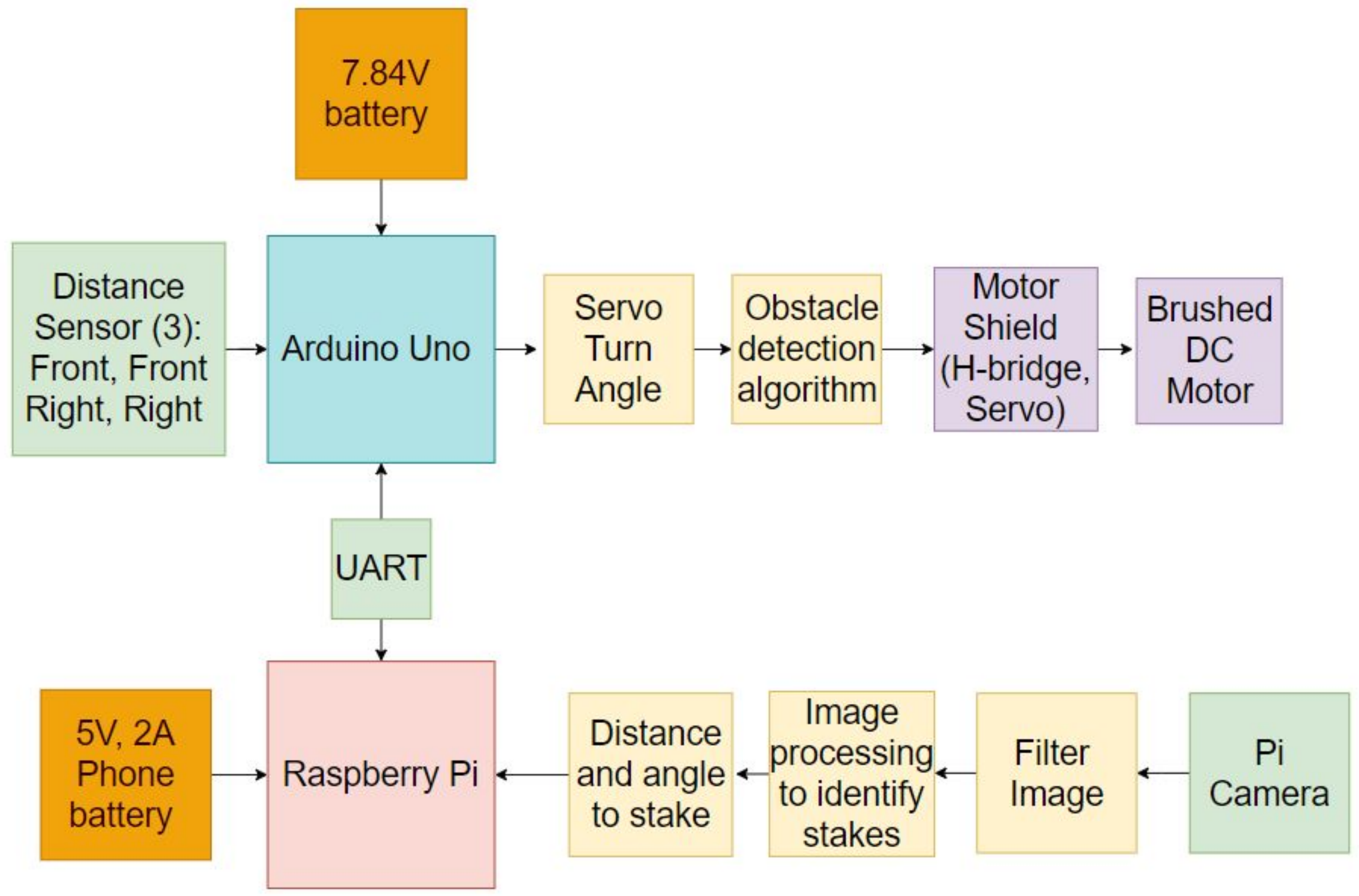

Figure 1: High Level Block Diagram of Autonomous RC Car System

Algorithm / Techniques

Motor Controls

Our Arduino's PID control loop outputs speed and direction directions to the DC brushed motor and Servo. Through the Adafruit Motor Shield, the RC car has duty cycle speed control and a roughly 90 degree steering span.

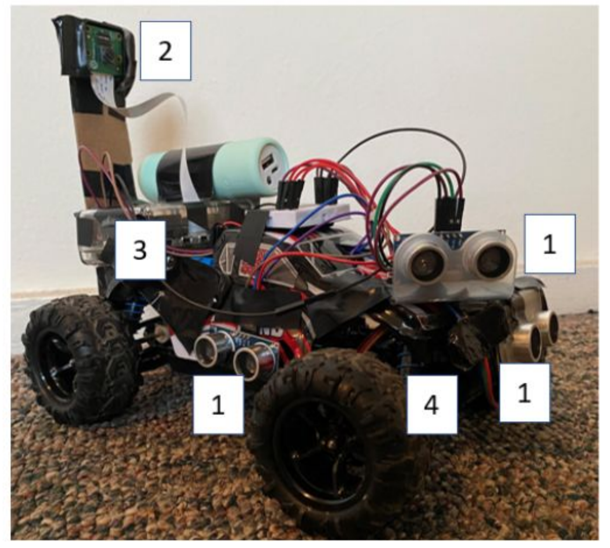

Figure 2: RC Car Architecture. (1) Distance Sensors, (2) Pi Camera (3) Raspberry Pi and Battery (4) Arduino Uno, Servo Motor, Brushed DC Motor

PID Design

We linearized the car system to design a PID Controller. To tune the PID Gains, we used the Ziegler Nichols Method as we viewed the car's behavior.

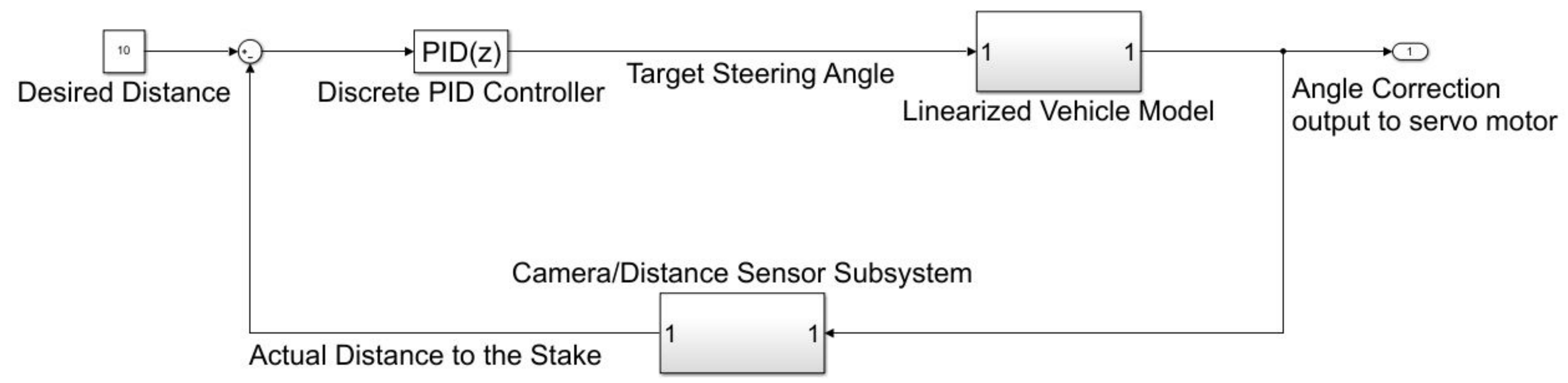

Figure 3: Simulink Block Diagram of PID Control Scheme used

Camera Vision Design

We used thresholding to detect and draw a line around the stake for detection. We then used focal length to calculate the location of the stakes from their position and apparent width in the image. The distance is calculated based on the perpendicular from the stake line to the camera.

$$\beta = \arctan(F/x)$$
$$\alpha = \arctan(F/(x + w_{detected})) - \beta$$
$$w = 2F(\tan(\alpha/2))$$

Figure 4: Distance from Focal Length Detection (Left). What we see versus what the camera sees (Right).

Results/Evaluation

The car was evaluated by plotting the stakes the car saw into a line using Linear Regression.

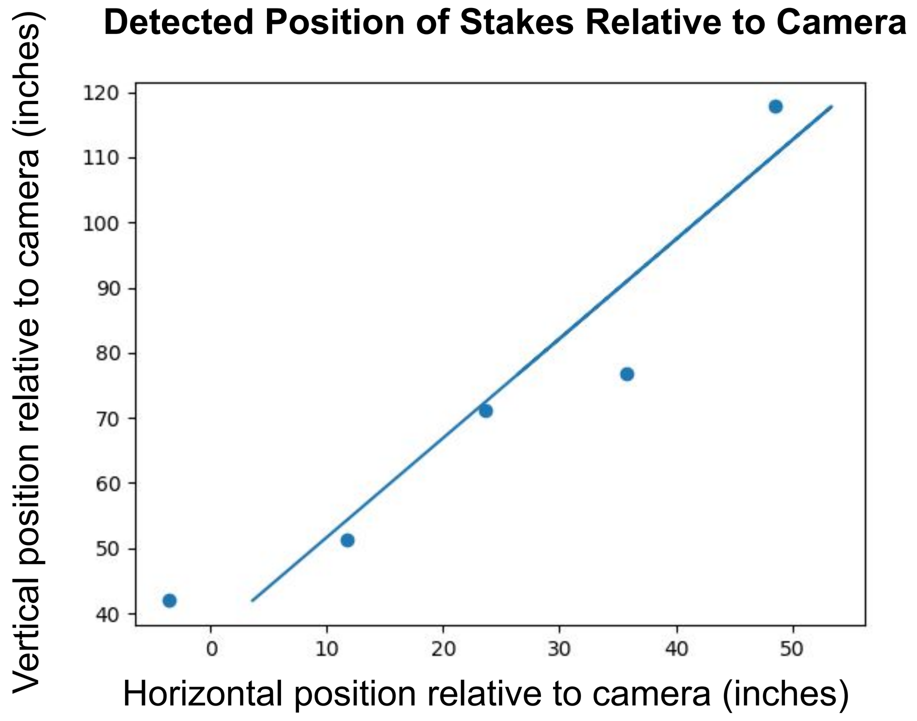

Figure 5: Car Linear Regression Line from the stakes

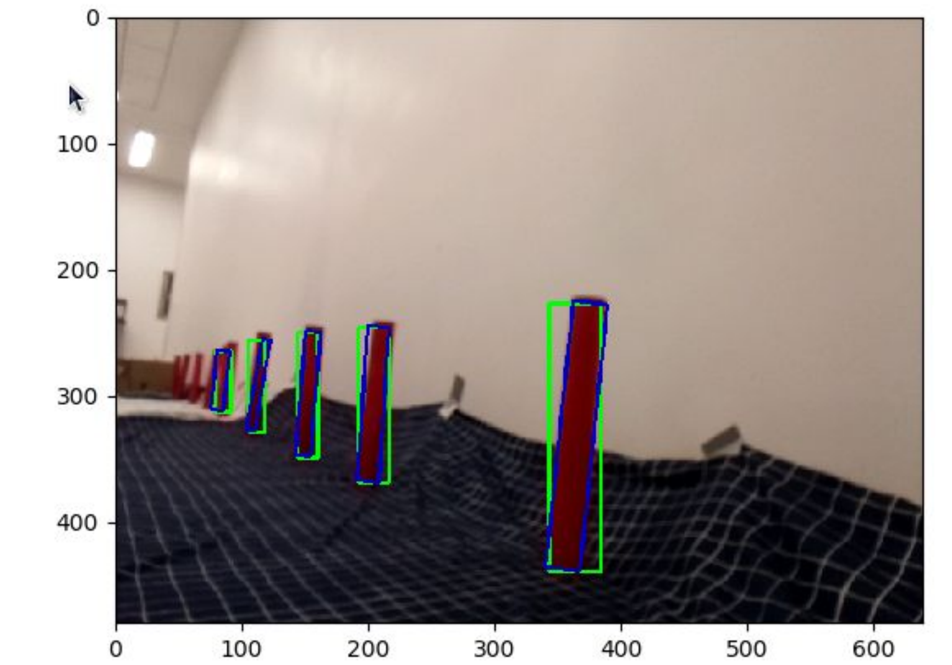

Figure 6: Car CV Stake Detection

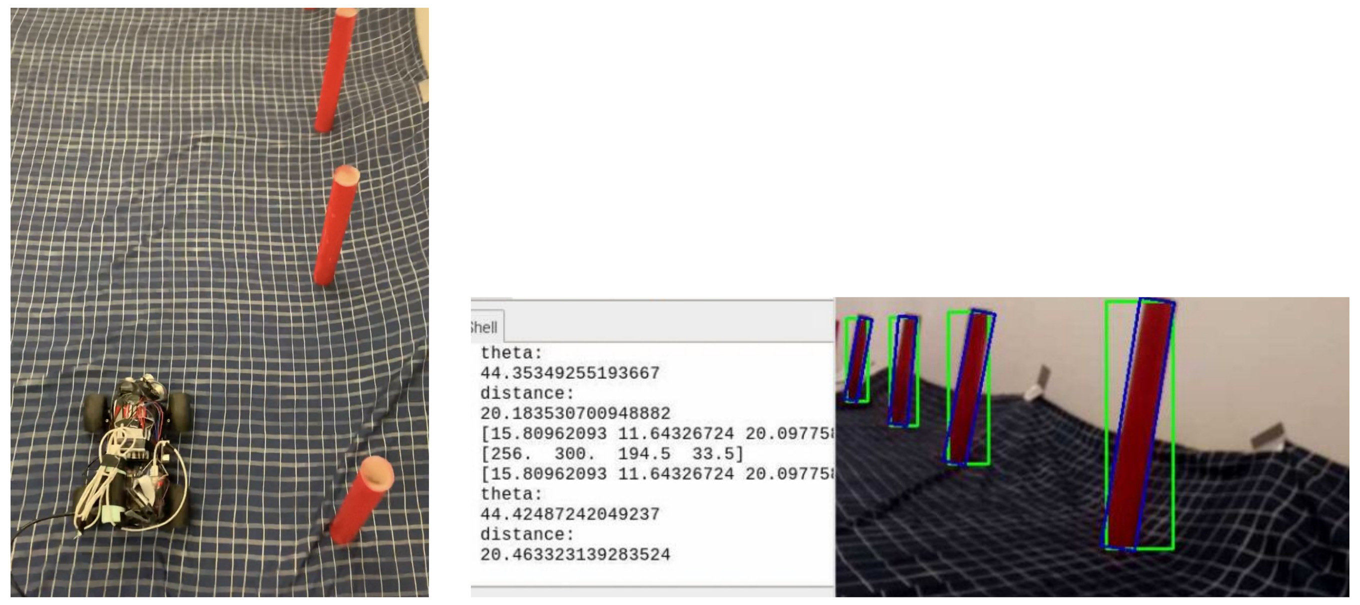

Figure 7: Car Driving along the stakes (Left). Distance Calculations (Right)

Acknowledgments

Thank you to Dr. Shai Revzen, David Kucher and Rishank Nair for their help on this project. We also would like to thank the EECS undergraduate program for funding our work.