Sameen Mahtab (260737048)

Chen Liu (260776963)

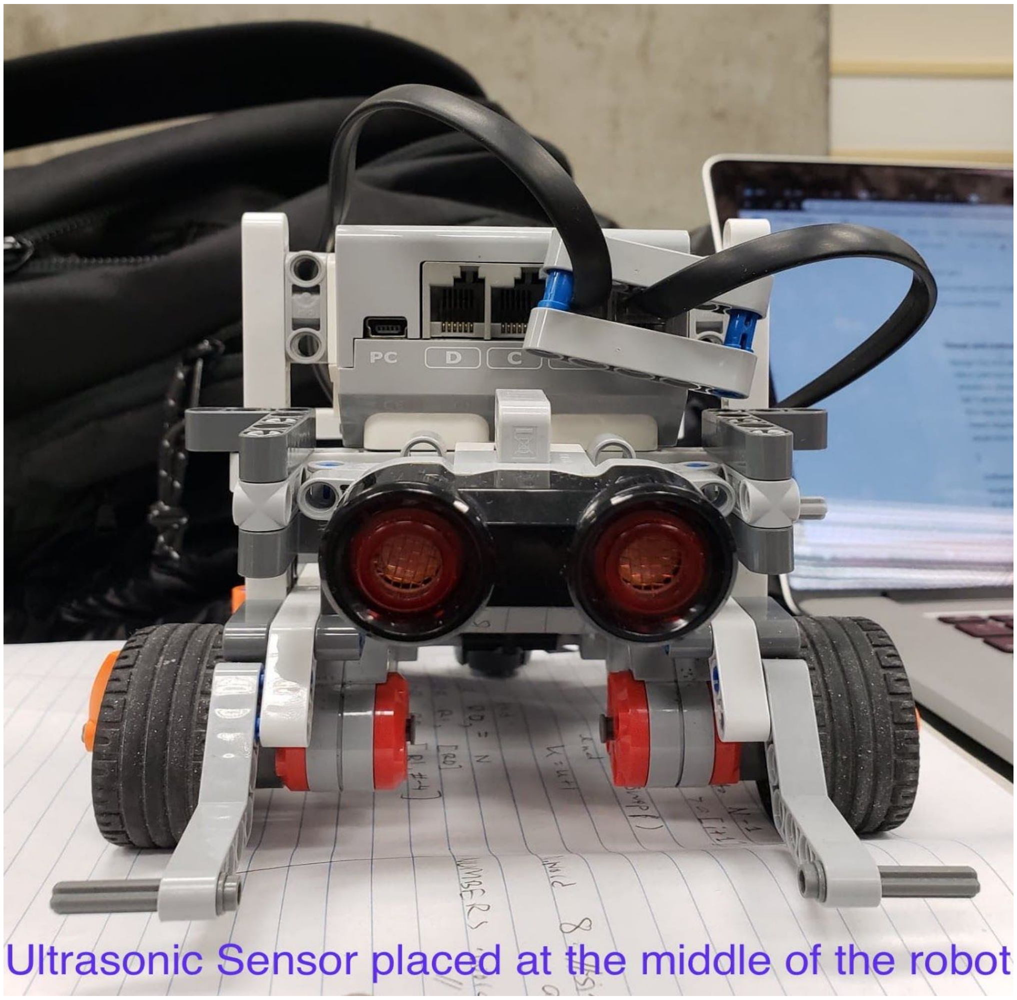
Lab 3

<u>Design and evaluation:</u>

Design: For this lab the main sensor used by our robot is the Ultrasonic Sensor. The robot uses two wheels with motors and one wheel used to balance the robot. The ultrasonic sensor is fixed close to the body at the center of the robot in contrast to lab 1 where the Ultrasonic sensor was fixed at the front left corner of the robot. This was done because in this lab the position of the obstacle is random so our robot requires a wider field of view. The sensor is held close to the body in order to attain the most accurate measure of distance between the robot and the wall.

Software: In the navigation class, the way points are fetched to calculate the current location of the robot and using trigonometry, the travelTo() class finds the distance that the robot has to travel to reach the next point is calculated. The odometer class runs in the background. Then the turnTo() method in the navigation class calculates the angle to which the robot has to turn to in order to face the next waypoint and subsequently rotates the robot. The the robot is moved to the next way point.

As for avoiding the obstacle, we added a condition where if the distance between the robot as returned by the Ultrasonic Sensor is less than 6, the robot will stop, perform a 90 degree turn go straight, perform another 90 degree turn, go straight again turn once more, at this point the robot is at the other side of the obstacle. The robot again turns towards its waypoint and keeps on travelling towards it.

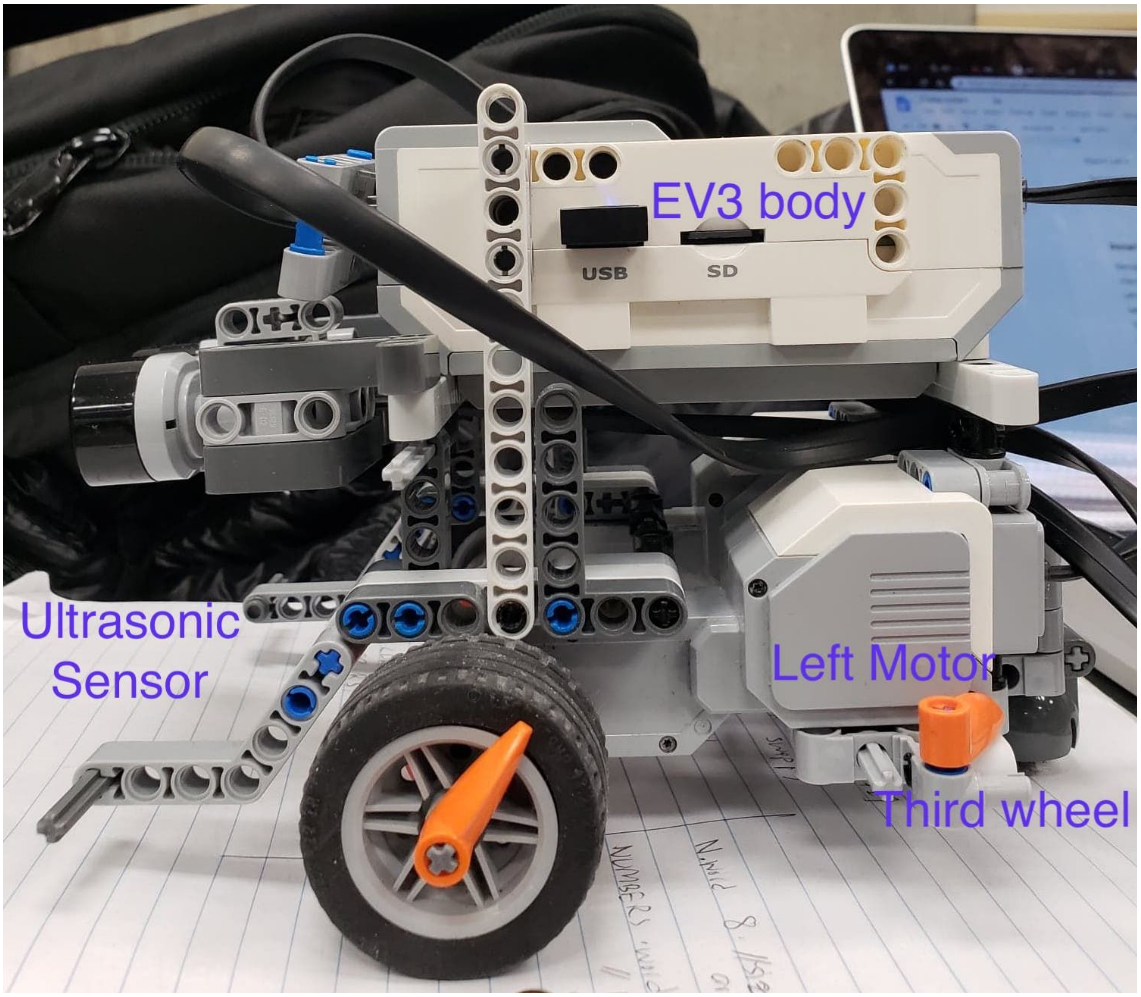


Ultrasonic Sensor placed at the middle of the robot

Formulas to calculate the minDistance and angle:

$$\text{minDistance} = \sqrt{(X - Xf)^2 + (Y - Yf)^2}$$

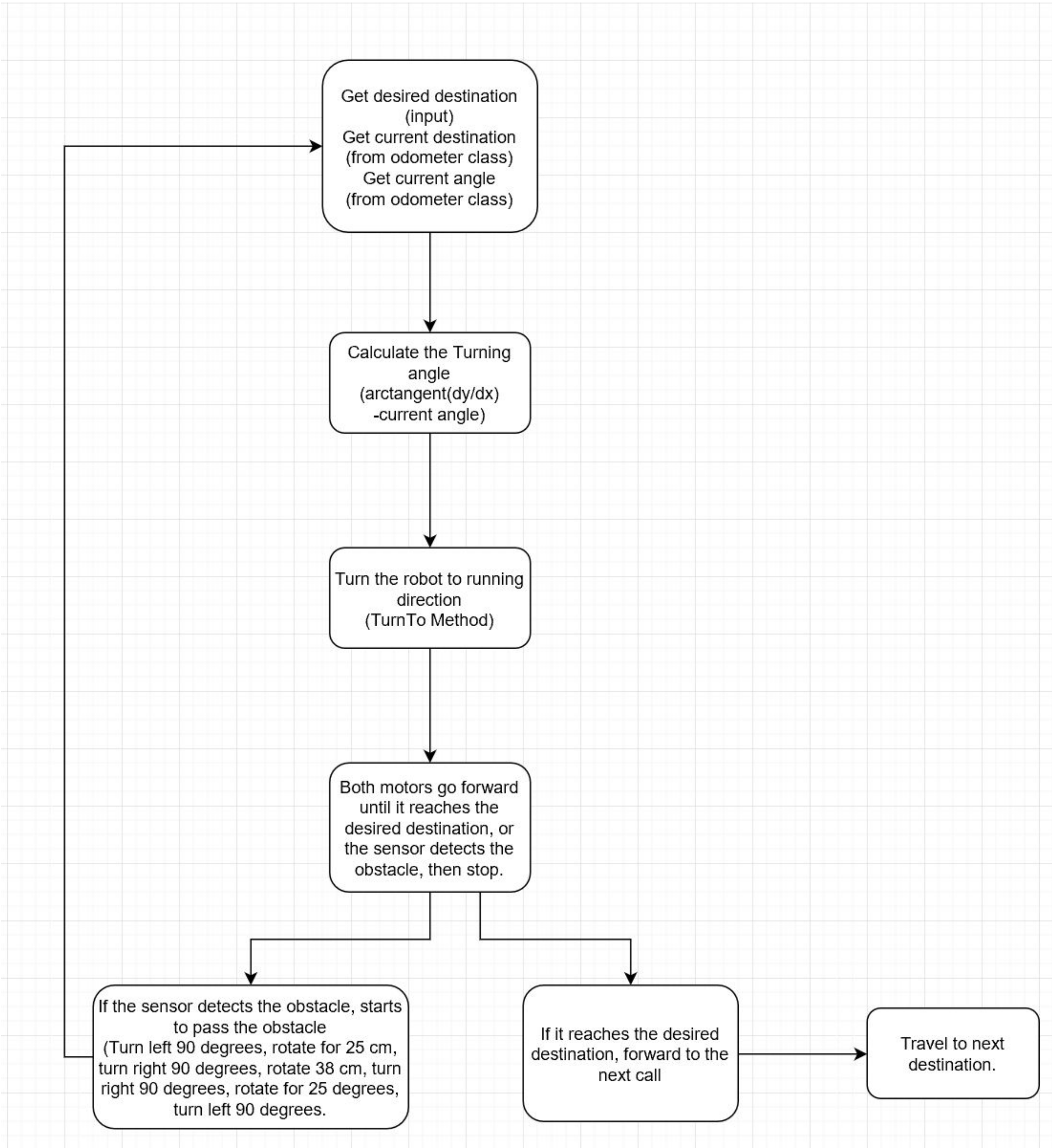
$$\text{Angle} = \arctan(\text{delta } X/\text{delta } Y)$$



The formulas used to calculate the distance and the angle are given below:

$minDistance = \sqrt{(X - Xf)^2 + (Y - Yf)^2}$

$Angle = \arctan(\text{delta } X / \text{delta } Y)$


Get desired destination
(input)
Get current destination
(from odometer class)
Get current angle
(from odometer class)
Calculate the Turning
angle
(arctangent(dy/dx)
-current angle)
Turn the robot to running
direction
(TurnTo Method)
Both motors go forward
until it reaches the
desired destination, or
the sensor detects the
obstacle, then stop.
If the sensor detects the obstacle, starts
to pass the obstacle
(Turn left 90 degrees, rotate for 25 cm,
turn right 90 degrees, rotate 38 cm, turn
right 90 degrees, rotate for 25 degrees,
turn left 90 degrees.
If it reaches the desired
destination, forward to the
next call
Travel to next
destination.

Test Data

10 independent trials were conducted to travel to the specified way points.

Trial	Xd(cm)	Yd(cm)	Xf(cm)	Yf(cm)	Error(cm)
1	60.96	0.00	64.29	3.44	
2	60.96	0.00	61.21	2.53	
3	60.96	0.00	63.53	4.28	
4	60.96	0.00	65.86	4.23	
5	60.96	0.00	62.49	2.35	
6	60.96	0.00	63.74	3.62	
7	60.96	0.00	63.95	1.97	
8	60.96	0.00	62.72	1.50	
9	60.96	0.00	61.74	4.00	
10	60.96	0.00	64.36	3.20	

The error is calculated using:

$\in = \sqrt{((X\mathrm{d} - Xf)^2 + (Y\mathrm{d} - Yf)^2)}$

Mean	3.1
Standard Deviation	.98

Formula to calculate mean:

$$\bar{x} = \frac{\sum x}{N}$$

=Summation of each error/number of trials

Formula to calculate Standard Deviation:

$$\sqrt{\frac{\Sigma(x-\bar{x})^2}{(n-1)}}$$

Observations and Conclusions:

Q: Are the errors you observed due to the odometer or navigator? What are the main sources?

A: The navigation class calls the odometer class, this is where the main error occurs. The values of current X,Y and theta can be inaccurate for several reasons; the wheels might not be completely parallel with each other, the wheels might skid, the radius if the wheels are not exactly what they were measured to be. The motors may have different delays when starting and stopping. The further the robot goes the errors keep on aggregating.

Q: How accurately does the navigation controller move the robot to its destination?

A: Although the navigation class moves the robot to the target coordinates, the robot is not perfectly at the coordinates most of the time. There would always be some error between where the robot actually stops and where the robot was supposed to stop.

Q: How quickly does the robot settle (stop oscillating on its destination)?

A: The robot settles almost immediately when it reaches the final point. The robot was not always accurate when stopping at the correct point.

Q: How would increasing the speed of the robot affect the accuracy of your navigation?

A: The main source of error for the robot is the skidding of the wheels on the surface. Increasing the speed will further increase this error. The wheels will skid even more due to more rapid acceleration.

Further improvements:

Hardware:

The ultrasonic sensor can be implemented in a way that makes it rotate from side to side. This will consequently increase its field of view and will improve barrier detection.

Better wheels could be used with more traction and a wider profile could be used. This will help reduce skidding if the wheels.

Software:

Software could be added to make the ultrasonic sensor swivel from side to side. The movement would have to be fast so that a blind spot ever exists for too long.