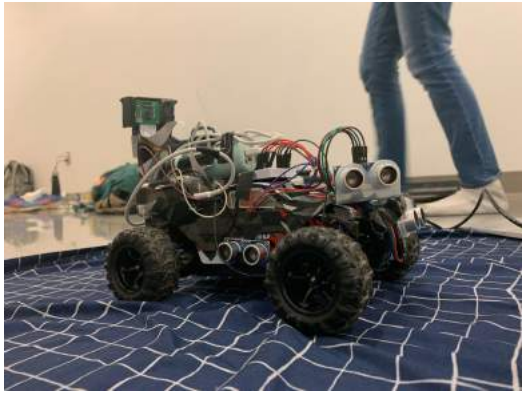
Robotic Car to Navigate Natural Landscapes Final Presentation

Team 8: John Wolvereene

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

Project Goal

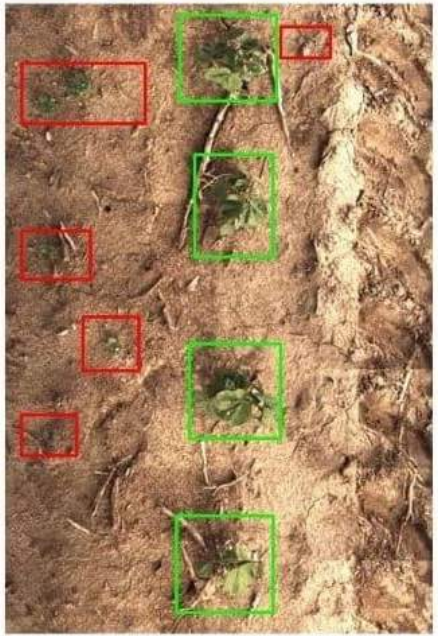
- This project demonstrated a precursor to an alternative smart harvesting technology by creating a vehicle capable of autonomously navigating a row of stakes



Our Autonomously Navigating Vehicle: John “Johnny” Wolvereene

Project Background

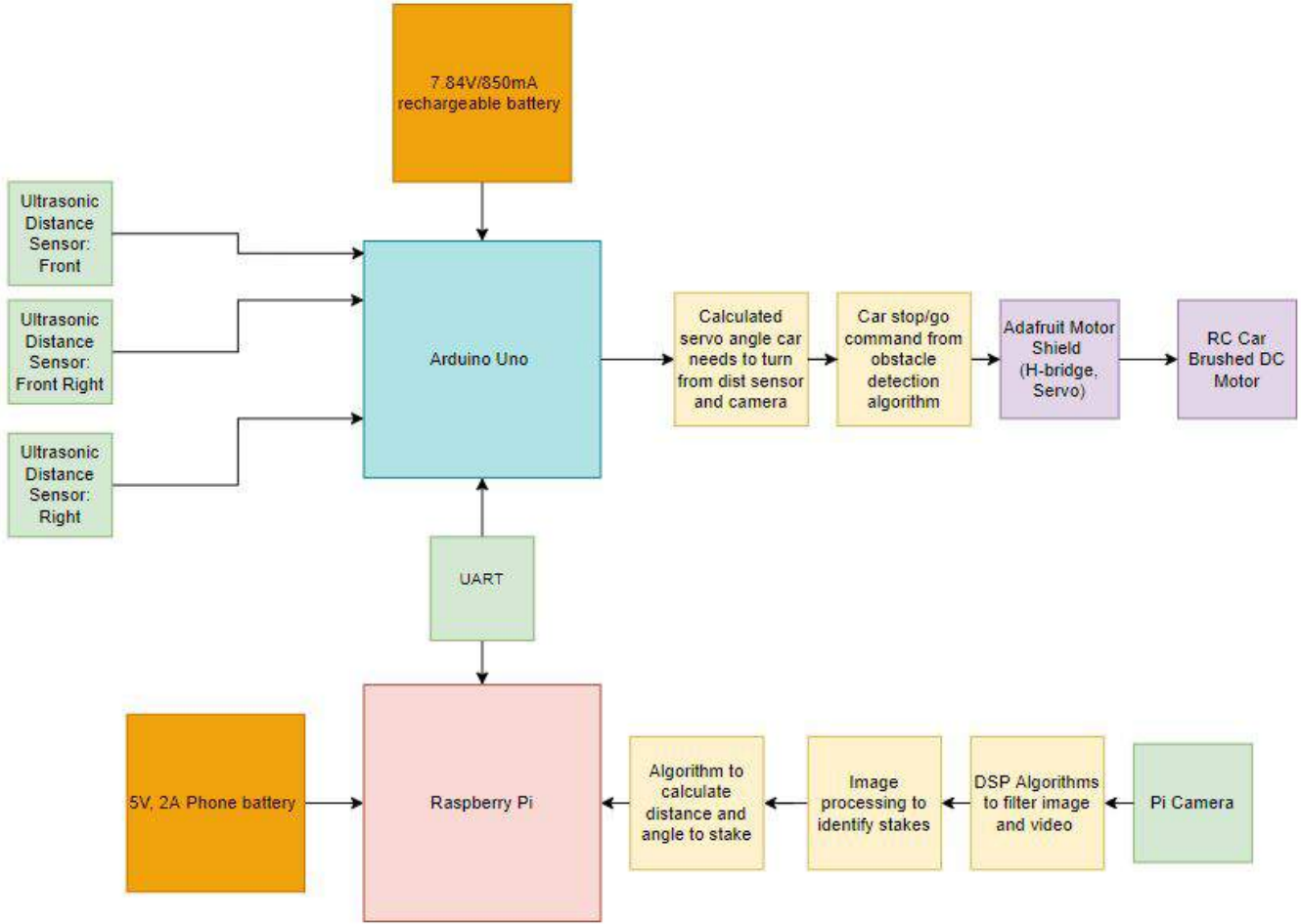
- Project Motivation Asparagus Harvesting
- Smart Harvesting
  - Prominent application of signal processing in agriculture
  - “See and Spray” technology for pesticide application
- The Asparagus Problem
  - Sporadic, continuous growth prevents one big harvest
  - Perennial plants, so important to avoid damage during harvest
  - Around 75% of production costs come from labor
- Current Asparagus Harvesters
  - Rail systems
  - Trench systems
  - No autonomously navigating vehicle systems (yet)



JOHN WOLVEREENE

Previous Work

- RC car hacking
  - Precedent for hacking similar vehicles and using an arduino for control
    - Instructable by Mjrovai
- PID control for the RC car
  - Precedence for PID controlled line and boundary following robots
    - Past senior design from Cornell
    - Arduino project hub information from Nova
  - MatLab toolboxes and Simulink to design PID controllers
- Computer vision for distance detection
  - Lab 2 as a starting point + additional object detection algorithms
    - Histogram of Oriented Gradients
    - Convolutional Networks
    - You Only Look Once
  - Using the focal length of the camera to calculate distance
    - Starter source code from Mahbub
    - If this is not sufficient, stereovision with two cameras is an option
  - UART communication with distance sensor
    - Instructions online for arduino to pi communication

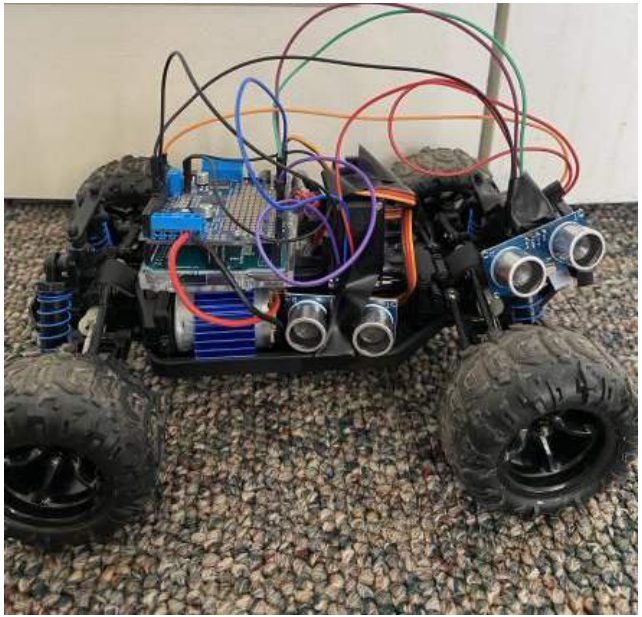


JOHN WOLVEREENE

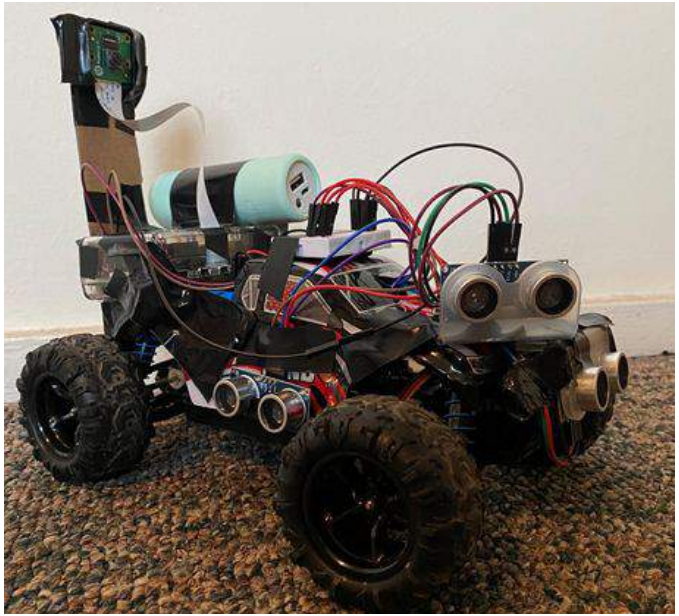
RC Car Hardware Method

- Drive System
  - Original RC Car Brushed DC Motor
  - Adafruit Motor Shield + Arduino provides speed control
  - Duty Cycle + Directional control via H-bridge
- Steering System
  - Original RC car steering linkage
  - Servo motor drives linkage
  - Roughly 90 degree steering span
- Wrote custom RC Car software library for easy interfacing

RC Car Hardware Method



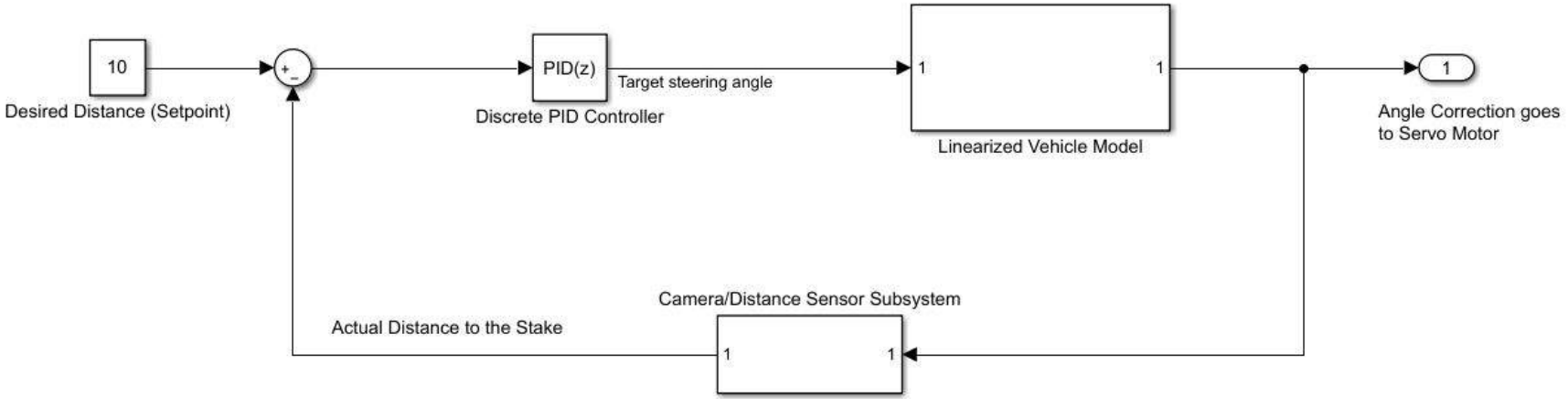
“Under-hood” view



“Completed” view

JOHN WOLVEREENE

Control System: Method



Control System: PID Equations

1 PID Equations

(1) $error = setPoint - Input$

(2) $PID\ \ Error = K_p * error + K_I * \int error + K_d * \frac{derror}{dt}$

(3) $Steering\ \ Angle = PID\ \ Error * \frac{15}{127}$

Where PID Error $\in [-127, 128]$ due to the 8-bit Arduino ADC

JOHN WOLVEREENE

Control System: PID Tuning

- Used Ziegler-Nichols Tuning Rules in our Matlab/Simulink Simulation to get PID gains to start with
- Tuned the gains as we viewed the car's behavior along the stakes/wall

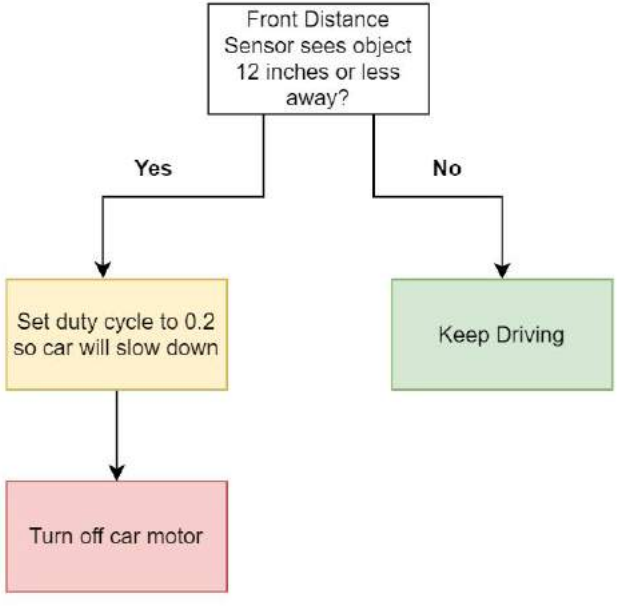
JOHN WOLVEREENE

Distance Sensor Library

- Wrote custom wrapper for distance sensor library
  - Three Sensors (Right, Front Right, Front)
- Measures the median time of 5 sample points
- Converted return time from sensors to a distance (units of inches)
- Equation:
  - Distance = (((Duration_measured) / 2)* speed_sound_cm) / 2.54

JOHN WOLVEREENE

Obstacle Detection



UART Method

- Wrote custom UART library
  - Used Start and End delimiters for packet definition
  - Arduino sends a GPIO High to Raspi to indicate Ready
    - Used as a fix for buffer overflow issue
    - A future solution is to use an interrupt routine
  - Library contains easy to interface functions for CV code

	To Arduino
Message	Value
Start Delimiter	0x1400
Data	Stake Position
End Delimiter	0xFFFF

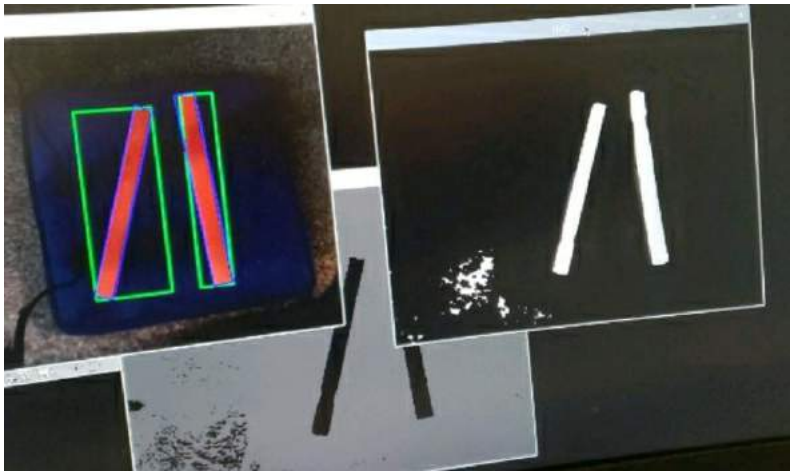
	To Raspi
Message	Value
Start Delimiter	0x1400
Sensor 1 Data	Distance Sensor
Sensor 2 Data	Distance Sensor
End Delimiter	0xFFFF

CV Stake Detection Method

Use findContours after using H, V, R, B thresholding on the image.

Draws a vertical box around the whole stake, and another box that will outline the stake (even if it is at an angle).

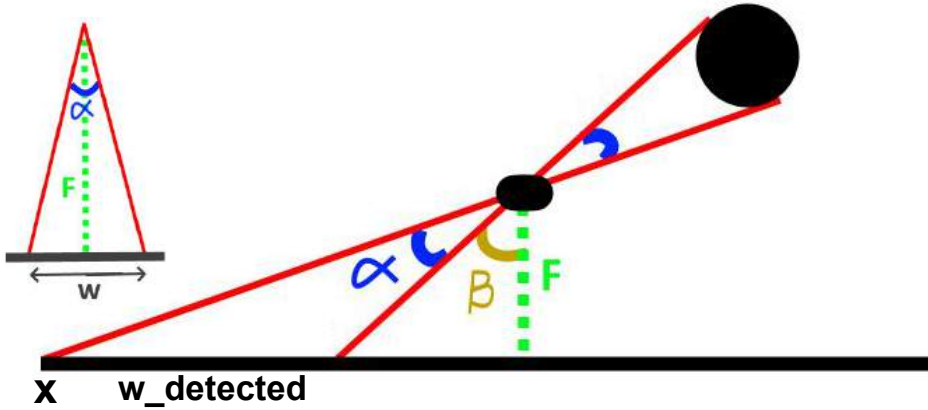
Gain values for the width of the stakes and the distance between the center of the stake to the center of the image for use in distance calculations



JOHN WOLVEREENE

CV Distance Calculation Method

- Used focal length to calculate the location of the stakes from their position and apparent width in the image
- Objects farther away appear smaller
- Must compensate for the fact that cameras project onto a plane rather than a cylinder



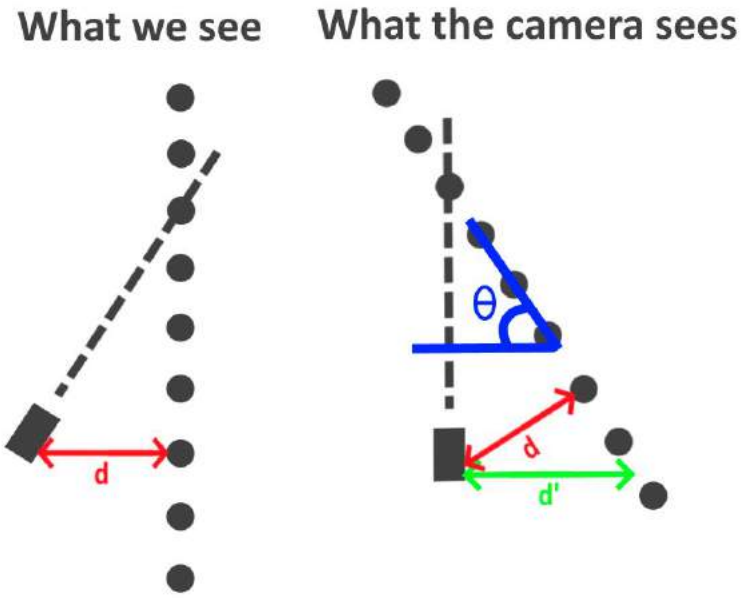
$$\beta = \arctan(F/x)$$
$$\alpha = \arctan(F/(x + w_{detected})) - \beta$$
$$w = 2F(\tan(\alpha/2))$$

CV Distance Calculation Method

- Calculate the distance (d) based on the perpendicular from the stake line to the camera
- The camera is the origin for all calculations
- A linear regression can approximate the stake line
- d' is found from the x intercept of the stake line

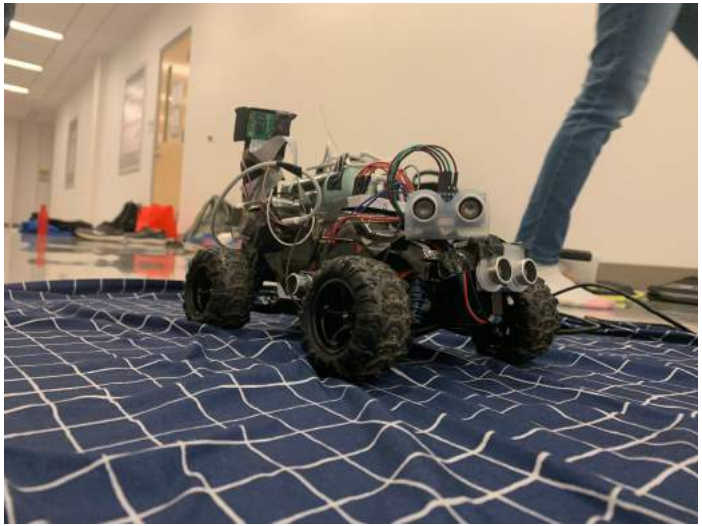
$$d = d' \sin(\theta)$$

$$\theta = \arctan(|slope|)$$



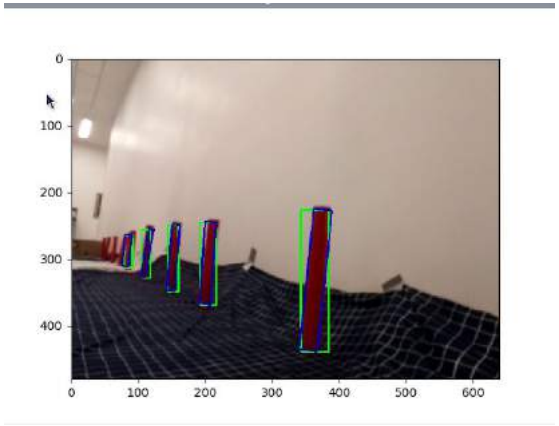
Demo!

- Car Headless With Obstacle Detection
- Car Run 1
- Car Run 2
- Car Diagonal
- Curve Run
- Camera Perspective

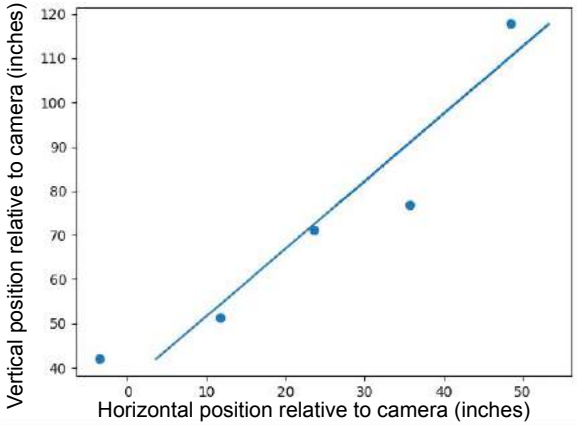


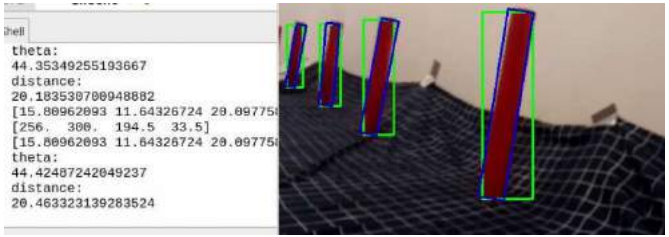
JOHN WOLVEREENE

Project Results



Detected Position of Stakes Relative to Camera





Technical Highlights

- Created our own custom PID loop without any Arduino Libraries
- Took a consumer RC Car and were able to successfully modify it to work with our system architecture
- Implemented robustness in our CV architecture by throwing out outliers and allowing the stakes to be detected even at angles
- Unlike a path following robot, our robot actually takes non-defined obstacles and creates a line to follow from the data
    - It did not have a clear defined path, it had to create its own path from environmental objects

JOHN WOLVEREENE

Technical Highlights

What we wish we knew going into the project!

- CV Detection is *extremely* sensitive to lighting conditions
- The PiCamera uses a pinhole camera model
- Approximating a nonlinear system as linear only gets you so far in control theory
  - Car Cannot correct horizontally
- Discovered UART buffer overflow on Arduino
  - Caused by the Raspi sending commands faster than the Arduino can process
  - Solved by Arduino sending a 1 on a GPIO to the Raspi when ready for new data

Project Poster

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY OF MICHIGAN

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

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

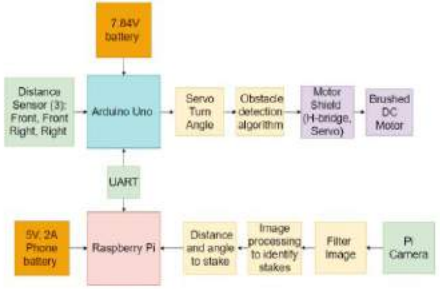


Figure 1: High Level Block Diagram of Autonomous RC Car System

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.

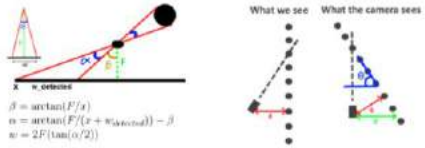


Figure 4: Distance from Focal Length Detection (Left). What we see versus what the camera sees (Right).

Results/Evaluation

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

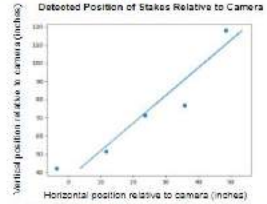


Figure 5: Car Linear Regression Line from the stakes

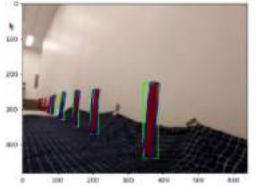


Figure 6. Car CV Stake Detection

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.

JOHN WOLVEREENE

References

[1] Chong, Bonnie, et al. “ECE 4760 The Autonomous Driving Car.” *Cornell ECE*, Cornell, people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2011/bcc44_acl84_my259/bcc44_acl84_my259/index.html.

[2]*GARotics*, European Union’s Seventh Framework Programme, 2012, www.amlight.eu/garotics.php.

[3] Khvoynitskaya, Sandra. “3 Types of Autonomous Vehicle Sensors in Self-Driving Cars.” *Itransition*, Itransition, 2 Nov. 2020, www.itransition.com/blog/autonomous-vehicle-sensors#:~:text=The%20majority%20of%20today%27s%20automotive,cameras%2C%20radars%2C%20and%20lidars.

[4] Mahbub, Istiaq. “Distance Measurement.” *MD Istiaq Mahbub*, Wix, istiaqmahbub.wixsite.com/embedded/distance-measurment.

References

[5] Mjrovai, “Hacking a RC Car With Arduino and Android.” *Instructables*, Autodesk, 28 Sept. 2017, www.instructables.com/Hacking-a-RC-Car-With-Arduino-and-Android/.

[6] Nova. “Line Follower Robot (with PID Controller).” *Arduino Project Hub*, Arduino, 11 Aug. 2020, create.arduino.cc/projecthub/anova9347/line-follower-robot-with-pid-controller-cdedbd.

[7] Satran, Joe. “Most Of Your Asparagus Comes From Abroad These Days. Here's Why.” *HuffPost*, HuffPost, 7 Dec. 2017, www.huffpost.com/entry/asparagus-farms-california_n_7029836.

[8] *See & Spray Agricultural Machines - Blue River Technology*, Blue River Technology, www.bluerivertechnology.com/.

[9] Zitter, Leah. “Inaho's Tireless Asparagus Picking Robot.” *Food and Farming Technology*, 15 Oct. 2020, www.foodandfarmingtechnology.com/news/yield-monitoring/inahos-tireless-asparagus-picking-robot.html.

JOHN WOLVEREENE

Thank you! Questions?