ASSIGNMENT 4

NOVEMBER 20, 2021

Submitted to: Prof. Dr. Kemalettin Erbatur

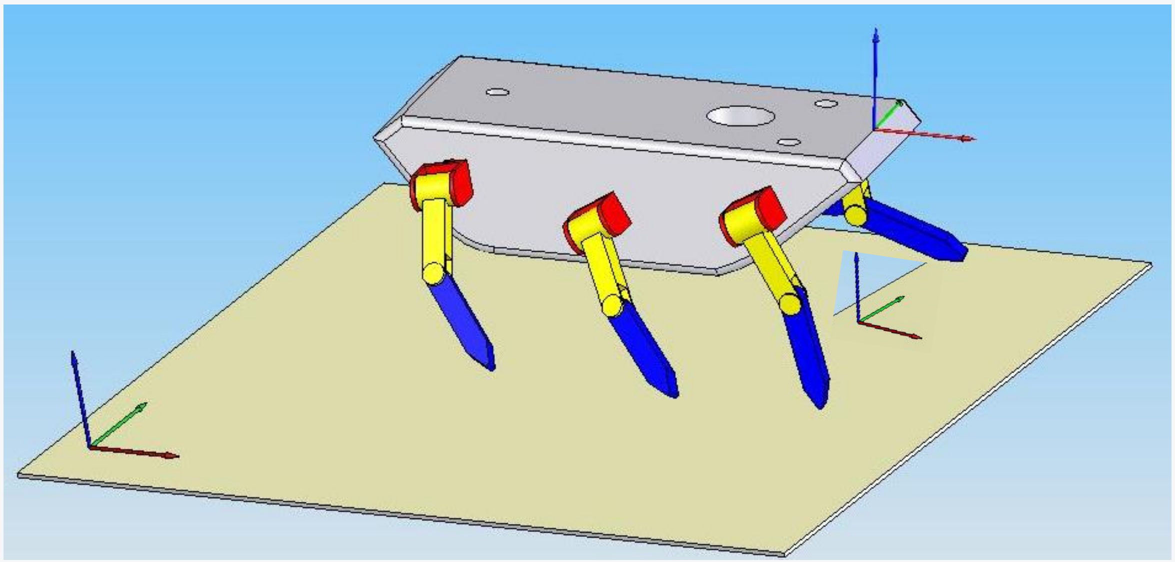
Name of Student: Moses Chuka Ebere
Student Number: 31491
Course: Autonomous Mobile Robotics (ME 525)

TABLE OF CONTENTS

INTRODUCTION

This is a computational assignment to determine and plot the body coordinate frame origin coordinates of a hexapod with respect to the world frame.

According to the question, at the robot's initial position, the world frame is located on the ground just below the body frame with all axes parallel to the axes of the initial body coordinate frame as shown below:



To determine the body frame origin points with respect to the world frame, we establish the following:

1. The robot does not swing to the left or right while moving. This implies that the y-coordinate remains fixed at $y = 0$.
2. At all times, the body of the robot is at the same height from the ground. This translates to a constant z-coordinate which is equal to the body height ($50cm = 0.5$ m).
3. With the above two conditions, the initial position of the body frame's origin viz-a-viz the world frame is $(x, y, z) = (0, 0, 0.5)$.

*****Just in case, I added another code in the appendix for the right and left foot references with respect to the world frame.

MATLAB CODE

```
m-file name – Assignment_4

clear all
close all
clc

% DATA ENTRY
% ----------------------------------------
% Simulation Parameters
step_time = 0.001; % We'd like to have data for every
millisecond

% Recall: T_walk = 2(T_single + T_double)
T_walk = 3;
T_single = 1;
T_double = 0.5;
Step_size = 0.1;
Step_height = 0.05;
Body_height = 0.5;
y_offset = 0.4;

% The simulation is performed for 15 seconds.
stop_time = 5*T_walk;

% Data Storage Dimension - The +1 is for time = 0s.
record_length = (stop_time/step_time) + 1;

% References for all axes for each leg.
x_right_list = zeros(record_length, 1);
x_left_list = zeros(record_length, 1);
y_right_list = zeros(record_length, 1);
y_left_list = zeros(record_length, 1);
z_right_list = zeros(record_length, 1);
z_left_list = zeros(record_length, 1);
time_list = zeros(record_length, 1);

% Body Coordinates
x_body_list = zeros(record_length, 1);
y_body_list = zeros(record_length, 1);
z_body_list = zeros(record_length, 1);
```

```
% Initial position of the body frame's origin with respect
to the world
% frame.
x_body = 0;
y_body = 0;
z_body = Body_height;

for iteration_index = 1:1:record_length
   time = (iteration_index - 1)*step_time;
   % Normalize the time to always be between 0 and 1
   time_ratio_of_walk_period = (time/T_walk) -
floor(time/T_walk);
   % Now, scale this value with the step period. This is to
ensure that the
   % reference at any time instant could be correctly
calculated.
   time_in_walk_period = time_ratio_of_walk_period *
T_walk;

   % There are four segments per period: 0 <= t <=
T_single,
   % T_single < t <= T_single + T_double, T_single +
T_double < t <=
   % 2*T_single + T_double, and 2*T_single + T_double < t.
   if (time_in_walk_period <= T_single)
       x_right = -Step_size + Step_size*(1 -
cos(time_in_walk_period*pi/T_single));
       x_left = Step_size - Step_size*(1 -
cos(time_in_walk_period*pi/T_single));
       y_right = -y_offset;
       y_left = y_offset;
       z_right = -Body_height + Step_height*0.5*(1 -
cos(time_in_walk_period*2*pi/T_single));
       z_left = -Body_height;
   end

   if (T_single < time_in_walk_period  &&
time_in_walk_period <= T_single + T_double)
       x_right = Step_size;
       x_left = -Step_size;
       y_right = -y_offset;
```

```
        y_left = y_offset;
        z_right = -Body_height;
        z_left = -Body_height;
    end

    if (T_single + T_double < time_in_walk_period  &&
time_in_walk_period <= 2*T_single + T_double)
        x_right = Step_size - Step_size*(1 -
cos((time_in_walk_period...
            - (T_single + T_double)) *pi/T_single));
        x_left = -Step_size + Step_size*(1 -
cos((time_in_walk_period...
            - (T_single + T_double))*pi/T_single));
        y_right = -y_offset;
        y_left = y_offset;
        z_right = -Body_height;
        z_left = -Body_height + Step_height*0.5*(1 -
cos((time_in_walk_period...
            - (T_single + T_double))*2*pi/T_single));
    end

    if (2*T_single + T_double < time_in_walk_period)
        x_right = -Step_size;
        x_left = Step_size;
        y_right = -y_offset;
        y_left = y_offset;
        z_right = -Body_height;
        z_left = -Body_height;
    end

    % Store the results in the list
    x_right_list(iteration_index) = x_right;
    x_left_list(iteration_index) = x_left;
    y_right_list(iteration_index) = y_right;
    y_left_list(iteration_index) = y_left;
    z_right_list(iteration_index) = z_right;
    z_left_list(iteration_index) = z_left;
    time_list(iteration_index) = time;

    % Establish a condition for the position of the x-
coordinate of the body
    % frame's origin
```

```
    if iteration_index == 1
        x_body_list(iteration_index) = x_body;
    else
        x_body_list(iteration_index) =
x_body_list(iteration_index - 1)...
            + abs(x_right_list(iteration_index) -
x_right_list(iteration_index - 1));
    end

    y_body_list(iteration_index) = y_body;
    z_body_list(iteration_index) = z_body;
end

figure %1
plot(time_list, x_right_list, 'r')
ylim([-0.12 0.12])
hold
plot(time_list, x_left_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

figure %2
plot(time_list, y_right_list, 'r')
ylim([-0.5 0.5])
hold
plot(time_list, y_left_list, 'b')
xlabel('Time [s]')
ylabel('y references [m]')
grid

figure %3
plot(time_list, z_right_list, 'r')
ylim([-Body_height -Body_height+4/3*(Step_height)])
hold
plot(time_list, z_left_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid

figure %4
subplot(3,1,1)
```

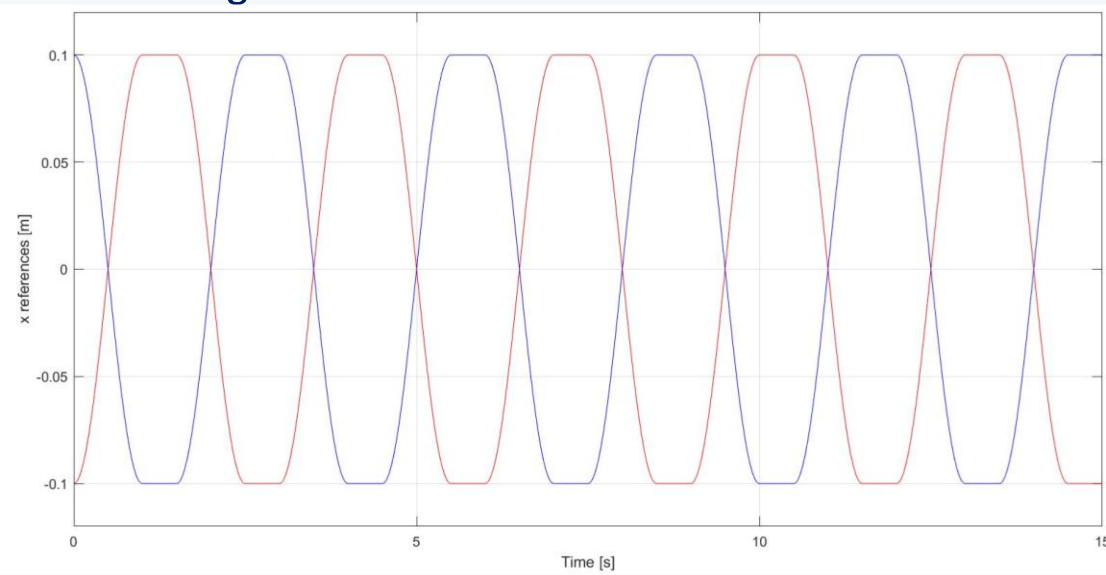
```
plot(time_list, x_right_list, 'r')
hold
plot(time_list, x_left_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

subplot(3,1,2)
plot(time_list, y_right_list, 'r')
hold
plot(time_list, y_left_list, 'b')
xlabel('Time [s]')
ylabel('y references [m]')
grid

subplot(3,1,3)
plot(time_list, z_right_list, 'r')
hold
plot(time_list, z_left_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid

% Body Frame Origin Coordinates in the World Frame
figure %5
plot(time_list, x_body_list)
title('Body Coordinate Frame wrt the World Coordinate
Frame')
xlabel('Time [s]')
ylabel('x [m]')
grid

figure %6
plot(time_list, y_body_list)
xlabel('Time [s]')
ylabel('y [m]')
grid

figure %7
plot(time_list, z_body_list)
xlabel('Time [s]')
ylabel('z [m]')
```

```
grid


figure %8
subplot(3,1,1)
plot(time_list, x_body_list)
title('Body Coordinate Frame wrt the World Coordinate
Frame')
xlabel('Time [s]')
ylabel('x [m]')
grid

subplot(3,1,2)
plot(time_list, y_body_list)
xlabel('Time [s]')
ylabel('y [m]')
grid

subplot(3,1,3)
plot(time_list, z_body_list)
xlabel('Time [s]')
ylabel('z [m]')
grid

% The following figure is added to compare the origin plot
with legs x and
% z trajectories.
figure %9
subplot(3,1,1)
plot(time_list, x_right_list, 'r')
title('X Trajectory of Right and Left Leg')
hold
plot(time_list, x_left_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

subplot(3,1,2)
plot(time_list, x_body_list)
title('X Coordinate of Body Frame Origin wrt the World
Coordinate Frame')
xlabel('Time [s]')
```

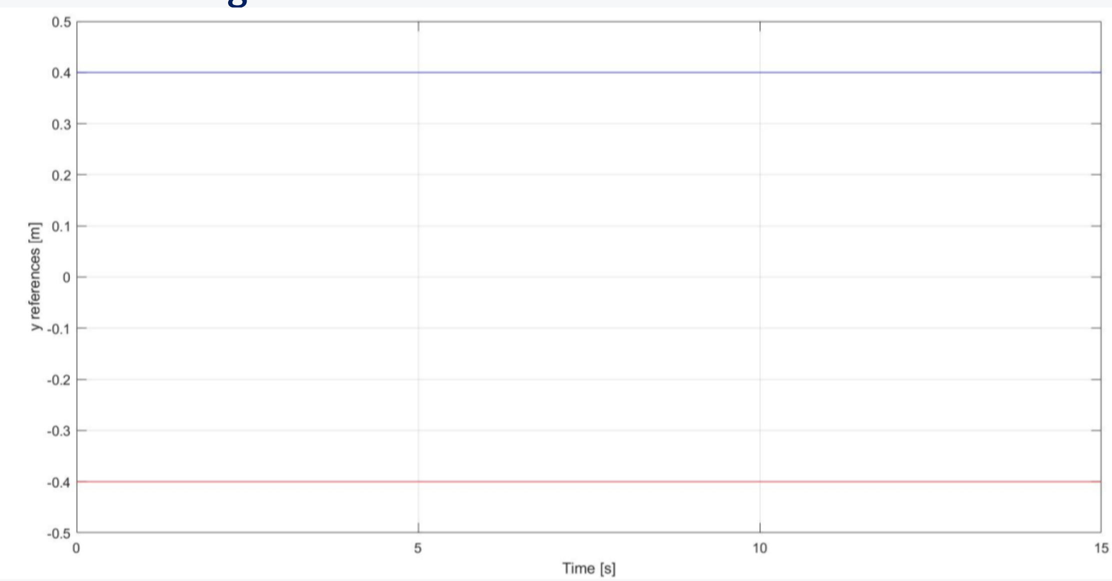
```
ylabel('x [m]')
grid

subplot(3,1,3)
plot(time_list, z_right_list, 'r')
title('Z Trajectory of Right and Left Leg')
hold
plot(time_list, z_left_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid
```

PLOTS

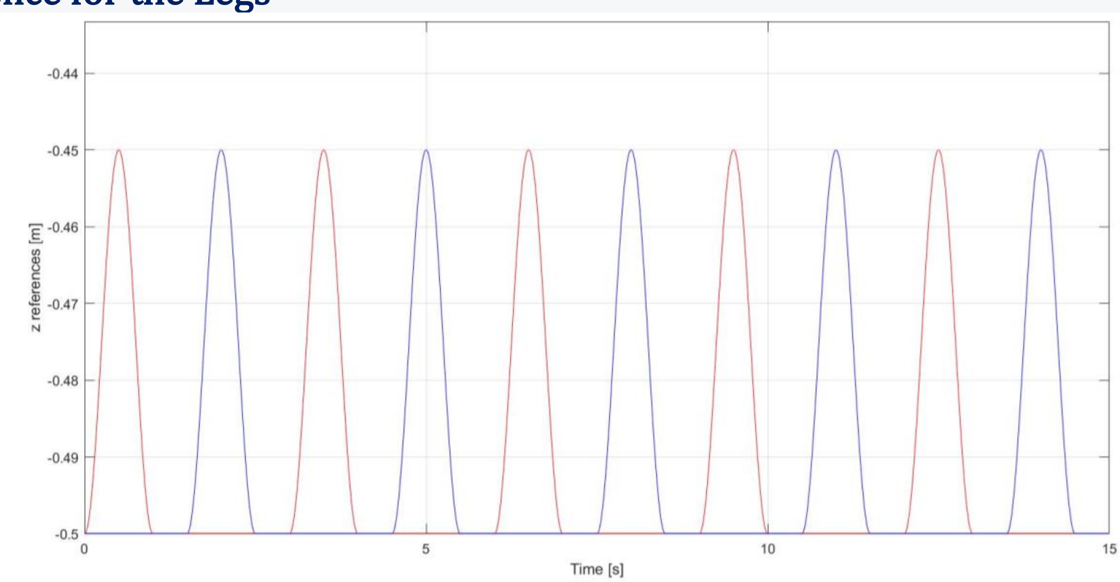
X Reference for the Legs



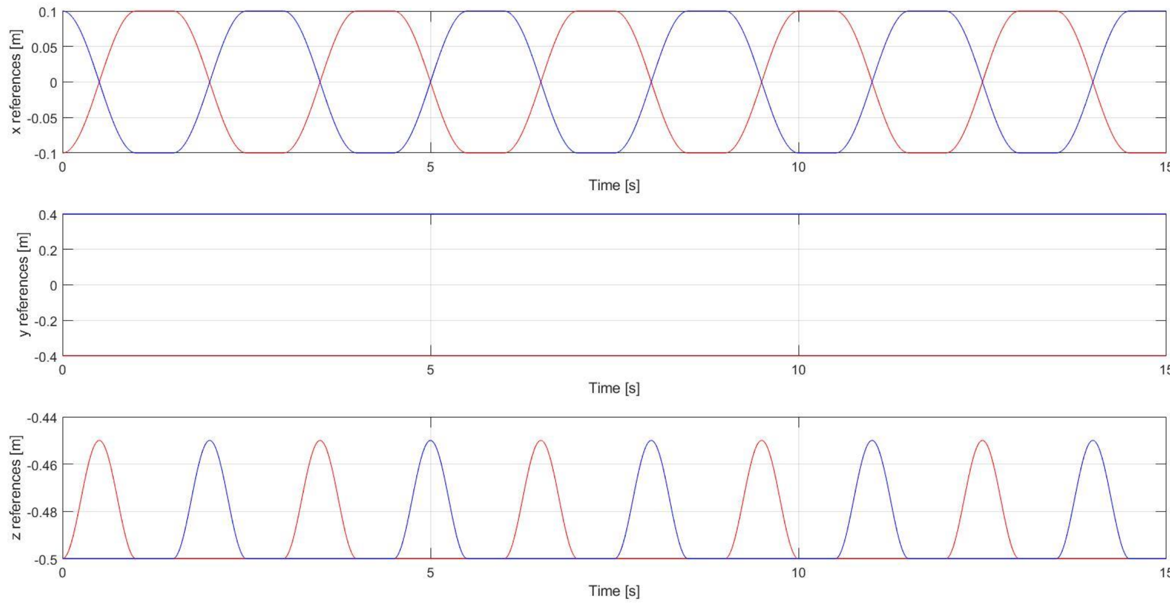
Y Reference for the Legs



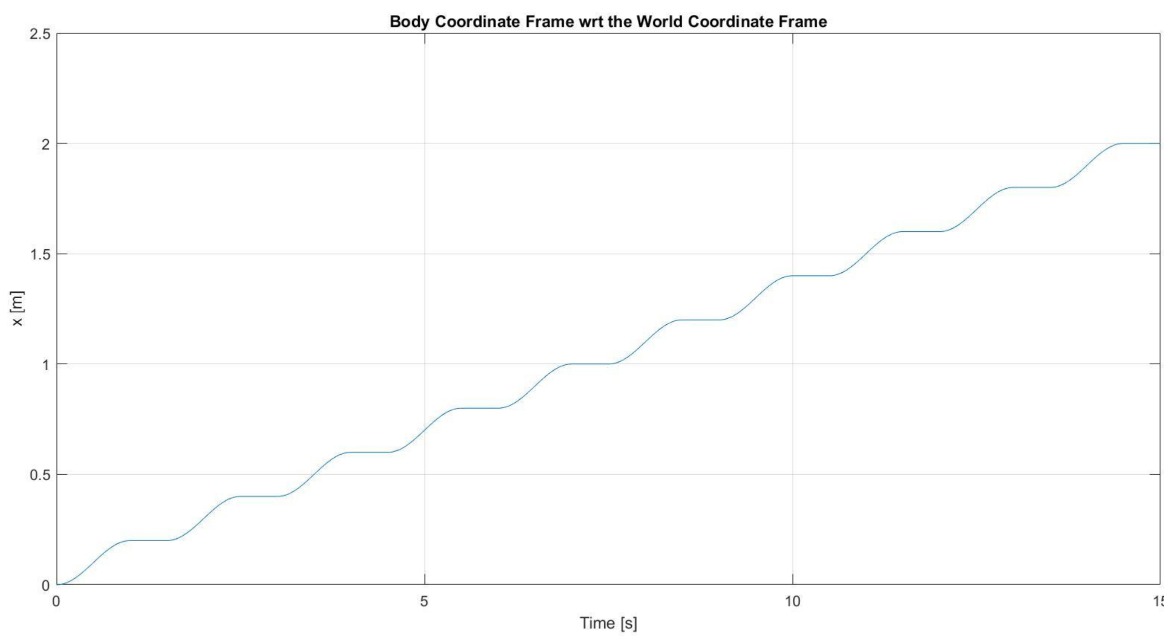
Z Reference for the Legs



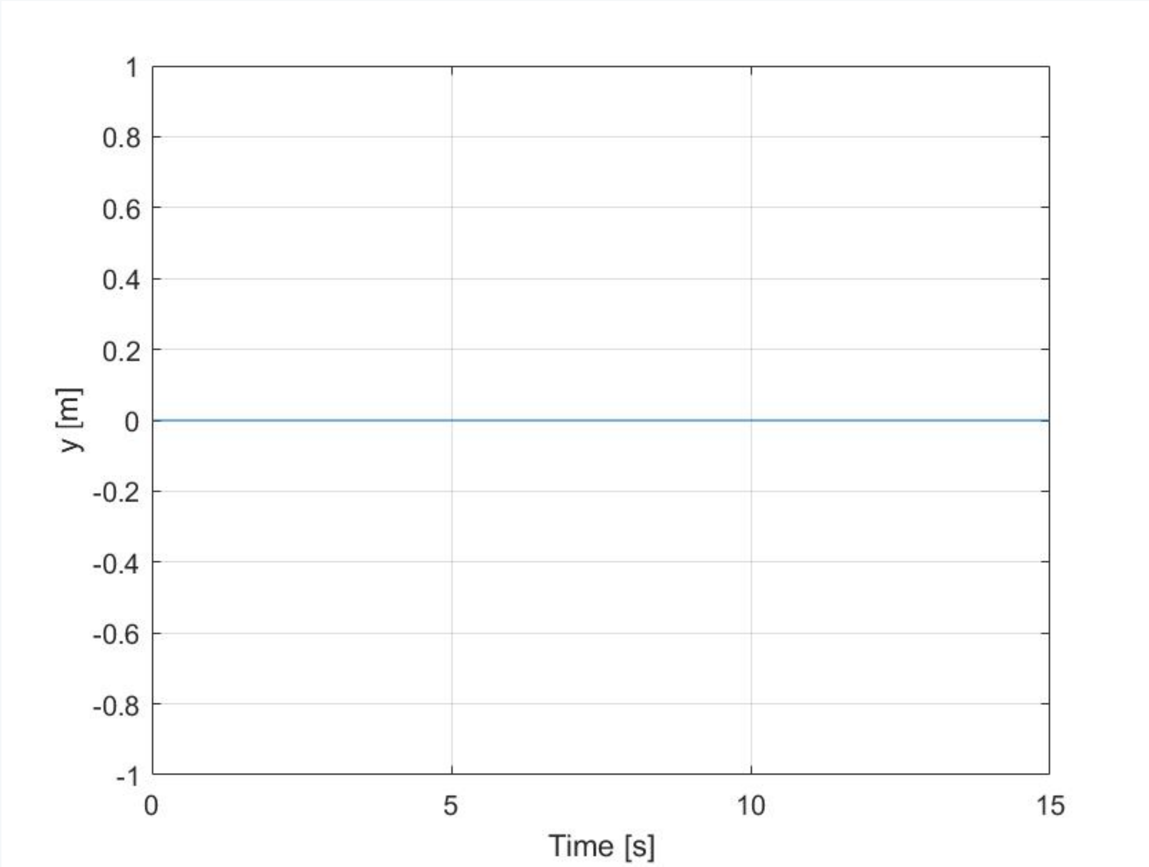
X, Y, and Z References for the Legs



Body Frame Origin Coordinates in the World Frame
X

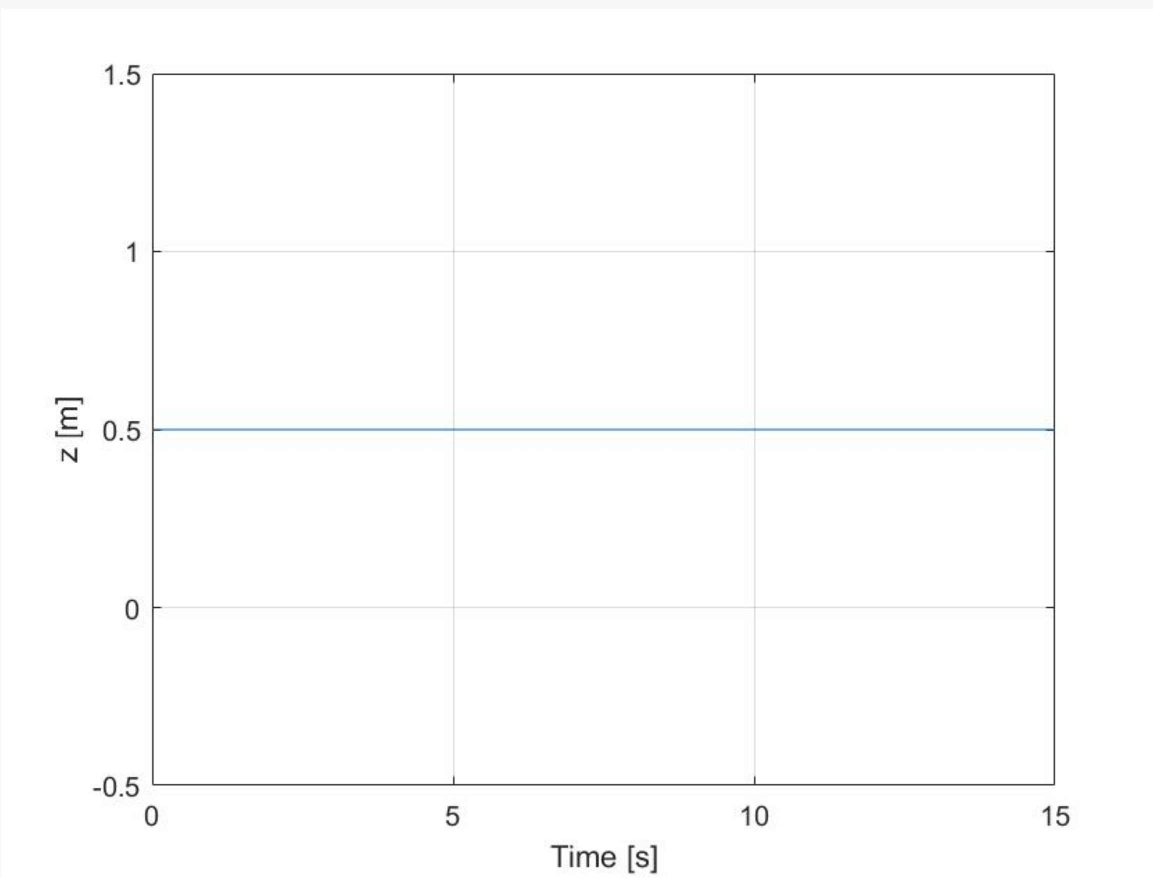


Y

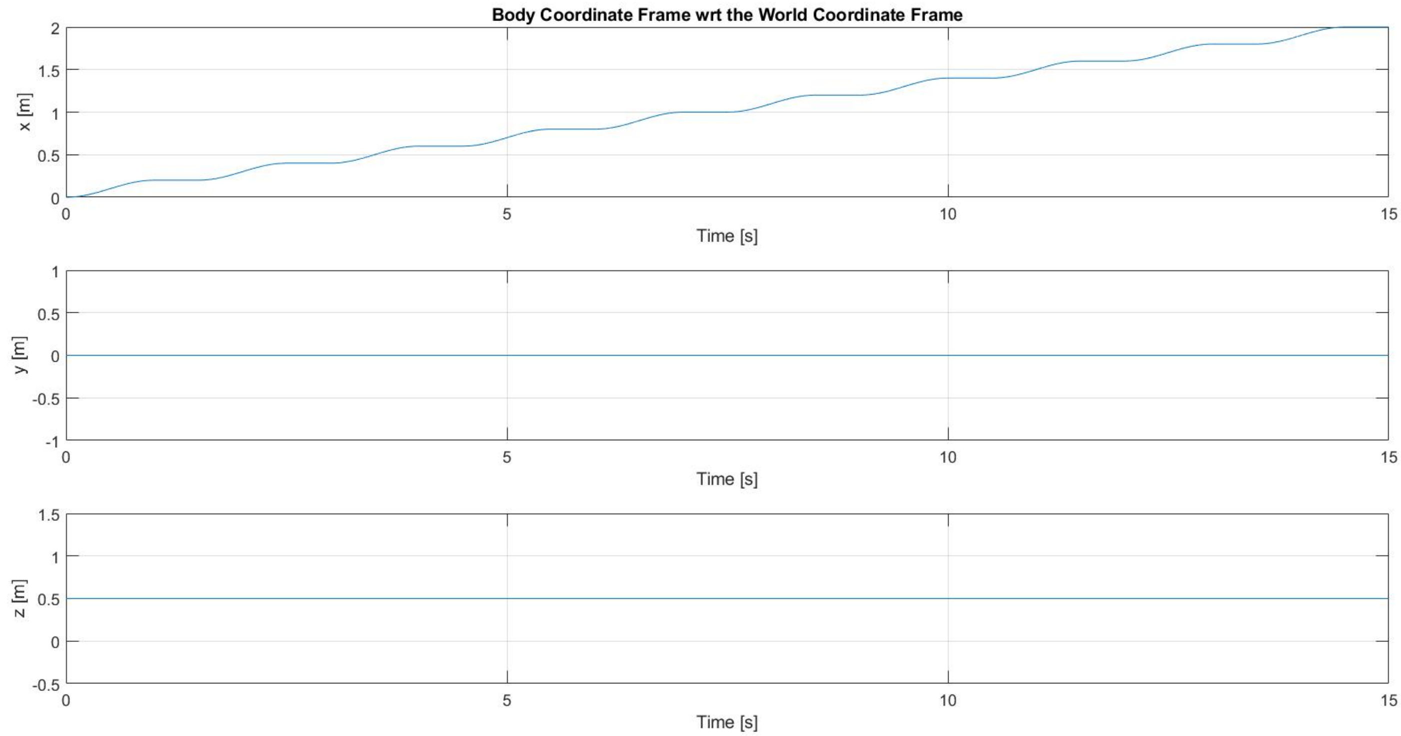


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y [m]
0
5
10
15
Time [s]

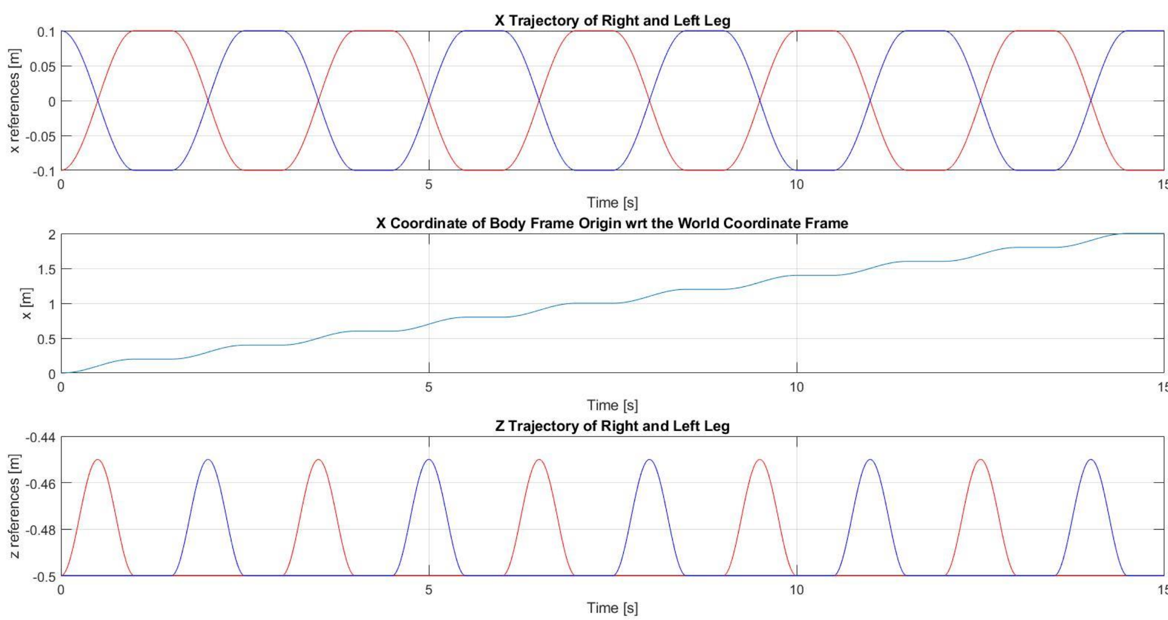
Z



1.5
1
0.5
0
-0.5
z [m]
0
5
10
15
Time [s]



Comparison Between Leg Trajectories and Origin Coordinate (wrt the world frame)



DISCUSSION

Notice: The x-coordinates of the body frame origin with respect to the world frame remains flat whenever the robot is on a double support by the two legs. This is realistic given that the body is momentarily at rest during the double support period and only moves while either leg is swinging.

APPENDIX

FOOT REFERENCE WITH RESPECT TO THE WORLD FRAME

****m-file name - Assignment_4_1

```
clear all
close all
clc

% DATA ENTRY
% ----------------------------------------
% Simulation Parameters
step_time = 0.001; % We'd like to have data for every
millisecond

% Recall: T_walk = 2(T_single + T_double)
T_walk = 3;
T_single = 1;
T_double = 0.5;
Step_size = 0.1;
Step_height = 0.05;
Body_height = 0.5;
y_offset = 0.4;

stop_time = 5*T_walk;

% Data Storage Dimension - The +1 is for time = 0s.
record_length = (stop_time/step_time) + 1;

% References for all axes for each leg.
x_right_list = zeros(record_length, 1);
x_left_list = zeros(record_length, 1);
y_right_list = zeros(record_length, 1);
y_left_list = zeros(record_length, 1);
z_right_list = zeros(record_length, 1);
z_left_list = zeros(record_length, 1);
time_list = zeros(record_length, 1);

% Foot Coordinates
x_left_foot_list = zeros(record_length, 1);
x_right_foot_list = zeros(record_length, 1);
```

```
y_left_foot_list = zeros(record_length, 1);
y_right_foot_list = zeros(record_length, 1);
z_left_foot_list = zeros(record_length, 1);
z_right_foot_list = zeros(record_length, 1);

x_l_list = zeros(record_length, 1);
x_r_list = zeros(record_length, 1);

for iteration_index = 1:1:record_length
   time = (iteration_index - 1)*step_time;
   % Normalize the time to always be between 0 and 1
   time_ratio_of_walk_period = (time/T_walk) - 
floor(time/T_walk);
   % Now, scale this value with the step period. This is to 
ensure that the
   % reference at any time instant could be correctly 
calculated.
   time_in_walk_period = time_ratio_of_walk_period * 
T_walk;

   % There are four segments per period: 0 <= t <= 
T_single,
   % T_single < t <= T_single + T_double, T_single + 
T_double < t <= 
   % 2*T_single + T_double, and 2*T_single + T_double < t.
   if (time_in_walk_period <= T_single)
       x_right = -Step_size + Step_size*(1 - 
cos(time_in_walk_period*pi/T_single));
       x_left = Step_size - Step_size*(1 - 
cos(time_in_walk_period*pi/T_single));
       y_right = -y_offset;
       y_left = y_offset;
       z_right = -Body_height + Step_height*0.5*(1 - 
cos(time_in_walk_period*2*pi/T_single));
       z_left = -Body_height;

       x_right_foot = Step_size*(1 - 
cos(time_in_walk_period*pi/T_single));
       x_left_foot = 0;
       y_right_foot = -y_offset;
       y_left_foot = y_offset;
```

```
        z_right_foot = Step_height*0.5*(1 -
cos(time_in_walk_period*2*pi/T_single));
        z_left_foot = 0;
    end

    if (T_single < time_in_walk_period  &&
time_in_walk_period <= T_single + T_double)
        x_right = Step_size;
        x_left = -Step_size;
        y_right = -y_offset;
        y_left = y_offset;
        z_right = -Body_height;
        z_left = -Body_height;

        x_right_foot = 0;
        x_left_foot = 0;
        y_right_foot = -y_offset;
        y_left_foot = y_offset;
        z_right_foot = 0;
        z_left_foot = 0;
    end

    if (T_single + T_double < time_in_walk_period  &&
time_in_walk_period <= 2*T_single + T_double)
        x_right = Step_size - Step_size*(1 -
cos((time_in_walk_period...
            - (T_single + T_double)) *pi/T_single));
        x_left = -Step_size + Step_size*(1 -
cos((time_in_walk_period...
            - (T_single + T_double))*pi/T_single));
        y_right = -y_offset;
        y_left = y_offset;
        z_right = -Body_height;
        z_left = -Body_height + Step_height*0.5*(1 -
cos((time_in_walk_period...
            - (T_single + T_double))*2*pi/T_single));

        x_right_foot = 0;
        x_left_foot = Step_size*(1 -
cos((time_in_walk_period...
            - (T_single + T_double))*pi/T_single));
        y_right_foot = -y_offset;
```

```
        y_left_foot = y_offset;
        z_right_foot = 0;
        z_left_foot = Step_height*0.5*(1 -
cos((time_in_walk_period...
            - (T_single + T_double))*2*pi/T_single));
    end

    if (2*T_single + T_double < time_in_walk_period)
        x_right = -Step_size;
        x_left = Step_size;
        y_right = -y_offset;
        y_left = y_offset;
        z_right = -Body_height;
        z_left = -Body_height;

        x_right_foot = 0;
        x_left_foot = 0;
        y_right_foot = -y_offset;
        y_left_foot = y_offset;
        z_right_foot = 0;
        z_left_foot = 0;
    end

    x_right_list(iteration_index) = x_right;
    x_left_list(iteration_index) = x_left;
    y_right_list(iteration_index) = y_right;
    y_left_list(iteration_index) = y_left;
    z_right_list(iteration_index) = z_right;
    z_left_list(iteration_index) = z_left;
    time_list(iteration_index) = time;

    x_r_list(iteration_index) = x_right_foot;
    x_l_list(iteration_index) = x_left_foot;

    if iteration_index == 1
        x_right_foot_list(iteration_index) = x_right_foot;
        x_left_foot_list(iteration_index) = x_left_foot;
    else
        x_right_foot_list(iteration_index) =
x_right_foot_list(iteration_index - 1)...
            + abs(x_r_list(iteration_index) -
x_r_list(iteration_index - 1));
```

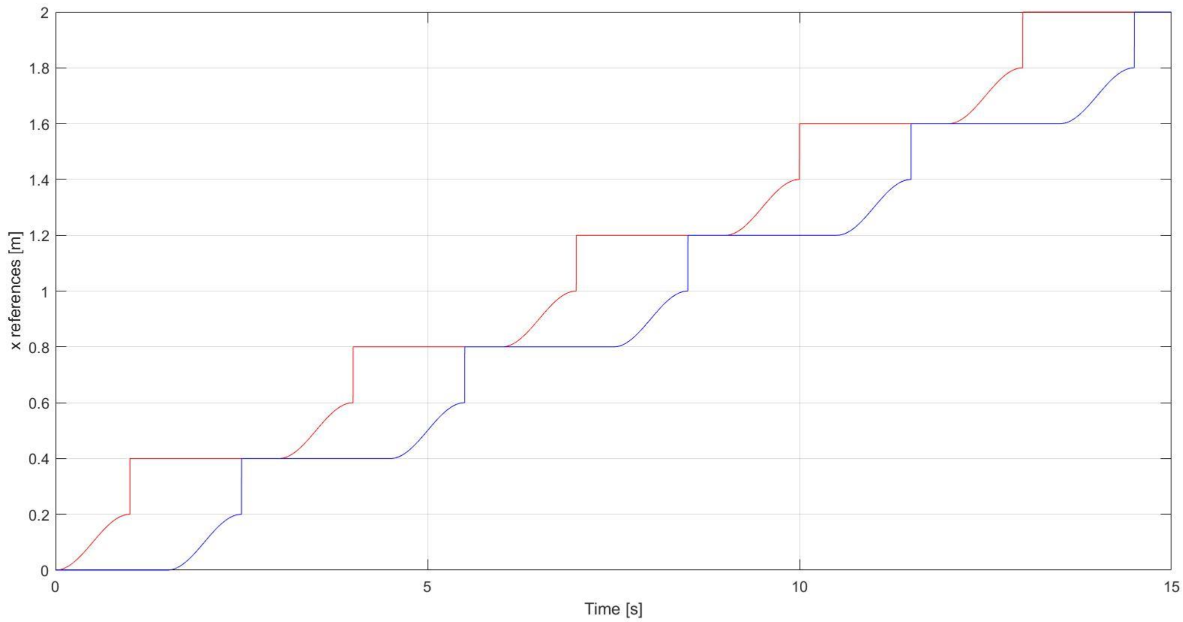
```
        x_left_foot_list(iteration_index) =
x_left_foot_list(iteration_index - 1)...
            + abs(x_l_list(iteration_index) -
x_l_list(iteration_index - 1));
   end

%    x_right_foot_list(iteration_index) = x_right_foot;
%    x_left_foot_list(iteration_index) = x_left_foot;
   y_right_foot_list(iteration_index) = y_right_foot;
   y_left_foot_list(iteration_index) = y_left_foot;
   z_right_foot_list(iteration_index) = z_right_foot;
   z_left_foot_list(iteration_index) = z_left_foot;

end

figure %1
plot(time_list, x_right_list, 'r')
ylim([-0.12 0.12])
hold
plot(time_list, x_left_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

figure %2
plot(time_list, y_right_list, 'r')
ylim([-0.5 0.5])
hold
plot(time_list, y_left_list, 'b')
xlabel('Time [s]')
ylabel('y references [m]')
grid

figure %3
plot(time_list, z_right_list, 'r')
ylim([-Body_height -Body_height+4/3*(Step_height)])
hold
plot(time_list, z_left_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid
```

```
figure %4
subplot(3,1,1)
plot(time_list, x_right_list, 'r')
hold
plot(time_list, x_left_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

subplot(3,1,2)
plot(time_list, y_right_list, 'r')
hold
plot(time_list, y_left_list, 'b')
xlabel('Time [s]')
ylabel('y references [m]')
grid

subplot(3,1,3)
plot(time_list, z_right_list, 'r')
hold
plot(time_list, z_left_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid

figure %5
plot(time_list, x_right_foot_list, 'r')
hold
plot(time_list, x_left_foot_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

figure %5
plot(time_list, y_right_foot_list, 'r')
hold
plot(time_list, y_left_foot_list, 'b')
xlabel('Time [s]')
ylabel('y references [m]')
grid

figure %6
```

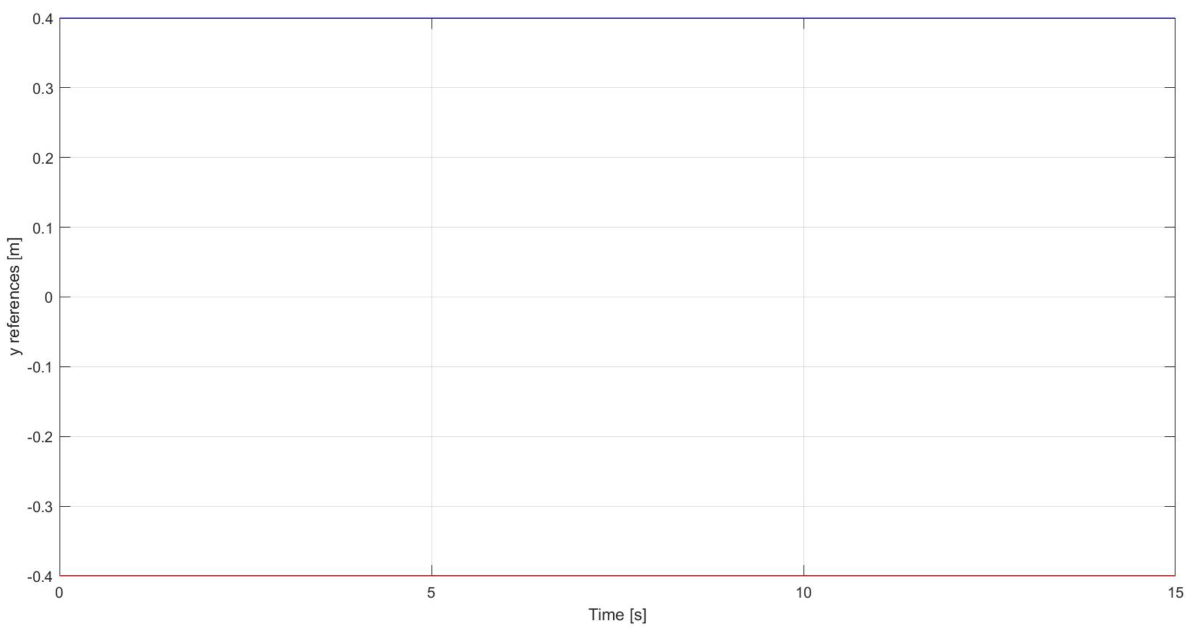
```
plot(time_list, z_right_foot_list, 'r')
%ylim([-Body_height -Body_height+4/3*(Step_height)])
hold
plot(time_list, z_left_foot_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid

figure %7
subplot(3,1,1)
plot(time_list, x_right_foot_list, 'r')
hold
plot(time_list, x_left_foot_list, 'b')
xlabel('Time [s]')
ylabel('x references [m]')
grid

subplot(3,1,2)
plot(time_list, y_right_foot_list, 'r')
hold
plot(time_list, y_left_foot_list, 'b')
xlabel('Time [s]')
ylabel('y references [m]')
grid

subplot(3,1,3)
plot(time_list, z_right_foot_list, 'r')
hold
plot(time_list, z_left_foot_list, 'b')
xlabel('Time [s]')
ylabel('z references [m]')
grid
```

Plots Foot References wrt the World Frame

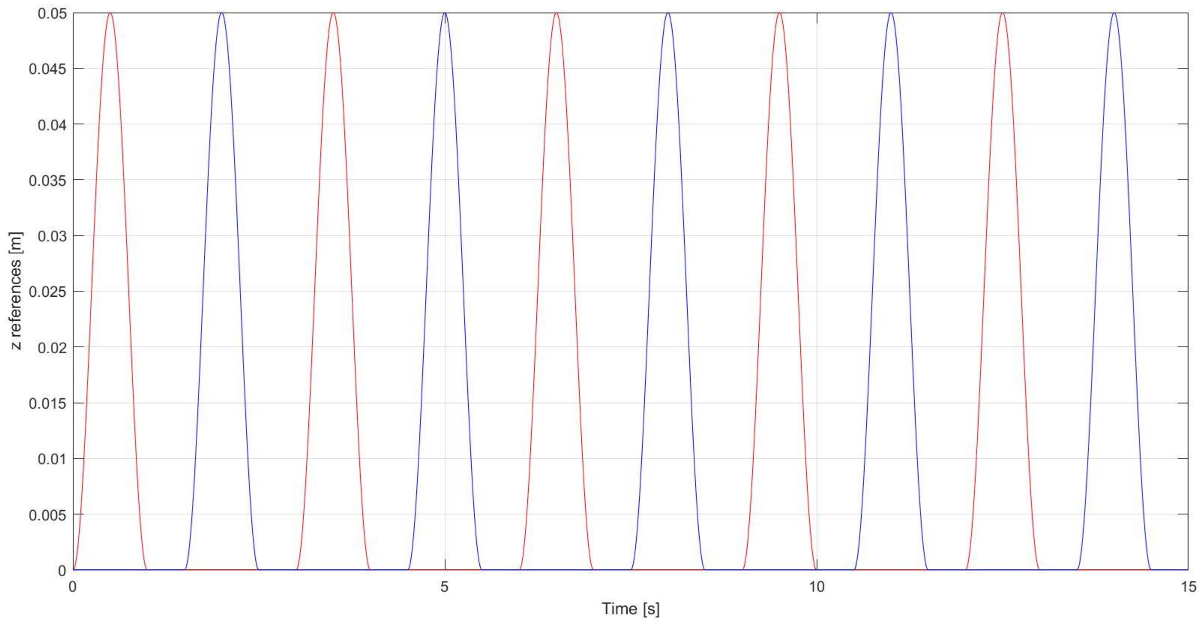
X

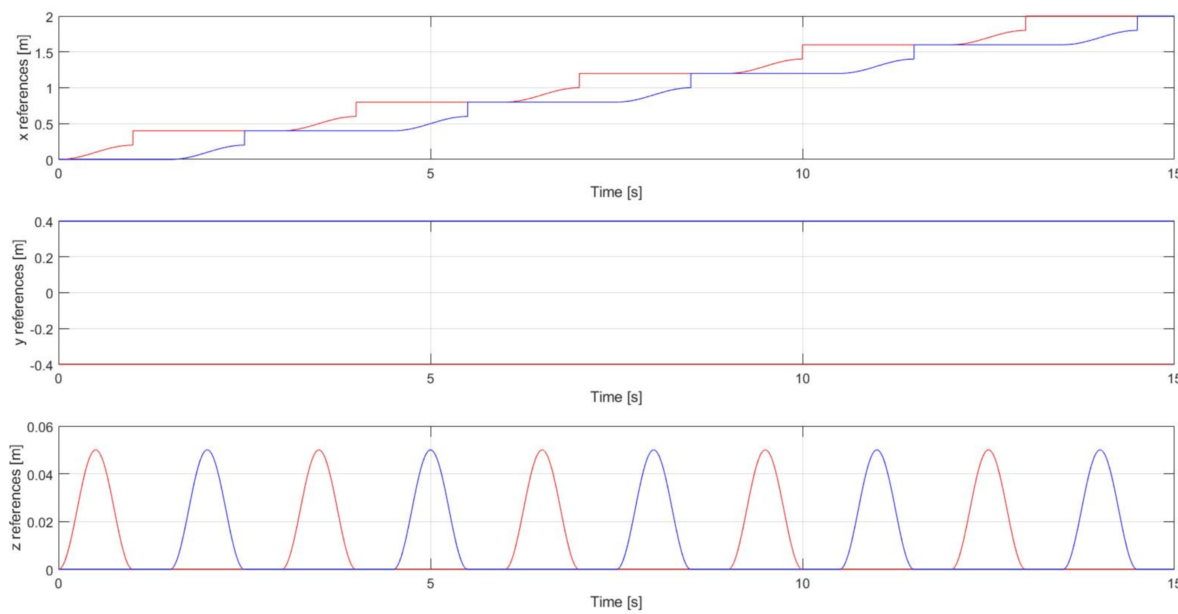


Y



Z





P.S. I couldn’t figure out how to determine accurate plots for the x coordinates.