ME 425 Post Lab Report 4

Bati Berk Ozata,Demir Demirel, Sinan Ibrahim Bayraktar

***Abstract*— This report include implementation of range finder sensor made with Lego Mindstorms^{TM} EV3 IR sensor and discussion on the test results gathered with MATLAB on real environment.**

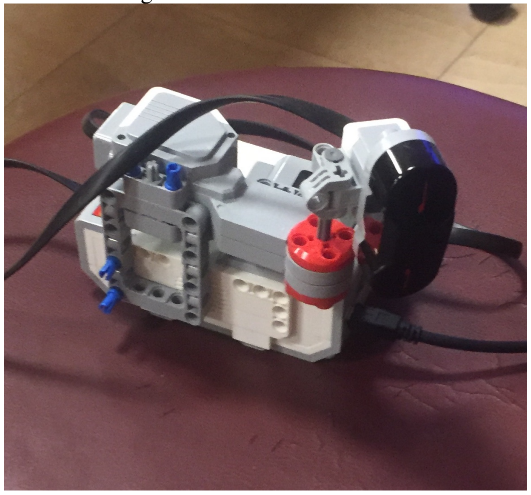
I. INTRODUCTION

In this laboratory exercise, we built a motor and an infrared sensor on top of it. We calculated at first to understand the units as cm. We programmed the motor to rotate 180 degrees back and forth. The sensor is used as a proximity distance mode and we used the data we gathered to map the obstacles around the system we built. We tried several different obstacles varying in shape, color and material. And those factors affected the sensor values.

II. PROCEDURE

We started the lab by building the robot using small number of components. The most important component is the infrared sensor. It is connected to a motor that can rotate 180 degrees. We used this system a range finder to detect the obstacles around the sensor within the limit of the sensor. When we were finished with building the robot, it looked as follows. In order to

Fig. 1. Polar Coordinate Transformation



detect the obstacles in an area, we have to locate them in a coordinate system using specific methods. We used a 2D coordinate system and we used two different methods to locate them in it. First method is the polar coordinates which is represented by an angle from the x axis and the distance from the origin. We can get these two information using the sensors in our robot. Infrared sensor will give the distance from the coordinate system center which is assumed to be the location of the sensor. Also, using the time when the infrared sensor detected an obstacle, we can obtain the rotation angle from the motor. So by using these information, we can convert it to the (X,Y) cartesian coordinates which is our main coordinate system to store the obstacles. To do that, we used two simple equations:

$$x = dcos(\theta) y = dsin(\theta) \quad (1)$$

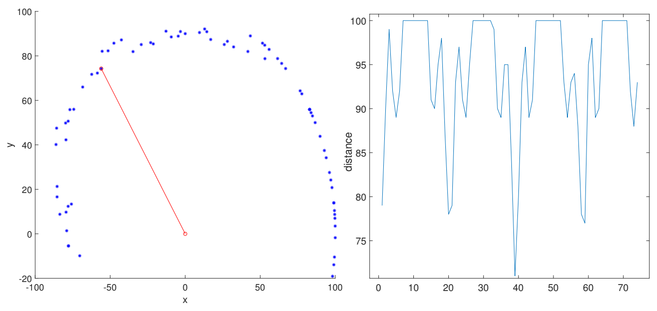
Now we can store the (x,y) coordinates of each obstacles and plot them. Next, we applied this mathematical model our algorithm. After we completed the construction, we started writing the code for the robot. First, we defined our robot, infrared sensor and motor. We set the motor speed to the speed variable and started the motor. Before entering the big loop of the algorithm, we defined our in-code variables which store the distance, rotation, x y positions and set the time step for each iteration which controls the resolution of the data. We reset the rotation measurement in order to obtain the angle difference starting from 0 degree. In the big loop, first, we got the measurements from the infrared sensor and motor encoder and, stored them in an array. Then by using the equations, we converted them to the cartesian coordinates. For each encoder reading, we checked whether motor reaches the 180 degrees or not. If yes, motor changes its direction and checks for the 0 degree. We plotted each obstacle after the sensor detected and showed it with a line as well. To complete one loop, we paused the system for the entered time step and iterate through the all process until the up button is pressed. Lastly, we saved the information gathered during the process in a mat file. When we were complete with the algorithm, we started the experiments with figuring out the relation between our measured unit and real-life distance in cm. Thus, we can calculate the multiplication factor between two measurement units and find the real distance in cm.

Next, we tested our algorithm with several different situations. In the experiments, we tried to understand the effects of obstacles, shape of the obstacles, color of the obstacles and the speed of the motor on the results. For each case, the tested our algorithm and understood the relationship between them. In the results part, you can see different experiments and their effects on the results.

III. RESULT

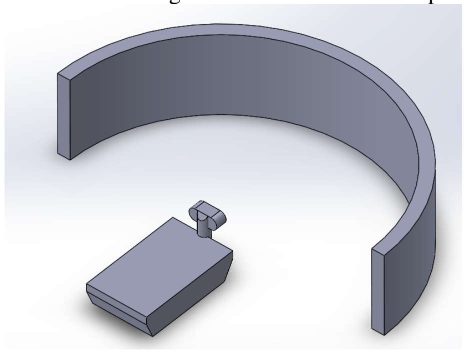
In this lab we observed the range finder sensor experiments in MATLAB. We create different scenarios with obstacles and colors. In the figure 2 we can observe that the empty measurement without any obstacles. We get the results with measure in the air. The distances we were observed fluctuate between 100 and 75 in the graph. Since the sensor is very primitive and the technology does not produced for measuring the distance, the fluctuation is very normal. But when we look at the x,y graph, the range is look like semicircle and without looking the distance,time graph it would seems like perfect sensor. Without any obstacles we

Fig. 2. Measurement without obstacles



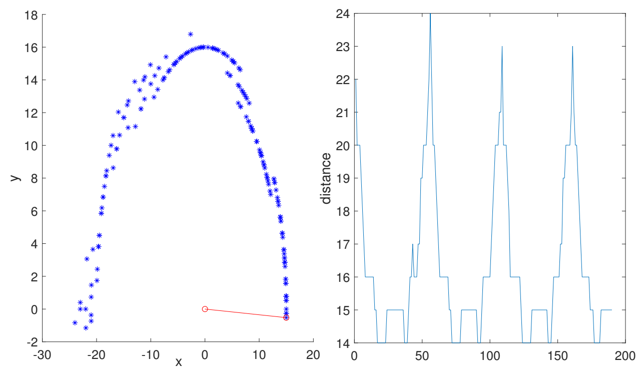
made some observations but we need to observe one more circular observation with closer object that should create a semi circle. The figure 3 shows us the obstacle

Fig. 3. Obstacles of the experiment



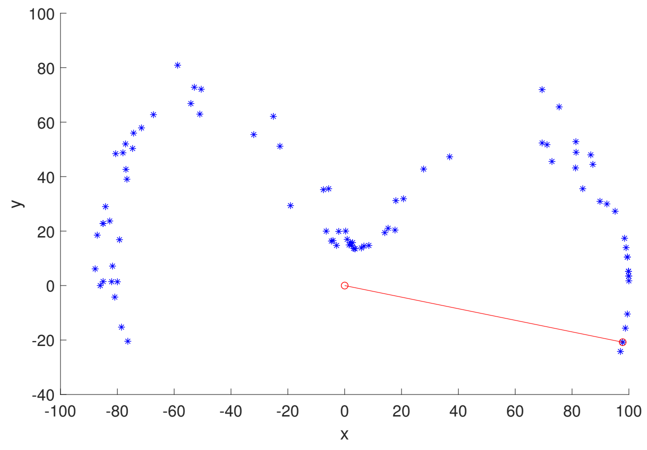
that we use in the experiment. When we look at the graphs on the figure 4 the shape look like semicircle which is what we expect. The distance, time graph shows us the error is 10 cm which is very small value. And the measurement looks very smooth.

Fig. 4. Measurement with closer circular obstacles



The circular object and there is no obstacle scenarios are discussed in previous part now we need to find out what happens if we change the Δ t. In the graphs below, figure 5 represent the data gathered with 0.1 second. In that figure, density of the blue points is less than from the figure 6. We can easily observe the more data points.

Fig. 5. Measurement time= 0.1



Since we get the data from different coloured obstacles, the data shown in the figures below.

In the white object we can see the minimum distance as 10 which is the location of the white obstacle in the front side.

Fig. 6. Measurement time =0.001

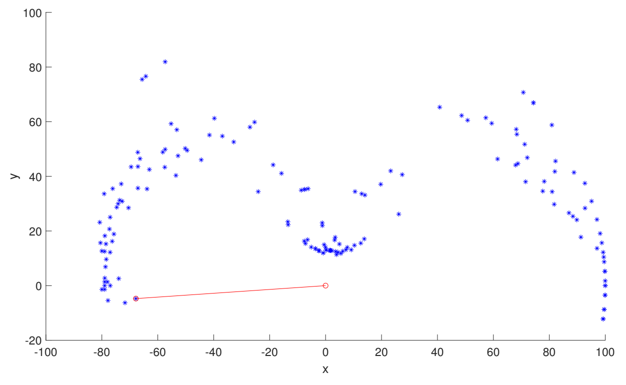


Fig. 7. Location of Obstacles

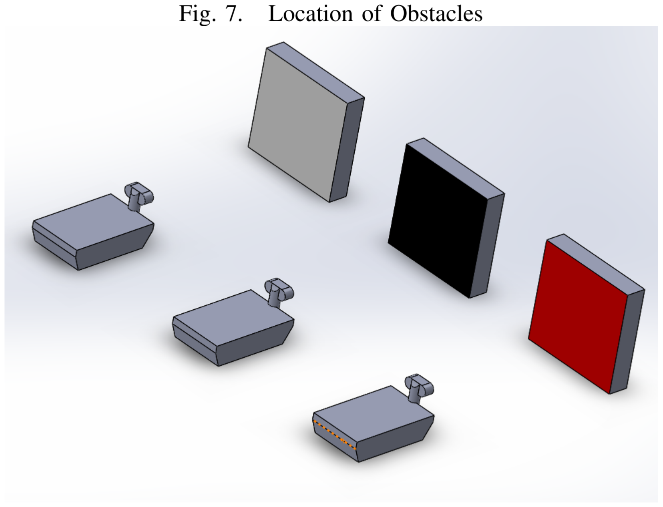
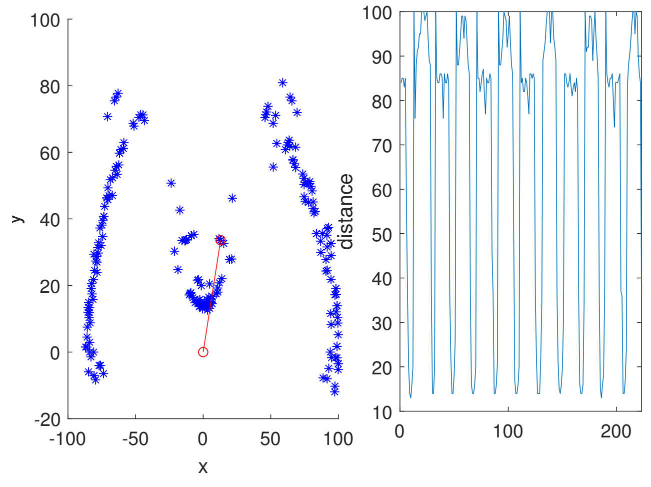


Fig. 8. With White Obstacles



With the black obstacle the distance seems like 30 while there is a same obstacle with different colors.

In this part we obtain use same objects that has same dimension and location and the results shows us the

Fig. 9. With Black Obstacles

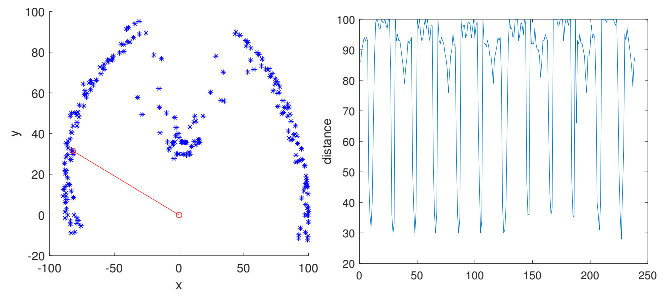
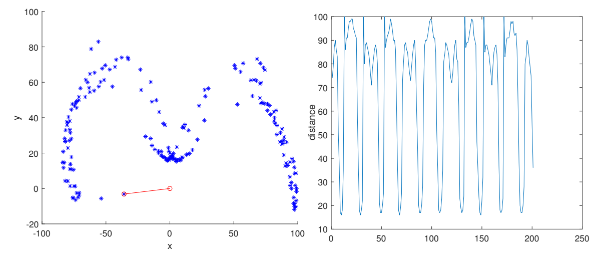
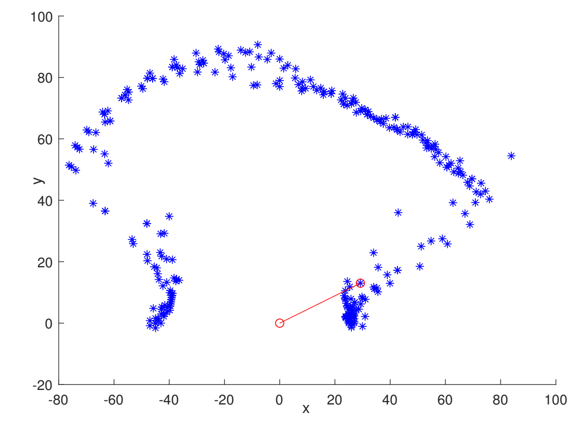


Fig. 10. With Red Obstacles



color affect the measurements. This is a expected result since the IR sensor is just send the light and take the time of reflection. But the thing what makes this system is unreliable is absorption of the colors. Since some of the lights were absorbed the time interval that signal returns may vary. And what we observe in those three figure is exactly this.

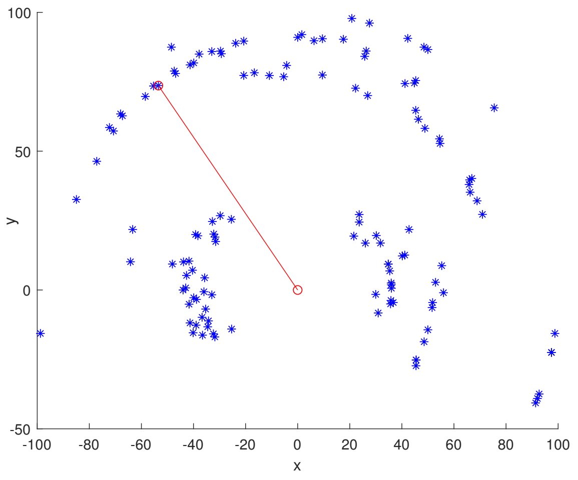
Fig. 11. Speed= 7



A. CONCLUSION

The range finder with IR Sensor of experimental platform LEGO MINDSTORMS EV3 is described in this lab report. The data gathered from the experiment discussed and reviewed in terms of the sensors and

Fig. 12. Speed = 50



the environmental factors. The theoretical expectations such as the absorption effects on different colours and shapes was satisfied well.

IV. Individuals

A. *Demir Demirel*

In this lab we are asked to do some test with the IR sensor using as a distance sensor with the data gathered with MATLAB. About the sensor: this sensor would not produced for distance measurement. The working principle is very different than the laser or sonar sensors. It transmit the light and receive the reflection of that light signal. When we come up to sensors there are some specifications that makes differences in the sensor. Resolution, accuracy, precision are the main consideration for the sensor selection. About the IR sensor resolution could be change with the various speed. We can observe this relation in the Figure 11 and Figure 12. The resolution in the 11 is very higher than the speed with 50. The other thing is the accuracy. Which is depends on the ambient light in the lab or the surface. From the definition of accuracy the data points were uncertain when the point that there is a change in shapes. And we can conclude that the sensor is not accurate. The other aspect is reproduce-ability which means precision. We can only decide whether its precision is high or not with looking the Figures 8 and 9, the edges of the graphs could represent precision is not too bad but not good as well. As we know from the high school physics courses, in summer we don't dress black cloths since black is absorb the light and the energy but we dress white cloths since it reflect the light. When we combine this old information with the experiment that we done in lab: the black object absorb the light and it makes the delay in reflection. The figure 8,9,10 could be evaluate with that idea. In white object we seen it reflect very fast and get the small distances while with black object we see, it absorb the light and the reflection is not fast and determine distance as long.

B. *Sinan Ibrahim Bayraktar*

This lab was about building a range finder system using the LEGO MINDSTORMS EV3 robot and its components. We used an infrared sensor and a motor as well as the robot. The idea was to cover a range of 180 degrees in front of the sensor within the maximum distance that sensor can read. We plotted the results for different configurations and parameters and, tried to understand their effects. After several experiments discussed in the Results section, we understood that the sensor does not work perfectly so, we should prepare the optimal environment in order to obtain better results. Also, we have to make a trade off between the precision and computational cost. Decreasing the timestep gives more precise results; however, MATLAB will have hard time to compute each timestep. Also, as we made experiments with and without the obstacles and, we saw that there is an outer ring at the plot that shows the range of the infrared sensor. When we added obstacles to the environment, we first measured the conversion coefficient between the unit in sensor and cm. Then we analyzed the color of the obstacle effect, shape of the obstacle effect, distance from the sensor effect and motor speed effect. As mentioned, we should prepare the environment considering these effects beforehand in order to obtain the best result.

C. *Bati Berk Ozata*

In this lab assignment, we created a system that calculates the range using an infrared sensor and rotates 180 degrees at the same time. We tried the sensor in different conditions. The color, shape and texture affects the sensor readings. Some are neater and some are lighter on the map. We used a4 papers to form a circular obstacle and we observed the graph to see whether we get a circular mapping or not and we got we wanted in the end.

APPENDIX

Listing 1. Range Finder main file

```
% Me 425
clear;
% Create a connection to the EV3 brick called myev3.
mylego = legoev3;

% Set up Infrared sensor on input port 3
myirsensor = irSensor(mylego);

% Create a connection to a motor.
% Motors are connected to A,B,C,D.
mymotor = motor(mylego,'A');

% Adjust the speed of the motor. (-100,100)
% Positive values - forward direction
% Negative values - backward direction
speed = 15;
mymotor.Speed = speed;
% Start the motor
start(mymotor)

% Reset the rotation value of the motor to zero.
resetRotation(mymotor);

dist = [];
rotation = [];
x = [];
y = [];
k = 1;
∆T = 0.1;

FileName = 'deneme8_empty';

while ¬readButton(mylego, 'up')

    rotation(k) = double(readRotation(mymotor));
    dist(k)   =      readProximity(myirsensor);
    x(k)    = -dist(k)*cosd(rotation(k));
    y(k)    = dist(k)*sind(rotation(k));

if rotation(k) ≤ 0
    mymotor.Speed = speed;
elseif rotation(k) ≥ 180
    mymotor.Speed = -speed;
end

% Plot settings
clf;
subplot(1,3,1);
    scatter(x,y,'b*')
    xlabel('x');
    ylabel('y');
    hold on;
    plot([x(end) 0], [y(end) 0], 'r-');
    plot([x(end) 0], [y(end) 0], 'ro');
subplot(1,3,2);
    plot(dist);
    ylabel('distance');
subplot(1,3,3);
    plot(rotation);
    ylabel('Rotation angle');
pause(ΔT);

k = k + 1;
end

% Stop the motor
stop(mymotor);
save(FileName,'x','y','rotation','dist','ΔT','speed');
```