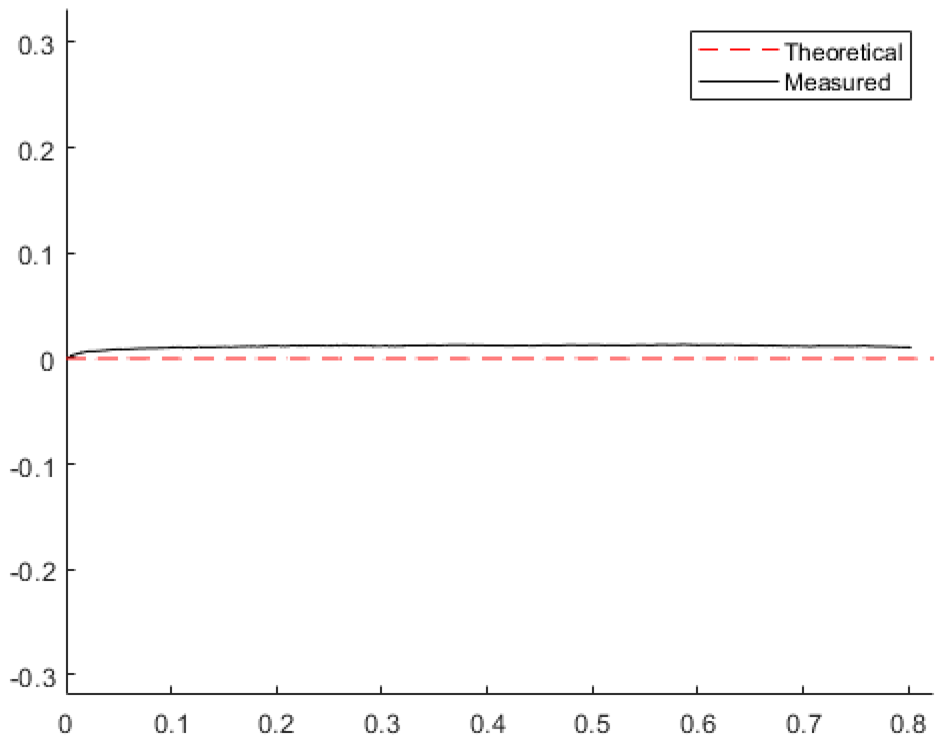
Table of Contents

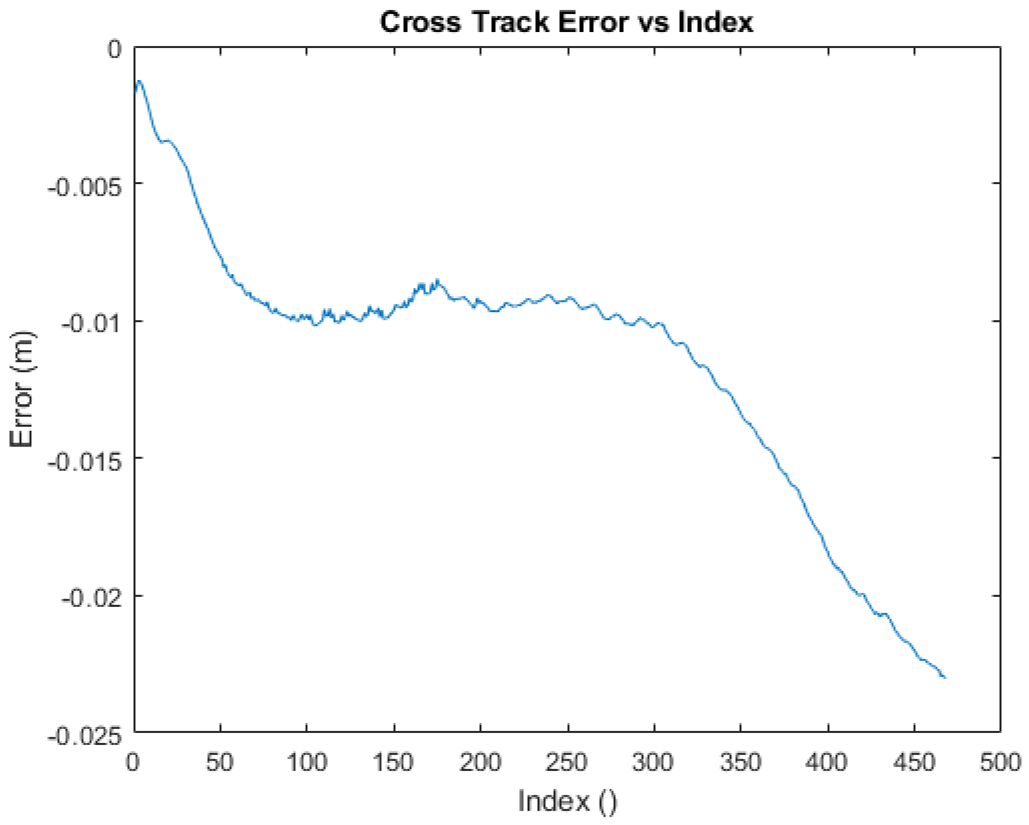
```
clearvars -except SLRobot PTRobot CWRobot CCWRobot
close all;
format compact;
clc;

dataBool = 1;

if dataBool
    SLRobot = readstruct("G2/G2StraightLine.json");
    PTRobot = readstruct("G2/G2PointTurn.json");
    CWRobot = readstruct("G2/G2CWNinety_001.json");
    CCWRobot = readstruct("G2/G2CCWNinety_001.json");
end
```

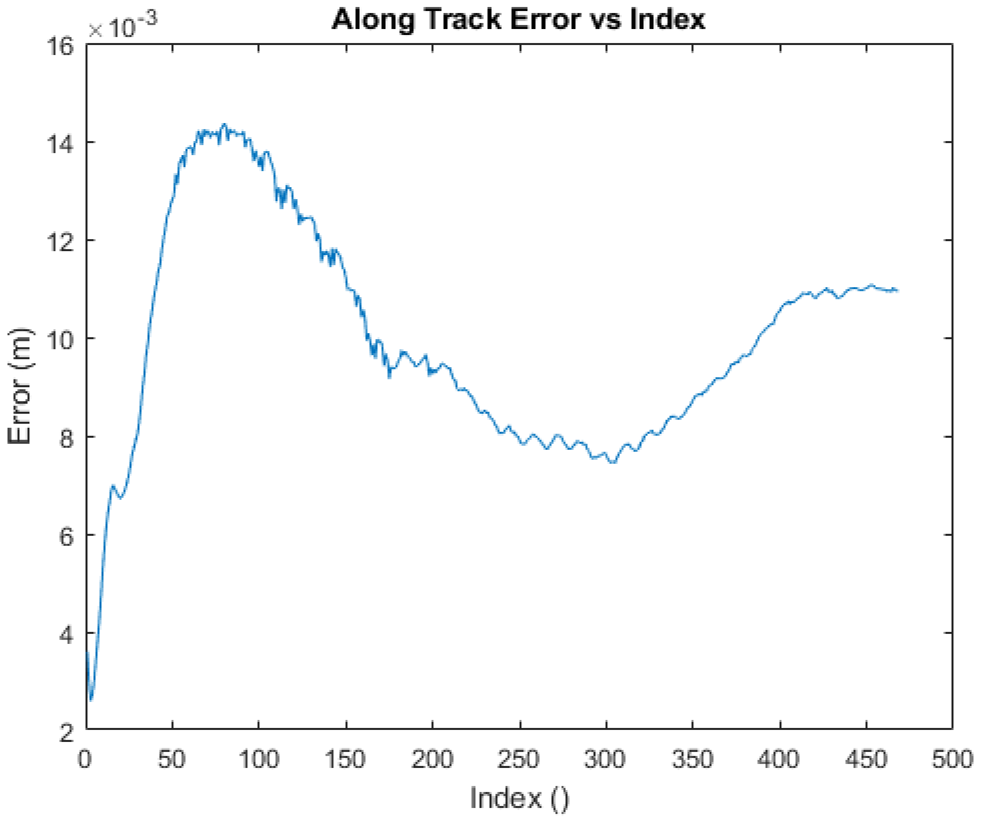
Straight Line

```
sample_length = numel(SLRobot.MobRobX);
dt= SLRobot.Time(2) - SLRobot.Time(1);
vx= (SLRobot.MobRobX(sample_length/2) - SLRobot.MobRobX(sample_length/2
-1)) / dt;
startIndex = 320;
endIndex = 788;

x_vals=0;
z_vals=0;
phi(1)=0;

for i = 2:(endIndex - startIndex)
    x_vals(i) = x_vals(i-1) + vx*cos(phi(i-1))*dt;
    z_vals(i) = z_vals(i-1) + vx*sin(phi(i-1))*dt;
    phi(i) = phi(i-1) - (0)/sample_length;

end

figure()
hold on
plot(x_vals, z_vals, '--r')
plot(SLRobot.MobRobX(startIndex:endIndex),
-SLRobot.MobRobZ(startIndex:endIndex), 'k');
legend("Theoretical", "Measured")
axis('equal')
hold off
```

```
saveas(gcf, "StraightLine", 'jpg');

[XTE, ATE] = trackErrorCalc(x_vals, z_vals, SLRobot.MobRobX(startIndex:
(endIndex-1)), SLRobot.MobRobZ(startIndex:(endIndex-1)), phi);

figure()
plot(XTE)
title("Cross Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "StraightLineXTE", 'jpg')

figure()
plot(ATE)
title("Along Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "StraightLineATE", 'jpg')

clear x_vals z_vals phi
```




Cross Track Error vs Index
Error (m)
Index ()
0
-0.005
-0.01
-0.015
-0.02
-0.025
0
50
100
150
200
250
300
350
400
450
500

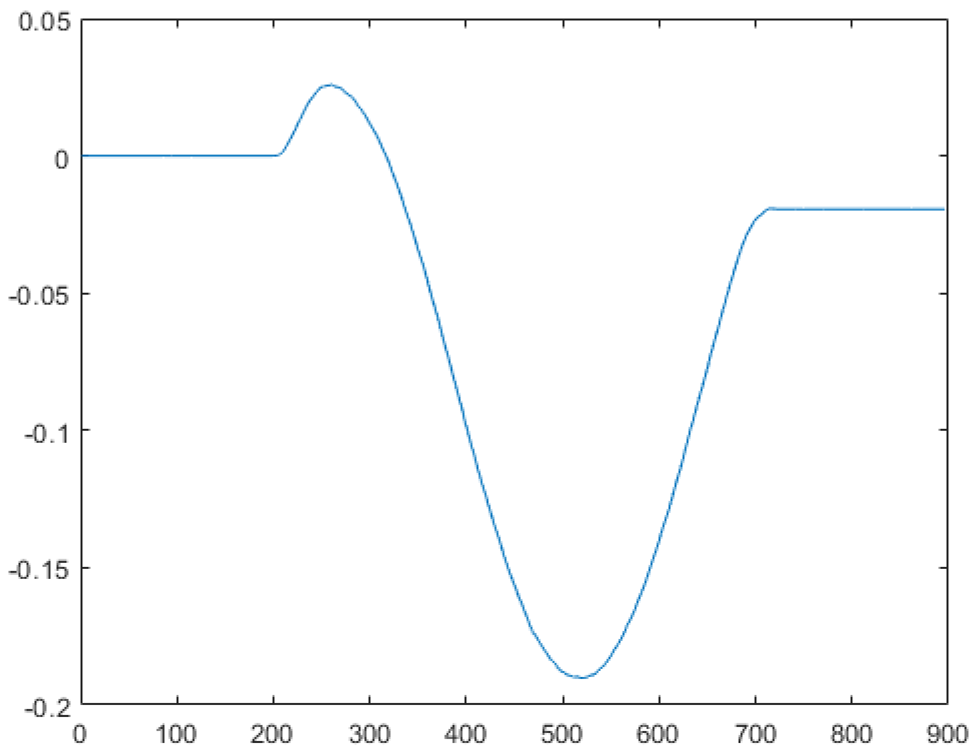

Along Track Error vs Index
×10^-3
Error (m)
Index ()
16
14
12
10
8
6
4
2
0
50
100
150
200
250
300
350
400
450
500

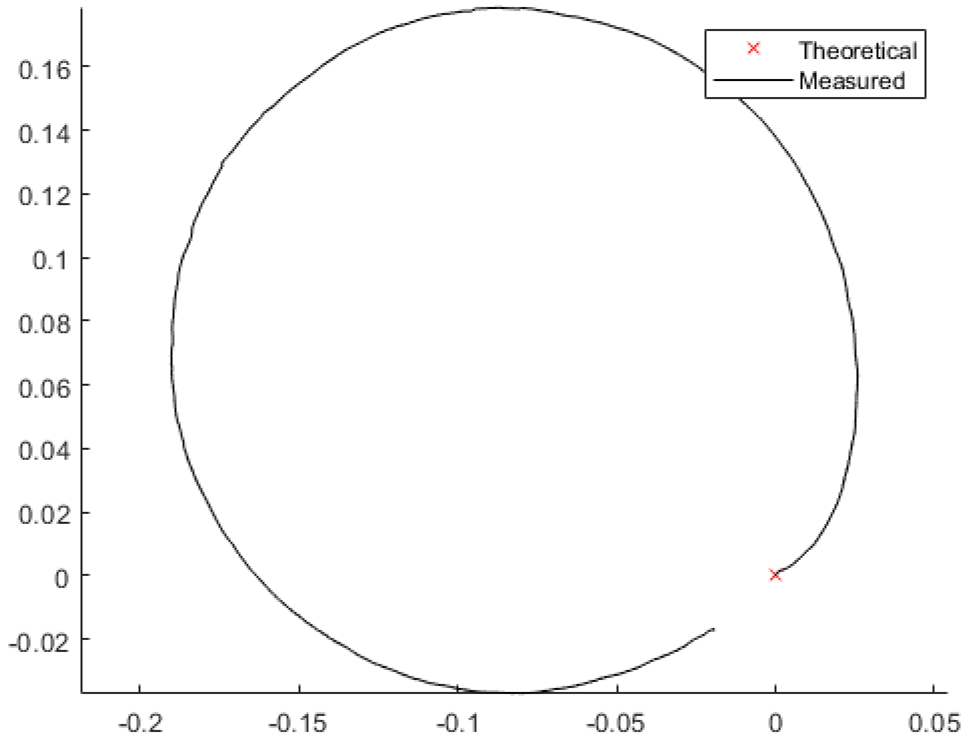
Point Turn

```
sample_length = numel(PTRobot.MobRobX);
dt= PTRobot.Time(2) - PTRobot.Time(1);
vx= (PTRobot.MobRobX(sample_length/2) - PTRobot.MobRobX(sample_length/2
-1)) / dt;
vz= (PTRobot.MobRobZ(sample_length/2) - PTRobot.MobRobZ(sample_length/2
-1)) / dt;
v = sqrt(vx^2 + vz^2);

figure();
plot(PTRobot.MobRobX);

startIndex = 210;
endIndex = 712;


x_vals(1)=0;
z_vals(1)=0;
phi(1)=0;

for i = 2:(endIndex - startIndex)
    x_vals(i) = x_vals(i-1) + 0*v*cos(phi(i-1))*dt;
    z_vals(i) = z_vals(i-1) + 0*v*sin(phi(i-1))*dt;
    phi(i) = phi(i-1) + (2*pi)/sample_length;
end

figure()
hold on
plot(x_vals, z_vals, 'xr')
plot(PTRobot.MobRobX, -PTRobot.MobRobZ, 'k');
legend("Theoretical", "Measured")
axis('equal')
hold off
saveas(gcf, "PointTurn", 'jpg');

[XTE, ATE] = trackErrorCalc(x_vals, z_vals, PTRobot.MobRobX(startIndex:
(endIndex-1)), PTRobot.MobRobZ(startIndex:(endIndex-1)), phi);

figure()
plot(XTE)
title("Cross Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "PointTurnXTE", 'jpg')

figure()
plot(ATE)
title("Along Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "PointTurnATE", 'jpg')
```

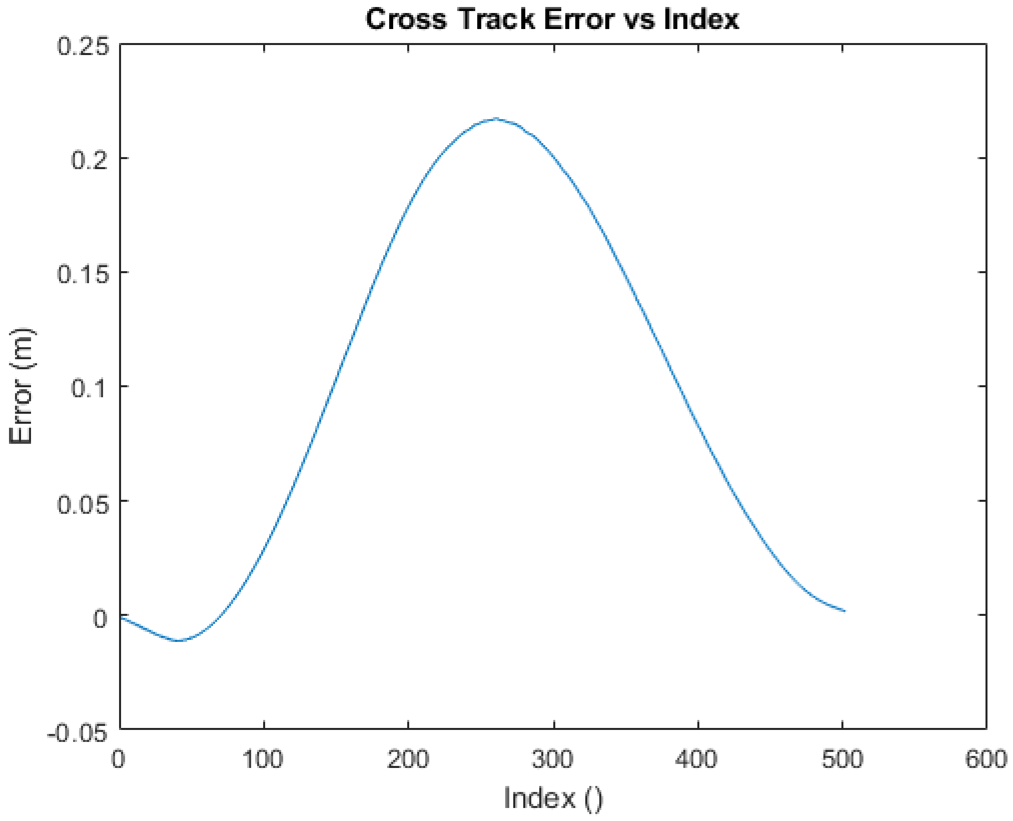
```
clear x_vals z_vals phi
```

```
Warning: Rank deficient, rank = 0, tol =  0.000000e+00.
```

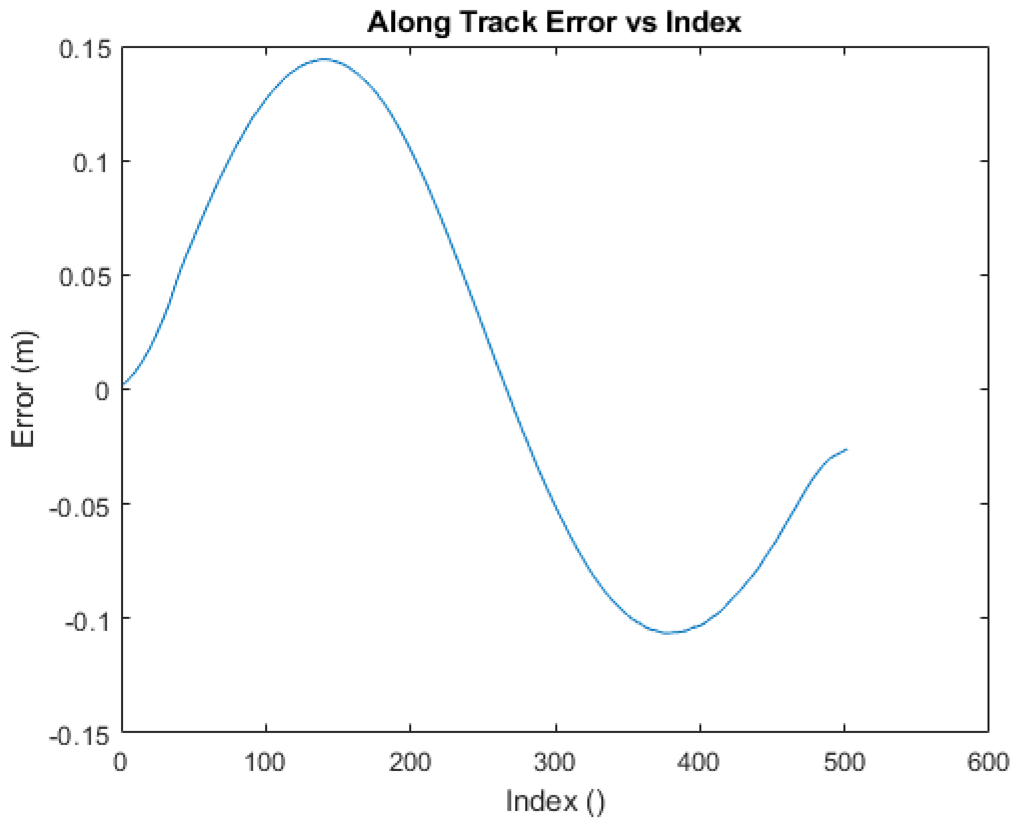




Theoretical
Measured
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.2
-0.15
-0.1
-0.05
0
0.05



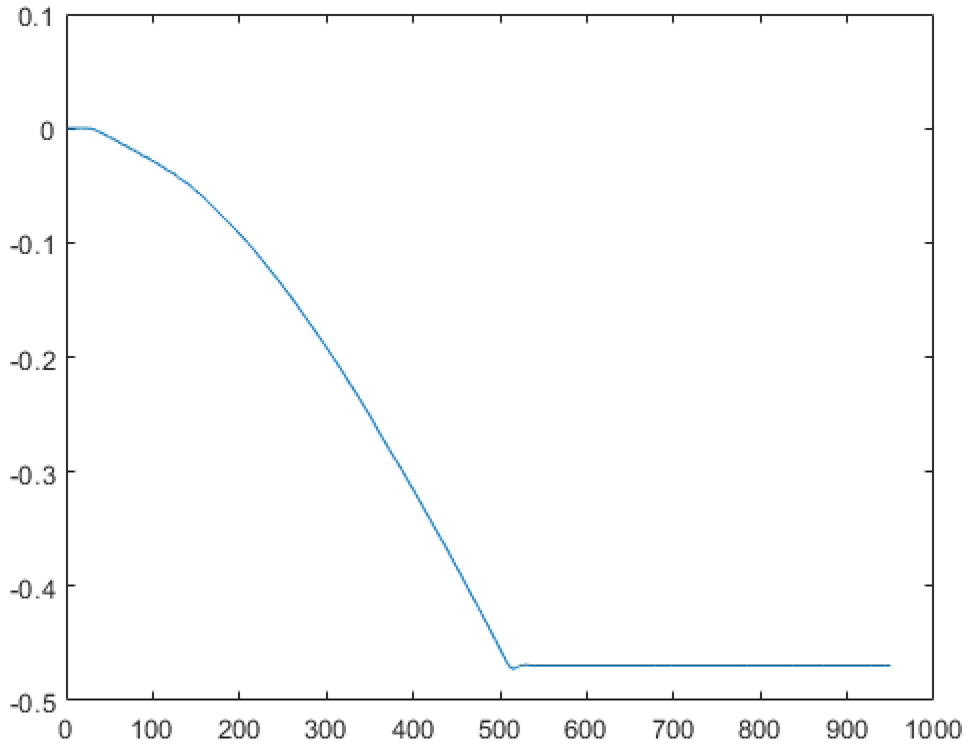
Cross Track Error vs Index
Error (m)
Index ()
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
100
200
300
400
500
600

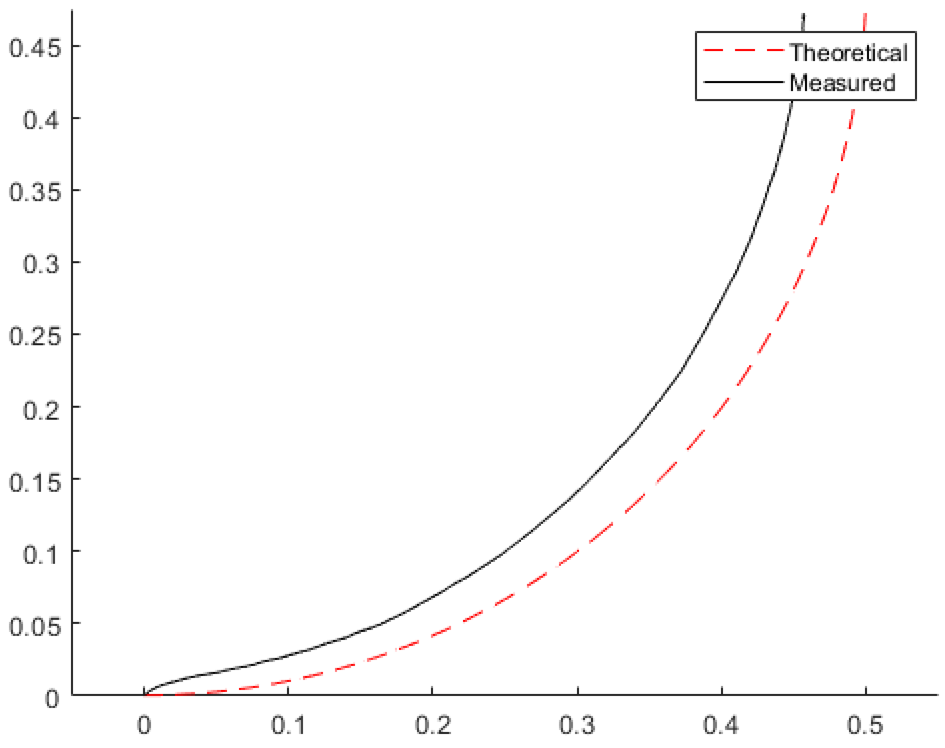


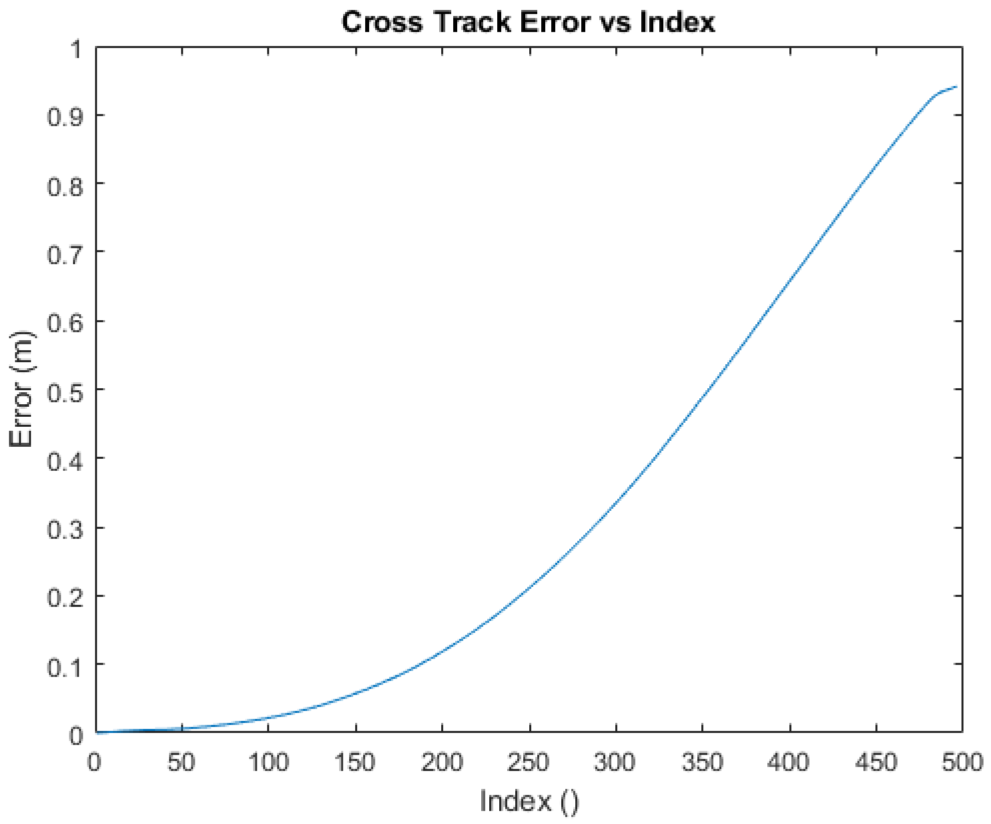
CCW Ninety

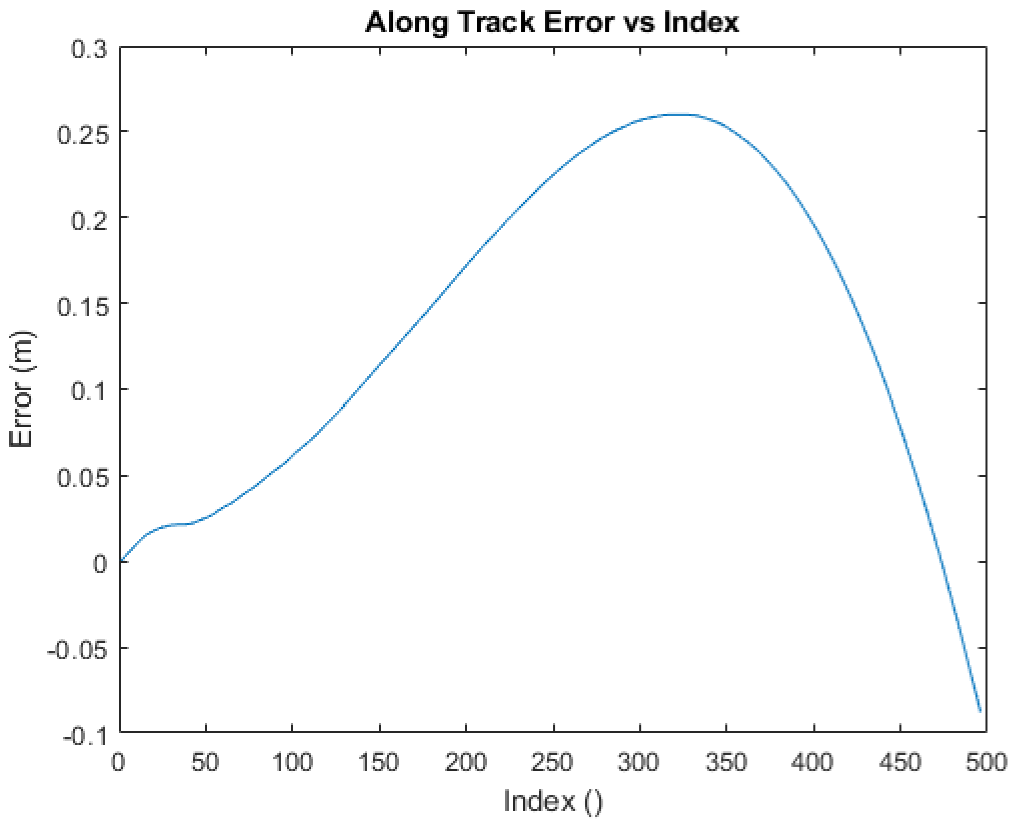
```
sample_length = numel(CCWRobot.MobRobX);
dt= CCWRobot.Time(2) - CCWRobot.Time(1);
vx= (CCWRobot.MobRobX(sample_length/2) - CCWRobot.MobRobX(sample_length/2
-1)) / dt;
vz= (CCWRobot.MobRobZ(sample_length/2) - CCWRobot.MobRobZ(sample_length/2
-1)) / dt;
v = sqrt(vx^2 + vz^2);

figure();
plot(CCWRobot.MobRobZ)

startIndex = 30;
endIndex = 526;

x_vals=0;
z_vals=0;
phi(1)=0;
circle_radius = .5;
AngVel= v/circle_radius;


for i = 2:(endIndex - startIndex)
     x_vals(i) = x_vals(i-1) + v*cos(phi(i-1))*dt;
```

```
    z_vals(i) = z_vals(i-1) + v*sin(phi(i-1))*dt;
    phi(i) = phi(i-1) + AngVel*dt;
end

figure()
hold on
plot(x_vals, z_vals, '--r')
plot(CCWRobot.MobRobX, -CCWRobot.MobRobZ, 'k');
legend("Theoretical", "Measured")
axis('equal')
hold off
saveas(gcf, "CCW", 'jpg');

[XTE, ATE] = trackErrorCalc(x_vals, z_vals, CCWRobot.MobRobX(startIndex:
(endIndex-1)), CCWRobot.MobRobZ(startIndex:(endIndex-1)), phi);

figure()
plot(XTE)
title("Cross Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "CCWXTE", 'jpg')

figure()
plot(ATE)
title("Along Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "CCWATE", 'jpg')

clear x_vals z_vals phi
```


0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
100
200
300
400
500
600
700
800
900
1000


Theoretical
Measured
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5

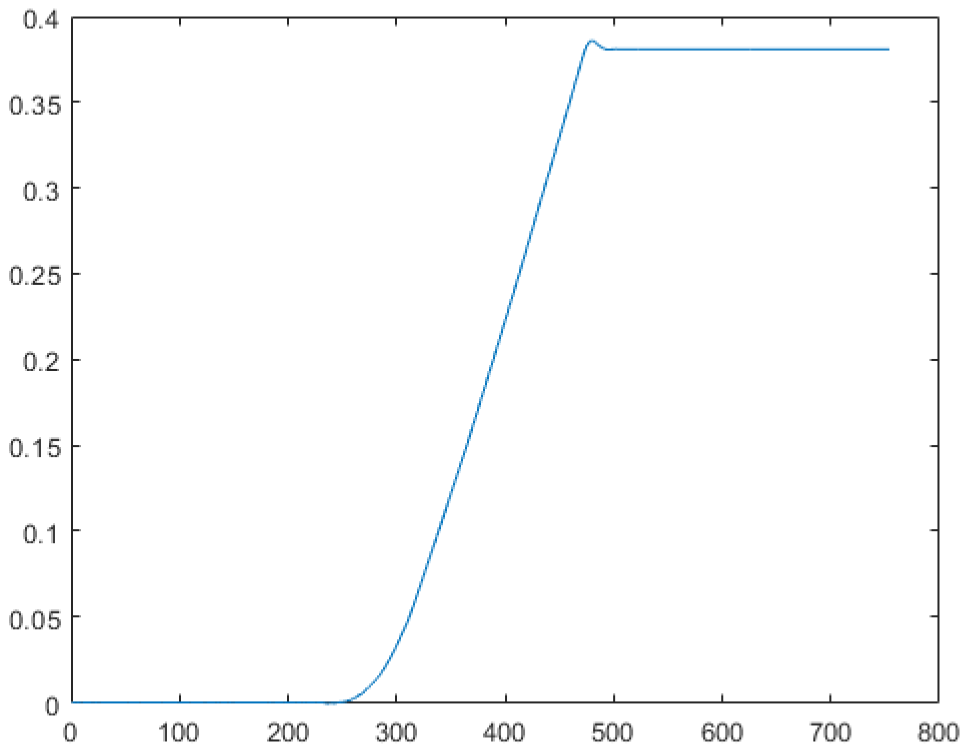

Cross Track Error vs Index
Error (m)
Index ()
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300
350
400
450
500

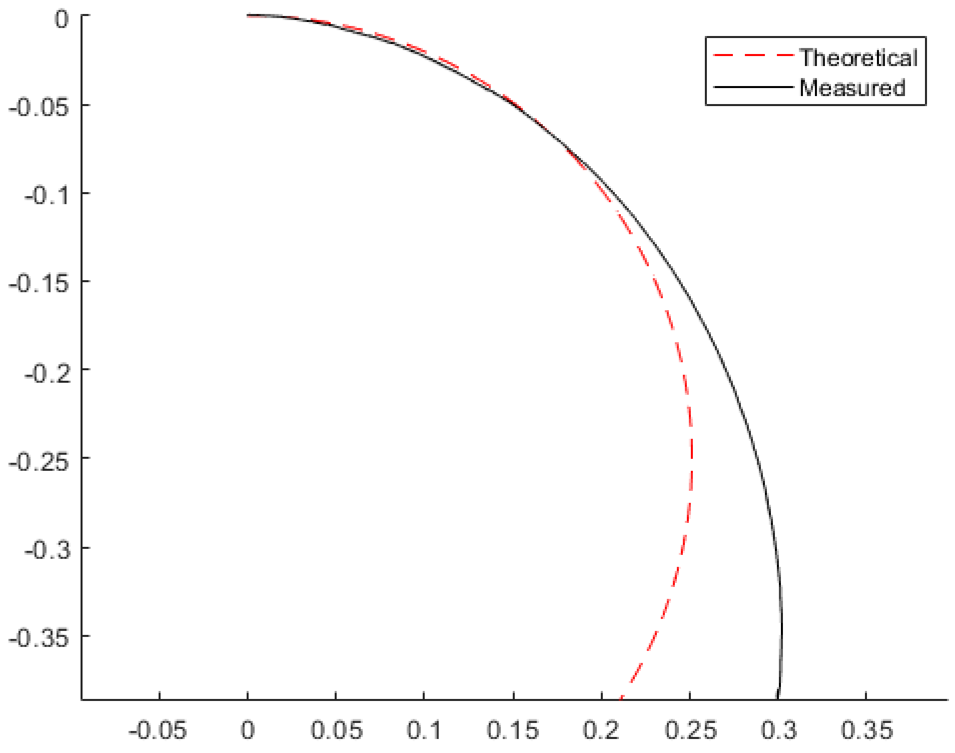

Along Track Error vs Index
Error (m)
Index ()
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3
0
50
100
150
200
250
300
350
400
450
500

CW Ninety

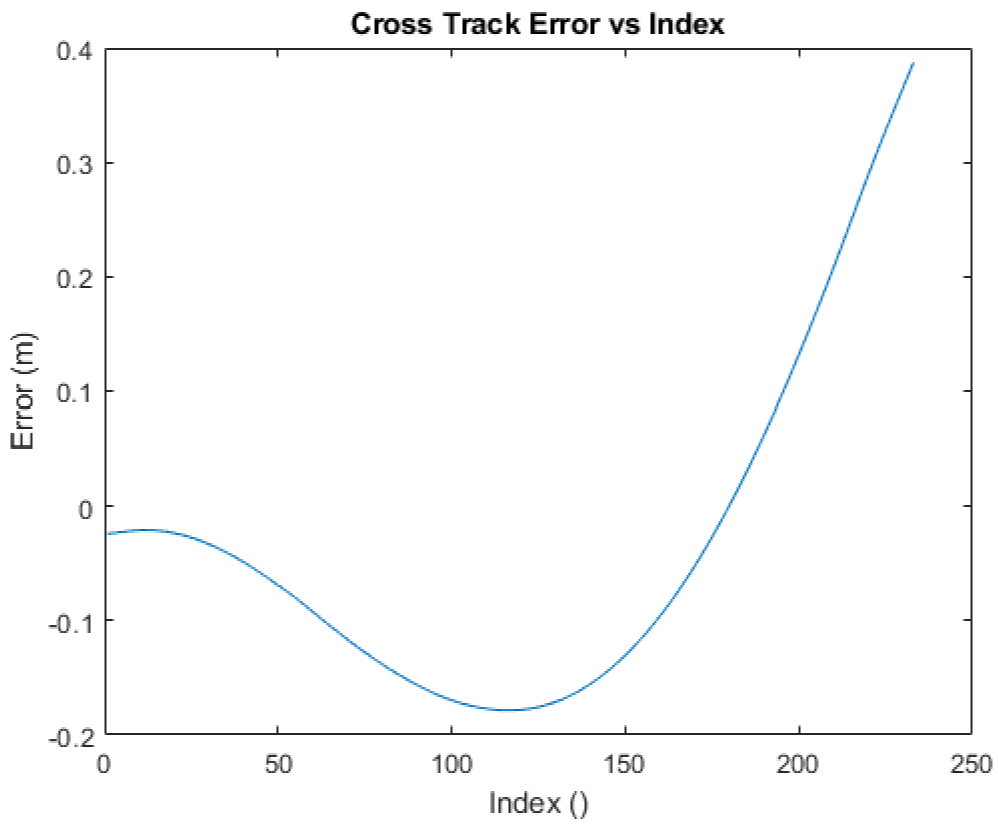
```
sample_length = numel(CWRobot.MobRobX);
dt= CWRobot.Time(2) - CWRobot.Time(1);
vx= (CWRobot.MobRobX(sample_length/2) - CWRobot.MobRobX(sample_length/2
-1)) / dt;
vz= (CWRobot.MobRobZ(sample_length/2) - CWRobot.MobRobZ(sample_length/2
-1)) / dt;
v = sqrt(vx^2 + vz^2);

figure();
plot(CWRobot.MobRobZ)

startIndex = 257;
endIndex = 490;

circle_radius = .25;

x_vals=0;
z_vals=0;
phi(1)=0;
AngVel= v/circle_radius;


for i = 2:(endIndex - startIndex)
     x_vals(i) = x_vals(i-1) + v*cos(phi(i-1))*dt;
    z_vals(i) = z_vals(i-1) + v*sin(phi(i-1))*dt;
    phi(i) = phi(i-1) - AngVel*dt;
end

figure()
hold on
plot(x_vals, z_vals, '--r')
plot(CWRobot.MobRobX, -CWRobot.MobRobZ, 'k');
legend("Theoretical", "Measured")
axis('equal')
hold off
saveas(gcf, "CW", 'jpg');

[XTE, ATE] = trackErrorCalc(x_vals, z_vals, CWRobot.MobRobX(startIndex:
(endIndex-1)), CWRobot.MobRobZ(startIndex:(endIndex-1)), phi);

figure()
plot(XTE)
title("Cross Track Error vs Index")
ylabel("Error (m)")
xlabel("Index ()")
saveas(gcf, "CWXTE", 'jpg')

figure()
plot(ATE)
title("Along Track Error vs Index")
ylabel("Error (m)")
```

```
xlabel("Index ()")
saveas(gcf, "CWATE", 'jpg')


clear x_vals z_vals phi
```

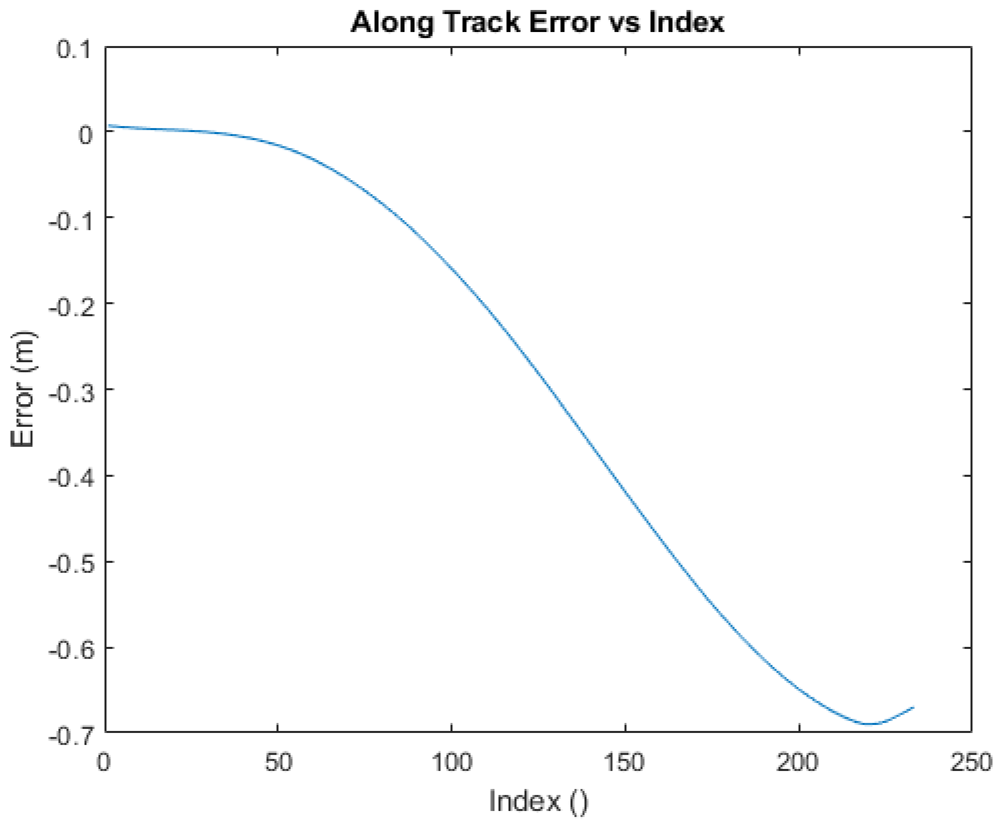




Theoretical
Measured
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35



Cross Track Error vs Index
Error (m)
Index ()
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
50
100
150
200
250



```
function [XTE, ATE] = trackErrorCalc(desX, desZ, robX, robZ, phi)

theta = atan(desZ - robZ / desX - robX);
B = sqrt((desX - robX).^2 + (desZ - robZ).^2);
alpha = 90 - theta - phi;
XTE = B.*cos(alpha);
ATE = B.*sin(alpha);

end
```

Published with MATLAB® R2024a