Introduction to Mobile Robotics with MATLAB and Simulink
Unit 2: Basic Robot Movements

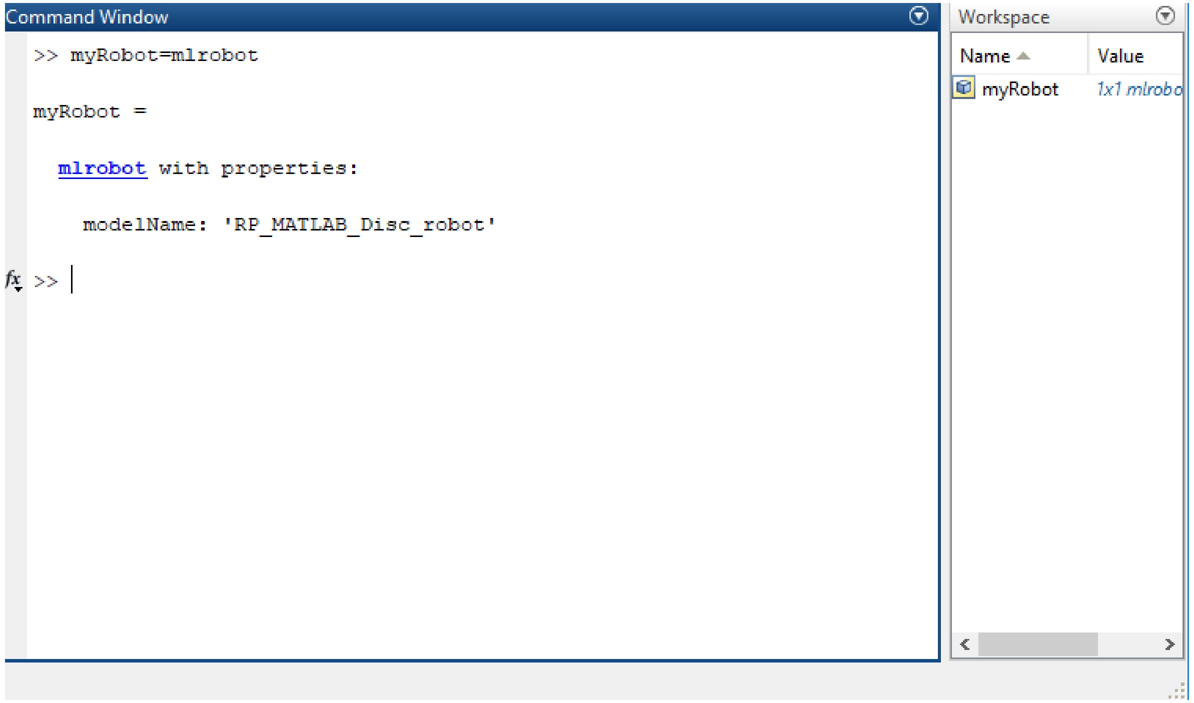
By MathWorks Student Competition team

Starting a Virtual Environment from MATLAB

- The Robotics Playground provides a MATLAB virtual environment for understanding how to program simple robot tasks such as controlling robot motion

- Use the “mlrobot” class to create a robot within a virtual environment and assign it to a variable.

- `>> myRobot=mlrobot;`



```
>> myRobot=mlrobot

myRobot = 

  mlrobot with properties:

    modelName: 'RP_MATLAB_Disc_robot'

>> 
```

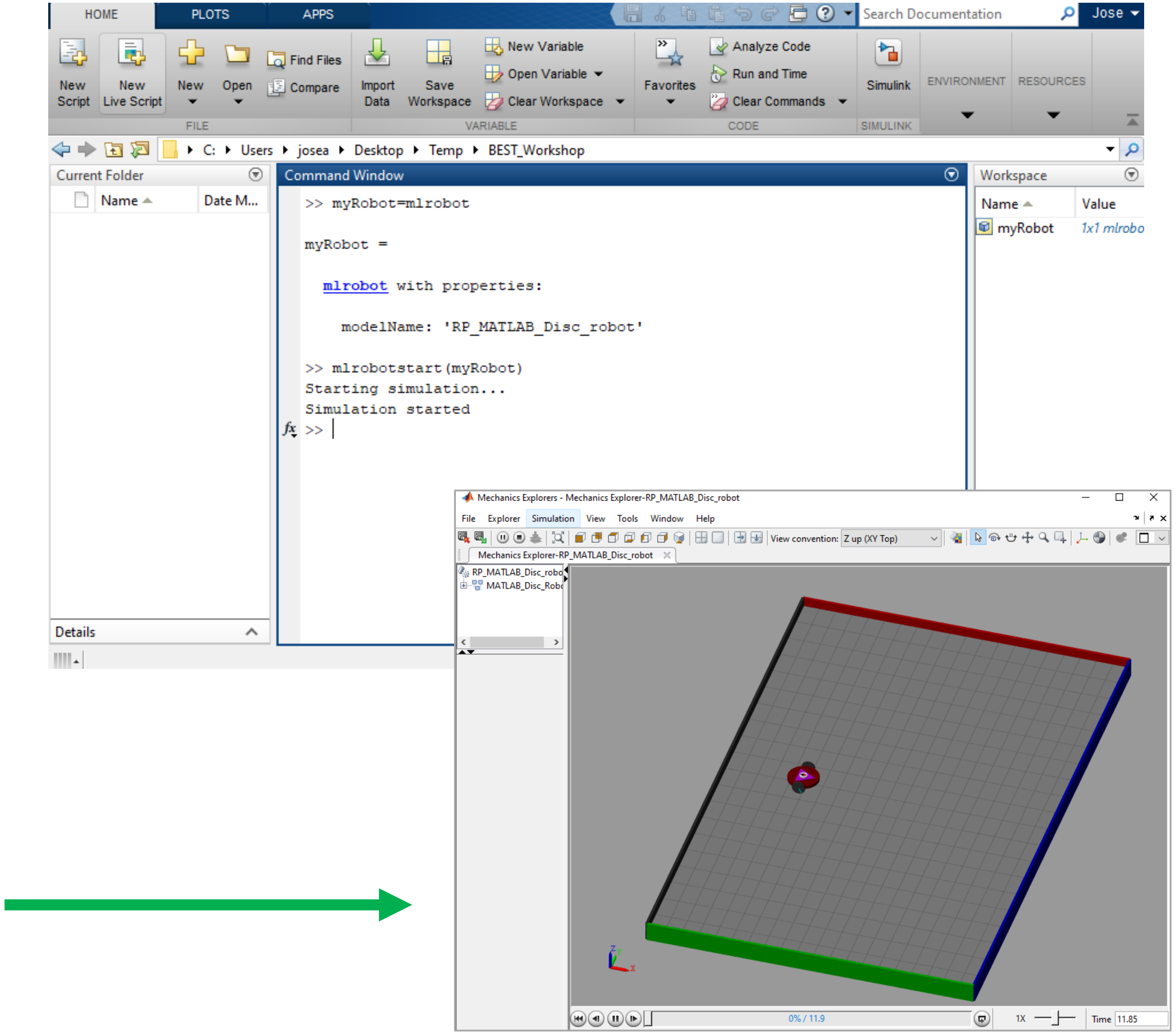
Name	Value
myRobot	1x1 mlrobo

Starting a Virtual Environment from MATLAB

- Start the execution of the virtual world using the function **“mlrobotstart”**

- `>>mlrobotstart(myRobot)`

- The following window with the environment will become available

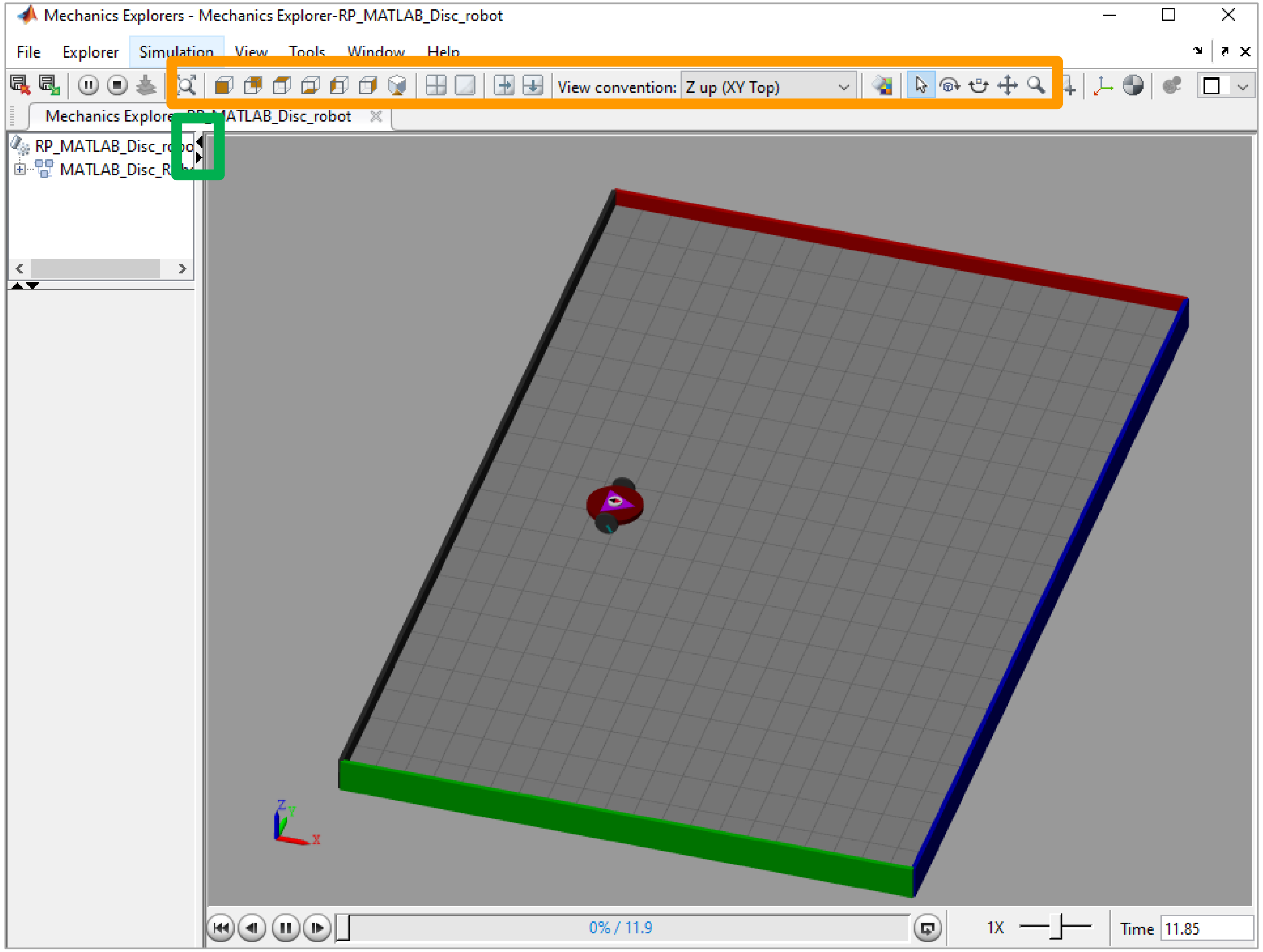


```
>> myRobot=mlrobot

myRobot =

  mlrobot with properties:

    modelName: 'RP_MATLAB_Disc_robot'

>> mlrobotstart(myRobot)
Starting simulation...
Simulation started
>>
```

Starting a Virtual Environment from MATLAB

- You can hide the environments parts list by using the minimizing arrows (Shown in Green)

- Simple 3D operations like zoom and rotate can be performed to reorient the view within the virtual environments (See toolbar in Orange)


Mechanics Explorers - Mechanics Explorer-RP_MATLAB_Disc_robot
File
Explorer
Simulation
View
Tools
Window
Help
Mechanics Explorer-RP_MATLAB_Disc_robot
RP_MATLAB_Disc_robot
MATLAB_Disc_R
Z
Y
X
1X
Time

View convention: Z up (XY Top)

0% / 11.9

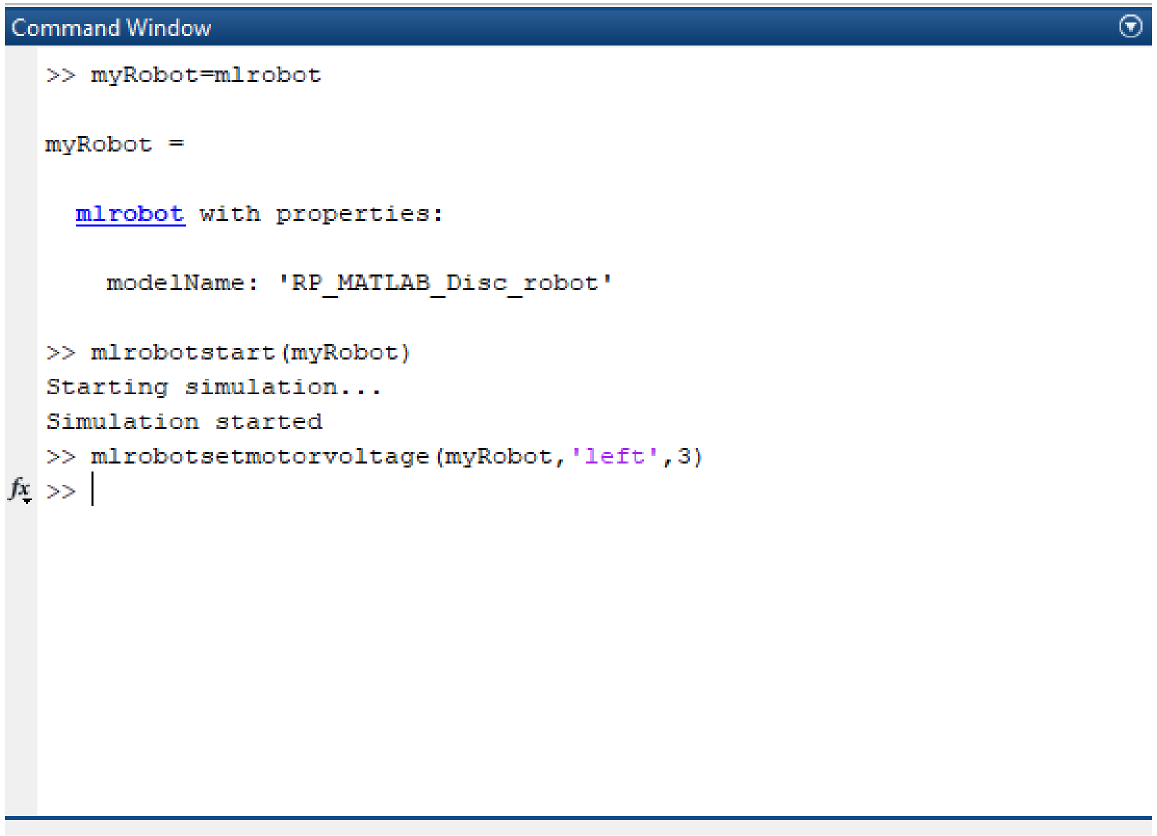
Time 11.85

Interfacing with the Virtual Robot

- How do we make the robot move?
- Each wheel in the robot has a motor, you can set the voltage of the left or right motor by using the command **“mlrobotsetmotorvoltage”**
- Execute the following:

```
>>mlrobotsetmotorvoltage(myRobot,’left’,3)
```

- Now the robot is moving ! (It should be rotating in a circle)



```
>> myRobot=mlrobot

myRobot =

  mlrobot with properties:

    modelName: 'RP_MATLAB_Disc_robot'

>> mlrobotstart(myRobot)
Starting simulation...
Simulation started
>> mlrobotsetmotorvoltage(myRobot,'left',3)
>> 
```

Interfacing with the Virtual Robot

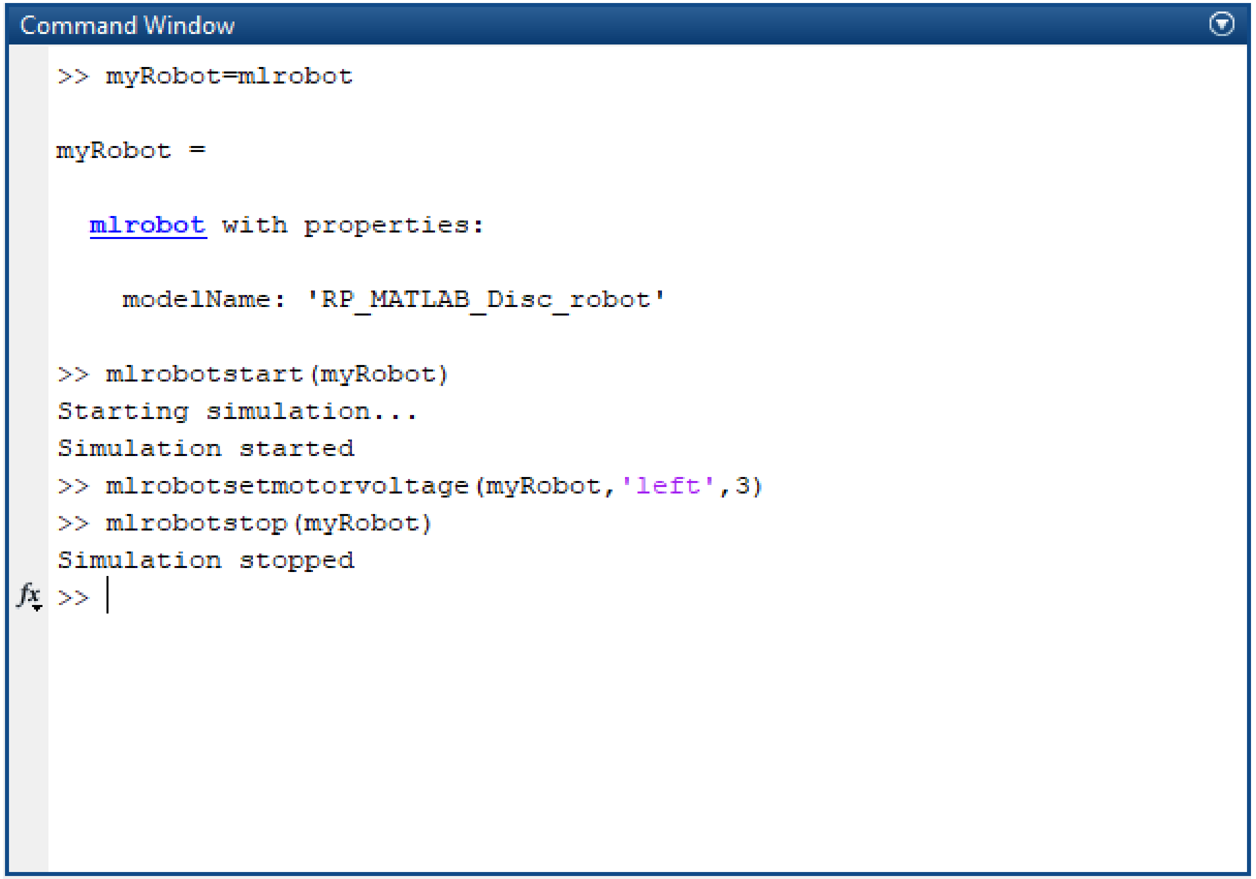
- How do we stop the robot?

- Stop the robot by setting both motor voltages to zero.

- `>>mlrobotsetmotorvoltage(myRobot,’left’,0)`
- `>>mlrobotsetmotorvoltage(myRobot,’right’,0)`

- If you want to stop the virtual environment use **“mlrobotstop”**

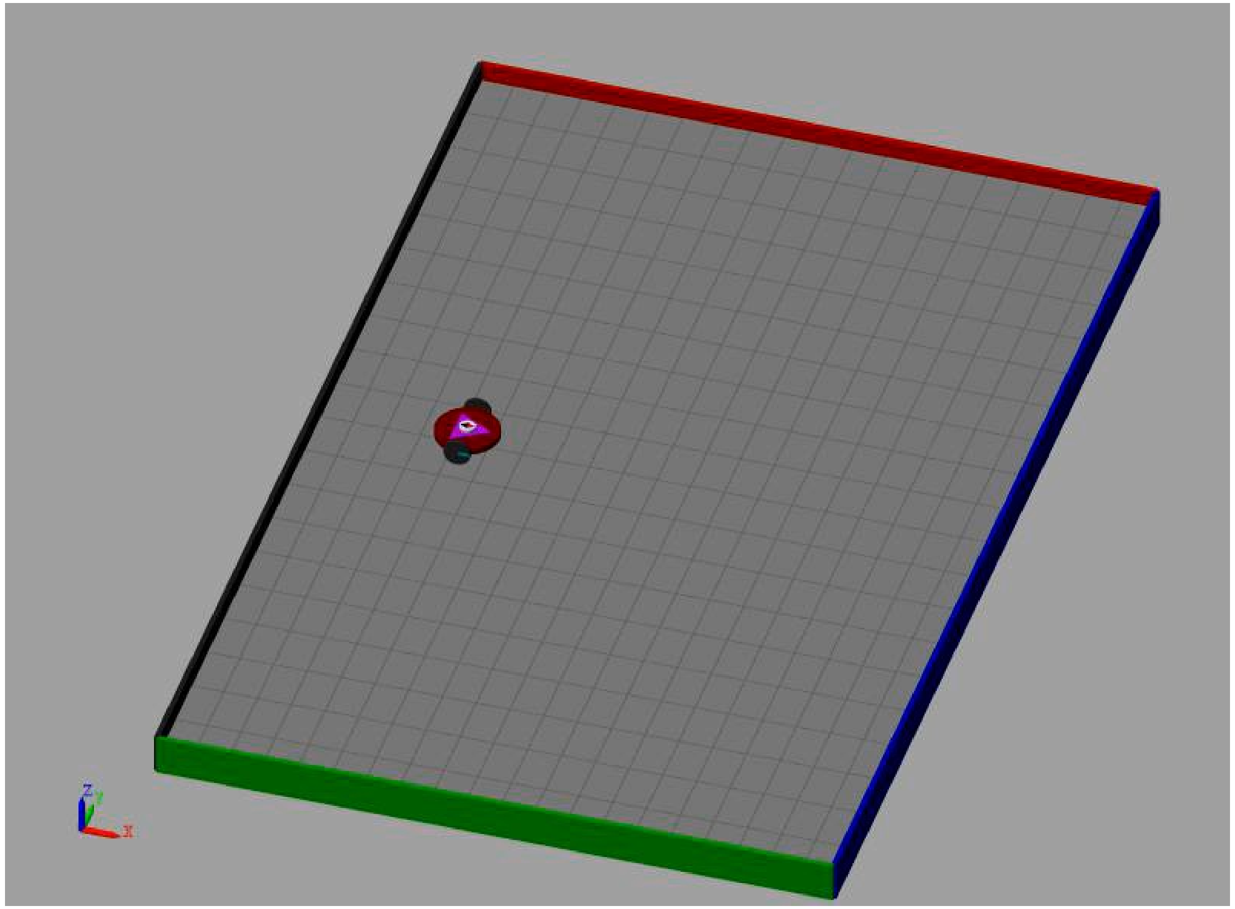
- `>>mlrobotstop(myRobot)`



```
>> myRobot=mlrobot

myRobot =

  mlrobot with properties:

    modelName: 'RP_MATLAB_Disc_robot'

>> mlrobotstart(myRobot)
Starting simulation...
Simulation started
>> mlrobotsetmotorvoltage(myRobot,'left',3)
>> mlrobotstop(myRobot)
Simulation stopped
>> 
```

Exercise 1: Move Forward and Stop

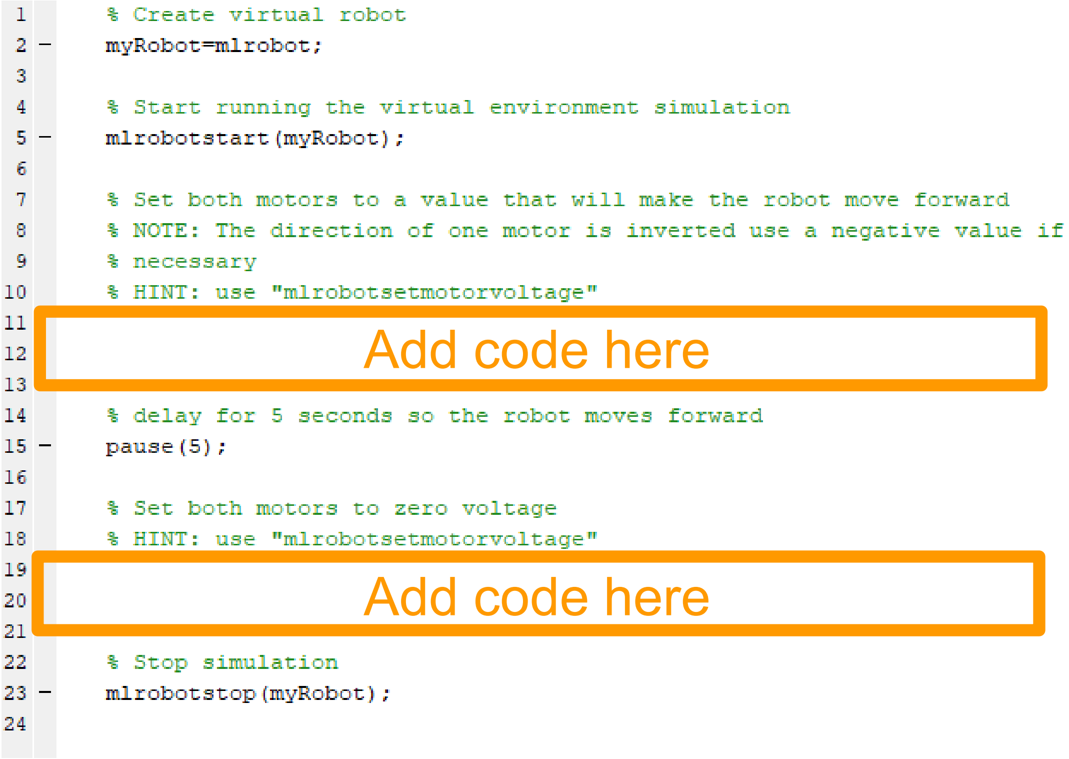
- The robot should move forward and stop as in the following video



Exercise 1: Move Forward and Stop

- Create a MATLAB script to make the virtual robot move forward for 5 seconds and then stop.

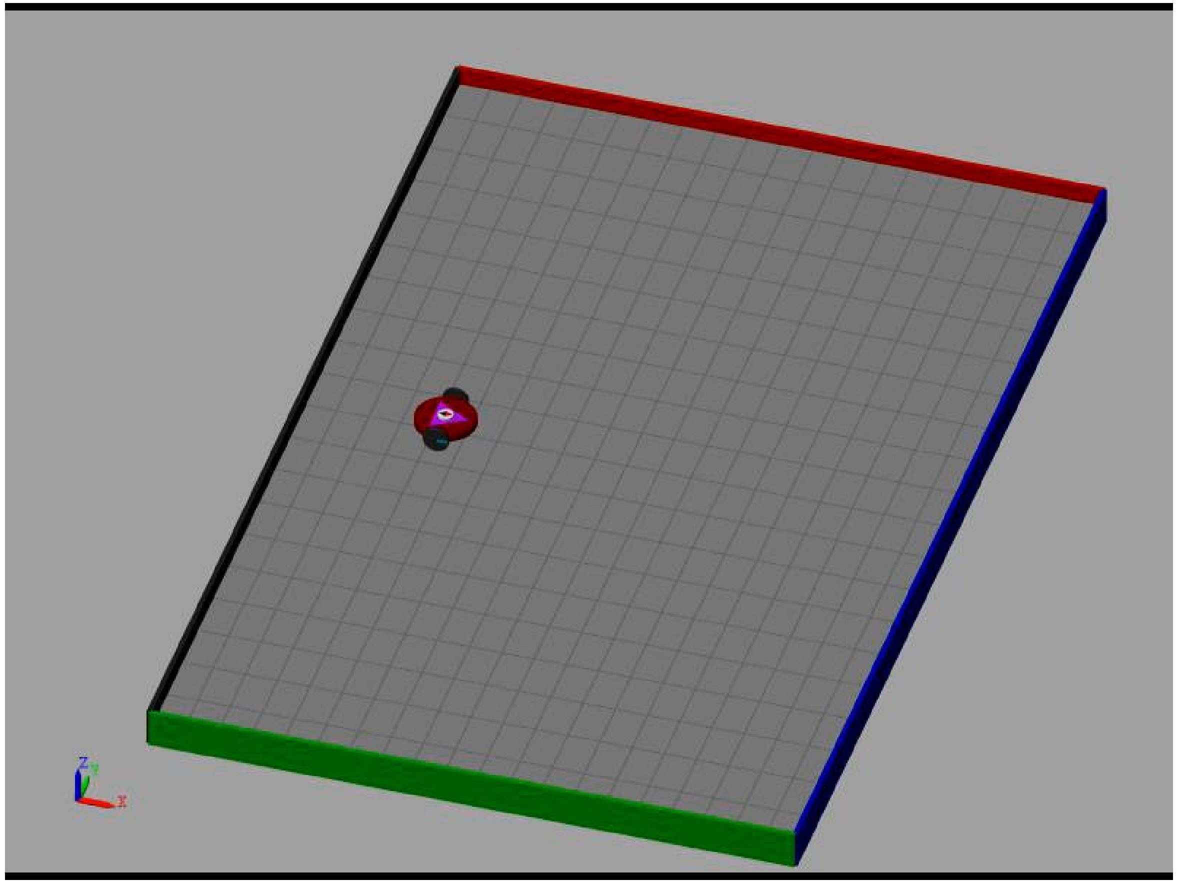
1. Use the “**MoveForward_start.m**” script as a template
2. Add commands to set motor voltages
3. Use the “**pause**” function to delay the execution of statements for a specified number of seconds.



```
% Create virtual robot
myRobot=mlrobot;

% Start running the virtual environment simulation
mlrobotstart(myRobot);

% Set both motors to a value that will make the robot move forward
% NOTE: The direction of one motor is inverted use a negative value if
% necessary
% HINT: use "mlrobotsetmotorvoltage"

                         Add code here

% delay for 5 seconds so the robot moves forward
pause(5);

% Set both motors to zero voltage
% HINT: use "mlrobotsetmotorvoltage"

                         Add code here

% Stop simulation
mlrobotstop(myRobot);
```

Exercise 2: Move, turn and return

- The final script should make the robot move similar to the following video.



Exercise 2: Move, Turn and Return

- Write a script to make the robot move forward, then turn around and drive back to its starting position.

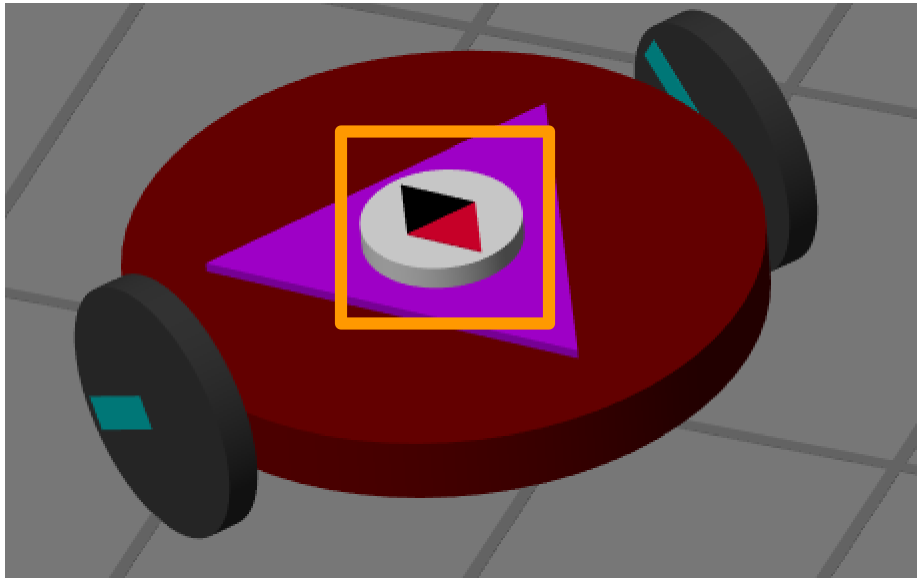
1. Use the script named **“MoveForwardAndBack_start.m”**
2. Fill in the statements to set motors and delay execution
3. Run script
4. Modify parameters until robot achieves desired motion

Exercise 3: Plot the Robot Orientation

- Obtain information from robot movements to understand its behavior or automate tasks.
- Plot the robot’s heading (Orientation) with respect to the stationary world.
- Get information from the compass located in the robot’s center.
- Use the function **“mlrobotreadcompassangle”**

- Use the MATLAB help to understand more about this function

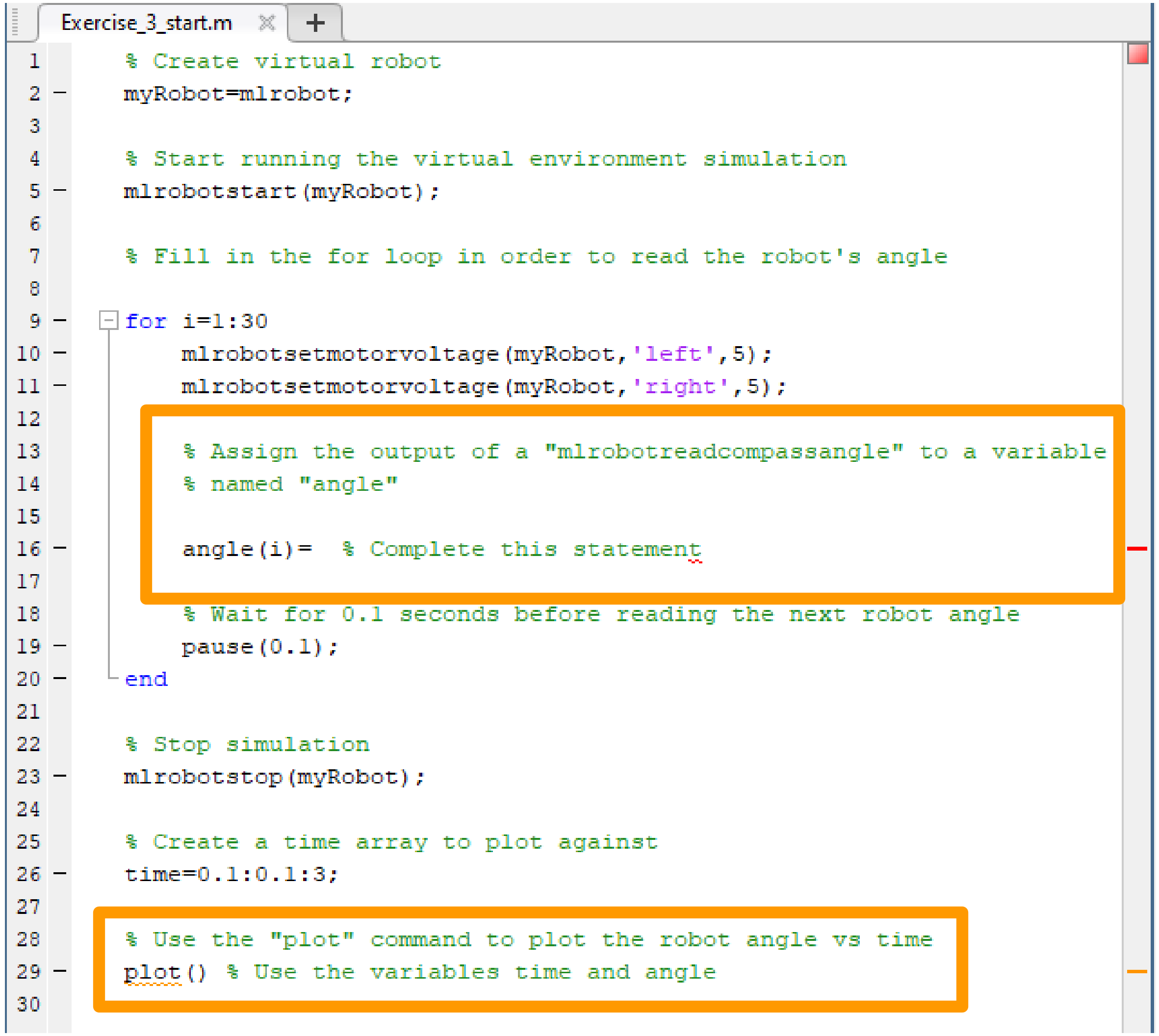
```
>>help mlrobotreadcompassangle
```



Exercise 3: Plot the Robot Orientation

1. Open the script named **“PlotOrientation_start.m”**

2. Fill in the highlighted regions to complete a program that plots the robot’s angle for 3 seconds

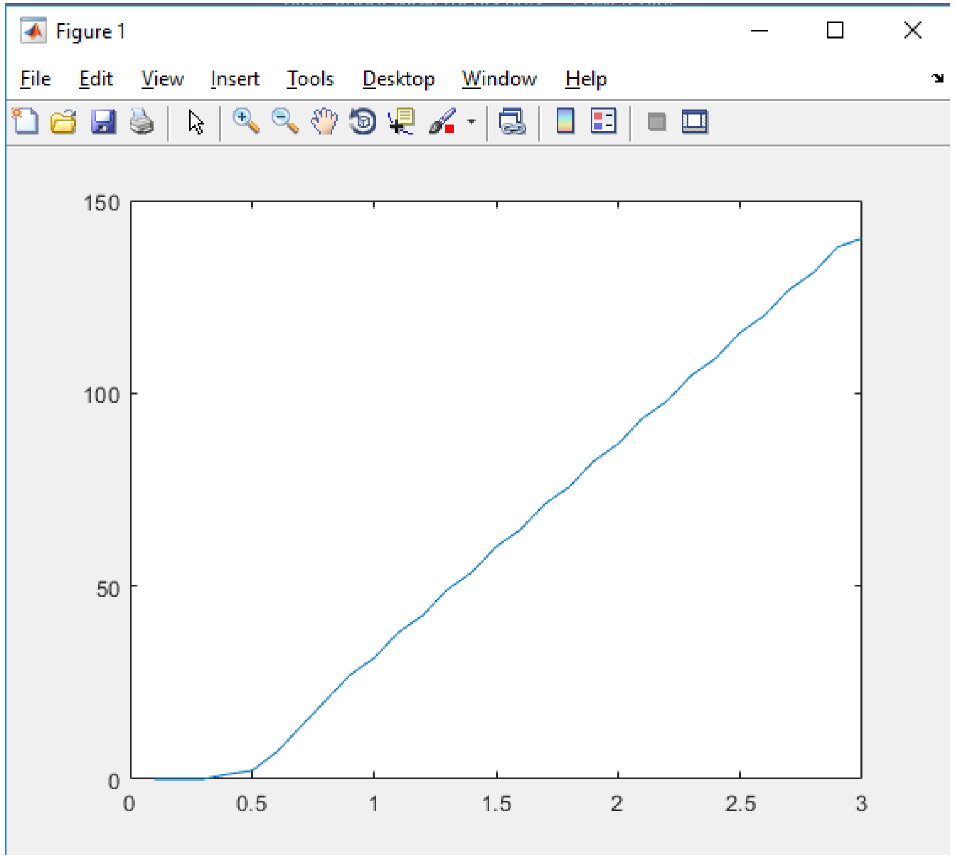


```
% Create virtual robot
myRobot=mlrobot;

% Start running the virtual environment simulation
mlrobotstart(myRobot);

% Fill in the for loop in order to read the robot's angle

for i=1:30
    mlrobotsetmotorvoltage(myRobot,'left',5);
    mlrobotsetmotorvoltage(myRobot,'right',5);

    % Assign the output of a "mlrobotreadcompassangle" to a variable
    % named "angle"

    angle(i)=  % Complete this statement

    % Wait for 0.1 seconds before reading the next robot angle
    pause(0.1);
end

% Stop simulation
mlrobotstop(myRobot);

% Create a time array to plot against
time=0.1:0.1:3;

% Use the "plot" command to plot the robot angle vs time
plot() % Use the variables time and angle

```

Exercise 3: Plot the Robot Orientation

- The output of your MATLAB script should be a figure as the one shown

- The robot angle should change from 0 to about 140 degrees



Exercise 3: Plot the Robot Orientation

1. Add a title to your plot using the "title" function

2. Add Axis labels to your graph using the "xlabel" and "ylabel" functions

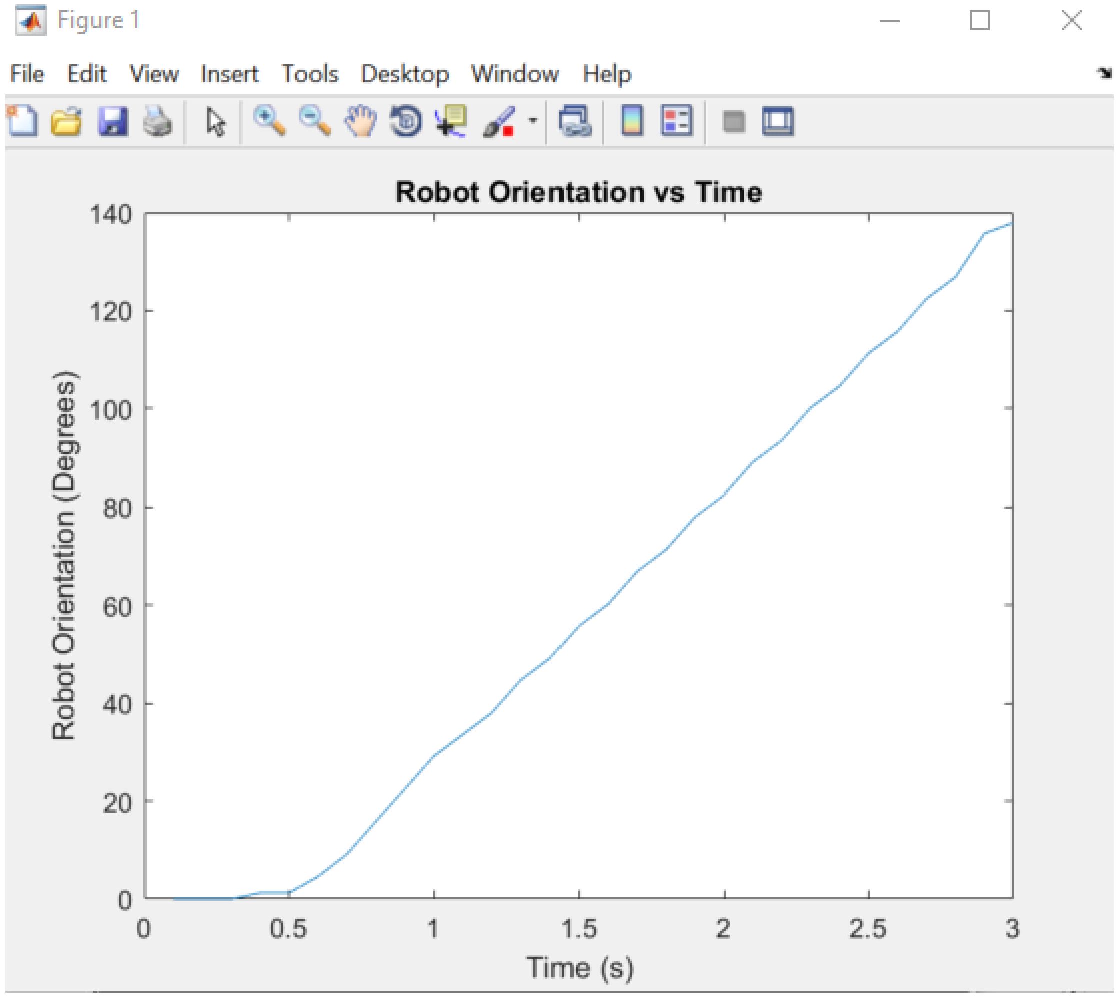


Figure 1
File Edit View Insert Tools Desktop Window Help
Robot Orientation vs Time
Robot Orientation (Degrees)
Time (s)
0
20
40
60
80
100
120
140
0
0.5
1
1.5
2
2.5
3

End of Unit 2: Basic Robot Movements

- Congrats !

- Here are some of learning outcomes from this unit:
  - How to use the Robotics Playground virtual environment
  - How to set motor speeds to achieve desired robot motion
  - How collect and analyze data from the robot movements