Survivor Identification and Retrieval Robot

Karun Koppula   Zachary Wasserman   Zhijie Jin

What Does it Solve?

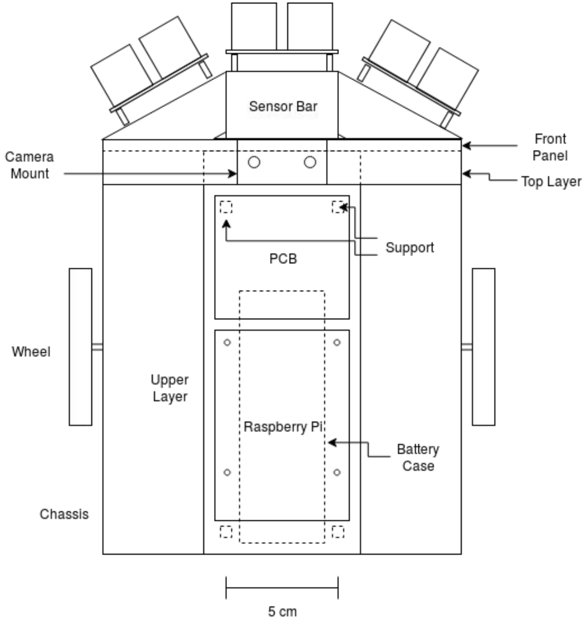
- Rescue people/objects in situations that would be life threatening
- Grab a drink and bring it back

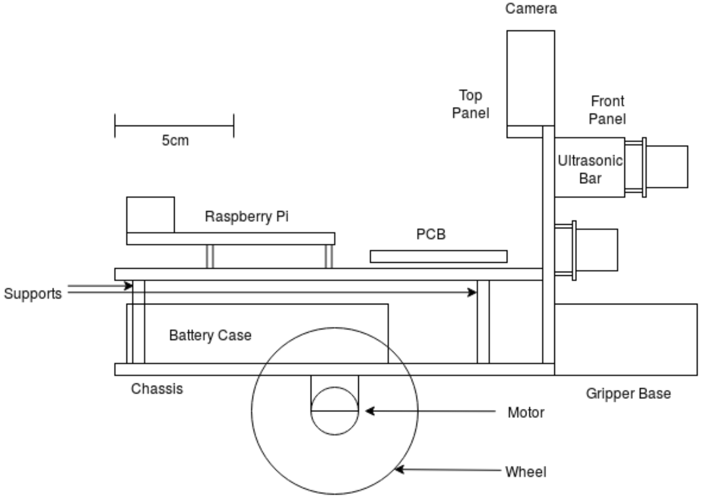
Does this by being:

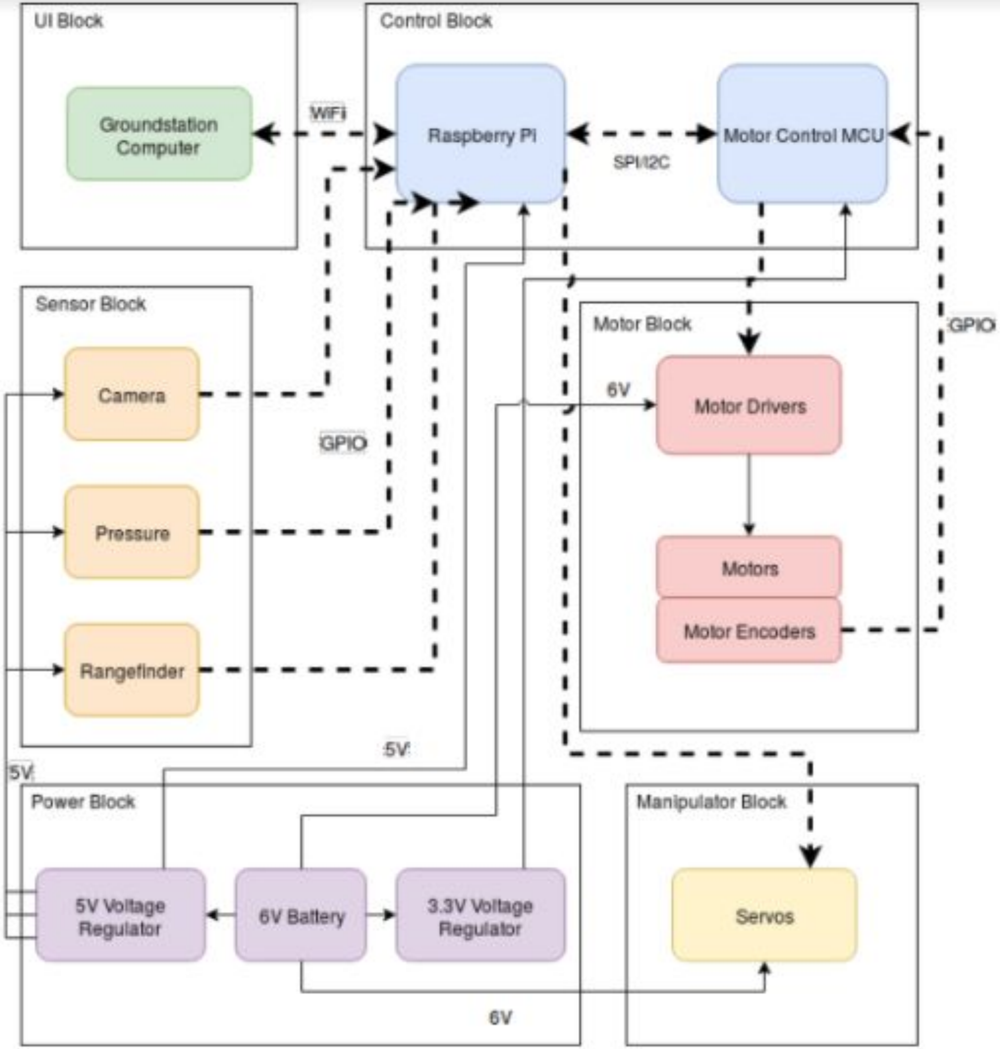
- An autonomous vehicle that can navigate a maze
- Capable of identification of objects
- Capable of retrieving objects

Original Design

- Originally wanted a robot that could solve a maze using a ML (Machine Learning) algorithm
- Identify objects using a ML algorithm






UI Block
Groundstation Computer
WiFi
Control Block
Raspberry Pi
SPI/I2C
Motor Control MCU
Sensor Block
Camera
Pressure
Rangefinder
GPIO
Motor Block
6V
Motor Drivers
Motors
Motor Encoders
GPIO
5V
5V
Power Block
5V Voltage Regulator
6V Battery
3.3V Voltage Regulator
6V
Manipulator Block
Servos

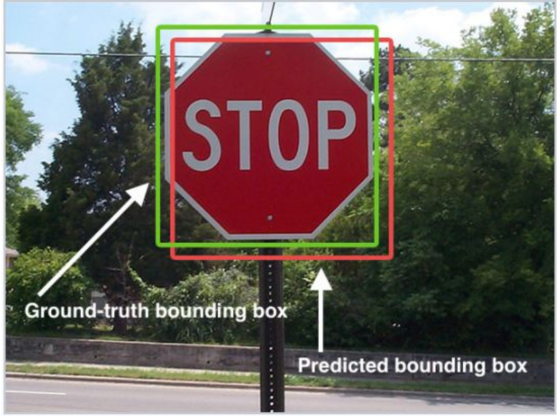
Detection/Classification

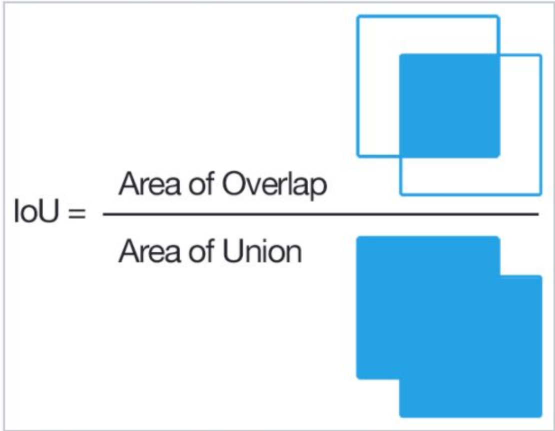
- Used Single Shot Detection (SSD)
- Need to know classification
- Need to know the localization
- Alternatives would include YOLO:9000 bigger,faster,stronger.

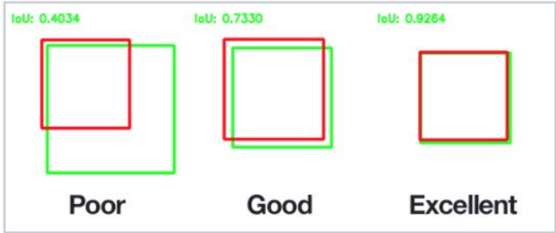
How Single Shot Detection Works

- First determining the features of the image
- For each location use a filter to evaluate the boxes
- For each box, predict box and class
- Modify prediction values s.t. IoU closest to 1
- Also modify values so that the classifier is more correct
- Box Choices

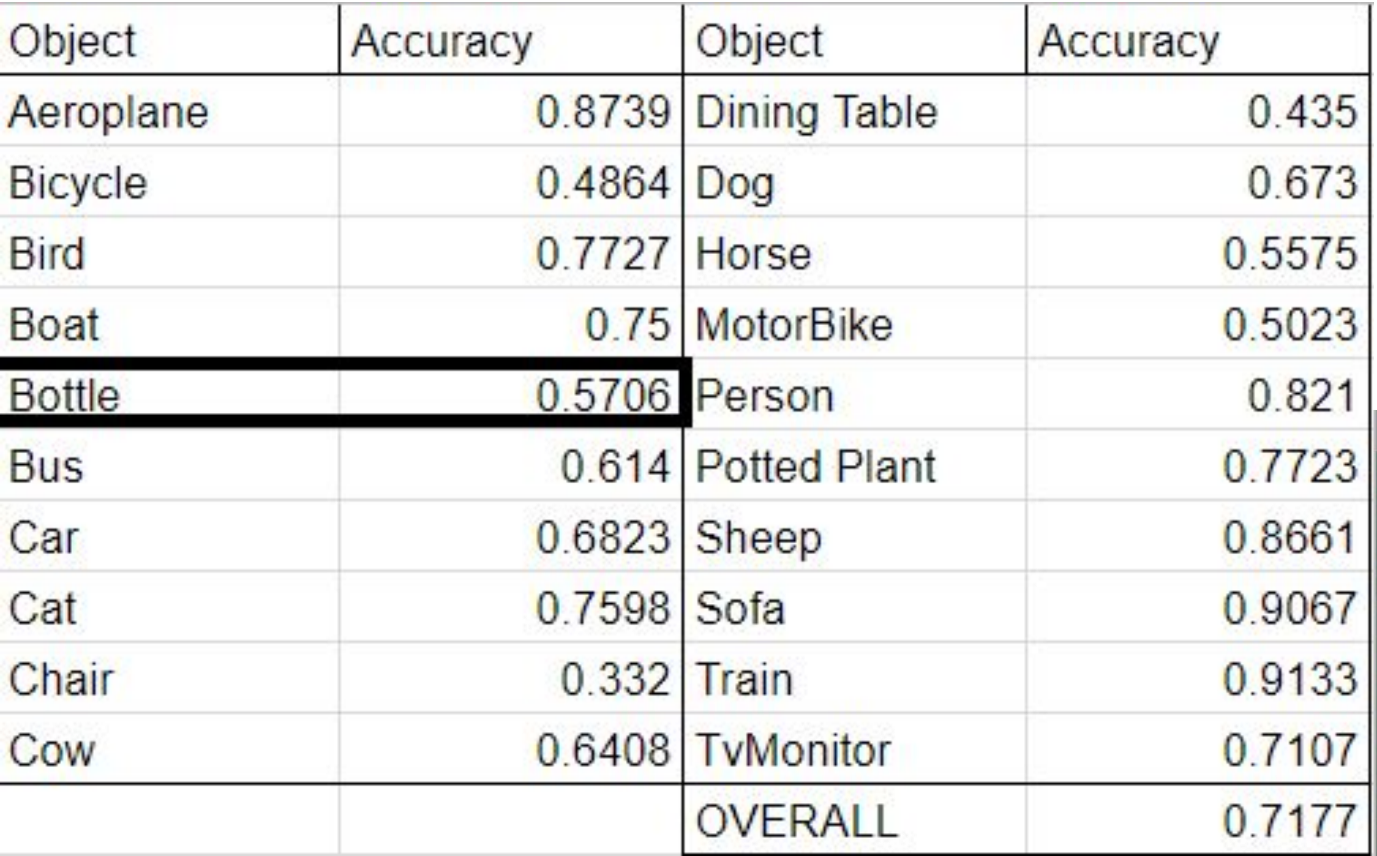
multiBoxLoss=confidenceLoss+alpha*locationLoss



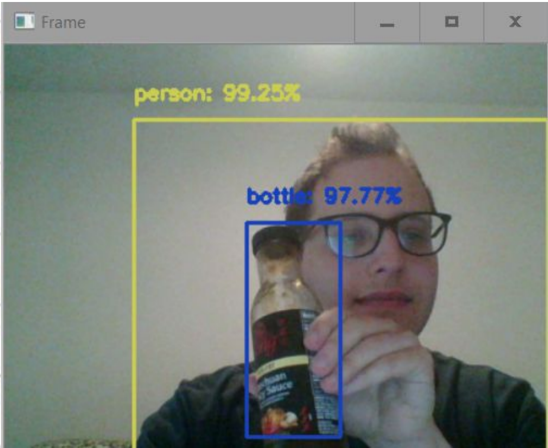




Verification of Single Shot Detection Model



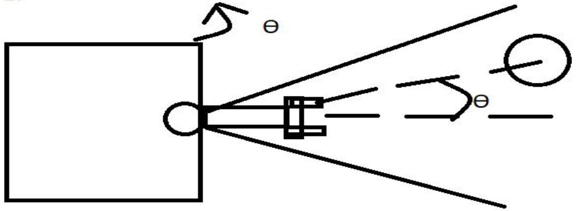
Object	Accuracy	Object	Accuracy
Aeroplane	0.8739	Dining Table	0.435
Bicycle	0.4864	Dog	0.673
Bird	0.7727	Horse	0.5575
Boat	0.75	MotorBike	0.5023
Bottle	0.5706	Person	0.821
Bus	0.614	Potted Plant	0.7723
Car	0.6823	Sheep	0.8661
Cat	0.7598	Sofa	0.9067
Chair	0.332	Train	0.9133
Cow	0.6408	TvMonitor	0.7107
		OVERALL	0.7177



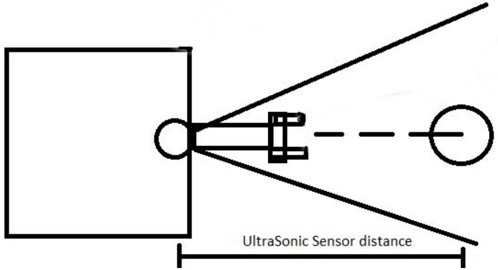
Retrieval

- Single Shot Detection passes the pixel coordinates of where the object exists on the screen
- Take these coordinates and calculate degrees to turn
- Rotate s.t. the two are lined up
- Use Ultrasonic sensor to move forward
- Use servo to pick up object

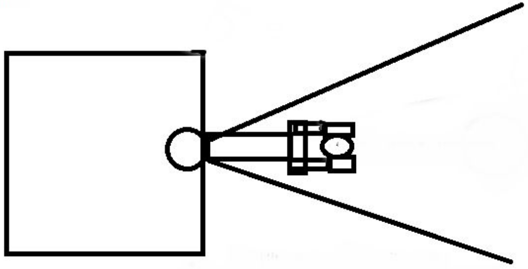
1.



2.

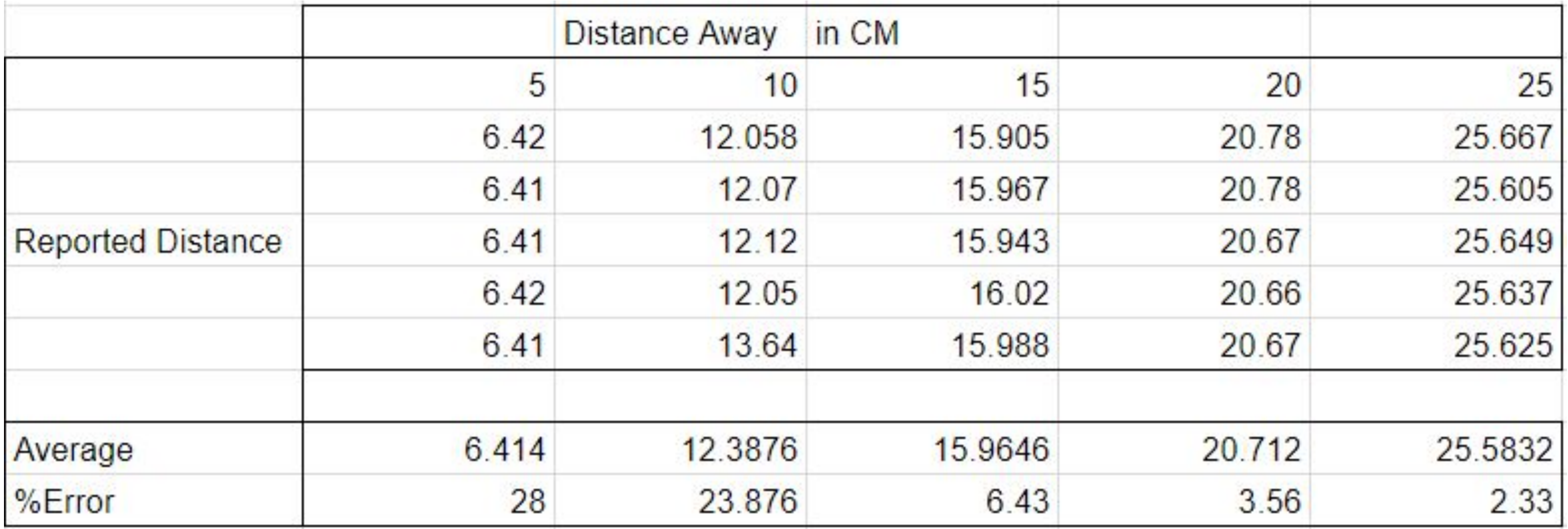


3.



Verification of Retrieval

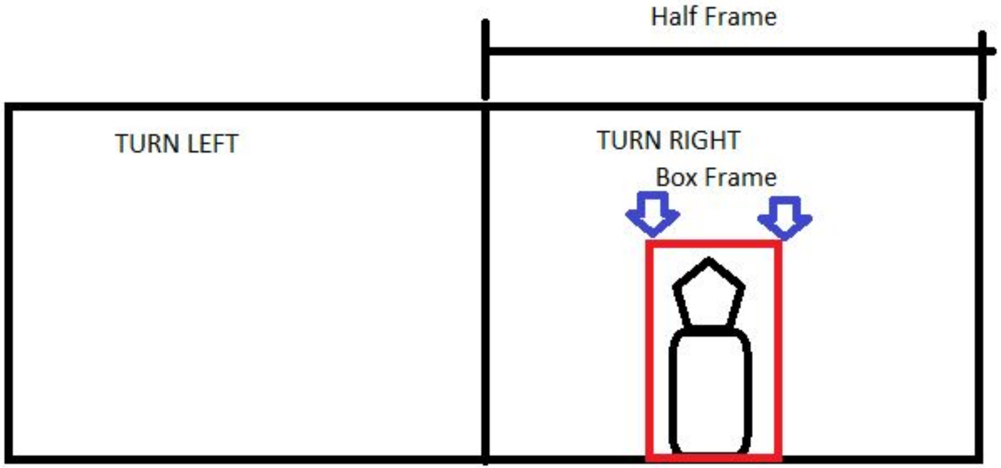
- Checked console to see if the output of the pixel coordinates was reasonable
- Servo can be verified by using python commands to control them
- Measure distances using UltraSonic sensor



		Distance Away	in CM		
	5	10	15	20	25
	6.42	12.058	15.905	20.78	25.667
	6.41	12.07	15.967	20.78	25.605
Reported Distance	6.41	12.12	15.943	20.67	25.649
	6.42	12.05	16.02	20.66	25.637
	6.41	13.64	15.988	20.67	25.625
Average	6.414	12.3876	15.9646	20.712	25.5832
%Error	28	23.876	6.43	3.56	2.33

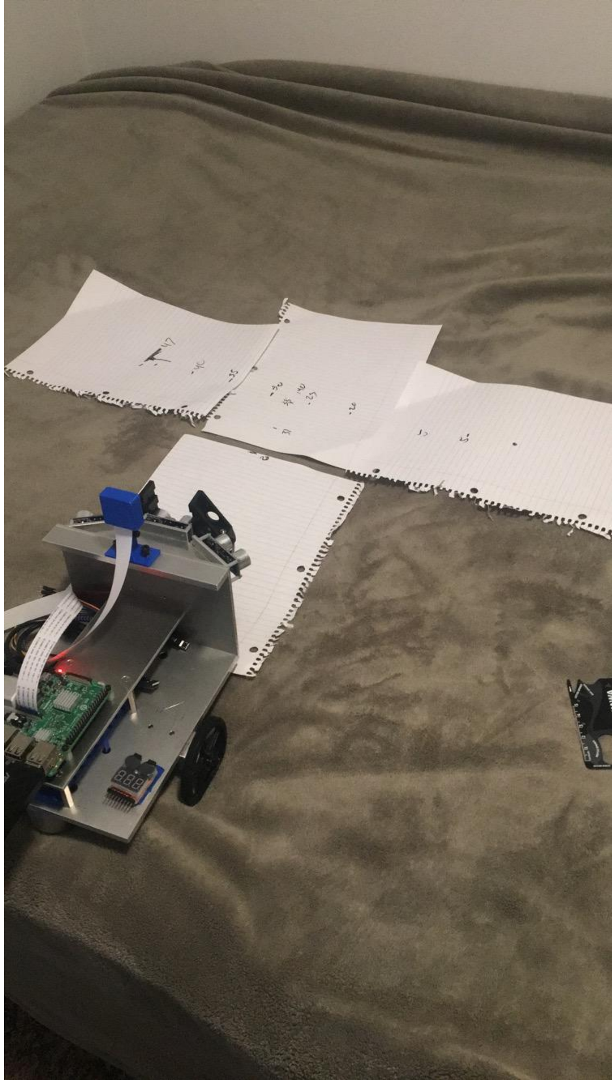
How to find the degrees to turn

1. Get the “Half Frame in pixel distance
2. Find the middle horizontal point of the box drawn around the object
3. Subtract value from 2 by value from 1
4. Multiply by ratio of degrees/pixel
5. If turn left, multiply by -1



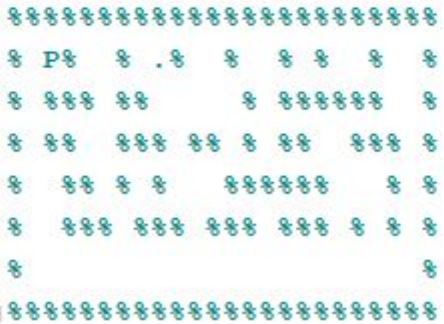
Verification of Degrees to Turn

- My field of view calculation: 30.43 degrees
- Documentation: 31.1 degrees
- The horizontal pixel distance can be done in python
- Tested against real object, did math to see if match

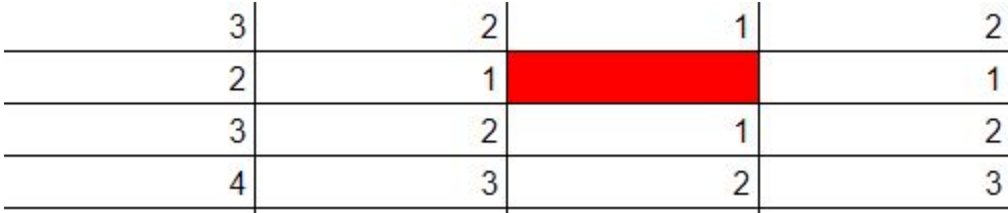


A* search algorithm

- First we analyze each node and determine distance to the end point from each node (Manhattan Distance)
- Follow path taking smallest weights at each node (sum of manhattan plus distance we have traveled to reach that point)
- When solved, passes the movement as ROS command



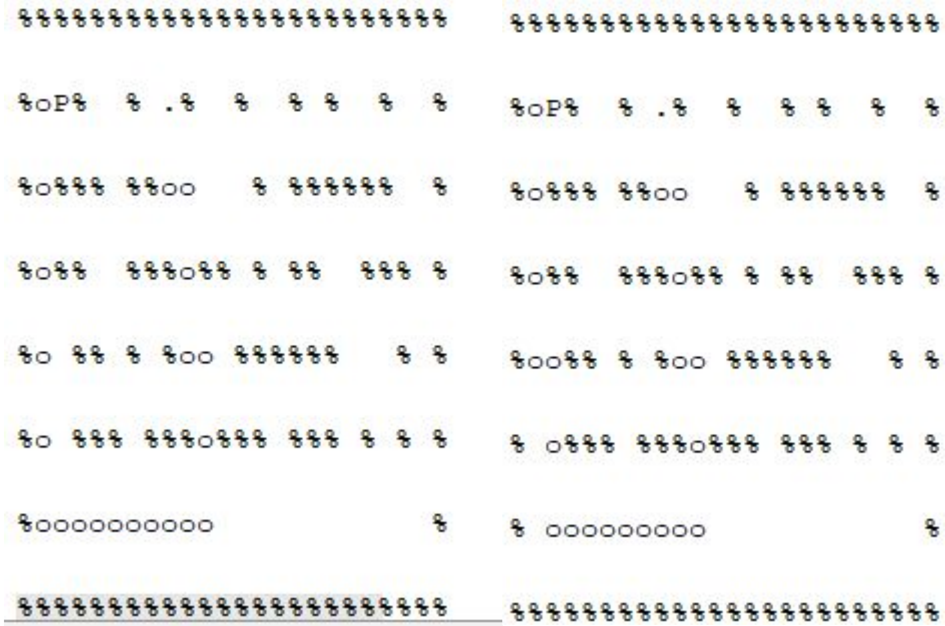
```
%%%%%%%%%%%%%%%%%%%%%%%%
% P%  % .%  %  % %  %  %
% %%% %%     % %%%%%%  %
% %%  %%% %% % %%  %%% %
%  %% % %   %%%%%%   % %
%  %%% %%% %%% %%% % % %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%
```



3	2	1	2
2	1	🟥	1
3	2	1	2
4	3	2	3

Verification of A*

- Detailed proofs that declare it is optimal if it is both Admissible and Consistent (lecture in itself)
- Check against a Breadth-first search solution (left BFS right A*)
- On 7x5, A* is 8 times faster than BFS
- On 24x7, A* is 22 times faster
- On 82x31,A* is 21 times faster



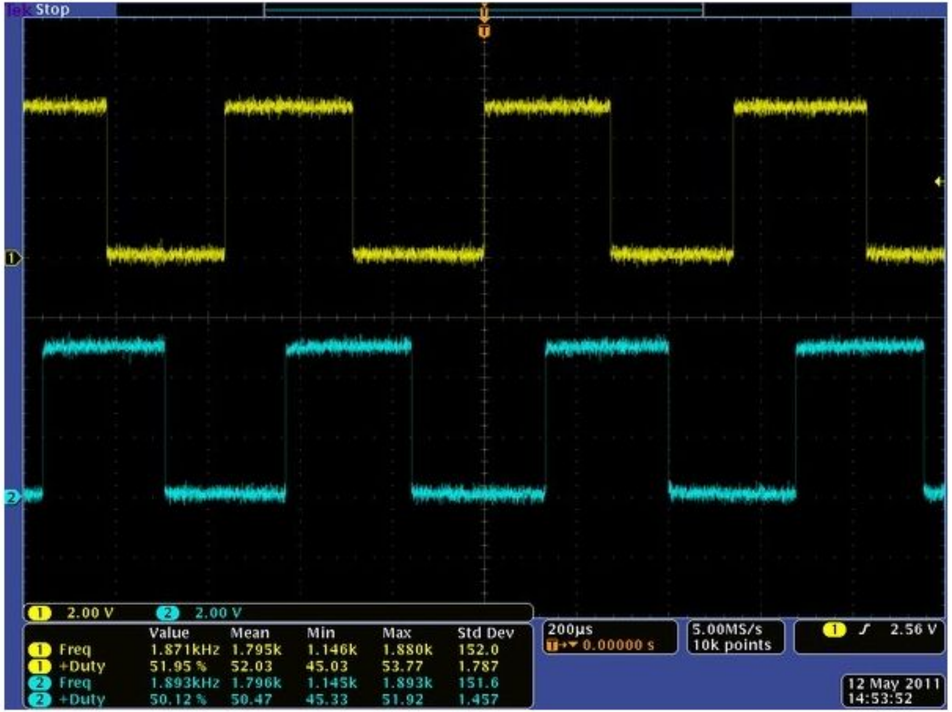
```
%%%%%%%%%%%%%%%%%%%%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%
%oP%  % .%  %  % %  %  %   %oP%  % .%  %  % %  %  %
%o%%% %%oo   % %%%%%%  %   %o%%% %%oo   % %%%%%%  %
%o%%  %%%o%% % %%  %%% %   %o%%  %%%o%% % %%  %%% %
%o %% % %oo %%%%%%   % %   %oo%% % %oo %%%%%%   % %
%o %%% %%%o%%% %%% % % %   % o%%% %%%o%%% %%% % % %
%oooooooooo            %   % ooooooooo            %
%%%%%%%%%%%%%%%%%%%%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%
```

Range Finding

- Ultrasonic Sensors
- Noisy
- Object Detection
- Domain

Rotary Encoder

- Hall Effect
- Two square waves with 90 degree phase offset
- Used for direction, speed
- Angular resolution of 0.002794 radians

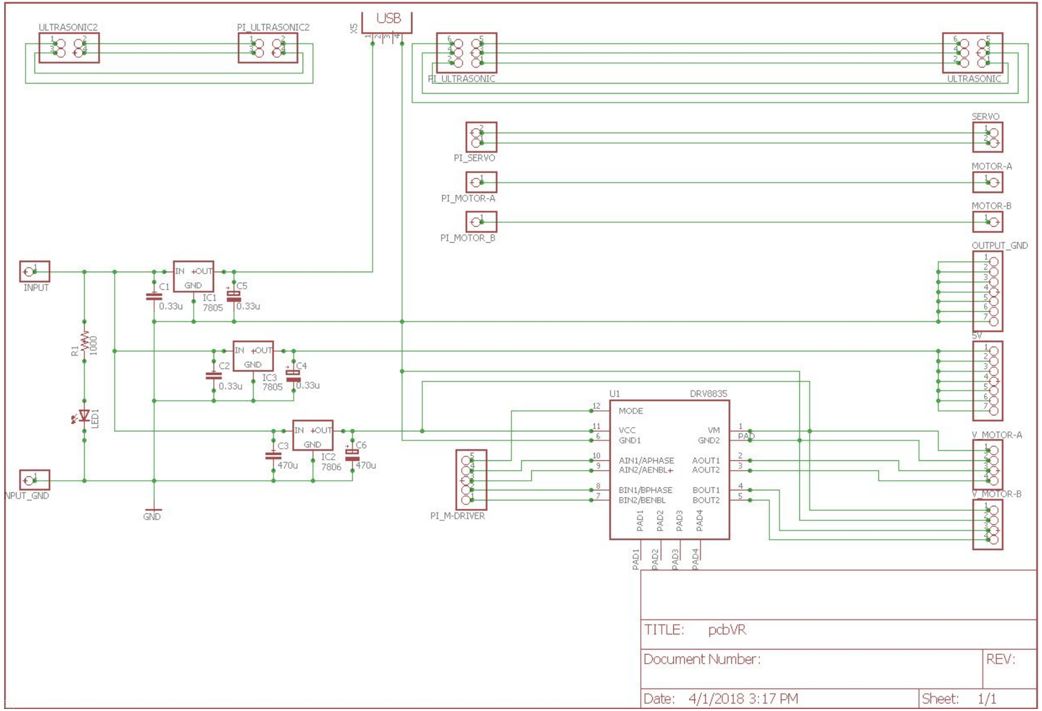


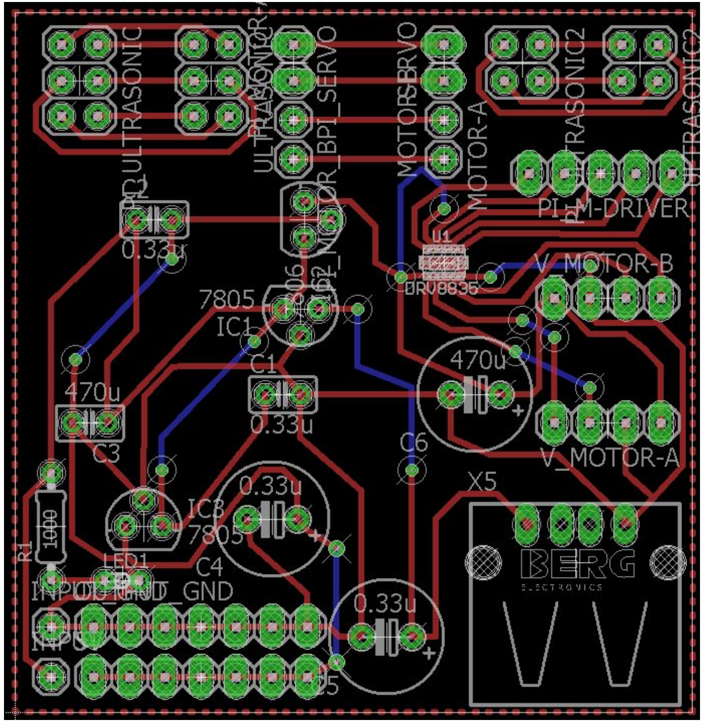
	Value	Mean	Min	Max	Std Dev
1 Freq	1.871kHz	1.795k	1.146k	1.880k	152.0
1 +Duty	51.95 %	52.03	45.03	53.77	1.787
2 Freq	1.893kHz	1.796k	1.145k	1.893k	151.6
2 +Duty	50.12 %	50.47	45.33	51.92	1.457

Motor Control

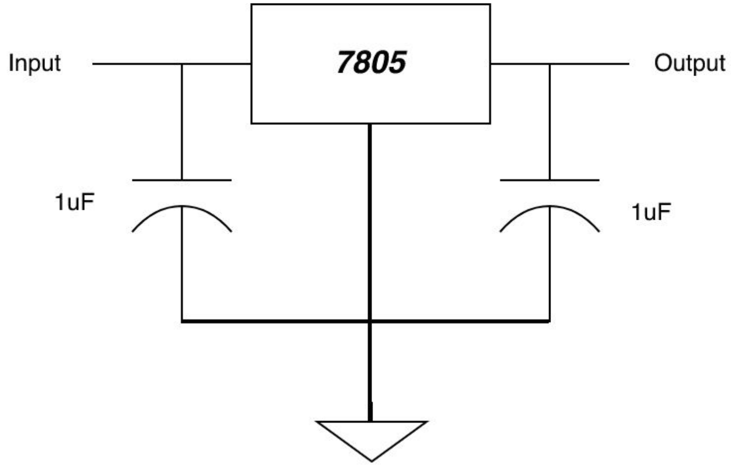
- PID Control
- Non-linearities of velocity estimation
- Raspberry Pi non-realtime

PCB Schematics & Layout



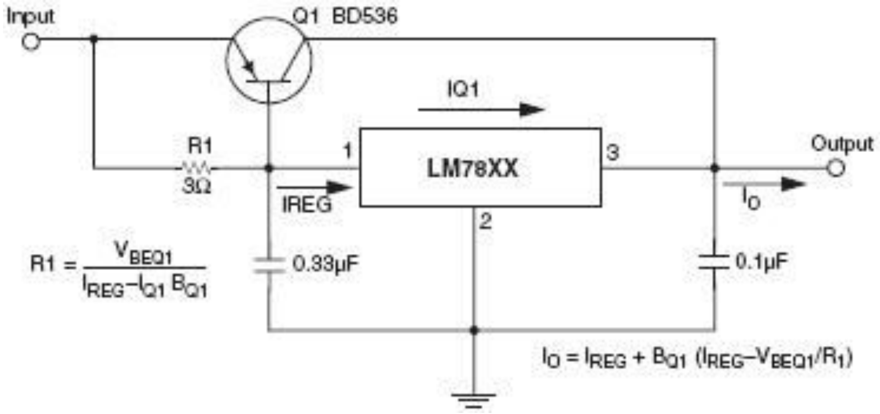


Voltage Regulator Circuit



Output voltage: 5V
Output current: 1.5A(max)

High Current Voltage Regulator Circuit



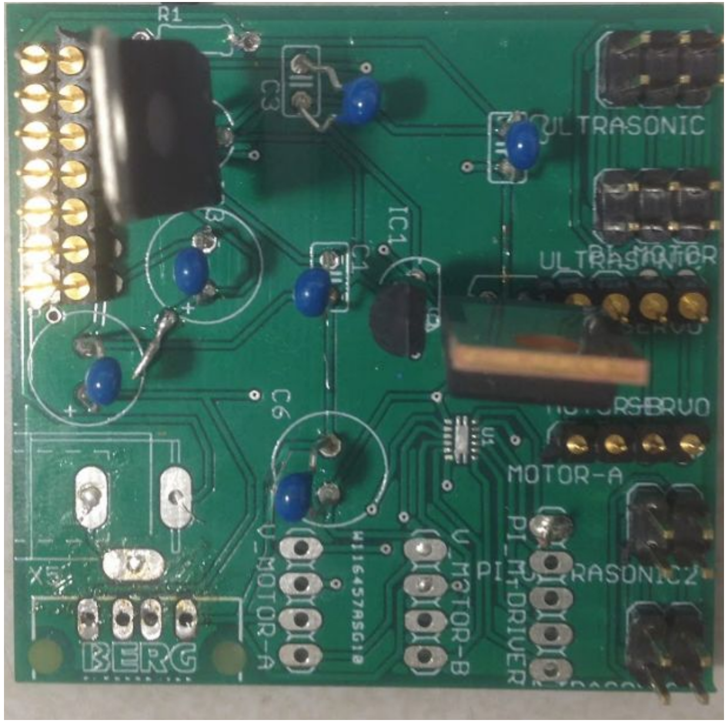
$$R1 = \frac{V_{BEQ1}}{I_{REG} - I_{Q1} B_{Q1}}$$

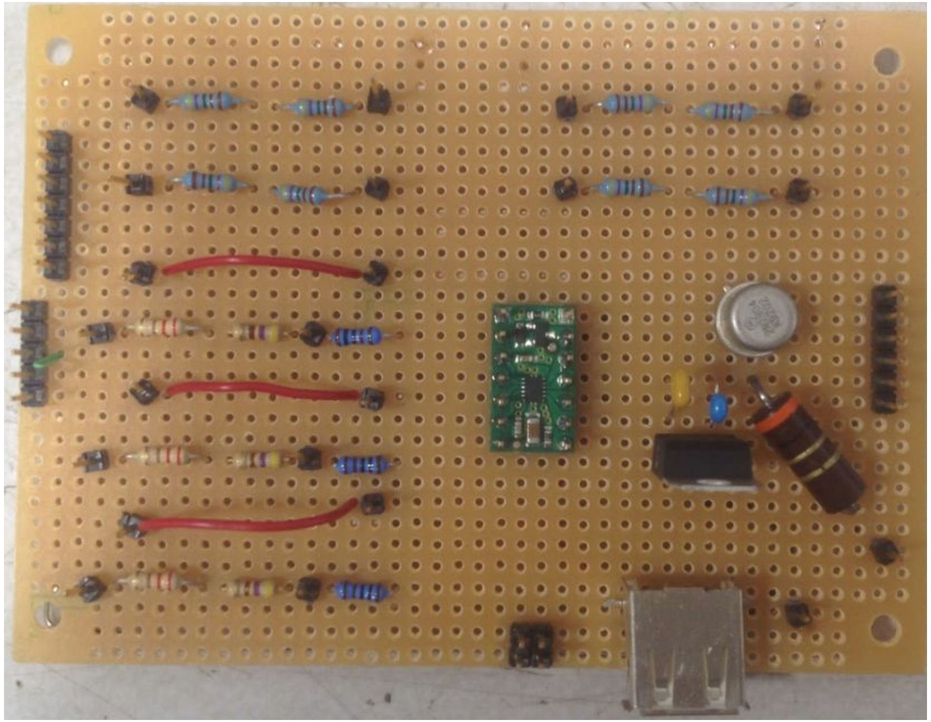
$$I_O = I_{REG} + B_{Q1} (I_{REG} - V_{BEQ1}/R_1)$$

From:

http://www.reuk.co.uk/wordpress/electric-circuit/high-current-voltage-regulation/

PCB and Perfboard Layout (with soldering)





Power Calculation

Total power from battery: Pb = 2.2Ah = 24.42 Wh

Two motors: 9 W

Two servos: 0.768 W

Three ultrasonic sensors: 0.225 W

Raspberry Pi: 12.5 W

Power wasted by voltage regulator: 23.956 W

Operation time: 30 minutes

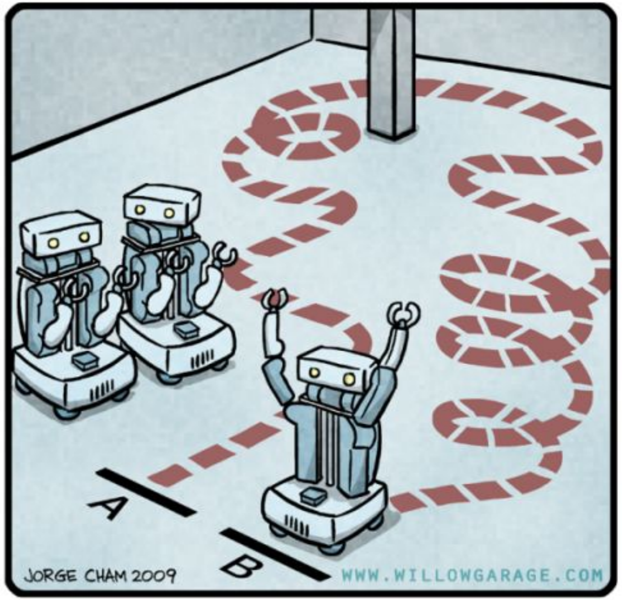
Successes and Challenges

Successes:

- Robot turning
- SSD

Challenges:

- ROS
- Robot moving forward
- Training of the SSD



R.O.B.O.T. Comics

"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

Future Work

- Getting the movement fully finished
- Stress testing
- Better designed bot:
    - Lighter
    - Better Processor
    - Better Gripper