ME 425 Post Lab Report 1

Busra Oz, Demir Demirel, Mustafa Kerem Erce

***Abstract*— This report include implementation of Quadrupedal Robot made with Lego Mindstorms™ EV3 and discussion on the test results gathered with MATLAB.**

I. INTRODUCTION

In this lab we are asked to design an algorithm to make movements with quadrupedal robot as seen in the Figure: 1. Quadruped robots are one of the examples of four legged locomotion. Most of the four legged locomotion robots were statically stable since the Centre Of Mass (COM) is always stays in the support polygon. In other words, there is always three leg on the surface to ensure stability. In statically stable robots it would be possible to freeze movement at any time. And it would give us a various advantages within. Hence the robots that are dynamically stable has two leg on the surface while the movement was done. In other words the support polygon is limited with a certain line between two leg on a surface. And it is very hard to locate projection of COM on that line. In contrast to statically stable systems when the robot is freeze, these systems will be fall. Dynamically stable systems also have advantages in the manner of velocity, since it moves continuously.

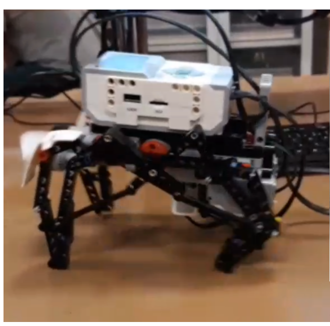


Fig. 1. Quadrupedal Robot

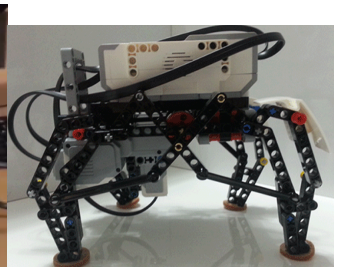


Fig. 2. Quadrupedal Robot From Lab Manual

II. PROCEDURE

A. Building Experimental Setup

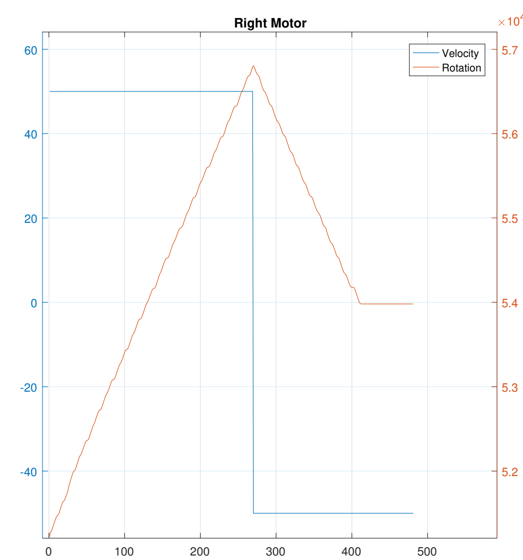
According to lab document, we followed the assembly steps and all group members made a part of quadruped robot. For example, one of the group member constructed the body structure, the other members made right and left legs parts. Then we assemblied the all of the parts and we used the CAT-6 cables to connect infrared sensor, touch sensor, motors and computer.

B. Contol of the Robot

After the computer connection, we wrote some matlab codes. First, we identified the lego, sensors and motors. Then, we set up the speed of the both motors as 50. Because, firstly we wanted that our robot should go to straight, when the motors run. Secondly, we build while loop. This while loop provides us, if up button of lego pushed, the motors will stop and rotation datas will be saved.
We defined the buttons and when we pushed the button 1, speed of left and right motors set up 50, thus the robot moves forward. When we pushed the button 2, speed of left and right motors set up -50, thus the robot moves backward. When we pushed the button 3, speed of left motor set up 50 and speed of right motor set up -50, thus the robot moves rightward. When we pushed the button 4, speed of left motor set up -50 and speed of right motor set up 50, thus the robot moves leftward.

III. RESULT

Fig. 3. On Surface Left-Right Movement



Result of these, we can observed movements of the robot. Our methodology was true, we can see the working principles of motors when the robot was reverse. However, because of the conditions like friction and calibration, our robot could not move clearly. The places friction was very less, thats why we observed the robot was sliding. For instance the graphs shown below Figure: III can be read as when the robot on the surface, the rotation of the motors are rippled but when it was reverse, the rotation graph was proper. The difference between two of the figures shows us the movement done by

Fig. 4. On Surface Left-Right Movement

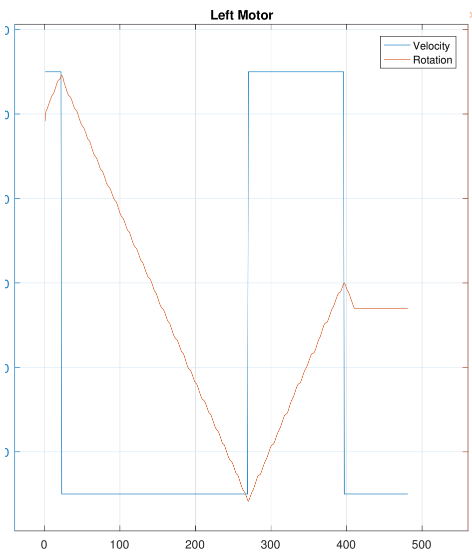
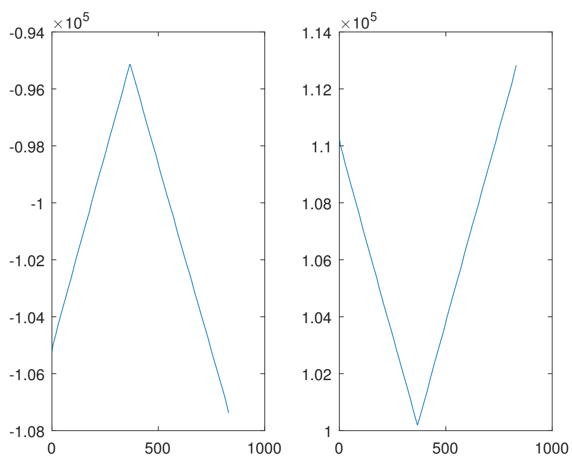


Fig. 5. Ideal Left-Right Movement



the robot is not guaranteed that what we expect from the theoretical results.

A. CONCLUSION

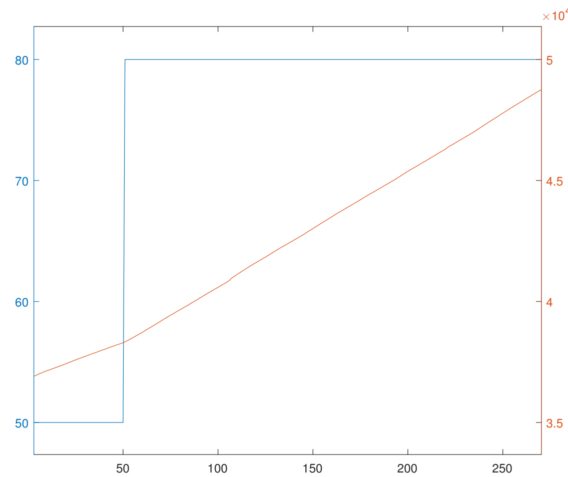
The quadropedal robotics experimental platform LEGO MINDSTORMS EV3 is described in this report. The data gathered from actuators, sensors and control hardware are reviewed in terms of graphs. The theoretical expectations satisfied when the data of the robot on air observed. However there are environmental constraints which prevent the accurate motion. Beside environmental constraints, speed of motors and first position of legs also affect the motion.

IV. INDIVIDUALS

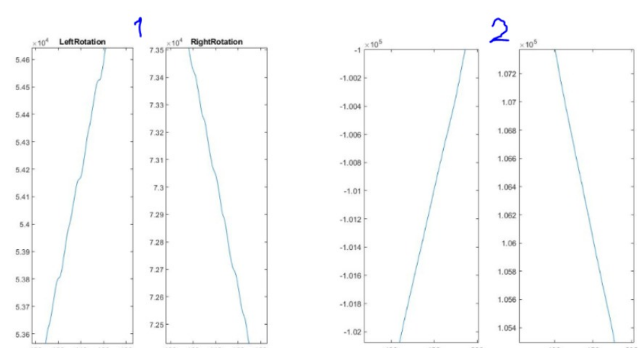
A. Busra Oz

In this lab, we divide the labour to save time. We constructed the parts of robot synchronously. At the end

Fig. 6. Ideal Forward with different velocity

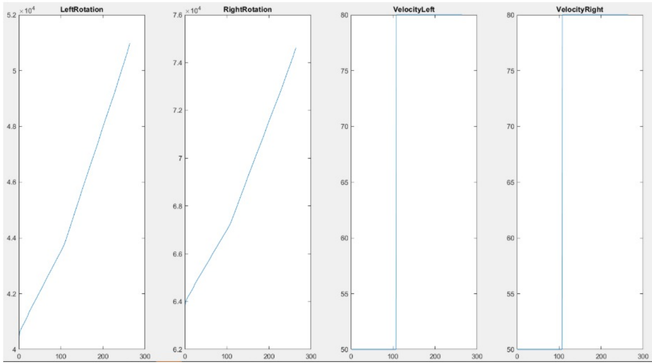


we combined the all parts and make connection between robot and computer. We wrote some MATLAB codes which is needed to give movement to robot forward, backward, leftward and rightward. First, we set the speed of motors as 90 and it was high for our robot, it could not remain standing and the leg parts started to disassembly from the robots body. Then we decrease the speed to 50 and robot moves more accurately but it was still not the proper movement. It is not only about speed, it is also about calibration and environmental factors. Surface is not proper for robot walking.

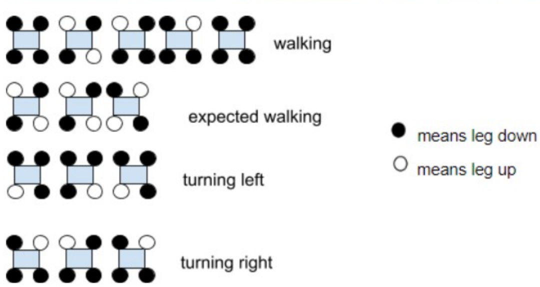


1 is rotation graph of motors while robot was on the surface and 2 is rotation graph of motors while robot was in the air. And these are from the moment turning right. We can see the difference, while it was in the air it has proper movement,it has linear line but while it was on the surface there is some deformation on graph. The friction prevents the rotation.

This is the graph from forward motion of robot with speed 50 and 80. We wanted the observe changes clearly, thats why we keep datas while robot was in the air.
When we increased the speed the slope of rotation graph

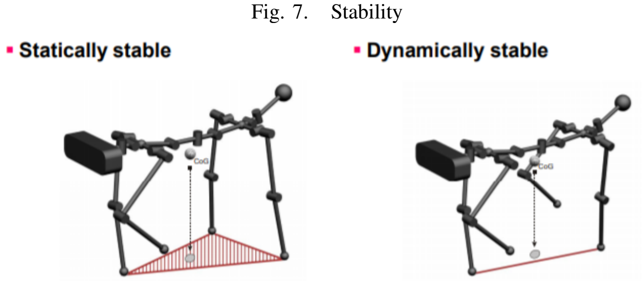


increased. This datas should be from on the surface to compare effect of speed on robot but I dont have them now, we did not save them. But I can say when speed increased the motion of robot will be more out of balance.
Our robot was standing without falling over while it is not moving and the center of mass is between the legs. Thats why our robot ensure its stability statically. Our robot seems like have 4 legs but they are dependent actually it has 2 independent legs. Thats why it can have 6 distinct event sequences from N = (2k 1)!, k is equal to number of legs, in this case it is 2. But our robot is just doing walking and turning right or left.



B. Demir Demirel

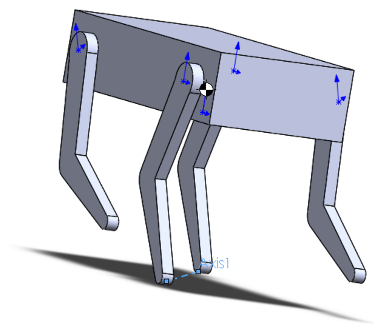
Fig. 7. Stability



In this lab our system is dynamically stable system. Hence quadrupeds movements are not certain as we expect.

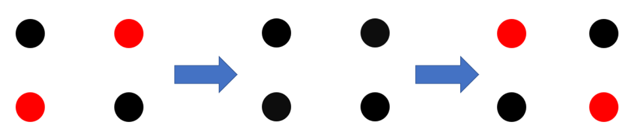
Since the robot has four leg and legs that are in same side are bonded with each other.So the gait of a robot should be as follow; one feet is on the surface while the other one is on air for legs that are in same side. The representative Figure: 8 shows us the leg positions clearly.
This type of gait enable us that; at a same instance there

Fig. 8. CAD Drawing of Quadruped



would be at most two leg on the surface.
So our system is dynamically stable and we can observe the footprint as shown in the Figure: 9 .In the figure red points represents legs in the air, black ones represent legs on the surface.
Since our robot was not calibrated, initial position of

Fig. 9. Footprint of Quadruped



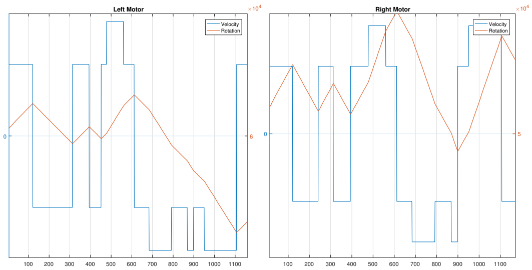
the robot could be different than what it should be. In other words, wrong start position leads wrong movements in terms of leg positions. For instance both front leg could be positioned higher. In that case, the robot make backlash movement and stays at the same position.

Also our robot could be controlled by remote. Motions and velocities could be change with that remote. In Figure:10 there is a data for representing response of the motors to remote. We can see the steps as follows:

- Move Forward
- Move Backward
- Turn Right
- Move Backward
- Turn Right
- Turn Left
- ...

This graph plotted with even surface (air). When we applied it in the table, movements were observed but the data is very noisy. The speed of movements were also very slow and it would be a problem for more sophisticated applications.
Also we can observe the speed of the robot from Figure:

Fig. 10. All Movements in different velocity



6. As it is seen when we change the speed, the slope of the rotation line change as a result. So we can say that, changing the speed has an effect on rotation. Theoretically when the speed increase it would end up with more accurate movements and trajectories. But it is true for only even terrain and well designed robots. When we applied more speed in our system it was ended up with breaking some parts of the robot.I think it is a result of reaction force of the surface.

To sum up this lab experiment shows us, both statically stable and dynamically stable systems have disadvantages and advantages at the same time. Speed of movement, operating surface and consistency are the major parts when designing that kinds of robots.

C. Mustafa Kerem ERCE

In lectures, we were touched the background information of what we do in Lab hours. However it becomes a little bit tricky while trying to learn and implement what we did no cover in lecture. In this lab we assemblied the legos and made a quadruped robot. We made the disturbation of work between us. When the Module, Body, Legs constractions finished, we wrote the MATLAB codes which are about the robots working principle such as when the user push the button 1 the robot goes forward. More over it include the how the robot can turn around or go back. Theoritically the codes work true but the robot did not work as we expected. Firstly we choosed the motor speed big value, thats why our robots legs out. Than we set the motor speed again and it worked smoothly. Our robot was static stability because it had four legs. According to proffesor, at least three legs are required for the static stability. When we pushed the button the motors would work as we expected but the robot could not make a pure movement and rotation. Because the friction between robot legs and surface is not enough make ideal movements. Moreover we did not calculate the calibration so our robot could not work well.

APPENDIX

Initialization

```
clear all;
mylego = legoev3;
myirsensor = irSensor(mylego,2);
mytouchsensor = touchSensor(mylego, 1);

mymotorleft = motor(mylego,'B');
mymotorright = motor(mylego,'C');

mymotorleft.Speed = 50;
mymotorright.Speed = 50;
Array_left = [];
Array_right = [];
Veloleft=[];
Veloright=[];
```

Movememts and Data Gathering

```

% Start the motor
start(mymotorleft);
start(mymotorright);
botton=1;

while ¬readButton(mylego, 'up')
    button = readBeaconButton(myirsensor,4);
    RotationLeft = readRotation(mymotorleft);
    RotationRight = readRotation(mymotorright);

    Array_left = [Array_left; RotationLeft];
    Array_right = [Array_right; RotationRight];
    Veloleft=[Veloleft;mymotorleft.Speed];
    Veloright=[Veloright;mymotorright.Speed];

    if button == 1 %% forward
        botton=1;
    elseif button == 2 %%back
        botton=2;
    elseif button == 3 %% right
        botton=3;
    elseif button == 4 %% left
         botton=4;
    end

    if button ==9
        speed1=80;
    else
        speed1=50;
    end

    if botton == 1 %% forward
        mymotorleft.Speed = speed1;
        mymotorright.Speed = speed1;
    elseif botton == 2 %%back
        mymotorleft.Speed = -speed1;
        mymotorright.Speed = -speed1;
    elseif botton == 3 %% right
        mymotorleft.Speed = speed1;
        mymotorright.Speed = -speed1;
    else %% left
         mymotorleft.Speed = -speed1;
        mymotorright.Speed = speed1;
    end

    subplot(1,4,1);plot(Array_left);
    title('LeftRotation');
    subplot(1,4,2);plot(Array_right);
    title("RightRotation");
     subplot(1,4,3);plot(Veloleft);
     title('VelocityLeft');
    subplot(1,4,4);plot(Veloright);
    title('VelocityRight');
    drawnow;
end

stop(mymotorleft);
stop(mymotorright);
```

```
filename="deneme1";
Data.Array_left  = Array_left;
Data.Array_right = Array_right;
save(filename,'Data');
```