DOGBOT: A Robot Responding to Laser Stimulus

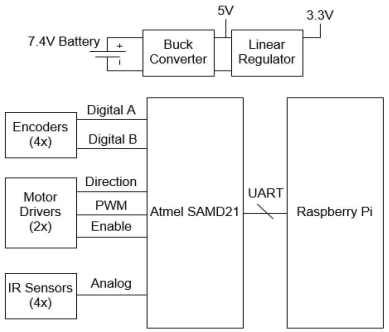
Andrew Levin, Shiva Mehta, Jonathan Zarger

EECS 452 - Digital Signal Processing Design Lab, Fall 2016

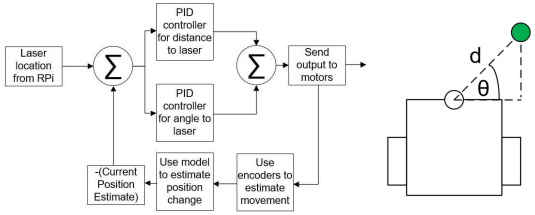
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Project Overview

DOGBOT is a robotic system that interacts with a laser pointer and a boundary. Like a dog, the robot chases the laser. Along with responding to the laser stimulus, the robot is constrained within a physical arena and must make sure not to bump into its walls. DOGBOT detects the walls of the arena and recognizes that it cannot continue to chase the laser.



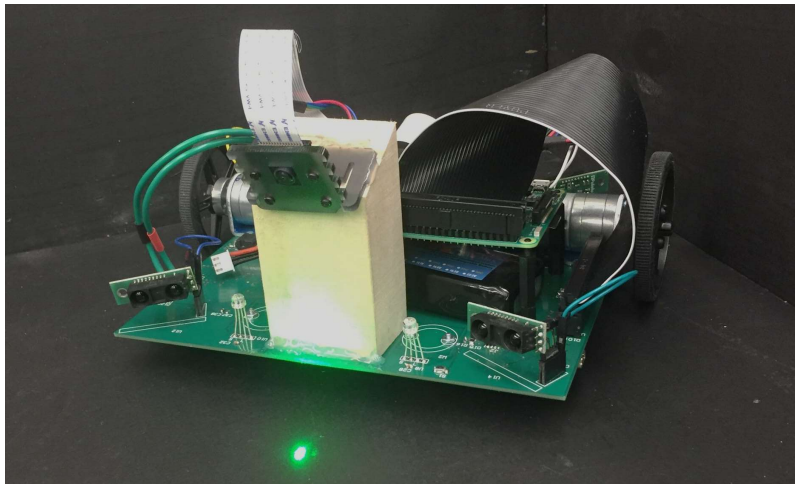
Control System Architecture



Control of the motors, estimation of the state of the robot, and wall avoidance are performed on an Atmel SAMD21 microcontroller (used in the Arduino Zero) running the Arduino platform.

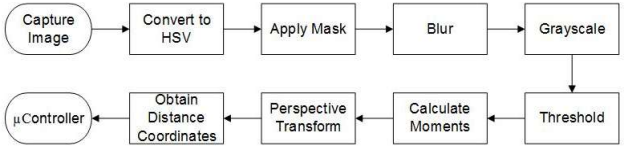
After the Raspberry Pi gives the microcontroller the location of the laser relative to the robot, it compares its current location estimate with the laser position to generate an error of location in distance and in angle to turn. These errors are fed into two Proportional-Integral-Derivative (PID) controllers, which attempt to minimize the error over time and generates outputs to send to the motor controller.

As the robot moves, it uses feedback from the wheels to estimate how its position has changed. If a wall is detected, it stops the robot and commands it to turn.

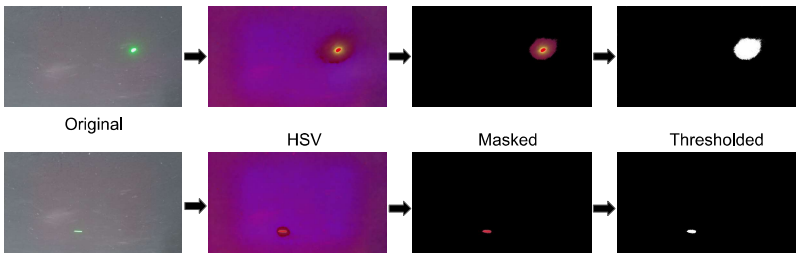


Objects shown in image not to scale

Image Processing



1. Image processing executes on the Raspberry Pi using OpenCV and Python.
2. After the Raspberry Pi camera captures the image, it is converted from the RGB to the HSV color space. In RGB; red, green, and blue values determine each pixel's appearance. In HSV, each pixel has a hue (color), saturation and brightness value. This distinction allows images to be filtered by the shade of color, saturation, and brightness.
3. Once in the HSV color space, each pixel is masked and thresholded. If it does not fall within a certain range of HSV values (HSV corresponding to green for DOGBOT), the pixel is set to black. All non-black pixels are set to white.
4. The average XY pixel coordinate (moment) of all white pixels is calculated and then passed through a perspective transform that converts the pixel coordinates into physical coordinates relative to the robot. These coordinates are then passed to the microcontroller.

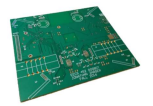


Hardware Overview

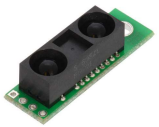
Raspberry Pi
Single-board Linux computer
Used for image processing
Interfaces with camera
Sends data over serial to chassis

Raspberry Pi Camera
8MP camera
Captures images at 10Hz

Chassis Printed Circuit Board
Places to mount all components
Built-in microcontroller
5V and 3.3V voltage regulators
Two 6A H-bridge motor drivers



IR Rangefinder
Sharp GP2Y0A60SZLF
10cm - 150cm range
Measures distance in front of robot
Keeps it from colliding with walls



Motors
1000RPM at 6V
39oz-in stall torque, 6.5A stall current
Has rotary encoders to measure speed

Miscellaneous Integrated Circuits
Atmel SAMD21 Microcontroller
Infineon TLE9201 H-Bridge
NXP 74HC4050PW Level Shifter
Diodes INC AP65502 Switching Reg.
Microchip TC2117-3.3V Lin. Reg.

Parts shown in images above are not to scale

Acknowledgements

We would like to thank Professor Gregory Wakefield, Dr. Kurt Metzger, and our GSI's Dom Calabrese and Sudheer Nuggehalli for their help and support throughout the semester. We would also like to thank the ECE Undergraduate Program and the Harris Corporation for funding our project, as well as Advanced Circuits for donating a printed circuit board.

References

[1] "OpenCV: Image Processing in OpenCV", *Docs.opencv.org*, 2016. [Online]. Available: http://docs.opencv.org/3.1.0/d2/d96/tutorial_py_table_of_contents_imgproc.html.

[2] "CS W4733 NOTES - Differential Drive Robots", 2016. [Online]. Available: https://chess.eecs.berkeley.edu/eecs149/documentation/differentialDrive.pdf.