University of California, Davis

Department of Aerospace and Mechanical Engineering

EME185AB A02 W2016

Preliminary Design: Winery Bot

Author:
Erick Campos
Kyle Cubba
Jerry Li
Dai Tran

Supervisor:
Jason Moore
Matthew Lefort
Farhad Ghadamli

March 13, 2016

Contents

List of Figures

List of Tables

1 Introduction

Wine-making is a messy process. Thousands of grapes must be crushed in a winery using heavy machinery to create wine. Through this process, grape debris, including grape stems, juice, and skins, can fall onto the floor and create a messy and sticky environment. Currently, this mess is cleaned by using a high powered hose to spray the debris down a drain because it is the easiest and most monetarily efficient way to do so. This process, however, wastes a significant amount of water which is costly to the environment. Although water is used in other parts of the wine-making process, cleaning takes up a significant percentage of the total usage and increases the ratio of water volume to wine volume to 6 to 1.

To reduce the environmental impact that wine-making has, the UC Davis Department of Viticulture and Enology tasked our senior design group to design an automated robot that could reduce water usage in the cleanup process. This robot, also known as the WEINbot, would pick up grape waste using a minimal amount of water while also requiring a minimal amount of human interaction. Uniquely for our team, the mechanical components of the WEINbot were manufactured and assembled the previous year. This solved the problem of how to clean up grape waste while using a minimal amount of water. The previous year's team, however, ran out of time and was unable to successfully develop a software that enables full automation.

The scope of our project involves developing a mapping and localization system for the WEINbot so that its operation can become fully automated. This means that we will assume that the previous year's WEINbot perfectly picks up grape waste and that it solves the problem of water usage. Using the mechanical structure and steering layout provided by the already designed robot, our goal is to design a fully automated robot that can best meet more of the sponsor needs listed in Table 1 Since our project focuses on the automation of the WEINbot prototype, some of the sponsor's needs will not be addressed by this project.The table of needs shown above was generated through discussion with our sponsor. Each need is scored based on its importance to the overall project with one star being the most important and three stars being the least. The needs were then translated into target specifications for the robot shown in the Table 2

2 Concept Description

2.1 Mapping

Mapping is the process of creating the accessible world a robot moves within. A properly constructed map is vital for the robot to know where obstacles are, where it can move safely, and what spaces have already been cleaned. For our project, an occupancy grid will be developed and will be used as a map for the robot. In an occupancy grid, the floor plan is divided into a matrix of equally sized squares. Each grid in the map will have a value of zero, one, or two. A zero in the grid represents an open space: there are no obstacles and the robot has not previously moved through this space (it is assumed to be dirty). A one in the grid means that the space has been cleaned previously. A two in the grid means that the robot cannot move into this grid: there is a hazard in this space or it has reached a boundary. The robot will try to move to other open spaces instead of moving over the same spaces over and over. The obstacle spaces (twos) must be different from the cleaned spaces (ones) because, if they were both assigned a value of one, it would possible for the robot to trap itself by being surrounded by cleaned spaces, which can be seen on Figure 1

In order for the robot to begin its operation, a preliminary initialized map will be coded into the robot. This will have information about the size of the floor plan and the locations of permanent

#	Need	Imp.	Met by WEINbot?	Within Scope of our project
1	The robot must clean the area of all crushed grape byproducts and residue	**	MAYBE	ASSUMED YES
2	While cleaning, the robot must operate without human aid	*	NO	YES
3	The robot must avoid hazards on the ground	**	NO	YES
4	The robot must avoid people that may walk in front of it	***	NO	YES
5	The robot must be easily cleaned	***	NO	NO
6	The robot must know when the waste container is full	***	NO	TIME PERMITTING
7	The robot must know when the water reservoir is empty	***	NO	TIME PERMITTING
8	The robot must clean quickly and thoroughly	***	NO	YES

Table 1: Sponsor Needs

obstacles such as walls. The robot's map will consist of a [5x6] matrix of 5x5 ft square grids (thirty total grid spaces). The size of each grid was chosen to be 5x5 ft because having the robot fitting inside of 1 grid makes the mapping and localization easier. All the grids in the matrix except locations with permanent obstacles will start with a value of zero. The map will cover approximately the entire 756 square feet of the winery crush pad as shown in the sketch found in Figure 2.

The robot will move randomly for the first ten minutes of operation. As it moves randomly across the floor, the map will be updated with twos as it cleans each area.
After the 10 minutes of random motion, a large portion of the room will be either 1's or 2's. In order, to make sure every space is visited by the robot, a secondary pathing code will be created that prioritizes the dirty zones and moves the robot in that direction. The robot will divide the room in half and calculate the average grid value in each half of the map. The robot will move towards the side with the lowest value as in Figure 3 (e.g. highest number of zeros).

There is a chance that an open space is surrounded by grids marked as obstacles, making the open space inaccessible. If a conflict such as this occurs, the open space inside of the inaccessible grids will be changed to an inaccessible grid itself. This will prevent the robot from trying to move to a point that it cannot reach as shown in Figure 4.

#	Need # Addressed	Specification	Val.	Unit
1	1. 2.	Maximum time to clean 750 sqft.	1	Hour
2	4. 5. 6.	Minimum sensor detection range	1	ft
3		Minimum distance between robot and object	0.5	ft
4	4. 5.	Object detection sweeping angle	180	degrees
5		LIDAR scanning radius	3.7	in

Table 2: Specification

2.2 Localization

Localization refers to the robot's position within inside its own environment. Localization is important because it makes it possible to obtain necessary information about the robot's movement inside the occupancy grid. For example, with localization, we can determine the distance to obstacles, where the robot has traveled, estimated time for cleaning crush pad, etc. Furthermore, by knowing the active location of the robot during the cleaning process, cleaning patterns can be optimized to ensure the robot performs as best as possible.

The robot will be localized using dead reckoning and landmarks. Dead reckoning uses the robot's speed, orientation, and time traveled to determine the displacement. The speed of the robot can be calculated by multiplying the wheel RPM by the circumference of the wheel. The wheel RPM can be calculated by the input voltage from the motors since they are directly proportional to each other. Distance can be found by multiplying speed by time. Orientation is measured by digital compass. The robot's displacement can be used to estimate which grid it is in.

Dead reckoning, however, is an error prone method. Wheel slippage, for example, can cause a difference between distance travelled and distance calculated. This error grows proportionally to the distance traveled and continually stacks up until the localization and mapping is completely useless and the robot crashes. In order to combat the issue of growing error in dead reckoning, our team will utilize landmarks as a secondary localization technique. Landmarks are points with known positions in the occupancy grid that can interact with the robot. When the robot reaches a landmark, its position and error can be reset to the known position.

Passive RFID tags will be used as the landmarks for the WEINbot. These tags are small thin squares of antenna and microchip with a covering made of paper or plastic and no internal power source. A reader attached to the WEINbot sends out an electromagnetic signal that will induce a current in the RFID chip and cause it to send out a signal. The distance an RFID tag can be read is determined

0	0	1	1
2	0	0	1
2	0	0	1
2	1	0	1

Figure 1: Example Occupancy Grid: 0 means the robot can move to that space, a 1 means the robot cannot, and a 2 means that space has been occupied previously

by the strength of the output signal from the reader as well as the number of coils on the antenna of the RFID chip itself. Using a high frequency RFID it is possible to receive a signal when the robot is within 5 inches of a chip [1]. Unlike for other applications, we desire a short read range so that we will have a much smaller error when the WEINbot triggers one. RFID can also work in harsh environments and can theoretically read a maximum depth of 68mm under salt water [2]. This is significantly deeper than the thin layer that grape juice applies. When the WEINbot triggers an RFID tag, the microcontroller will read that it has triggered an RFID at location (i,j) in the occupancy grid and reset the robot's position to (i,j).
To determine an appropriate landmark density in the occupancy grid, the dead reckoning error must be estimated. Once the robot is functional, we will run the robot in a straight line for one minute and the distance traveled will be measured. The difference between the theoretical distance traveled (wheel RPM*circumference*time) and the actual measured distance is the dead reckoning error. Dividing the dead reckoning error by the theoretical distance will results in an error per unit distance traveled value. This will repeated thirty times. A mean and standard deviation from the error data can be used to create a 95% confidence interval with the t-distribution. The landmarks will then be placed so that 95% of the time, the robot will trigger a landmark before reaching its maximum allowable error.

3 Safety and Obstacle Navigation

Safe obstacle navigation requires a sensor that can detect objects from an acceptable distance. Our initial design used multiple IR sensors on the front of the robot but we quickly realized that this led to a combinatorially increasing number of potential sensors triggering. What this means is that the robot would need to know what to do for each combination of sensors triggering. With only 4 sensors, 16 potential conditions would have to be programmed into the robot. At our programming skill level, this

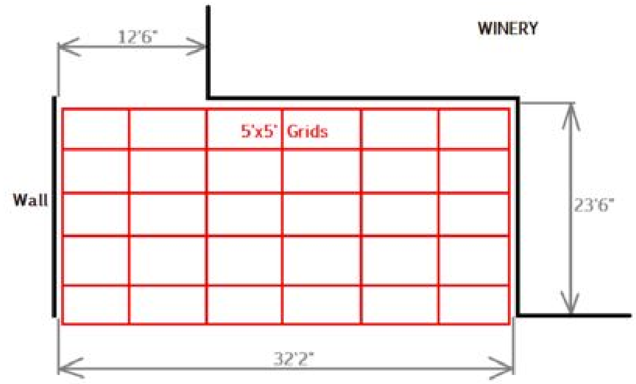

Figure 2: Occupancy grid (red) superimposed on sketch of crush pad (black). Dimensions of crush pad shown in gray.

1	0	0	2	1	1
0	2	0	2	2	2
0	2	0	0	2	2
0	2	2	2	1	2
0	0	0	0	0	0

Figure 3: Example of prioritized pathing. The map is divided into two halves and the robot will move towards the side with the higher density of zeros.

would quickly become an overwhelmingly impossible problem. Multiple sensors also made it difficult to scan in a way that is compatible with our turning radius.Each sensor scans an arc of designated radius and 3 degrees. It is impossible to create a smooth circular clearance area for the robot like that shown in Figure 5.

With these problems in mind, our final concept is to have a single LIDAR directly above the center of the axis of rotation that scans the forward 180 degrees. This LIDAR will be mounted on a servo so that it can rotate. Using this concept it will be possible to set the minimum detection radius to accommodate for the turning radius of the robot as well as an extra distance to allow time for the LIDAR to make a full oscillation and a safety zone for unforeseen circumstances. Adding all of these values together gives us a detection radius of 5ft. Since the robot will only move forward and theoretically always have enough space to make a full rotation, the 180 degree sweep of the LIDAR is enough to navigate safely. When the LIDAR does detect an object, the robot will be instructed to turn in 1 degree increments clockwise until the object is no longer within the sweep area. This will mean that the robot will restart forward motion after the object is at the 180 degree position with

1	1	1	0
1	0	1	0
1	1	1	0
0	0	0	0

Figure 4: An open space (0) surrounded by inaccessible spaces (1) will be changed to a 1 in order to avoid conflicts.

respect to the right side of the robot in the Figure 6.

4 Concept Justification

Using a programmed prototype, we have seen that we can use and apply the information that is output from sensors. The prototype was a simple setup with an infrared sensor and a microcontroller that activated an LED when an obstacle was a specified distance away. This could directly be applied to functions in our robot where the LED could be swapped out with a motor and object detection could turn a motor off. A large portion of our concept depends on our ability to pull data out of the sensors used and this prototype shows that we can do that. Although this prototype was made with an IR sensor instead of the LIDAR, the output from both sensors is the same and thus similar code can be used for both.

To further demonstrate that it is possible to achieve this concept, pseudo-code was written to show how random motion in the WEINbot could be safely achieved. This code is shown in Figure 7. The void loop is a function that will loop until a certain, currently unspecified, condition is met. Each time the loop runs, it will check if the "if" statement is true. If it is true it will execute the commands listed below it. Otherwise the loop will run the "else" statement which means that there is an obstacle and the robot will need to turn until there is no more obstacle detected. The second if statement nested within the first "if" is to insert randomness into the robot's choice of direction. The numbers in this pseudocode are just an example of a potential way to insert randomness into the robot's motion and the exact method has not yet been decided.

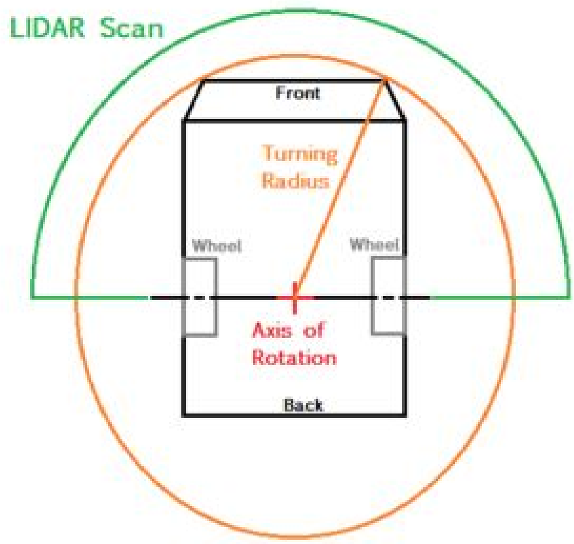

Figure 5: Sketch of WEINbot turning radius and LIDAR scan

5 Planning

In order to successfully execute the design, a strict plan must be followed in the upcoming quarter. Below is a table with dates running through the end of Spring 2016. The table is also divided into 3 categories and a primary goal for the two week period. Also, there is color coded key (EKJD) that shows the initial of assignee(s). Detail on planning is shown on Table 3

6 Financing

Since there is an existing grape cleaning robot, the cost of materials is significantly reduced and can be directed to the autonomous portion of the project. Last years senior design group also purchased and installed a LIDAR sensor and a pressure sensor bar, which also cut down the cost for this year's project. Therefore, the majority of spending will be on the new infrared sensors and RFID tags, both essential components to the mapping and localization. The Table 4 shows an estimated total cost of the materials required.

The Table 4 shows an accurate estimate of the cost of materials needed for the winery roomba project. The metal required for the battery cover might be slightly less but will depend on type of material and availability. The sponsor for this project will be able to cover all of the costs as there is an existing budget of $4,000 and according to the estimated bill of costs, less than $300 is needed.

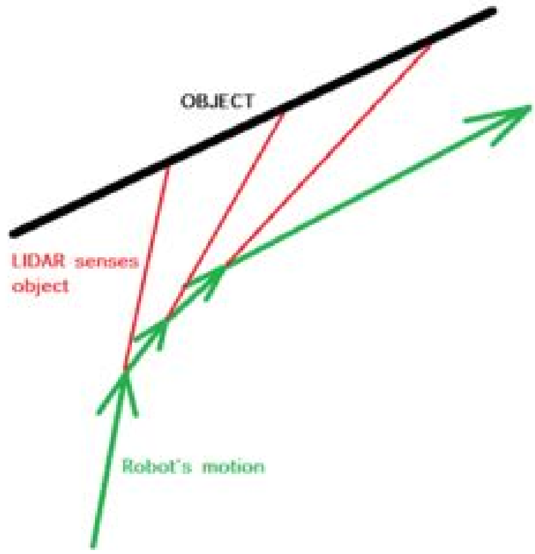

Figure 6: When the LIDAR detects an object in front of the robot, it will signal the robot to turn in one degree increments until there is no object obstructing the robot's motion.

7 References

References

[1] Barcodes Inc. (2016). *Choosing the Right RFID Technology.* `https://www.barcodesinc.com/info/buying-guides/rfid.htm.`.

[2] Giuliano Benelli and Alessandro Pozzebon (2013). *RFID Underwater: Technical Issues and Applications.* Univ. of Siena, Siena, Italy DOI: 10.5772/53934

```
void loop() {
  if LIDAR outputs value>5ft
  i=i+1
      Motor high
      Gather new LIDAR information
      if i divisible by random number between 7 and 12
        turn some random angle


  else Lidar outputs value<5ft
      Motor low
      rotate robot 1 degree
      Gather new LIDAR information
```

Figure 7: Priliminary Code

	March 14-27	March 28-April 1	April 2-17	April 18-30	May 1-15	May 16-31
Primary Goal	Spend time researching selected methods to further understanding	Continue building prototypes and wire WEINBOT	Connect as many components as available and begin running tests	Statistical analyses for robot movement and place landmarks throughout winery	Have a successful run where robot moves autonomously	Robot moves effectively in boundary. Improve grape pickup process
			E, K, J,D			
Learning	Dead reckoning method for robots	Beaglebone reading to connect	Improving robot movement	Programing multiple sensors on beaglebone	Programming map into robot	How to optimize robot
	E, K, J,D	E, K, J,D	E, K, J	E, K, J,D	D	
Purchasing	Finalize list of exact sensors and components we will order	Begin purchasing components needed	Order any missing parts	Last minute orders should be placed by this time	Grapes to test functionality of robot in crush pad	
	ED	E	E	E		
Engineering	Build small prototypes to test with microcontroller	Making a schematic for building and calibrate wheels	Analysis and testing for robot sensors	Use microcontroller to sync sensors with robot movement	Boundary and matrix are finalized	Optimize robot
	D	K, J	E, K,	D	E, J	E, K, J,D

Table 3: Planning Schedule: Erick-Green, Kyle-Blue, Jerry-Orange, Dai-Red

Item	**Cost**	**Quantity**	**Subtotal**	**Tax(8.5%)**	**Shipping**
Infrared Sensor	$15	7	$105	$8.92	$7.30
RFID Tags	$2	20	$40	$3.40	Free
RFID Reader	$35	1	$35	$2.98	$12
Cable	$10	n/a	$10	$0.85	Free*
Metal for Battery	$3 per lb	15 lbs	$50	$4.25	Free*
* Will buy locally		Total:	$240.00	$20.40	$19.30
		Grant Total		**$279.70**	

Table 4: Financing

#	Need	Imp.
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 5: Full List of Sponsor Needs

A Appendix

A.1 Needs and Specifications

A table of the sponsor's needs was generated at the beginning of the quarter. After the decision was made to continue development of the WEINbot instead of designing a new prototype. The list of needs was shortened to reflect the scope of our project is shown in A.1.

Similarly, a list of specifications was developed for the complete list of needs. The WEINbot prototype met most of the design specifications regarding the electromechanical components.

A.2 Concept Screening

In order to choose a concept for our project, a table was made and certain aspects were rated: "1" is a positive score, "0" is a neutral score, and "-1" is a negative score. Table 7 shows a comparison of the top three mapping options and top three localization options.

The three mapping options are bumping into objects, gray-scale occupancy grid, and a 0-1-2-occupancy grid. The "bumping" method requires the robot to sense an object by physically hitting an obstacle (vacuum roombas use this method). This method would require sensors, such as strain gauges, piezoelectric gauges, or pressurized air bags in the front and sides of the robot. The gray-scale occupancy grid would fill the grid spaces in the map with a value between 0 and 1 instead of a simple true or false. This allows greater accuracy and each grid can have a degree of accessibility. The 0-1-2 occupancy grid is the concept described in this report; each value refers to an open space, obstacle, or previously cleaned space.

The three localization methods considered were an ultrasonic transmitter system, dead-reckoning, and starting at same position. Ultrasonics involves the use of high frequency emitters and receivers to pinpoint the robot's location in the map. There would be three or more transmitters mounted on the ceiling of the crush pad and receiver mounted on the WEINbot. The distance data collected from the ultrasonic sensors are used to triangulate the robot's position. Dead-reckoning uses the robot's velocity and time to calculate displacement. The "starting at the same position" method would require a path to be preprogrammed into the robot. The robot would have to be placed at the same

Sensor Type	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
LASEREmits	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	Very accurateCompletes imaging in a timely matter Generate accurate 3D image in large scale	Expensive compared to other sensors
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 6: Specifications of Winery Floor Cleaning Robot

Selection Criteria	Concept					
	Mapping			Localization		
	1	2	3	4	5	6
	Bumping Into Objects	Gray Scale Occupancy Grid	0 1 2 Occupancy Grid	Ultrasonic Transmitter System	Dead Reckoning	Start at Same Position
Ease to Develop	1	-1	1	0	1	1
Safety	-1	1	0	1	0	-1
Avoiding Obstacles	-1	1	0	1	0	-1
Low Cost	0	-1	1	-1	1	1
Accuracy	1	1	0	1	0	-1
Effectiveness	-1	1	1	1	0	0
ADD 1's	2	4	3	4	2	2
ADD 0's	1	0	3	1	4	1
ADD –1's	-3	-2	0	-1	0	-3
Net Score	-2	2	3	3	2	-1
Rank	3	2	1	1	2	3
Continue?	NO	MAYBE	YES	YES	NO	NO

Table 7: Concept Selections

starting point each time it was turned on in order for it to follow the same path.

For mapping, the 0-1-2 occupancy scored the best. We chose to stick with this concept because it is relatively simple and easy and will be sufficient for our application. The bumping method would be too dangerous given how big the robot is and the gray scale occupancy grid was deemed too difficult to complete in our given time frame.

For localization, the ultrasonic system scored the best, but we decided to forego this for dead reckoning in order to give ourselves a realistic project to complete this Spring. We added the landmark concept to dead reckoning after this comparison table was generated in order to improve the positioning accuracy. The landmarks do not require a significant increase in workload to implement.

A.3 Pictures

Figure 8: Crush Pad at UC Davis Winery

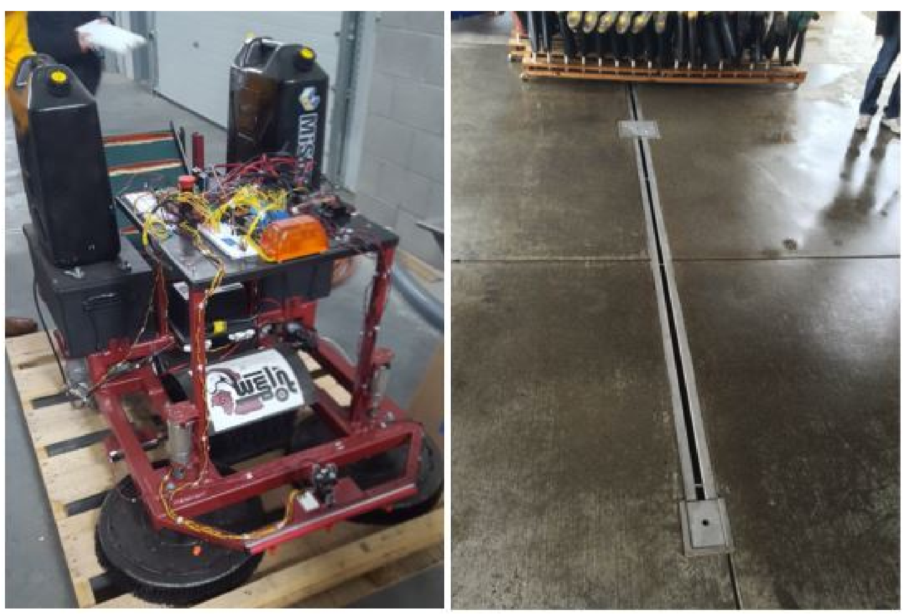

Figure 9: Left: Existing WEINBOT robot. Right: Drain used to collect grape debris