Project Proposal

Department of Aerospace and Mechanical Engineering

EME185A - Team Winery Bot

Erick Campos, Kyle Cubba, Jerry Li, Dai Tran

January 29, 2016

Abstract

Water efficiency has become an increasingly large concern in the winemaking industry. Over 80% of the water used in winemaking is allocated towards cleaning purposes.To combat this problem, the Department of Viticulture and Enology at the University of California, Davis has tasked the Winery Roomba team to create a robot that can functionally clean a winery while using significantly less water. The robot must safely and autonomously clean a specified area with minimal human interaction. This report illustrates potential paths that the Winery Roomba team can follow to achieve the desired goals.

Table of Contents

Section **Page**

List of Figures

Figure	Title	Page

List of Tables

Table	Title	Page

1 Mission Statement

There is an assembled WEINbot prototype available as a starting point for this project. There are two proposed solutions: the team will focus on the programming of the existing WEINbot to be an autonomous machine or the team will modify the WEINbot to maximize its cleaning efficiency. Whether or not the WEINbot is used, modified, or completely reconstructed, the design must meet these specific needs:

2 Identified Customer Needs

The assembled WEINbot prototype from the previous year will be used as a starting point for this project. Although there is already a previously constructed robot, the basic needs that this robot must meet have not changed. These basic needs and additional ones are listed in the following table with their importance shown on the far right. A value of 1 star means this need is very important while 3 stars means not very important.

#	**Need**	**Important**
1	The robot must use a minimal amount of water	*
2	The robot must clean the area of all crushed grape byproducts and residue	**
3	While cleaning, the robot must operate without human aid	*
4	The robot must avoid hazards on the ground	**
5	The robot must avoid people that may walk in front of it	***
6	The robot must be safe to handle	*
7	The robot should clean the entire area without having to recharge the batteries	**
8	The robot must be easily cleaned	***
9	The robot must be battery powered	**
10	The robot must know when the waste container is full	***
11	The robot must know when the water reservoir is empty	***
12	The robot must clean quickly and thoroughly	***

Table 1: Customer Need Priority

3 Refined Target Specifications

To address the needs developed by the sponsor and team, a set of target specifications are listed below.

#	Need # Address	Specification	Value	Unit
1	12.	Maximum time to clean 800 sqft	1	Hour
2	7.9.	Minimum battery charge	300	Ah
3	7.9.	Minimum battery voltage	12	V
4	6.7.9.	Maximum battery weight	100	lbf
5	4.5.6.	Minimum sensor detection range	1	ft
6	1.	Maximum water usage for cleaning	10	Gallon
7	6.	Minimum structural factor of safety	2	NA
8	8.	Cleaning tool removal time	60	s
9	12.	Floor adhesive strength after cleaning	1	psi
10	10.	Sensor placement height in waste basket	2.5	ft
11	11.	Sensor placement height in water container	6	in
12	4.5.	Minimum number of sensors placed along perimeter	4	sensors
13	9.	Maximum battery recharge time	3	Hours

Table 2: Quantity of the Needs

4 External Concepts and Existing Solutions

The roomba vacuum provided some insight into possible sensors that can be used on the winery robot. The roomba uses an infrared beam and photocell sensor to detect objects in front of it. There are also infrared sensors on the bottom of the robot which detect ?cliffs? and on the sides which allow the robot to follow a wall. Roombas are also equipped with a bumper sensor that tells the robot to stop and turn around if it hits something. Furthermore, ?virtual walls? can be set up to stop the roomba from leaving a certain area. These ?walls? are infrared emitters that, when the roomba detects, tell it to turn around.
After searching the internet for other kinds of sensors, there are many possible types that can be used to aide the Winery Roomba in safely navigating the workable environment. Infrared sensors can detect objects and the distances between them by emitting and detecting infrared radiation. Similarly, LASER sensors and optical sensors emit and detect visible light waves. Passive infrared sensors detect warm bodies (e.g. people) that move inside the detection zone. Ultrasonic sensors emit high frequency sound waves (beyond what humans can hear) and receive the echoes

that bounce off objects. RADAR sensors can detect moving objects by emitting microwaves and analyzing the doppler shift of the wave. LIDAR uses a scanning laser to detect objects.
External concepts were also generated for potential ways to improve the method of cleaning on

Sensor	Pros	Cons
Infrared	Relatively accurate Cheap ($10-$20)	Sunlight can negatively affect sensor Color of object being detected may cause poor results Particles in air (e.g. dust) can interrupt sensor
Ultrasonic	Very accurate Can be used inside or outside	More expensive than Infrared Have a minimum detection distance Materials such as cloth absorb sound waves and cannot be detected accurately Susceptible to false trips from loud noises
Passive Infrared (PIR)	Detects motion of people High range ($20 feet) Cheap cost ($10)	Can only detect motion (no stationary object detection) Does not give any data about distance motion is from sensor
LASER Emits	Very accurate and precise over large distances	Expensive compared to other sensors
Light Sensor	Cheap ($10) Simple (Light dependent variable resistor)	Possible interference from ambient light sources Highly dependent on the reflectability of its surroundings
RADAR	Can detect speed and velocity of object	Under control of government on the rank of frequencies Interference with other devices Affected by weather condition Minimum range may be too large for this application
LIDAR	Completes imaging in a timely matter Generate accurate 3D image in large scale	Expensive compared to other sensors Few applications

Table 3: Pros and Cons of Sensors

the WEINbot as shown in Table 4.

5 Initial Design Concept

The first design option is to focus on programming the WEINbot to be completely autonomous. This means adding sensors to allow the WEINbot to navigate the cleaning area without human control as well as developing or finding software to use these sensors.
The second design option is to modify the existing WEINbot. Preliminary testing of the WEINbot needs to be completed to determine its ability to remove grape waste from the concrete floor. Depending on these results, one or more features of the WEINbot may be modified or replaced. Ideas include replacing one or more of the brushes with a rotating mop, and replacing the conveyor with a vacuum unit.

Type	Use	How It Works
Car Wash	Cleans surface of vehicles	For a cloth friction wash, the car is first locked into a conveyer track. The system engages either the front or rear wheel. It then passes through an infrared beam located between two sensors. The sensors send signals to the computer which can accurately determine the length of the car. The car then proceeds onto a pre-soak where small nozzles spray a solution onto the car that lifts dirt. After, the car passes through a mitter curtain that hangs from a motorized frame. The strips from the curtain are moved in a circular motion to create friction and clean the car before scrubbing. The scrubbers are vertical cylinders that rotate anywhere from 100-500 rpm. The cylinders have cloth strips perpendicularly attached to them that cause a whipping motion. The conveyer system then moves the car into a high-pressure washer that has water jets spraying the car at a rate of about 1,000 psi. Finally the car reaches the rinsing arch, where nozzles are arranged in an arch shape so that it could clean off any excess dirt, soap, or debris. The rinse arches are located in various stages throughout the car wash and serve the same purpose.
Roomba	Vacuum cleaner	The Roomba cleans by first detecting the size of the room. It uses an infrared signal and calculates the time is takes for the signal to bounce back. Then it moves in a spiral motion outwards until it hits an obstacle.
Street Sweeper	Picks up leaves and trash, and washes floor	Sweeper uses two brushes that spin inwards. The trash is then moved by a rolling brush onto a conveyer belt that lifts the waste into a containment basket. The original WEINbot replicates this design
Spinning Mop	Same as regular mop except it dries faster making task faster	These mops work the same as a regular mop except rinsing the mop is rinsed done by pushing mop into a strainer that spins as you exert force downwards. The spinning motion and downward force cause the water on the cloth strips to be squeezed out.

Table 4: External Concepts

5.1 Modify the WEINBot

The current WEINbot utilizes different brushes to move the grapes onto a conveyor which drops the grapes into a waste container as shown in the Figure .

A detailed list of potential design adjustments are as follows:
A spinning mop which rotates fast enough to break debris from sticky surface and simultaneously brush the debris towards the back. One possible adjustment to the WEINbot design would be to

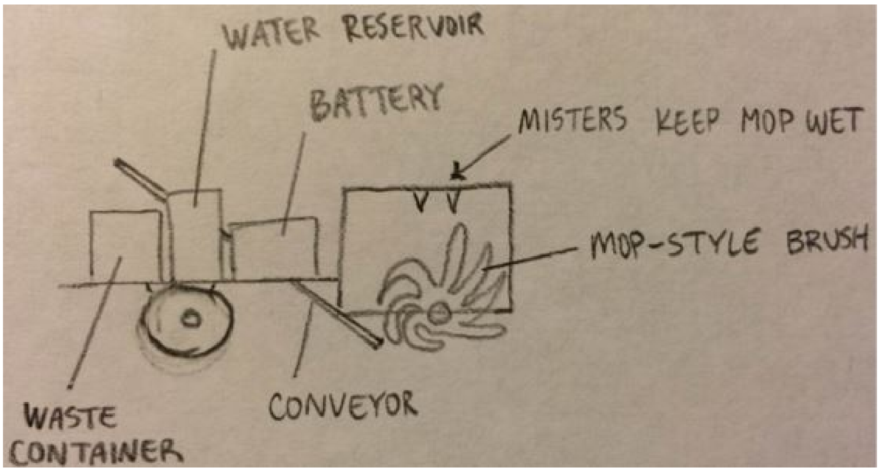

Figure 1: Spinning Mop Concept

replace the conveyor system with a vacuum unit. The brushes in front would push the grape waste towards the vacuum nozzle.
Another concept involves replacing the conveyor with a static waste container. The waste container can be removable with the opening facing the brushes (similar to a lawn mower bag). The brushes will push the grape waste into the waste container
A final adjustment that can be made to the WEINbot is changing the method it applies water. The current misting design would be replaced by a small high pressure concentrated stream. This would better remove sticky residue from the floor and reduce moisture lost to the air.

6 Preliminary Planning and Scheduling

At this point in time, it is important to note that it may not be possible to accomplish all of the final design goals and concepts. As a team, it has been elected that effort will either be focused on developing full automation or enhancing the mechanical components. With this in mind, two schedules have been created detailing each path that can be taken. If the team is able to somehow fully complete one path, then steps will be taken towards completing the second path.

February	March	April	May
Primary Goal: Make final decisions about what modifications need to be done to the WEINbot Tasks: Re-wire WEINbot Preliminary testing and assessment of WEINbot?s ability to pick up grapes Additional: Start practicing coding with python	Primary Goal: Finalize design concept Tasks: Sensor selection Microcontroller selection Assess feasibility of any proposed modifications Additional:	Primary Goals: Complete mechanical assembly of WEINbot (e.g. sensors and any modifications) Tasks: Purchase items Begin coding Develop schematic for wiring electronics Connect and test sensors	Primary Goal: Finalize automation of WEINbot

Table 5: Planning and Schedule

7 Preliminary budget

The preliminary budget for this project is similarly tied to which design path is chosen. Automation development and mechanical enhancement each require different materials. The estimated budget for automation is about $665. This includes:

- Microcontroller $60
- Sensors $50-$500
- Supporting equipment for electronics $100
- Wiring $5
- Batteries $200-300

The necessary amount of money may go down if old electronic parts can be reused. The estimated budget for mechanically upgrading the unit is $735. This includes:

- Brush cloth $20
- Metal $100
- Batteries $200-$300
- Vacuum $300
- Nozzle $15
- Motor $50

The amount of money specified in this report is just an estimate. There is always potential for unforeseen costs that will need to be added to these values.

8 References

1. Karl T. Ulrich, Steven D. Eppinger. *Product Design and Development.* McGraw-Hill International Edition.
2. *HowStuffWorks - Learn How Everything Works!.* Web. 29 Jan. 2016. http://http://www.howstuffworks.com/.

A Apendix

A.1 Software Develop

A.1.1 Sensors

In order for robot becomes smart, the robot must be equipped with lots of sensors. The first sensor is infrared detectors. By putting infrared transmitters and receivers around the robot, continuously sending infrared signals out and checking if the receivers getting any data. Based on the incoming signals of the receivers, microcontroller can control the movement of robot accordingly. The infrared concept is shown in Figure 2

Infrared detector is good in short distance, but weak in detecting long distance objects. The alternative solution for that is LIDAR. LIDAR can be used to get the distance from objects with longer range and more accurate than infrared detector. LIDAR concept is shown in Figure 3.

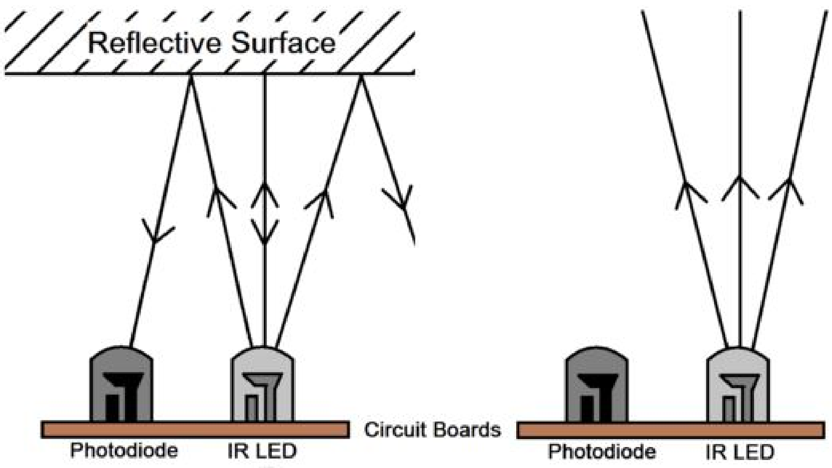

Figure 2: Infrared Detector

The above sensors may be good for robot to avoid obstacles, however, what happen if there is no obstacle at all ? The area of cleaning is open to a road that cars can get in from parking garage and the land field as shown in Figure 4. We decide to paint a yellow line in the open area. Four light sensors will be put under the four corners of the robot as shown in Figure 5.

All of sensors above may be good, but not enough for a stable and safe robot. For example, at 10 feet away, one person goes pass the robot, and LIDAR detects as an obstacle. It is ridiculous if the robot stops when a person just goes pass it. In my own experience, my robot went crazy when my friend secretly use the TV remote control (infrared transmitter) to "hack" my robot's infrared detectors. Therefore, our team decides that the roll of those fancy sensors are to slowdown the speed of robot instead completely stop it. To stop the robot, we wrap four air bags around robot and put four pressure sensors on those air bags as shown in Figure . Normally, air bag will have pressure of 10 psi. If it hits an obstacle, the pressure goes up to 12 psi, stops moving, saves that location into the map as a "unclear possition," then moves to other directions. If the pressure goes

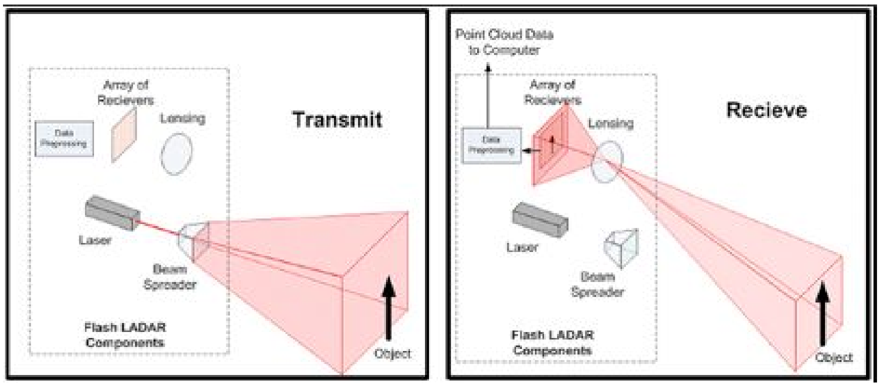

Figure 3: LIDAR Scanning

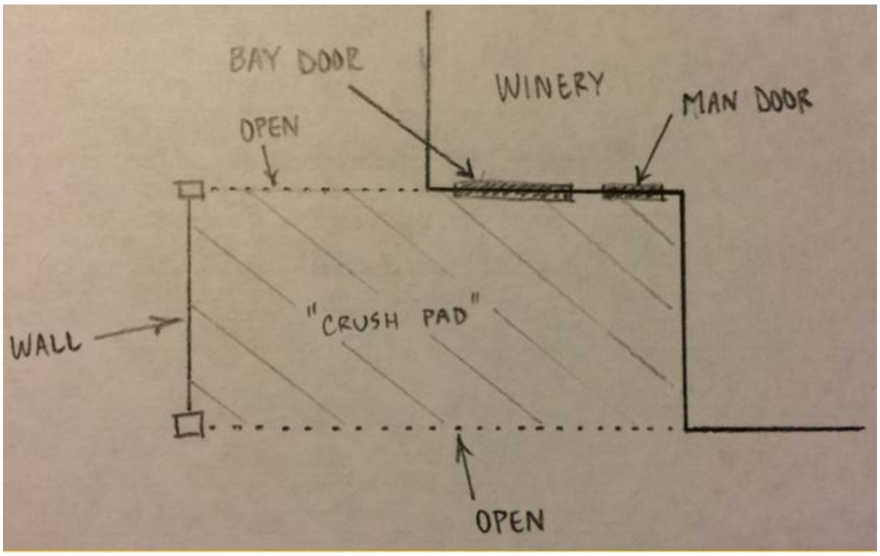

Figure 4: The Robot Needs to Clean the Crush Pad But Cannot Leave This Area While Cleaning

under 10 psi, robot will stop all moving, alert to operator to check air bag pressure. By doing that, the robot is safe if it hits obstacles and stable with noise and undesirable signals.

A.1.2 Mapping

By creating a matrix relative to the area of the floor, in which robot will be put, we can program the robot to learn the different floor area. For example, if the floor is 30'x30', it will be mapped to a matrix 30x30 in the program (assume we will use numpy package for Python programming). Every time robot hits one cell (1 square foot), robot will adds 1 into the value of that cell and move to other cell. Based on the values of cells, robot can decide if that cell has permanent objects, or just a temporary object. We decide that if robot hits a cell three times, that cell is not empty. By knowing which cells have obstacles, robot will avoid touching that cells in advance. Using that algorithm, after three time "learning," robot will have the map of new area. However, we will make robot reset its map memory every 50 runs in order to accommodate the physically changing of the

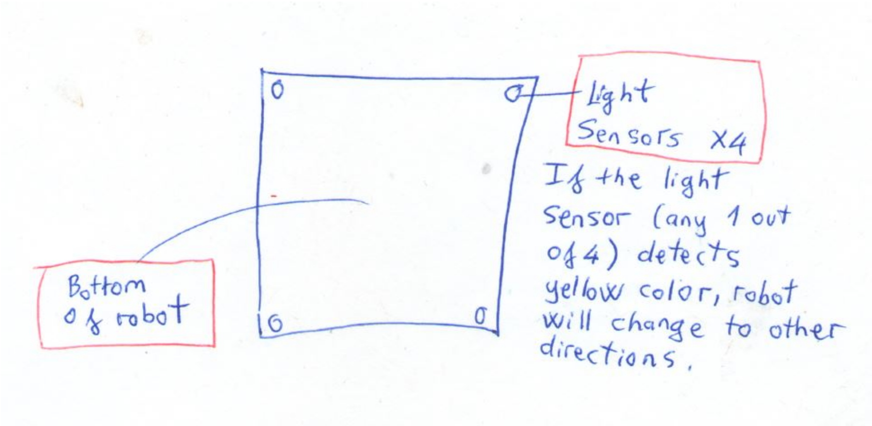

Figure 5: Using Light Sensors for Open Area

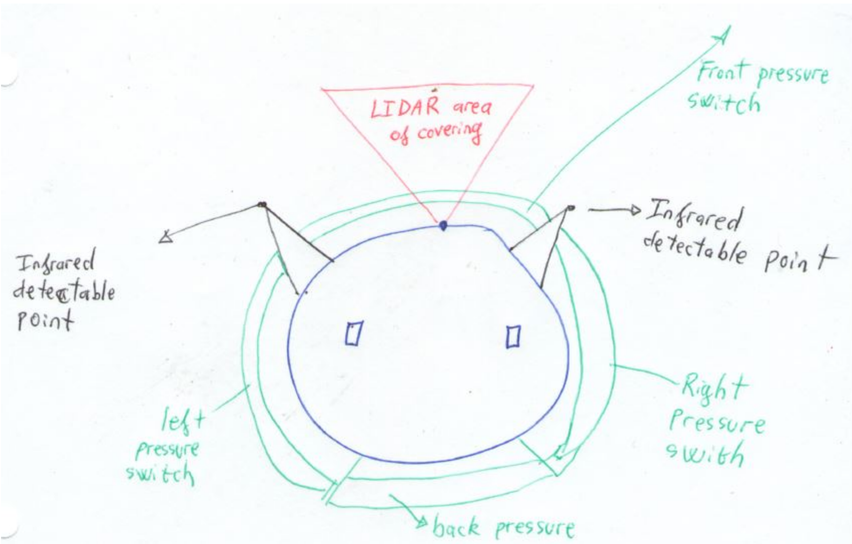

Figure 6: Sensory Planing for the Robot

area. The mapping content can be seen in Figure 7.

A.1.3 Localization

In order for robot knows its current position relative to the map, the robot is equipped a digital compass and a system of ultrasonic transmitters and receivers as shown in Figure 8. Four transmitters will be put on four corners of the floor area. First, the transmitter T_1 sends a signal S_1 to the robot, based on the time delay between sending and receiving the signal S_1, we can calculate the distance d_1. The process repeats for three other transmitters to get distances of d_2, d_3, and d_4. With those four data, we can estimate the position of the robot relative to the mapping matrix in Python program. Together with position, digital compass also give us the current direction of robot relative to the floor area.

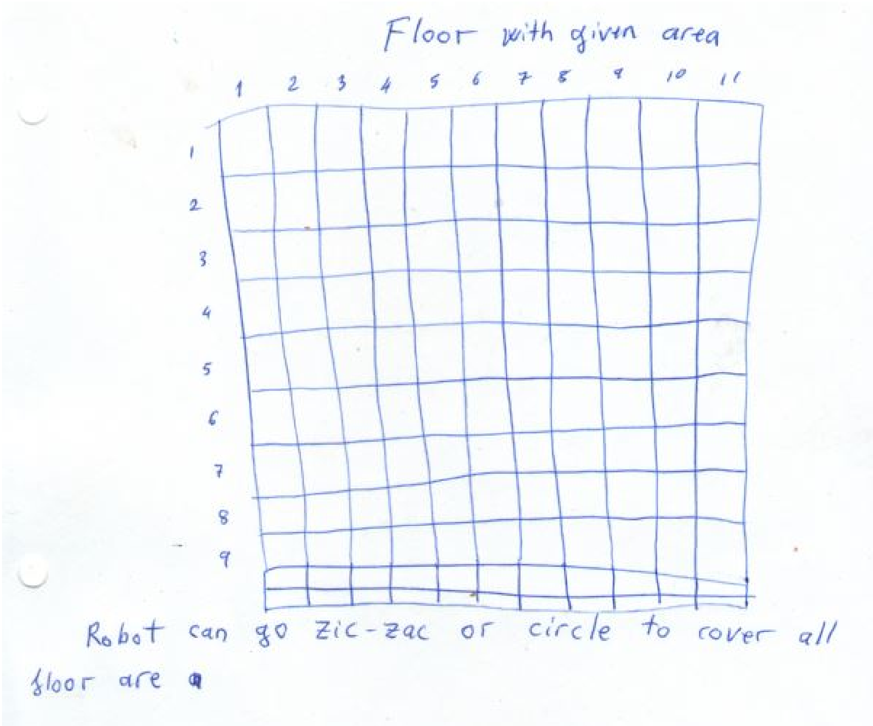

Figure 7: Mapping Floor into a Matrix to Create Moving Path for Robot

A.1.4 Software Structure

We will create a class call Robot using Python language. The main program is a loop of three functions:

```
import Robot as rb
remapping =50 # Robot will reset after 50 runs
rb.initilize() # checking if all sensors working and self-calibration
for i in remapping:
    rb.checkSensors() #Check all sensors, possition, direction, battery level ...
    rb.calculation() #Based on Sensors' data, making calculations and decisions ...
    rb.reaction() # send the output signal to actuators motors, conveyer ....
```

That simplify version program structure will allow our team breaks down the problem and tackle it step by step.

A APENDIX

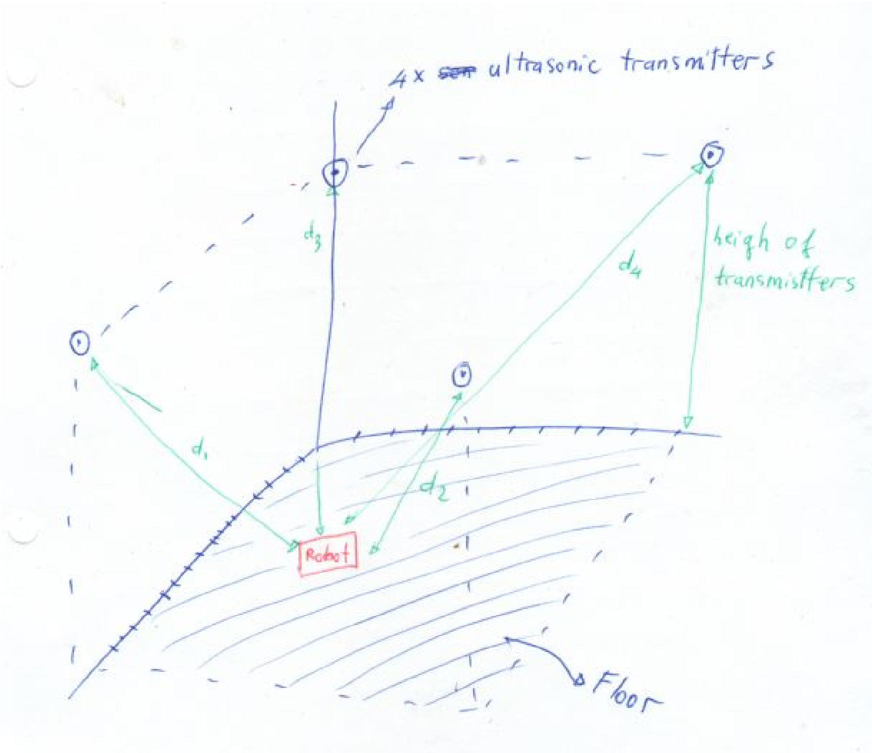

Figure 8: Possitioning the Robot Using Ultrasonic Sensors

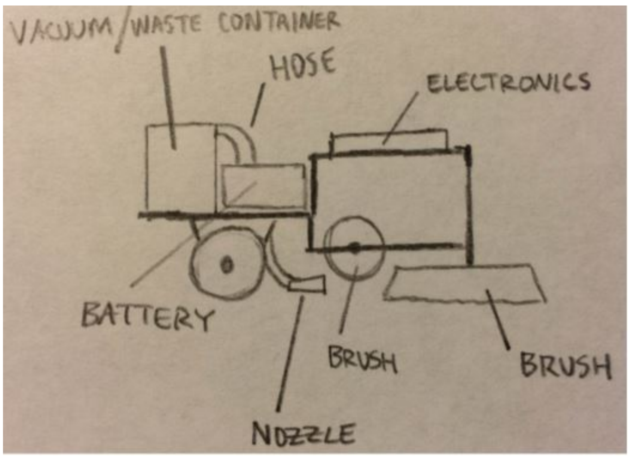

Figure 9: Vacuum System Replaces Conveyor to Pick up and Move Grape Debris into Waste Container

A APENDIX

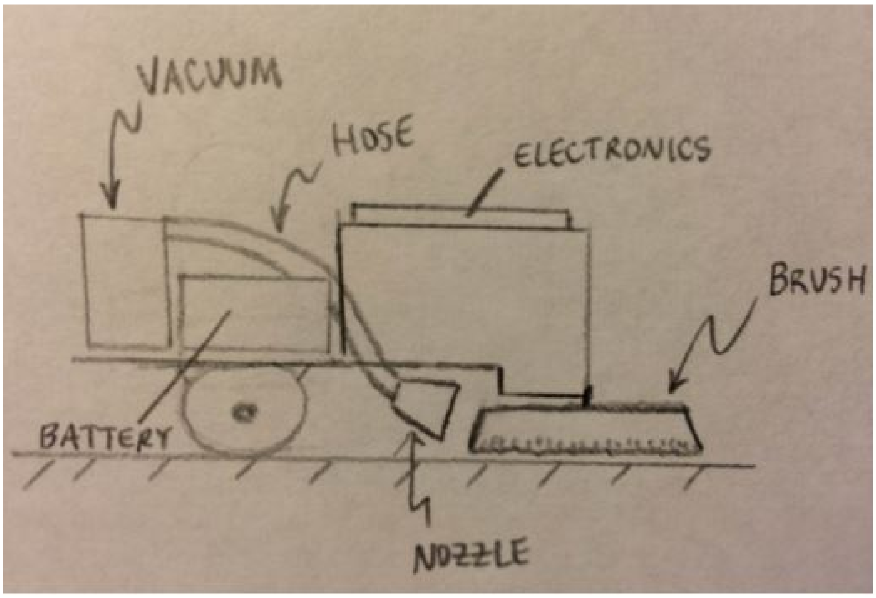

Figure 10: Vacuum System Replaces Conveyor and Central Brush

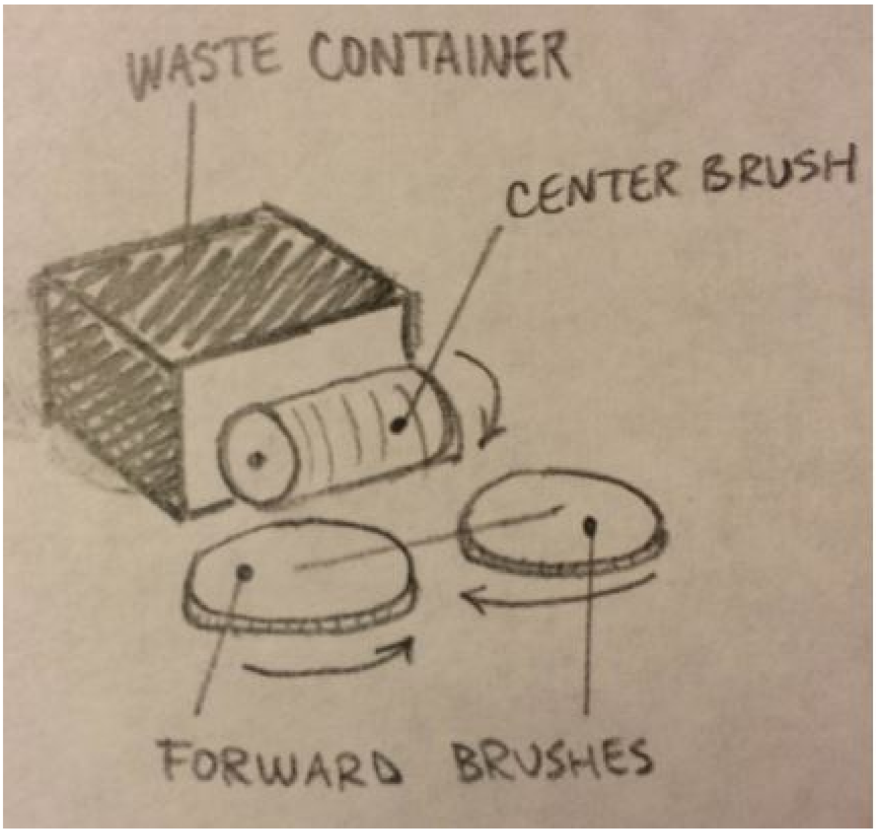

Figure 11: Static Waste Container Concept

A APENDIX

Figure 12: Roomba Sensors and Bumper Location