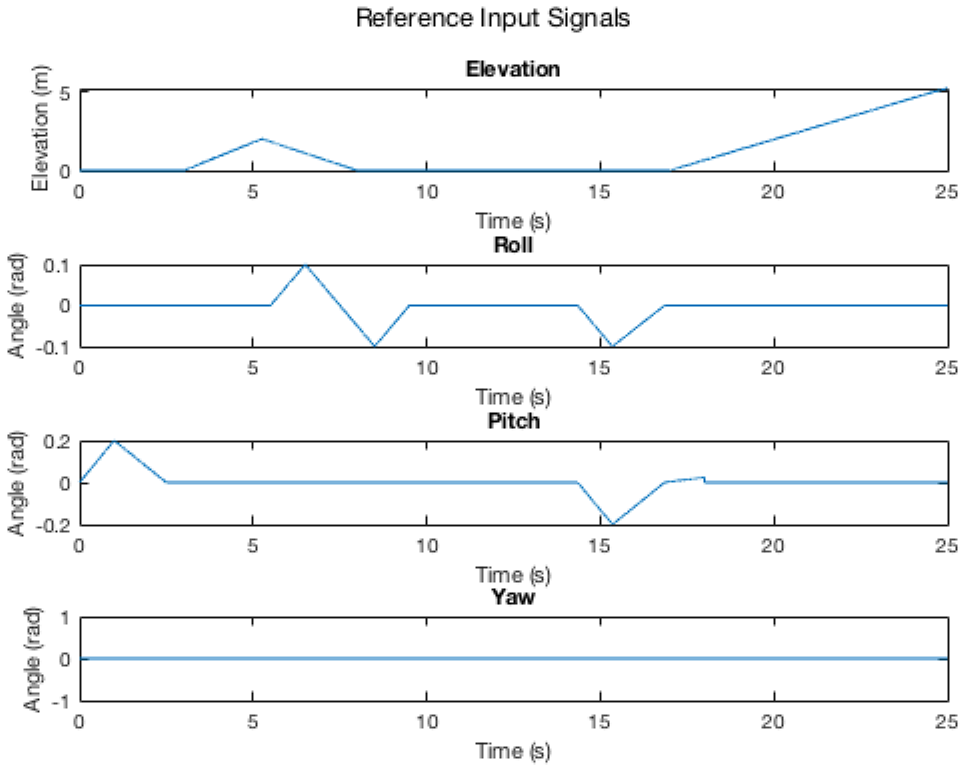
Table of Contents

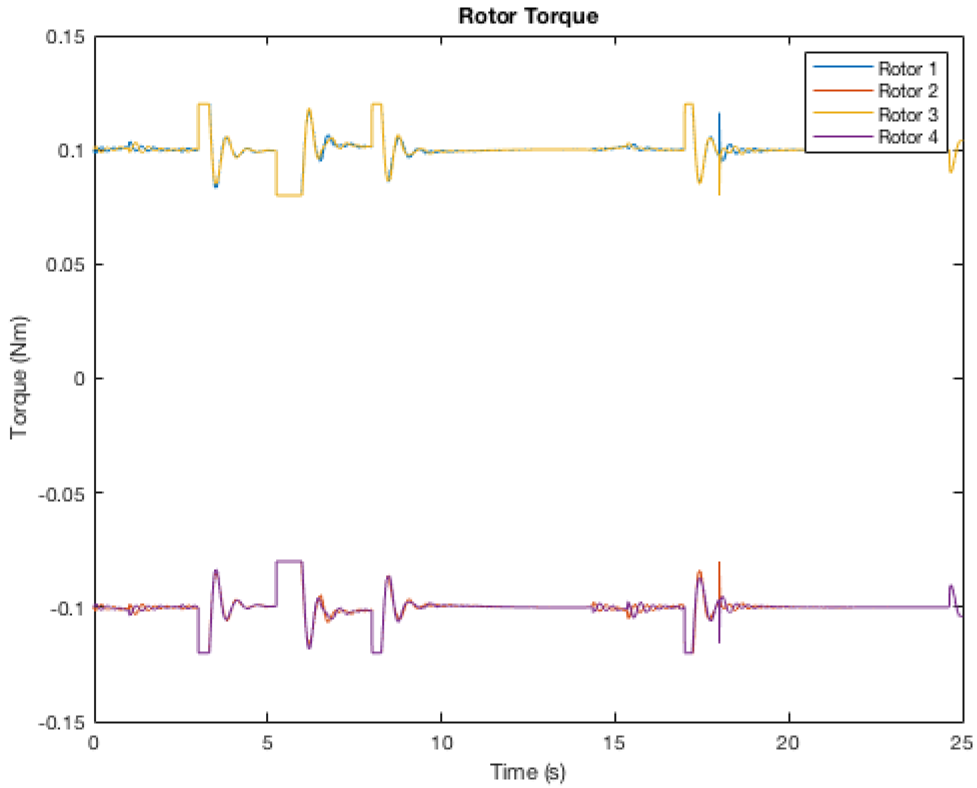
Input Signals

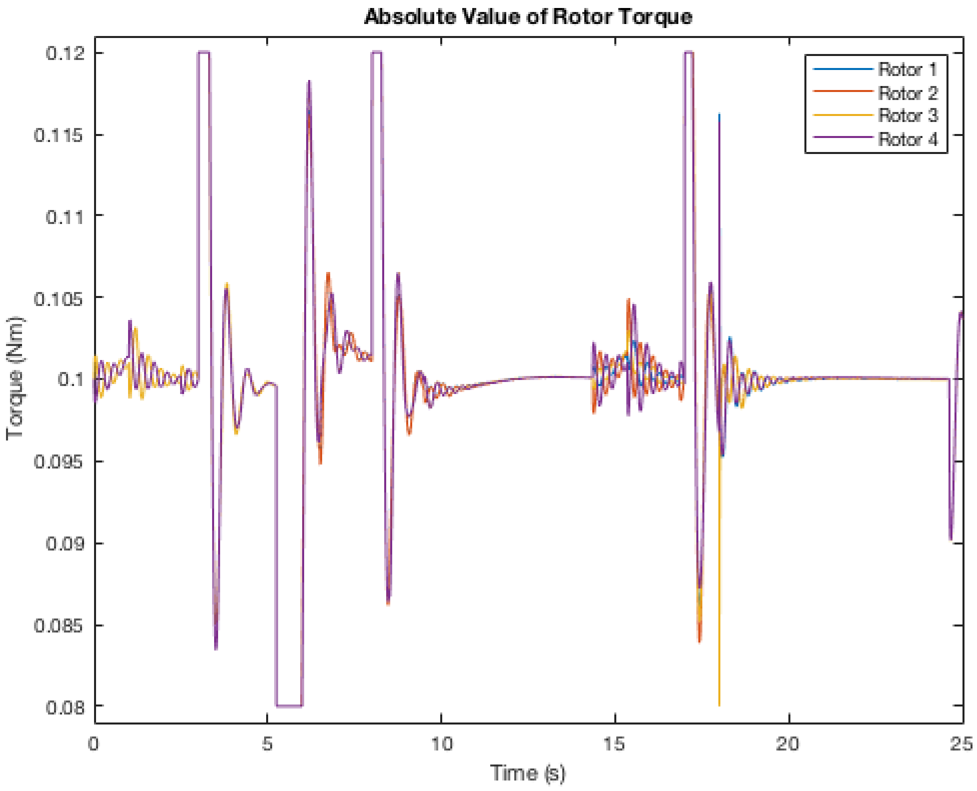
```
figure(1);
subplot(4,1,1); plot(Elevation.Time,Elevation.Data);
 title('Elevation'); xlabel('Time (s)'); ylabel('Elevation (m)');
subplot(4,1,2); plot(Roll.Time,Roll.Data); title('Roll'); xlabel('Time
 (s)'); ylabel('Angle (rad)');
subplot(4,1,3); plot(Pitch.Time,Pitch.Data); title('Pitch');
 xlabel('Time (s)'); ylabel('Angle (rad)');
subplot(4,1,4); plot(Yaw.Time,Yaw.Data); title('Yaw'); xlabel('Time
 (s)'); ylabel('Angle (rad)');
sgtitle('Reference Input Signals')
```



Torque Plots

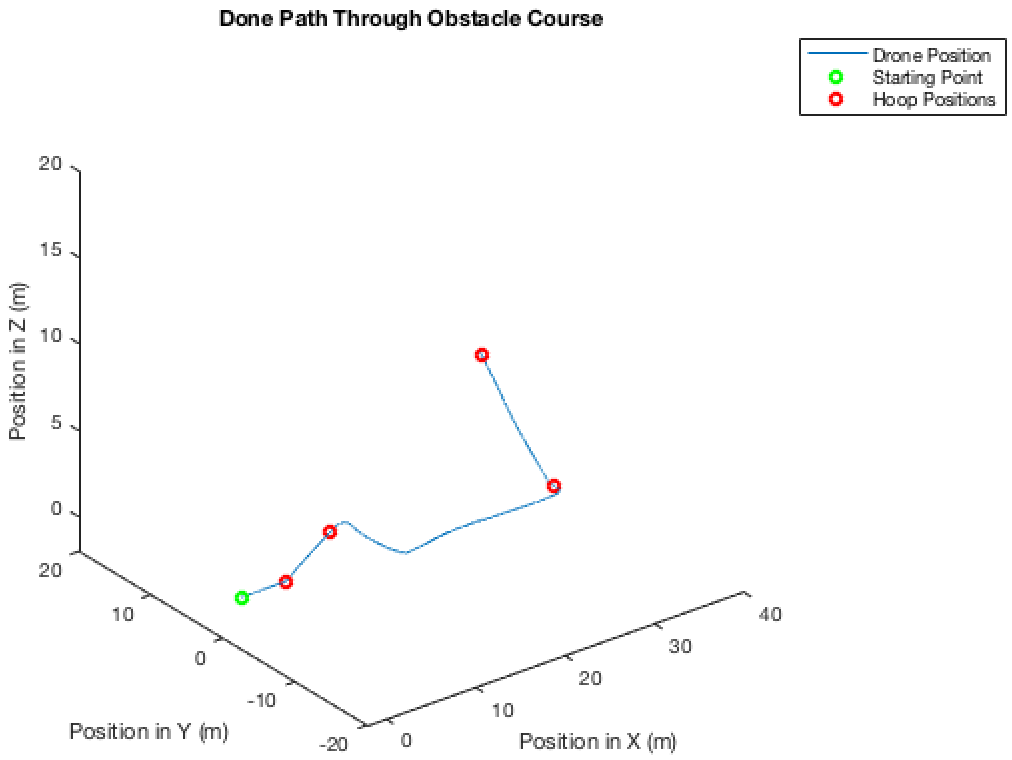
```
figure(2);
plot(rotorTorque1.Time,rotorTorque1.Data,rotorTorque2.Time,rotorTorque2.Data,rotor
title('Rotor Torque');xlabel('Time (s)');ylabel('Torque (Nm)');
legend('Rotor 1','Rotor 2','Rotor 3','Rotor 4');

figure(3);
plot(rotorTorque1.Time,rotorTorque1.Data,rotorTorque2.Time,abs(rotorTorque2.Data),
title('Absolute Value of Rotor Torque');xlabel('Time
 (s)');ylabel('Torque (Nm)');
legend('Rotor 1','Rotor 2','Rotor 3','Rotor 4'); ylim([.079 .121]);
```


Rotor Torque
Torque (Nm)
Time (s)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
5
10
15
20
25
Rotor 1
Rotor 2
Rotor 3
Rotor 4


Absolute Value of Rotor Torque
Torque (Nm)
Time (s)
0.12
0.115
0.11
0.105
0.1
0.095
0.09
0.085
0.08
0
5
10
15
20
25
Rotor 1
Rotor 2
Rotor 3
Rotor 4

Drone Navagation through Obstacle Course

```
figure(4);
plot3(droneX.Data,droneY.Data,droneZ.Data);
hold on; plot3(0,0,0,'o','Color','g');
plot3(5,0,0,'o','Color','r'); plot3(10,0,2,'o','Color','r');
plot3(35,0,0,'o','Color','r');plot3(35,10,5,'o','Color','r');
 hold off;
title('Done Path Through Obstacle Course');
legend('Drone Position','Starting Point','Hoop Positions');
xlabel('Position in X (m)');ylabel('Position in Y
 (m)');zlabel('Position in Z (m)');
xlim([-2 40]); ylim([-20 20]); zlim([-2 20]);
```



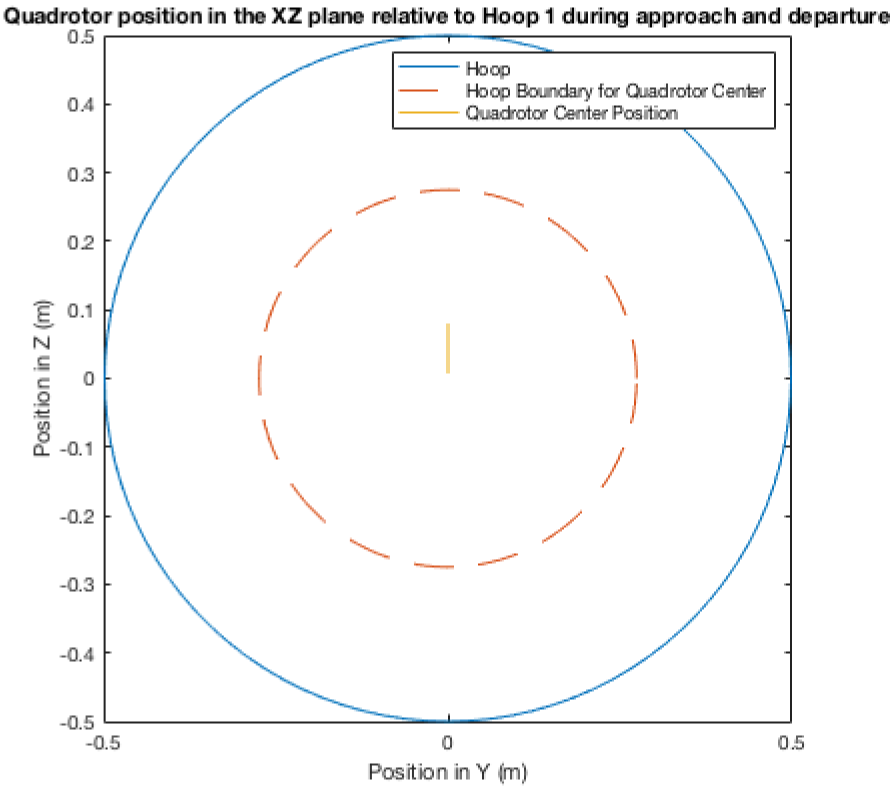
Hoop Validation

```
% Hoop Info
hoop1Time = 3.191423; hoop2Time = 5.18386;
hoop3Time = 16.57262; hoop4Time = 24.6207;
hoop1Pos = [5,0,0]; % Normal to X plane
hoop2Pos = [10,0,2]; % Normal to Z plane
hoop3Pos = [35,0,0]; % Normal to Y plane
hoop4Pos = [35,10,5]; % Normal to Z plane
droneBoundarySphereR = .225; % in m
hoopRadius = .5; hoopBoundaryRadius = hoopRadius-droneBoundarySphereR;
```

```
% The quadrotor could be contained in a sphere of radius 22.5 cm
 located at
% the center of the drone. If the hoop radius is shrunk by the same
 radius
% and the location of the center of the drone is plotted 22.5 cm
 before and
% after the time when it passes through the hoop in the plane of the
 hoop
% and the drone center stays within the bounds of the shrunken hoop,
 then
% it can be concluded that the drone passes thought the hoop without
% touching it.
```

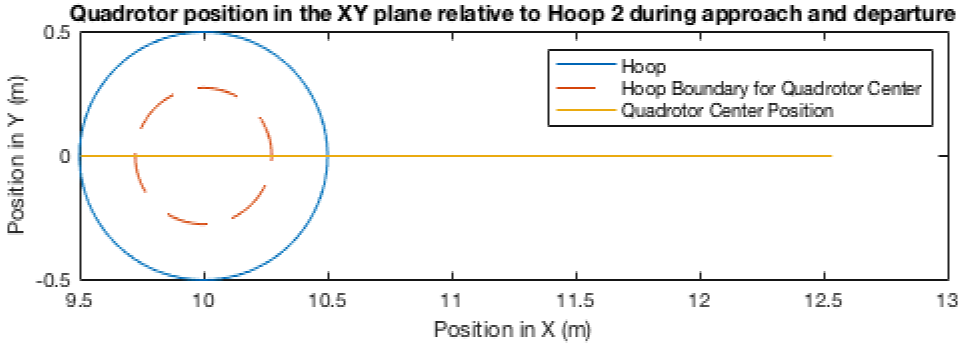
Hoop 1 in YZ Plane

```
th = 0:pi/50:2*pi;
yHoop = hoopRadius * cos(th);
zHoop = hoopRadius * sin(th);
yBoundary= hoopBoundaryRadius * cos(th);
zBoundary = hoopBoundaryRadius * sin(th);


i = find((hoop1Pos(1)-.225)<droneX.Data &
 droneX.Data<(hoop1Pos(1)+.225));
xH1 = droneX.Data(i(1):i(end));
yH1 = droneY.Data(i(1):i(end));
zH1 = droneZ.Data(i(1):i(end));

figure(5);
plot(yHoop, zHoop, yBoundary, zBoundary,'--',yH1,zH1);
title('Quadrotor position in the XZ plane relative to Hoop 1 during
 approach and departure')
legend('Hoop', 'Hoop Boundary for Quadrotor Center','Quadrotor Center
 Position');
xlabel('Position in Y (m)');ylabel('Position in Z (m)');
daspect([1 1 1]);
```



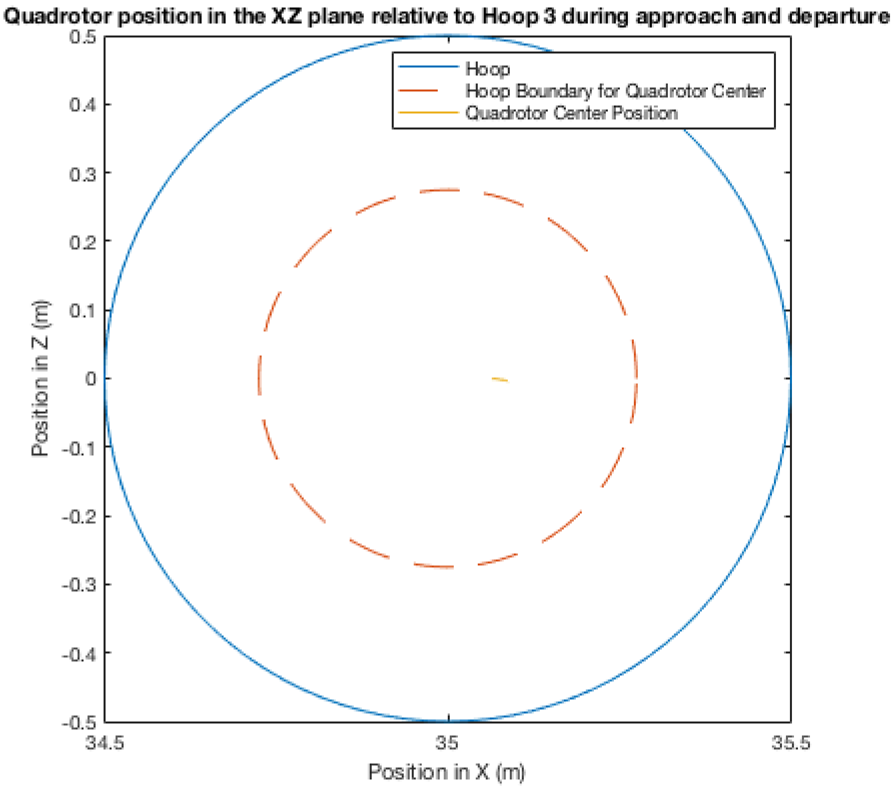
Hoop 2 in XY Plane

```
th = 0:pi/50:2*pi;
xHoop = hoopRadius * cos(th) + hoop2Pos(1);
yHoop = hoopRadius * sin(th) + hoop2Pos(2);
xBoundary= hoopBoundaryRadius * cos(th) + hoop2Pos(1);
yBoundary = hoopBoundaryRadius * sin(th) + hoop2Pos(2);

i = find((hoop2Pos(3)-.225)<droneZ.Data &
 droneZ.Data<(hoop2Pos(3)+.225));
xH2 = droneX.Data(i(1):i(467));
yH2 = droneY.Data(i(1):i(467));
zH2 = droneZ.Data(i(1):i(467));

figure(6);
plot(xHoop, yHoop, xBoundary, yBoundary,'--',xH2,yH2);
title('Quadrotor position in the XY plane relative to Hoop 2 during
 approach and departure')
legend('Hoop', 'Hoop Boundary for Quadrotor Center','Quadrotor Center
 Position');
xlabel('Position in X (m)');ylabel('Position in Y (m)');
daspect([1 1 1]);
```



Hoop 3 in XZ Plane

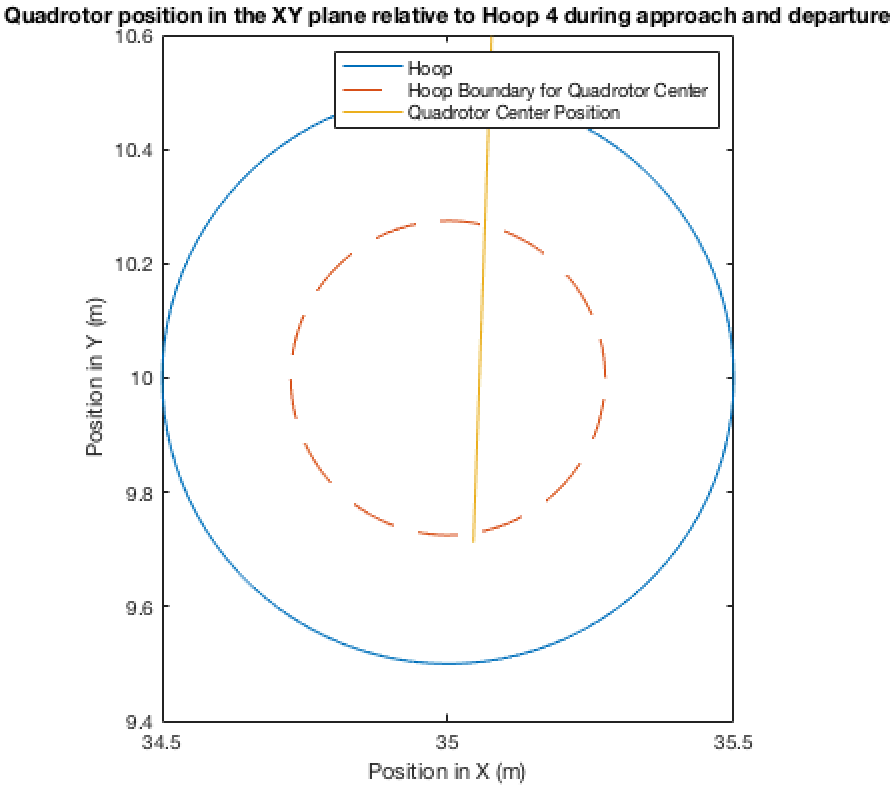
```
th = 0:pi/50:2*pi;
xHoop = hoopRadius * cos(th) + hoop3Pos(1);
zHoop = hoopRadius * sin(th) + hoop3Pos(3);
xBoundary= hoopBoundaryRadius * cos(th) + hoop3Pos(1);
zBoundary = hoopBoundaryRadius * sin(th) + hoop3Pos(3);

i = find((hoop3Pos(2)-.225)<droneY.Data &
 droneY.Data<(hoop3Pos(2)+.225) ...
    & (hoop3Time-1)<droneY.Time & droneY.Time<(hoop3Time+1)); %only
 consider times around hoop 3
xH3 = droneX.Data(i(1):i(end));
yH3 = droneY.Data(i(1):i(end));
zH3 = droneZ.Data(i(1):i(end));

figure(7);
plot(xHoop, zHoop, xBoundary, zBoundary,'--',xH3,zH3);
title('Quadrotor position in the XZ plane relative to Hoop 3 during
 approach and departure')
legend('Hoop', 'Hoop Boundary for Quadrotor Center','Quadrotor Center
 Position');
xlabel('Position in X (m)');ylabel('Position in Z (m)');
daspect([1 1 1]);
```



Hoop 4 in XY Plane

```
th = 0:pi/50:2*pi;
xHoop = hoopRadius * cos(th) + hoop4Pos(1);
yHoop = hoopRadius * sin(th) + hoop4Pos(2);
xBoundary= hoopBoundaryRadius * cos(th) + hoop4Pos(1);
yBoundary = hoopBoundaryRadius * sin(th) + hoop4Pos(2);

i = find((hoop4Pos(3)-.225)<droneZ.Data &
 droneZ.Data<(hoop4Pos(3)+.225));
xH4 = droneX.Data(i(1):i(end));
yH4 = droneY.Data(i(1):i(end));
zH4 = droneZ.Data(i(1):i(end));

figure(8);
plot(xHoop, yHoop, xBoundary, yBoundary,'--',xH4,yH4);
title('Quadrotor position in the XY plane relative to Hoop 4 during
 approach and departure')
legend('Hoop', 'Hoop Boundary for Quadrotor Center','Quadrotor Center
 Position');
xlabel('Position in X (m)');ylabel('Position in Y (m)');
daspect([1 1 1]);

% The plots for Hoops 2 and 4 are inconclusive because the quadrotor
 does
```

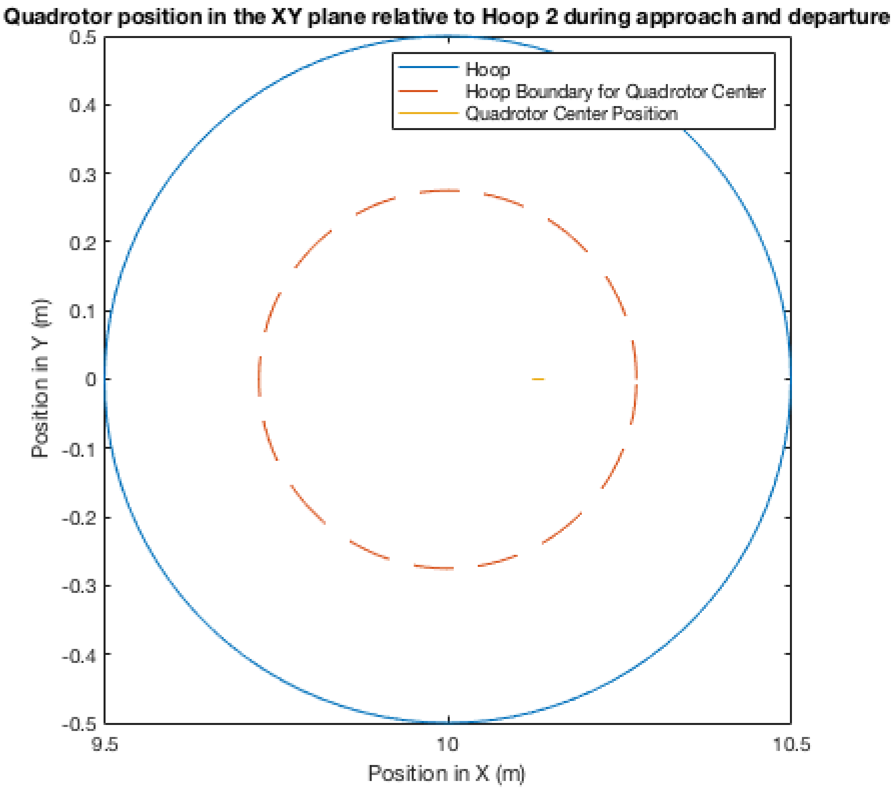
```
% not extend as far above and bellow its center as it does to each
 side.
% The assumption that the quadrotor is contained within a bouding
 sphere is
% a bad assumttion when considering the bottom and top of the
 quadrotor.
% Because of this, the quadrotor can cut the turns around the hoops
% oriented normal to the world Z axis slightly closer. I will
 replicate the
% plots with the same size bounding sphere, but only consider when
 center
% of the quadrotor is .505 bellow or above the hoop
```



Hoop 2 in XY Plane (Relaxed Boundary)

```
th = 0:pi/50:2*pi;
xHoop = hoopRadius * cos(th) + hoop2Pos(1);
yHoop = hoopRadius * sin(th) + hoop2Pos(2);
xBoundary= hoopBoundaryRadius * cos(th) + hoop2Pos(1);
yBoundary = hoopBoundaryRadius * sin(th) + hoop2Pos(2);

i = find((hoop2Pos(3)-.00505)<droneZ.Data &
 droneZ.Data<(hoop2Pos(3)+.00505) ...
        & (hoop2Time-.5)<droneZ.Time & droneZ.Time<(hoop2Time
+.5)); %only consider times around hoop 2
xH2 = droneX.Data(i(1):i(end));
```

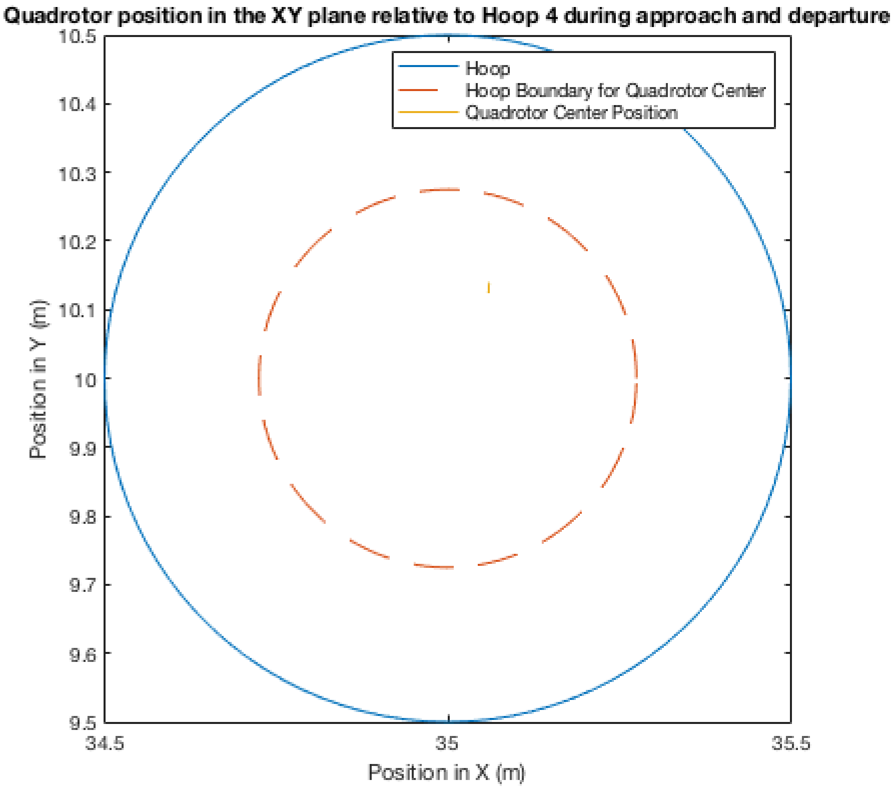
```
yH2 = droneY.Data(i(1):i(end));
zH2 = droneZ.Data(i(1):i(end));

figure(6);
plot(xHoop, yHoop, xBoundary, yBoundary,'--',xH2,yH2);
title('Quadrotor position in the XY plane relative to Hoop 2 during
 approach and departure')
legend('Hoop', 'Hoop Boundary for Quadrotor Center','Quadrotor Center
 Position');
xlabel('Position in X (m)');ylabel('Position in Y (m)');
daspect([1 1 1]);
```



Hoop 4 in XY Plane (Relaxed Boundary)

```
th = 0:pi/50:2*pi;
xHoop = hoopRadius * cos(th) + hoop4Pos(1);
yHoop = hoopRadius * sin(th) + hoop4Pos(2);
xBoundary= hoopBoundaryRadius * cos(th) + hoop4Pos(1);
yBoundary = hoopBoundaryRadius * sin(th) + hoop4Pos(2);

i = find((hoop4Pos(3)-.00505)<droneZ.Data &
 droneZ.Data<(hoop4Pos(3)+.00505));
xH4 = droneX.Data(i(1):i(end));
yH4 = droneY.Data(i(1):i(end));
zH4 = droneZ.Data(i(1):i(end));
```

```
figure(8);
plot(xHoop, yHoop, xBoundary, yBoundary,'--',xH4,yH4);
title('Quadrotor position in the XY plane relative to Hoop 4 during
 approach and departure')
legend('Hoop', 'Hoop Boundary for Quadrotor Center','Quadrotor Center
 Position');
xlabel('Position in X (m)');ylabel('Position in Y (m)');
daspect([1 1 1]);

% Visual Checks will be included in the report to supplement this
 analysis
```



Published with MATLAB® R2018b