Assignment 3

Shounak Chakraborty

1. We can see from the plots given that the data approximately follows a Gaussian distribution centered about zero. ∴ the zero-mean Gaussian seems reasonable.

For the motion model,

Using the translational and rotational speed variance :-

$$w_k \sim N(0, Q_k)$$

$$Q_k = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.00442026 & 0 \\ 0 & 0.00818609 \end{bmatrix}$$

For the observation model,

Using the range and bearing measurement variance :-

$$n_k^l \sim N(0, R_k^l)$$

$$R_k^l = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.00090036 & 0 \\ 0 & 0.0006714 \end{bmatrix}$$

2. In case of motion model, inorder to find the Jacobians, we need to compute F_{k-1} and w'_k

$$F_{k-1} = \left.\frac{\partial f(x_{k-1}, v_k, w_k)}{\partial x_{k-1}}\right|_{\hat{x}_{k-1}, v_k, 0}$$

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + T \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + w'_k \right)$$

$$F_{k-1} = \begin{bmatrix} \frac{\partial x_k}{\partial x_{k-1}} & \frac{\partial x_k}{\partial y_{k-1}} & \frac{\partial x_k}{\partial \theta_{k-1}} \\ \frac{\partial y_k}{\partial x_{k-1}} & \frac{\partial y_k}{\partial y_{k-1}} & \frac{\partial y_k}{\partial \theta_{k-1}} \\ \frac{\partial \theta_k}{\partial x_{k-1}} & \frac{\partial \theta_k}{\partial y_{k-1}} & \frac{\partial \theta_k}{\partial \theta_{k-1}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \frac{\partial}{\partial \theta_{k-1}}\left(x_{k-1} + T v_k \cos\theta_{k-1}\right) \\ 0 & 1 & \frac{\partial}{\partial \theta_{k-1}}\left(y_{k-1} + T v_k \sin\theta_{k-1}\right) \\ 0 & 0 & \frac{\partial}{\partial \theta_{k-1}}\left(\theta_{k-1} + T \omega_k\right) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -T v_k \sin\theta_{k-1} \\ 0 & 1 & T v_k \cos\theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix}$$

$$w'_k = \frac{\partial f(x_{k-1}, v_k, w_k)}{\partial w_k}\Bigg|_{\hat{x}_{k-1}, v_k, 0} \cdot w_k$$

$$w'_k = \frac{\partial}{\partial w_k}\left(\cancel{\begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix}}^{\to 0} + \frac{\partial}{\partial w_k} T \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} w_k \right) \cdot w_k$$

$$= T \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} w_k$$

In case of observation model, in order to find the Jacobians, we need to compute G_k and n'_k.

$$G_k = \frac{\partial g(x_k, n_k)}{\partial x_k}\Bigg|_{\check{x}_k, 0}$$

$$\begin{bmatrix} r_k^l \\ \phi_k^l \end{bmatrix} = \begin{bmatrix} \sqrt{(x_l - x_k - d\cos\theta_k)^2 + (y_l - y_k - d\sin\theta_k)^2} \\ \operatorname{atan2}(y_l - y_k - d\sin\theta_k, x_l - x_k - d\cos\theta) - \theta_k \end{bmatrix} + n_k^l$$

for the 1st term

$\therefore$ We obtain (by applying the chain rule):

let $q = \delta_x^T \delta_x$

$$\therefore \frac{\partial \sqrt{q}}{\partial x} = \frac{1}{\cancel{2}} \frac{1}{\sqrt{q}} \cancel{2}\delta_x(-1) = \frac{-\delta x}{\sqrt{q}}$$

Using the above concept,

$$\frac{\partial r_k^l}{\partial x_k} = -\frac{X}{r_k^l} \quad ; \quad \frac{\partial r_k^l}{\partial y_k} = -\frac{Y}{r_k^l}$$

where; $X = x_L - x_k - d\cos\theta$

$Y = y_L - y_k - d\sin\theta$

$$\frac{\partial r_k^L}{\partial \theta_k} = \frac{1}{2\sqrt{X^2+Y^2}} \frac{\partial}{\partial \theta_k}\left(X^2+Y^2\right)$$

$$= \frac{1}{2r_k^L} \frac{\partial}{\partial \theta_k}\Big(\cancel{x_L^2} + \cancel{x_k^2} + d^2\cos^2\theta_k - \cancel{2x_Lx_k} - 2x_Ld\cos\theta_k + 2x_kd\cos\theta_k + \cancel{y_L^2} + \cancel{y_k^2} + d^2\sin^2\theta_k - \cancel{2y_Ly_k} - 2y_Ld\sin\theta_k + 2y_kd\cos\theta_k \Big)$$

$$= \frac{1}{\cancel{2}r_k^L}\Big(\cancel{2}x_Ld\sin\theta_k - \cancel{2}x_kd\sin\theta_k - \cancel{2}d\cos\theta_k\sin\theta_k - \cancel{2}y_Ld\cos\theta_k + \cancel{2}y_kd\cos\theta_k + \cancel{2}d\sin\theta_k\cos\theta_k\Big)$$

$$= \frac{d}{r_k^L}\Big(\underbrace{\cancel{(x_L - x_k)}}_{X}\sin\theta_k - \underbrace{\cancel{(y_L - y_k)}}_{Y}\cos\theta_k\Big).$$

for the second row,

Using $\frac{\partial}{\partial x}\arctan f(x) = \frac{f'(x)}{1+f^2(x)}$

$$\frac{\partial \phi_k^l}{\partial x_k} = \frac{1}{1+\left(\frac{Y}{X}\right)^2}\cdot\frac{Y}{X^2} = \frac{Y}{X^2+Y^2} = \frac{Y}{(r_k^L)^2}$$

$$\frac{\partial \phi_k^l}{\partial y_k} = \frac{1}{1+\left(\frac{Y}{X}\right)^2}\cdot\frac{(-1)}{X} = \frac{-X}{X^2+Y^2} = \frac{-X}{(r_k^L)^2}$$

$$\frac{\partial \phi_k^L}{\partial \theta_k} = \frac{1}{1+\left(\frac{y}{x}\right)^2} \cdot (-d) \left(\frac{\cos\theta_k}{x} + \frac{y}{x^2} \sin\theta_k \right) - 1$$

$$= -d \, \frac{x\cos\theta_k + y\sin\theta_k}{x^2+y^2} - 1$$

$$= \cancel{d} \; -d \, \frac{x\cos\theta_k + y\sin\theta_k}{(r_k^L)^2} - 1$$

$$n_k' = \left.\frac{\partial g(x_k, n_k)}{\partial n_k}\right|_{\tilde{x}_k, 0} \cdot n_k$$

$$= \cdot I$$

3. Original equations of EKF ;

Predictor :- $\check{P} = F_{k-1} \hat{P}_{k-1} F_{k-1}^T + Q_k'$

state-check $\check{X} = f(\hat{x}_{k-1}, v_k, 0)$

Now, for each observation measurement, we need to calculate the Jacobian of the observation model G_k^L, and the innovation $(y_k^n - g(\check{x}, 0))$.

Original Equation of Kalman Gain & the corrector ;

Kalman Gain :- $K_k = \check{P}_k G_k^T (G_k \check{P}_k G_k^T + R_k')^{-1}$

state-hat $\hat{P}_k = (1 - K_k G_k)\check{P}_k$

Corrector :- $\hat{X}_k = \check{X}_k + K_k (y_k - g(\check{x}, 0))$.

$$\therefore G_k = \begin{bmatrix} G_k^1 \\ G_k^2 \\ \vdots \\ G_k^L \end{bmatrix} \qquad Y_k - g(\check{x}, 0) = \begin{bmatrix} y_k^1 - g(\check{x}, 0) \\ y_k^2 - g(\check{x}, 0) \\ \vdots \\ y_k^L - g(\check{x}, 0) \end{bmatrix}$$

and the measurement noise ; $R_k' = \begin{bmatrix} R_k^{1'} & 0 & \cdots & 0 \\ \vdots & R_k^{2'} & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & R_k^{L'} \end{bmatrix}$

```
clc
close all
clear
warning off;


load dataset2
who
T= 0.1
K=12609;
r_max = 1;

s_hat = zeros(K,3); %state
x_hat = zeros(K,1);
y_hat = zeros(K,1);
th_hat = zeros(K,3);
P_hat = zeros(3,3,K);


s_hat(1,:) = [x_true(1) y_true(1) th_true(1)].';
x_hat(1) = s_hat(1,1);
y_hat(1) = s_hat(1,2);
th_hat(1) = s_hat(1,3);
P_hat(:,:,1) = diag([1, 1, 0.1]);

for k = 2:1:K

    %Motion Model

    %Jacobians, F_k-1 and Q_k
    F_k_1 = [1 0 -T*v(k)*sin(th_hat(k-1));
         0 1 T*v(k)*cos(th_hat(k-1));
         0 0 1];

    %Process noise matrix
    Q_k = [v_var*(T*cos(th_hat(k-1)))^2, v_var*((T^2)*cos(th_hat(k-
1))*sin(th_hat(k-1))), 0;
         v_var*((T^2)*cos(th_hat(k-1))*sin(th_hat(k-1))),
v_var*(T*sin(th_hat(k-1)))^2, 0;
         0, 0, om_var^2];

    %Predictor
    P_check = F_k_1*P_hat(:,:,k-1)*F_k_1.' + Q_k;
    x_check = x_hat(k-1) + T*v(k)*cos(th_hat(k-1));
```

```
    %x_check = x_true(k-1) + T*v(k)*cos(th_true(k-1));
    y_check = y_hat(k-1) + T*v(k)*sin(th_hat(k-1));
    %y_check = y_true(k-1) + T*v(k)*sin(th_true(k-1));
    th_check = th_hat(k-1) + T*om(k);
    %th_check = th_true(k-1) + T*om(k);

    s_check = [x_check, y_check, th_check].';

    %Observation Model

    % For each landmark we need to calculate the jacobian of the
    % observation model G_k and and the innovation matrix
    landmarks = length(l);
    R_k = zeros(2,2);
    innovation = zeros(2,1);
    G_k = zeros(2,3);
    j = 0;

    for i = 1:1:landmarks
        if r(k,i) ~=0 && r(k,i) < r_max
            j = j+1;

            dr_x = -(l(i,1)-x_check -d*cos(th_check))/(( (l(i,1)-
x_check-d*cos(th_check))^2 + (l(i,2)-y_check-
d*sin(th_check))^2 )^(1/2) );
            dr_y = -(l(i,2)-y_check -d*sin(th_check))/(( (l(i,1)-
x_check-d*cos(th_check))^2 + (l(i,2)-y_check-
d*sin(th_check))^2 )^(1/2) );
            dr_th = d*( sin(th_check)*(l(i,1)-x_check -d*cos(th_check))
- cos(th_check)*(l(i,2)-y_check -d*sin(th_check)))/(( (l(i,1)-x_check-
d*cos(th_check))^2 + (l(i,2)-y_check-d*sin(th_check))^2 )^(1/2) );

            dphi_x = (l(i,2)-y_check -d*sin(th_check))/ ( (l(i,1)-
x_check-d*cos(th_check))^2 + (l(i,2)-y_check-d*sin(th_check))^2 );
            dphi_y = -(l(i,1)-x_check -d*cos(th_check))/( (l(i,1)-
x_check-d*cos(th_check))^2 + (l(i,2)-y_check-d*sin(th_check))^2 );
            dphi_th = -d*( cos(th_check)*(l(i,1)-x_check -
d*cos(th_check)) + sin(th_check)*(l(i,2)-y_check
-d*sin(th_check)))/(( (l(i,1)-x_check-d*cos(th_check))^2 + (l(i,2)-
y_check-d*sin(th_check))^2 )^(1/2) ) - 1;

            G_k(2*j-1 :2*j, 1:3) = [dr_x  dr_y dr_th;
                                    dphi_x dphi_y dphi_th];

            %measurement noise
```

```
            R_k(2*j-1 :2*j, 2*j-1 :2*j) = diag([r_var;
                                                v_var]);

            %innovation -- given(range and phi data) - calculated(range
and phi data)
            r_l = ( (l(i,1)-x_check-d*cos(th_check))^2 + (l(i,2)-
y_check-d*sin(th_check))^2 )^(1/2);
            phi_l = atan2((l(i,2) - y_check - d*sin(th_check)),(l(i,1)-
x_check-d*cos(th_check))) - th_check;

            if phi_l > pi
                phi_l = phi_l-2*pi;

            elseif phi_l < pi
                phi_l = phi_l +2*pi;
            end

            innovation(2*j-1: 2*j, 1) = [r(k,i), b(k,1)] - [r_l, phi_l];
        end
    end

    %Kalman Gain
    % K_G = P_check*G_k.'/(G_k*P_check*G_k.' + R_k)
    % 3xn     3x3   3xn    nx3  3x3    3xn     nxn
    K_G = P_check*G_k.'/(G_k*P_check*G_k.' + R_k);


    %Corrector
    if j == 0
        P_hat(:,:,k) = P_check;
        s_hat(k,:) = s_check;
        x_hat(k) = s_hat(k,1);
        y_hat(k) = s_hat(k,2);
        th_hat(k) = s_hat(k,3);
    else
        P_hat(:,:,k) = (eye(3) - K_G*G_k)*P_check;
        s_hat(k,:) = s_check + K_G*innovation;
        x_hat(k) = s_hat(k,1);
        y_hat(k) = s_hat(k,2);
        th_hat(k) = s_hat(k,3);
    end

    th_hat(k) = wrapTo2Pi(th_hat(k));
end
%Covariance matrix
```

```
for m = 1:1:length(x_true)
    x_var(m) = P_hat(1,1,m);
    y_var(m) = P_hat(2,2,m);
    th_var(m) = P_hat(3,3,m);
end


% a. Error Plots
figure(1)
plot(t, x_hat-x_true)
hold on
plot(t, 3*sqrt(x_var), '--')
hold on
plot(t, -3*sqrt(x_var), '--')
xlabel("Time (s)")
ylabel("Error_x (m)")
legend("Error", "3\sigma Uncertainity Envelope", "-3\sigma Uncertainty
Envelope")'

hold off


          % a. Error Plots
figure(1)
plot(t, y_hat-y_true)
hold on
plot(t, 3*sqrt(y_var), '--')
hold on
plot(t, -3*sqrt(y_var), '--')

xlabel("Time (s)")
ylabel("Error_y (m)")
legend("Error", "3\sigma Uncertainity Envelope", "-3\sigma Uncertainty
Envelope")'

hold off
```



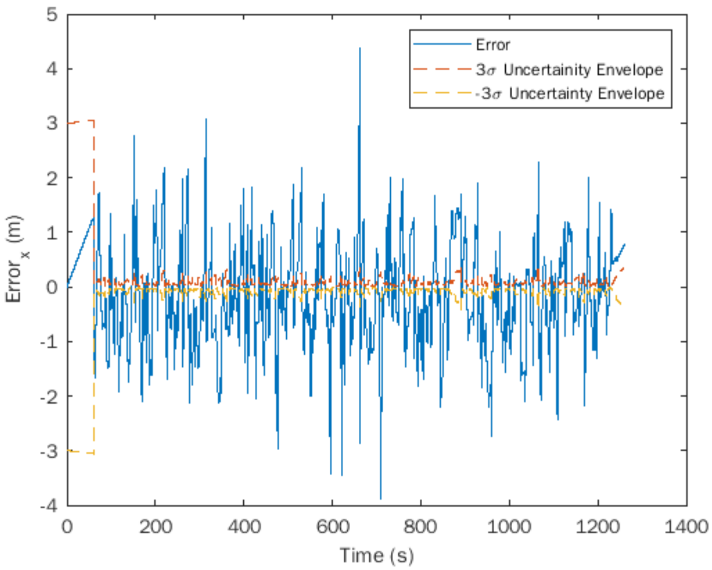
```
% a. Error Plots
figure(1)
plot(t, wrapToPi(th_hat-th_true))
hold on
```

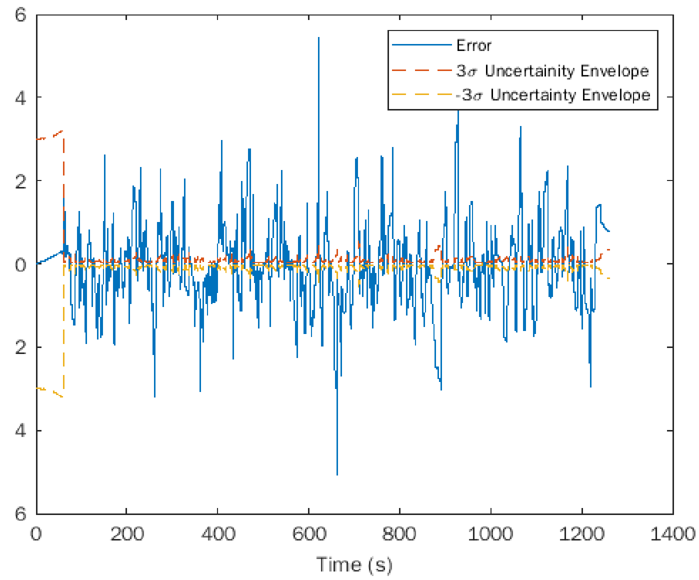
```
plot(t, 3*sqrt(th_var), '--')
hold on
plot(t, -3*sqrt(th_var), '--')

xlabel("Time (s)")
ylabel("Error_theta (rad)")
legend("Error", "3\sigma Uncertainity Envelope", "-3\sigma Uncertainty
Envelope")'

hold off
```

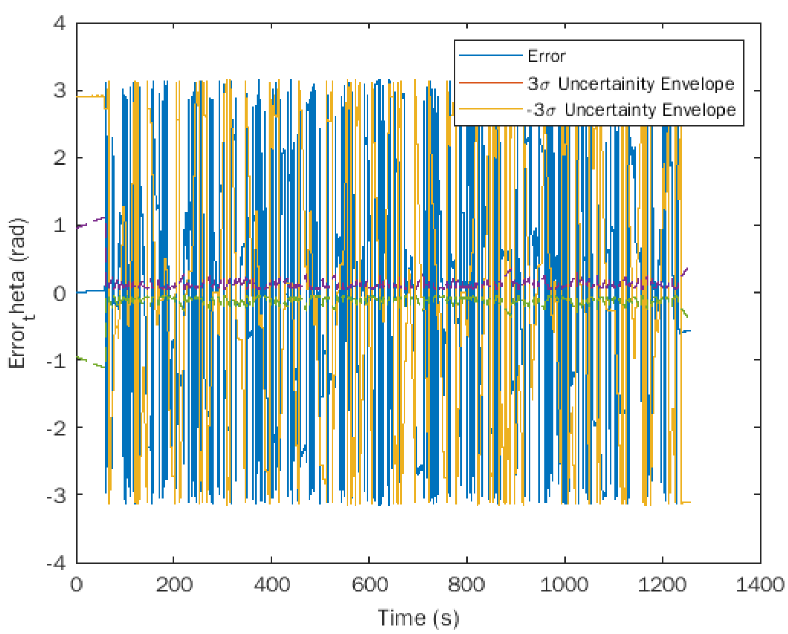
Assignment 3

Question 4.

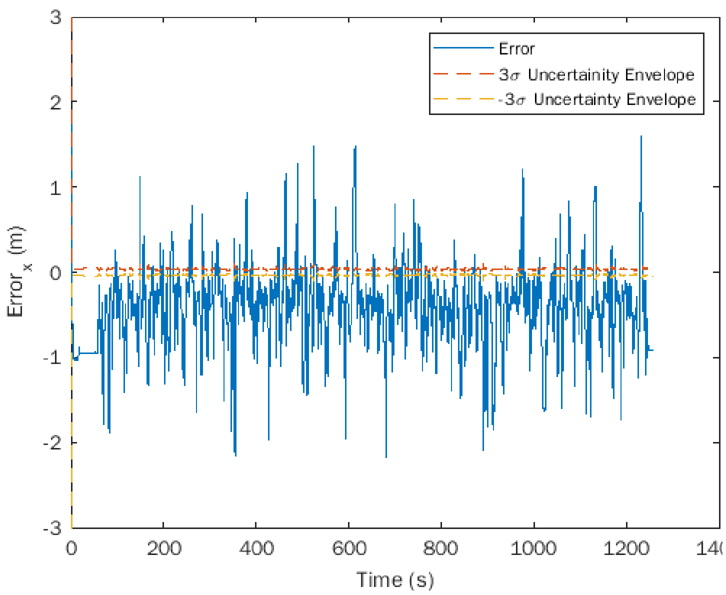
a. For r_max =1



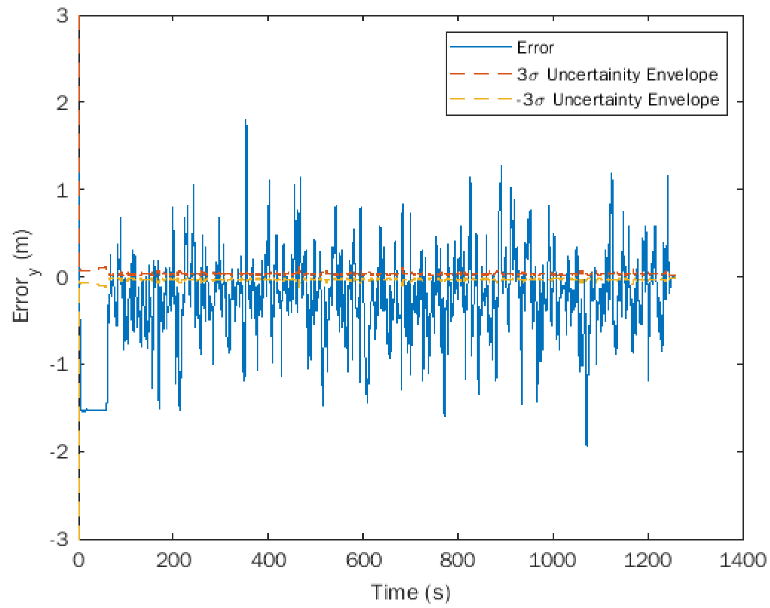




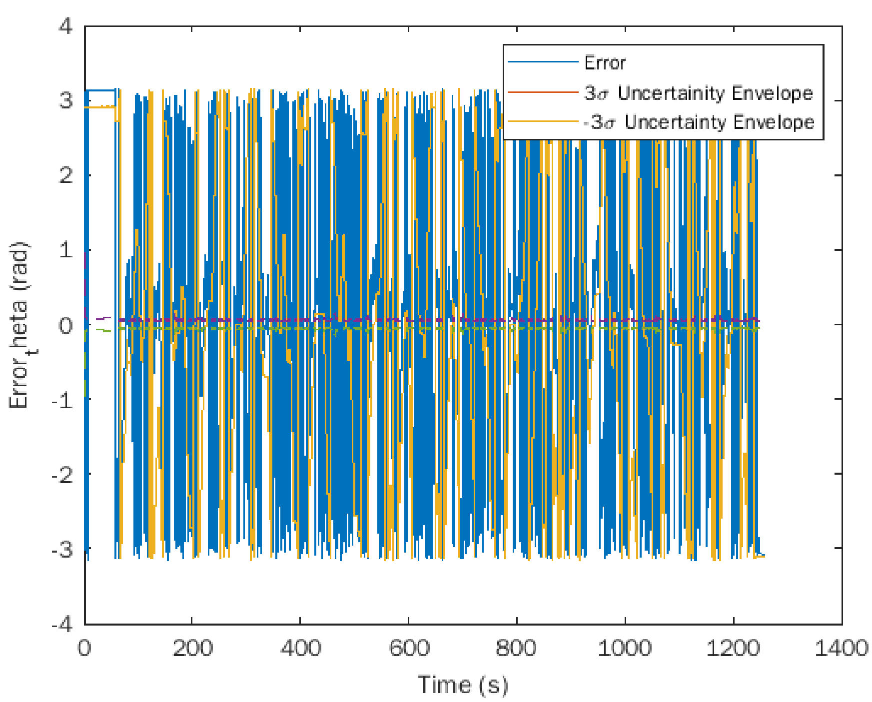
For r_max = 3:



Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_x (m)
Time (s)

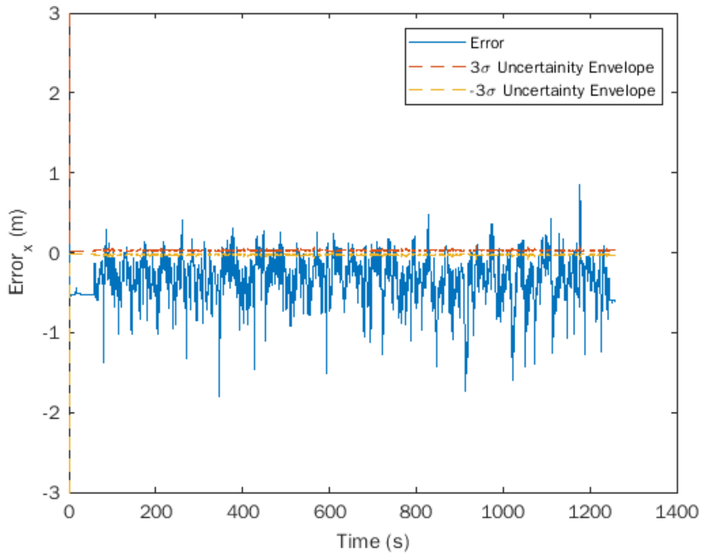


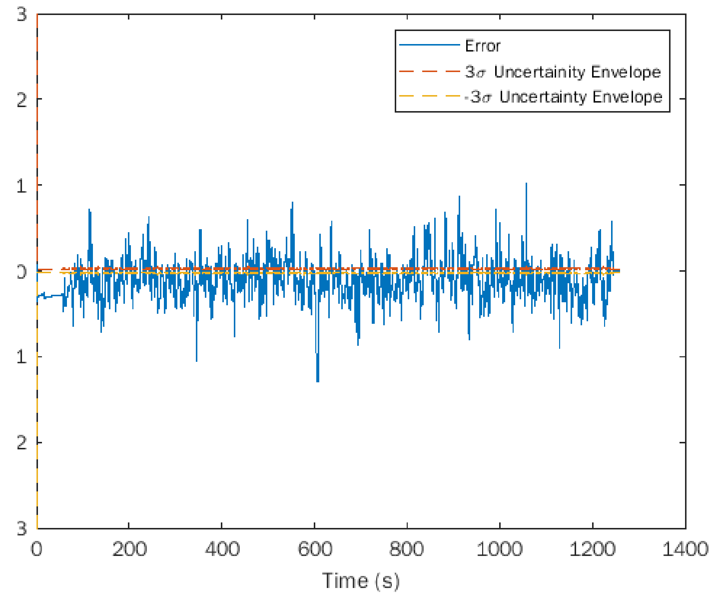
Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_y (m)
Time (s)

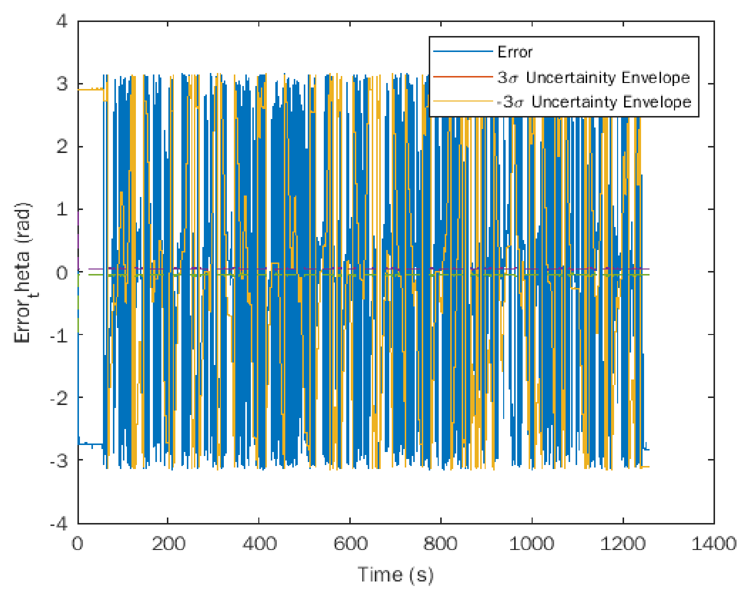


Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_theta (rad)
Time (s)

For r_max = 5:

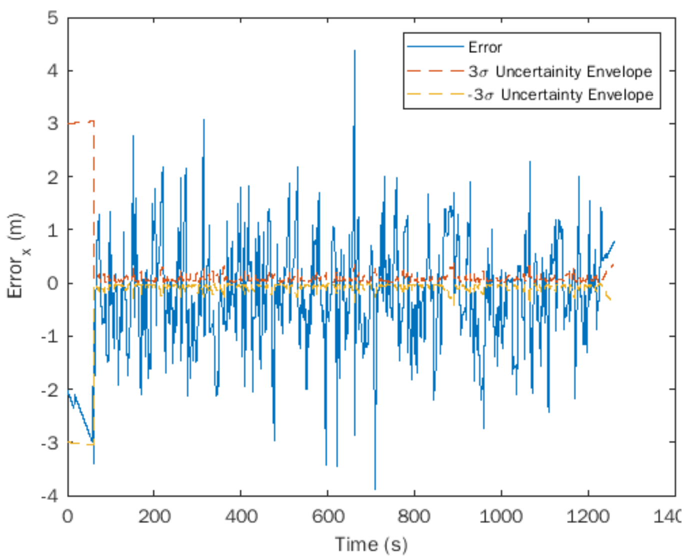

Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_x (m)
Time (s)

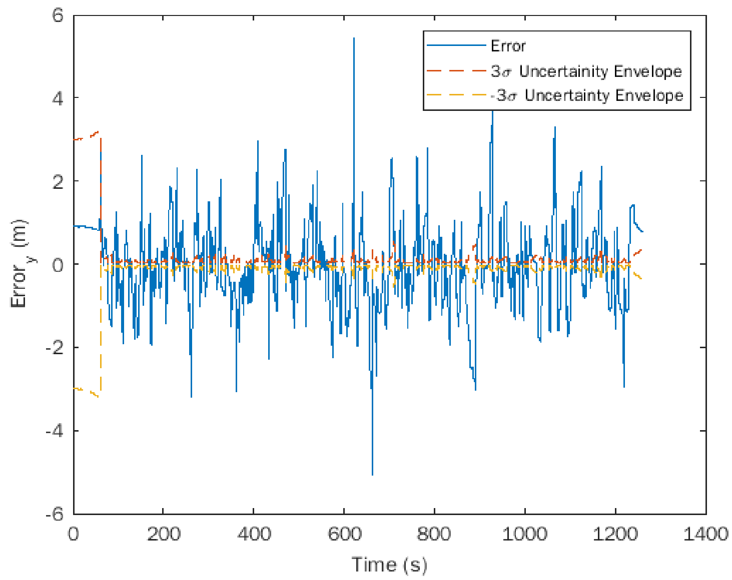

Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Time (s)

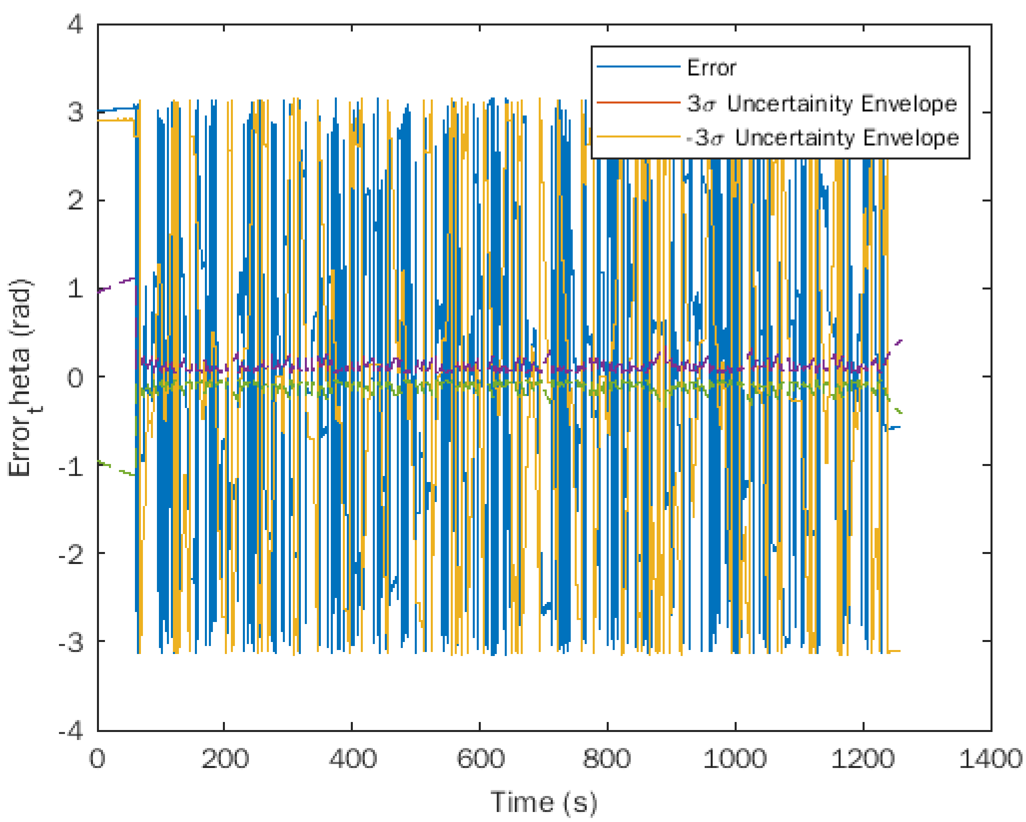

Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_theta (rad)
Time (s)

b. With ˆx0 = (1, 1, 0.1)

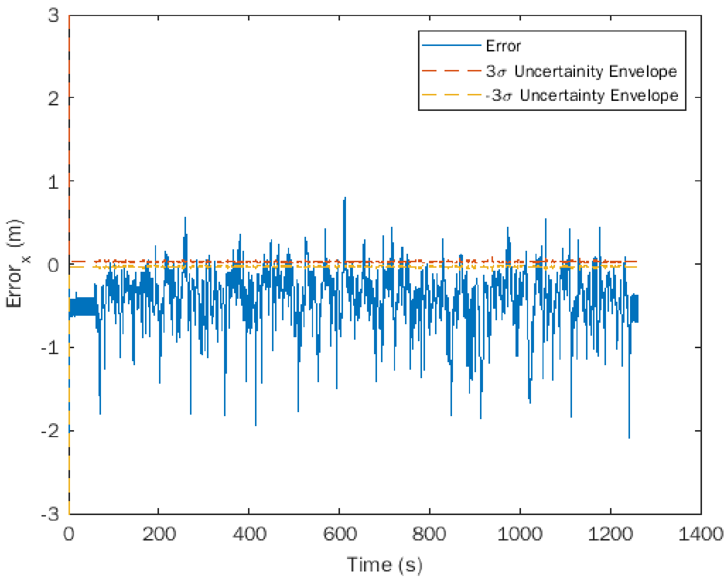
For r_max:1



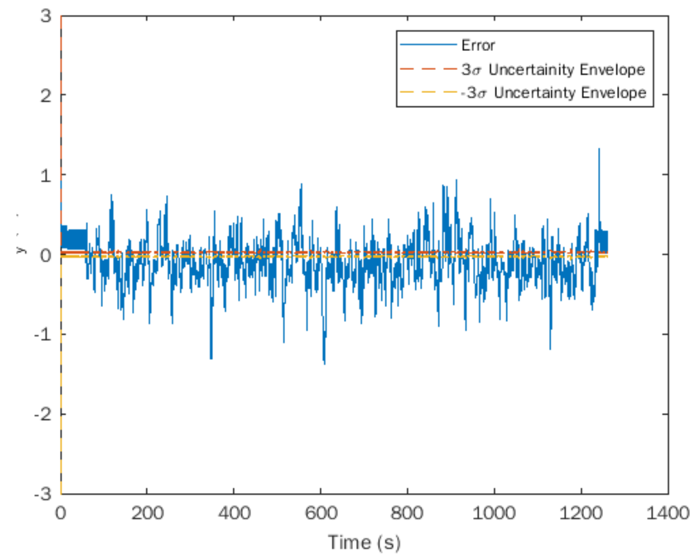




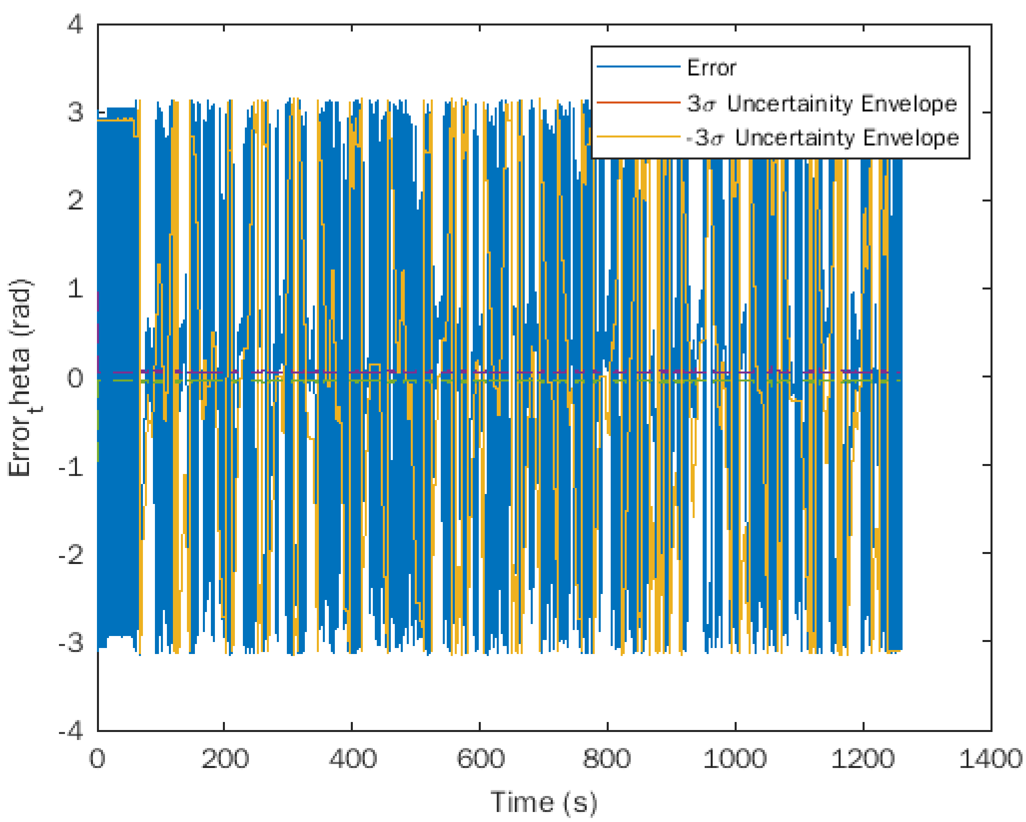
For r_max :3



Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Error_x (m)
Time (s)

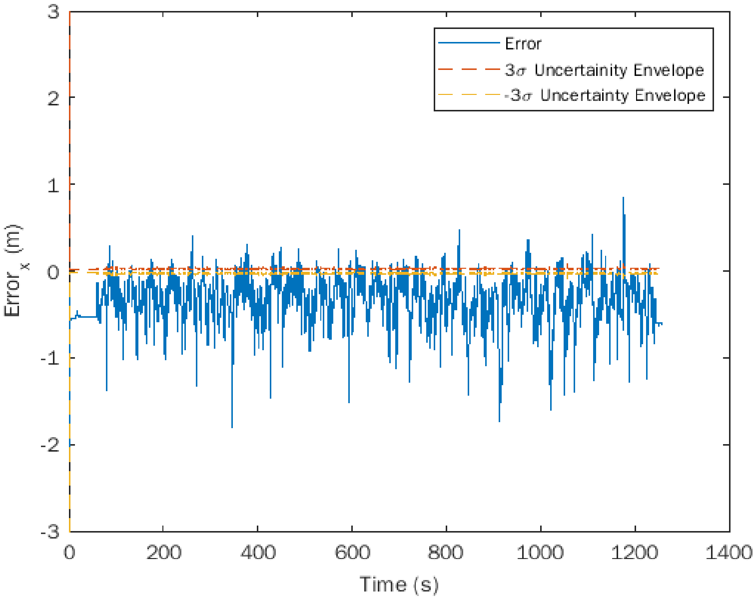


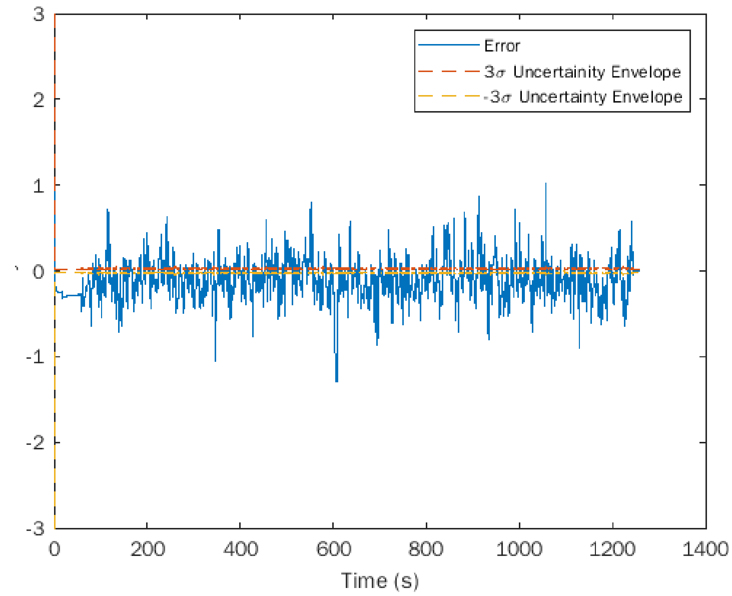
Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Time (s)

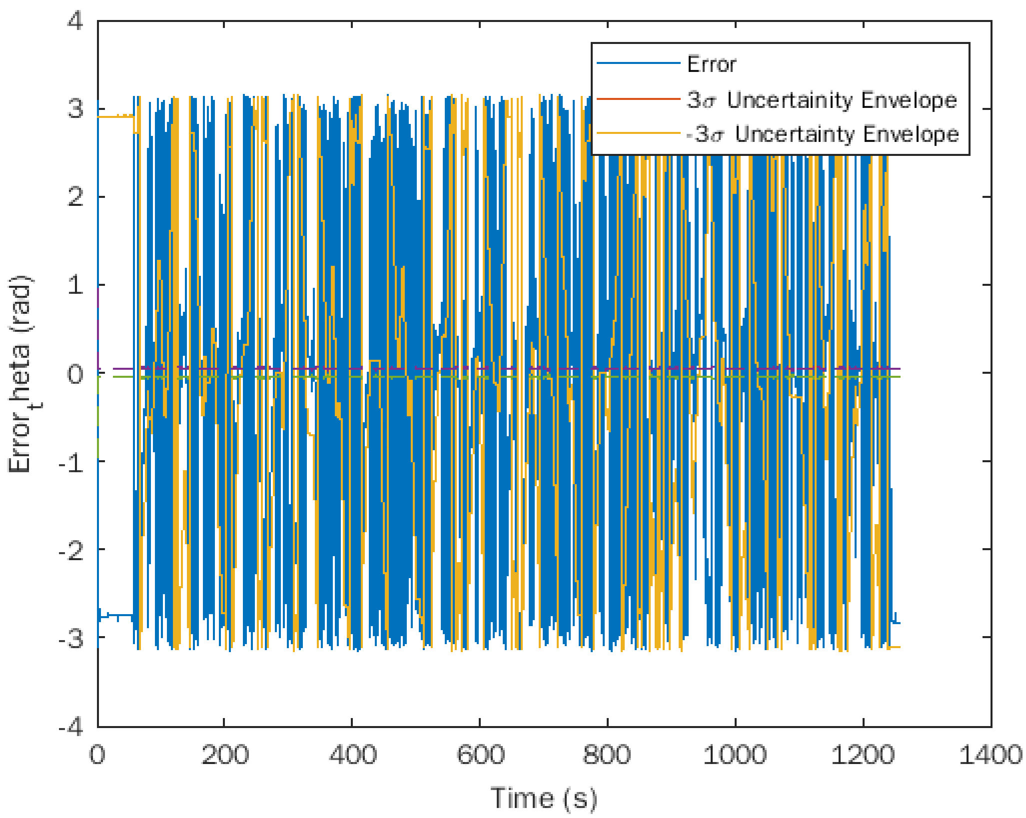


Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_theta (rad)
Time (s)

For r_max:5

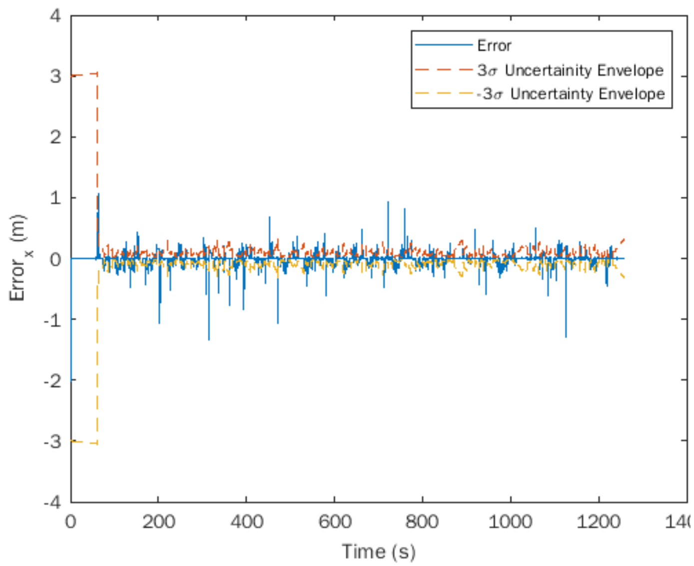

Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Error_x (m)
Time (s)

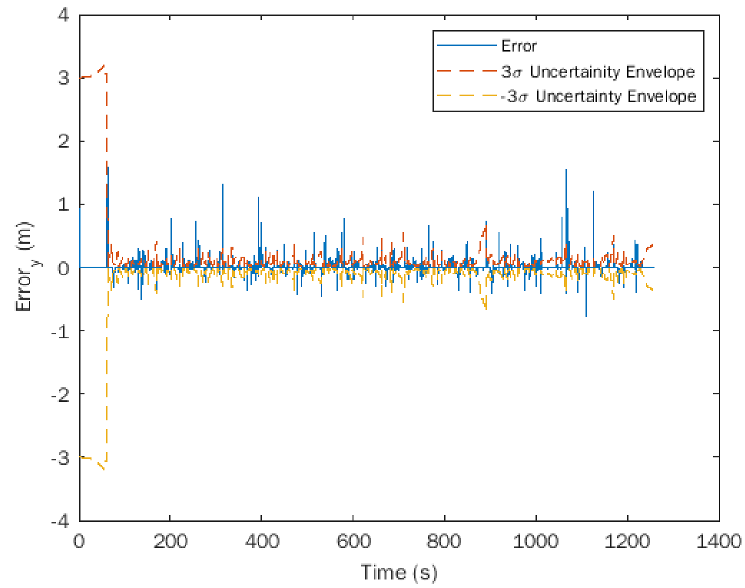

Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Time (s)

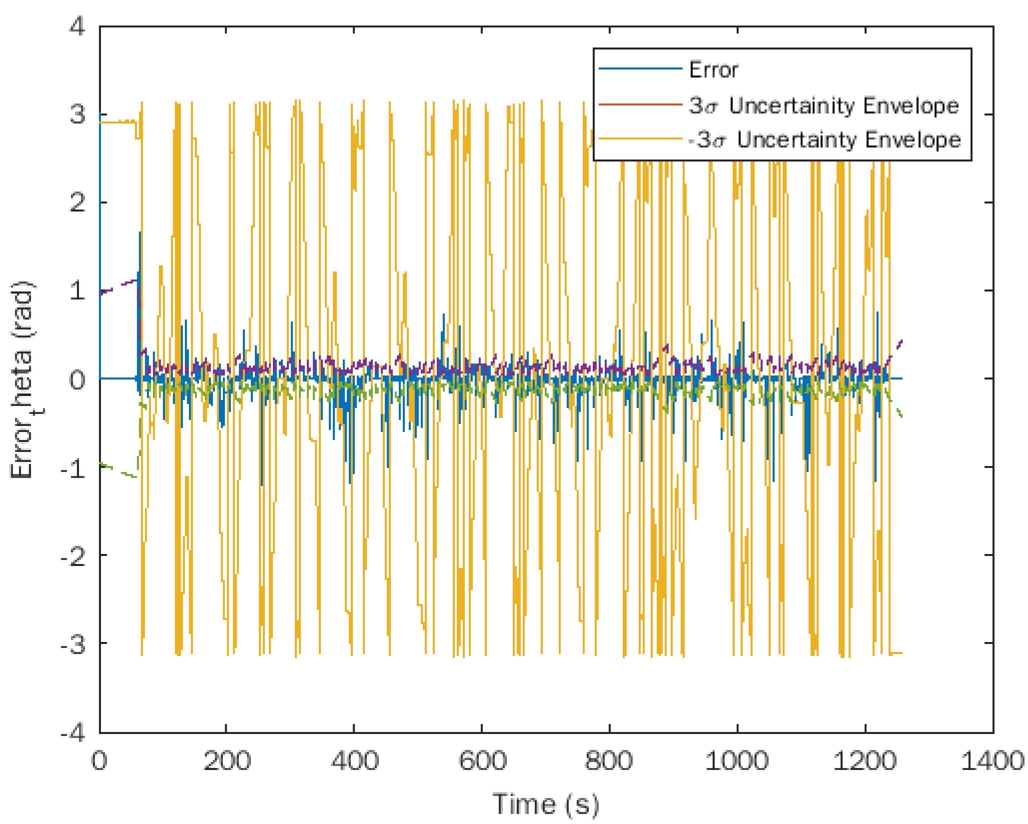

Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Error_theta (rad)
Time (s)

c. Evaluate all the Jacobians at the true robot state, xk, instead of the estimated robot state, ˆxk.

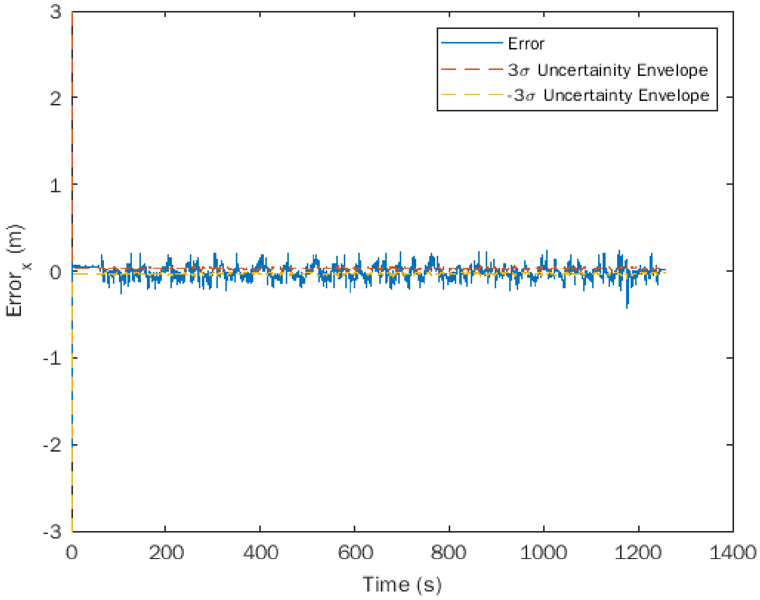
For r_max:1

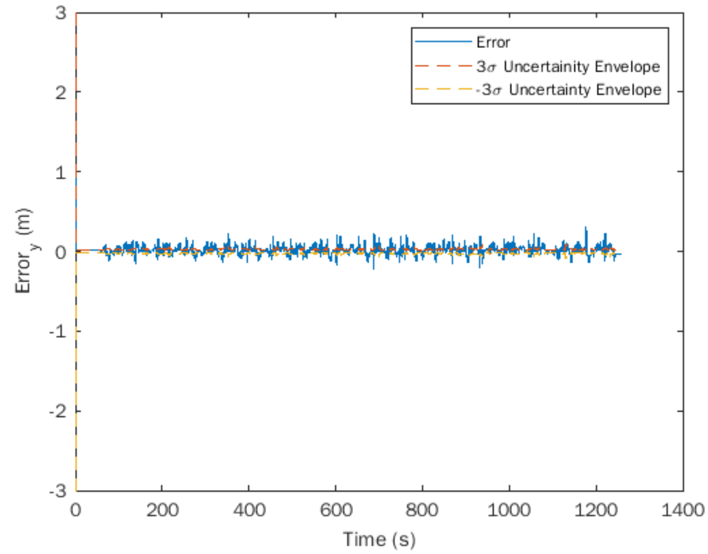


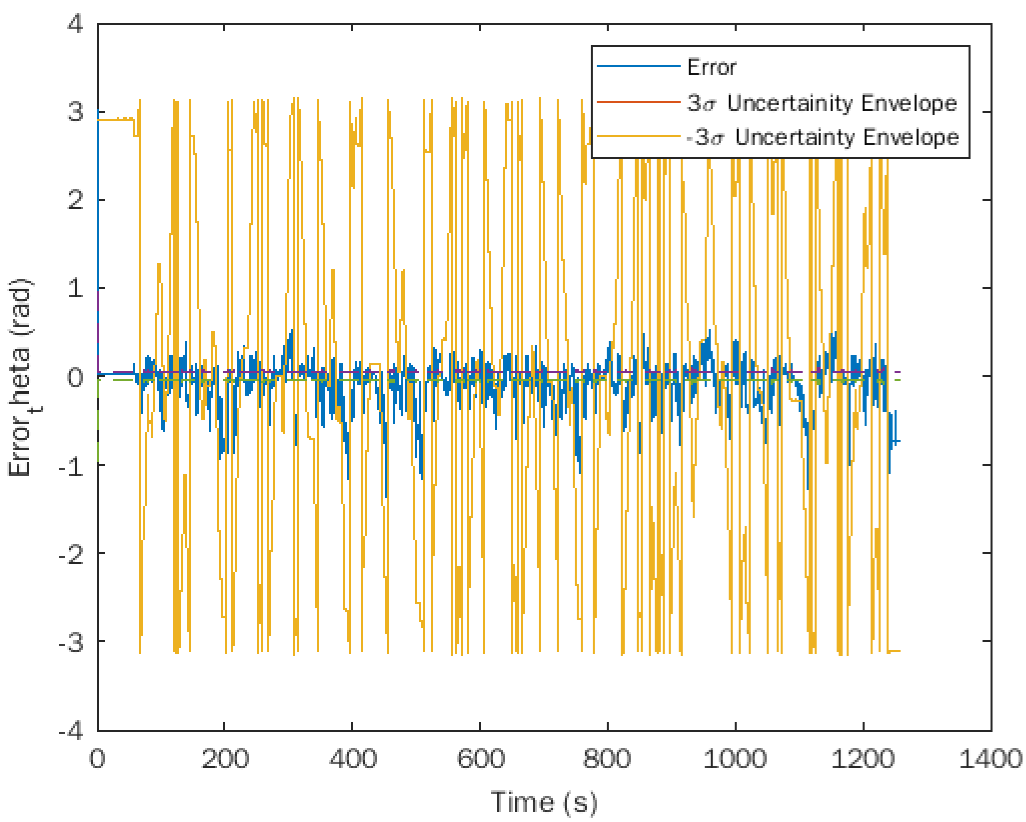




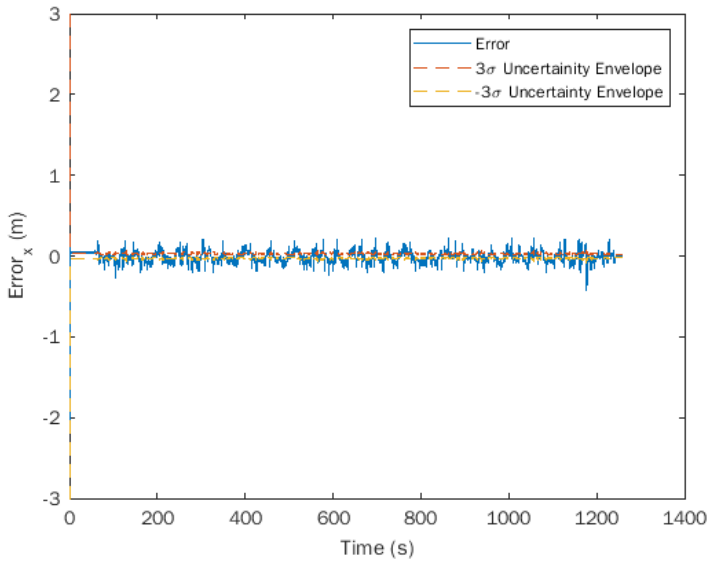
For r_max:3


Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Error_x (m)
Time (s)

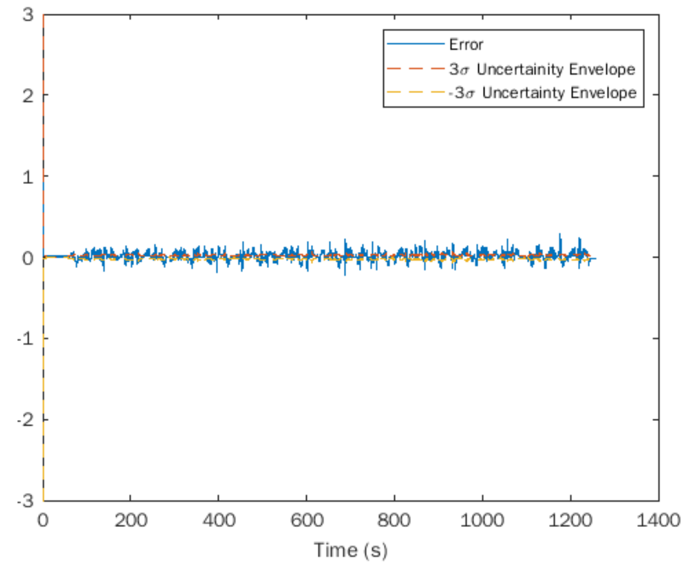

Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Error_y (m)
Time (s)


Error
3σ Uncertainity Envelope
-3σ Uncertainity Envelope
Error_theta (rad)
Time (s)

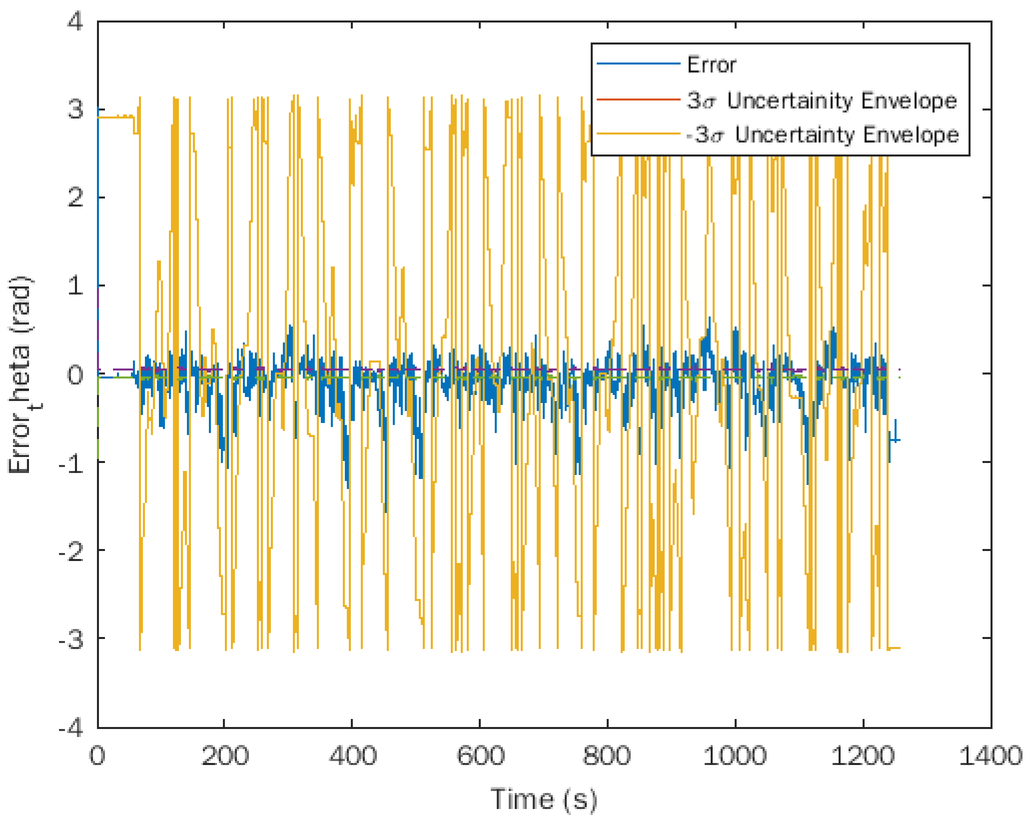
For r_max : 5



Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_x (m)
Time (s)



Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Time (s)



Error
3σ Uncertainity Envelope
-3σ Uncertainty Envelope
Error_theta (rad)
Time (s)