AER1513 Assignment 1 Report

Kevin Hu

October 11, 2023

1 Introduction

For this assignment we use batch linear-Gaussian algorithm to estimate a robot's position based on using both the odometry and laser measurements. Based on the histograms of the laser rangefinder's range error and the wheel odometer's velocity error, we can see that the errors are roughly Gaussian and their means are roughly centered at zero, so using the assumption of zero-mean Gaussian noise is reasonable. Assuming the Gaussians outlined in red are the best fit Gaussians, the variances σ_q^2 and σ_r^2 should be the square of the standard deviations so $\sigma_q^2 = 0.047554^2 = 0.002261$ and $\sigma_r^2 = 0.019155^2 = 0.000367$.

The batch linear-Gaussian objective function that we seek to minimize is:

$$J(x_{1:K}|u_{1:K}, y_{1:K}) = \sum_{k=1}^{K} \frac{1}{2\sigma_q^2}(x_k - x_{k-1} - Tu_k)^2 + \frac{1}{2\sigma_r^2}(y_k - x_k)^2 \tag{1}$$

In matrix form the objective function looks like:

$$J(\mathbf{x}|\mathbf{u}, \mathbf{y}) = \mathbf{\frac{1}{2}}(\mathbf{z} - \mathbf{Hx})^\top \mathbf{W^{-1}}(\mathbf{z} - \mathbf{Hx}) \tag{2}$$

$\mathbf{x}$ is a K+1 by 1 vector for the state that can be expressed as $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_K \end{bmatrix}$.

$\mathbf{z}$ is a 2(K+1) by 1 vector that can be expressed as $\begin{bmatrix} Tu_1 \\ Tu_2 \\ \vdots \\ Tu_K \\ y_1 \\ y_2 \vdots \\ y_K \end{bmatrix}$.

$\mathbf{H}$ is a 2K by K+1 matrix that looks like

$$\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

$\mathbf{W}$ is a 2K by 2K square matrix that looks like: $\begin{bmatrix} \sigma_q^2 & & & & & \\ & \ddots & & & & \\ & & \sigma_q^2 & & & \\ & & & \sigma_r^2 & & \\ & & & & \ddots & \\ & & & & & \sigma_r^2 \end{bmatrix}.$

Note that all the blank entries are zero.

The equation we find after taking the derivative with respect to $\mathbf{x}$ and setting it equal to zero is:

$$(\mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H})\hat{\mathbf{x}} = \mathbf{H}^\top \mathbf{W}^{-1}\mathbf{z} \tag{3}$$

where $\hat{\mathbf{x}}$ is the posterior state vector. Note that the equation is in the form of a linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A} = \mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H}$ and $\mathbf{b} = \mathbf{H}^\top \mathbf{W}^{-1}\mathbf{z}$. We can use linear least squares to solve this linear system. The linear least squares solution to the linear system $\mathbf{Ax} = \mathbf{b}$ is $\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{b}$. So an expression for the optimal position estimates derived using the method of least squares is:

$$\begin{aligned} x^*_{1:K} &= \underset{x}{\operatorname{argmin}}\, J(x_{1:K}|u_{1:K}, y_{1:K}) \\ &= ((\mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H})^\top \mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H})^{-1}(\mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H})^\top \mathbf{H}^\top \mathbf{W}^{-1}\mathbf{z} \\ &= (\mathbf{H}^\top \mathbf{W}^{-\top}\mathbf{H}\mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H})^{-1}\mathbf{H}^\top \mathbf{W}^{-\top}\mathbf{H}\mathbf{H}^\top \mathbf{W}^{-1}\mathbf{z} \end{aligned}$$

where $\mathbf{W}^{-\top} = (\mathbf{W}^{-1})^\top$.

The sparsity pattern of the left hand side of the systems of equations $\mathbf{H}^\top \mathbf{W}^{-1}\mathbf{H}$ is shown below for a 10 by 10 matrix which can be extrapolated to a matrix of size K+1 by K+1 or 12710 by 12710 where K = 12709.

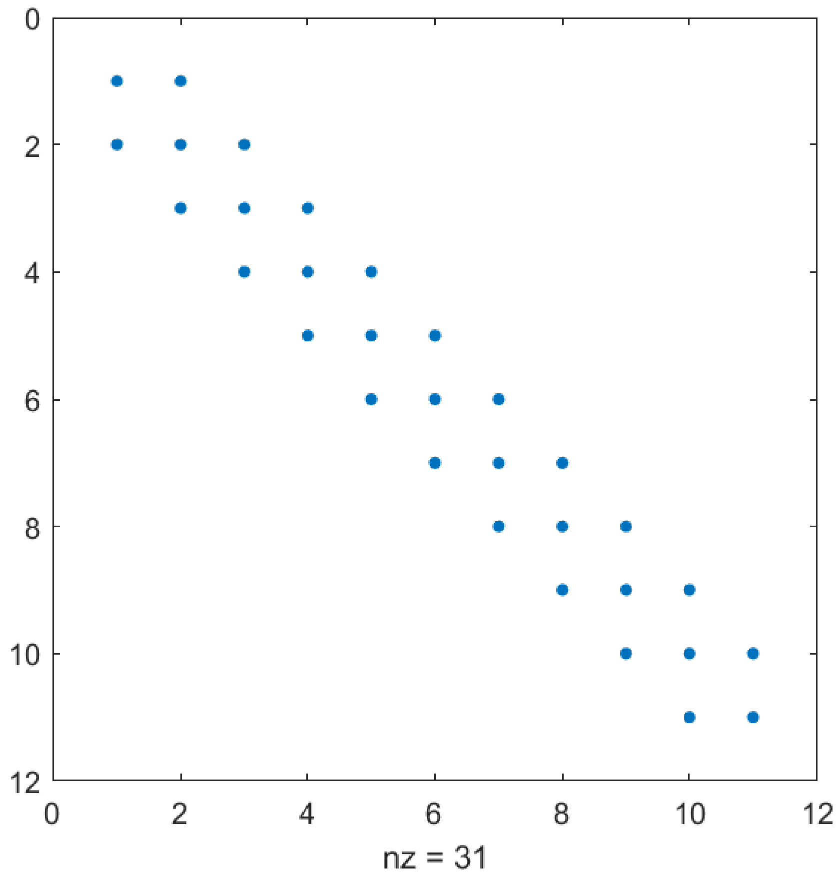

Figure 1: Matrix Sparsity Plot for K=10 or 11x11 Matrix

For this matrix sparsity plot we can see that all the entries on the main diagonal and all the entries on the two diagonals above and below the main diagonal (upper and lower diagonals) are filled. This type of matrix is called a tridiagonal matrix. Methods that can be use to solve this linear system of equations as efficiently as possible are Cholesky factorization or the Rauch-Tung-Striebel (RTS) smoother equations because they can solve the problem in $O(K)$ time instead of $O(K^3)$ time which is how long it normally takes to solve a linear system of equations.

2 Results

Here are the plots of the difference between the predicted state compared to the true state ($x_k^* - x_k$) for delta = 1, 10, 100 and 1000, where delta is the number of range readings between each range reading used over time steps t_k. The uncertainty up to 3 standard deviations above and below the mean of zero error is shown in dotted lines. Underneath is the histogram of errors of the predicted state compared to the true state ($x_k^* - x_k$) for each delta.

Delta = 1

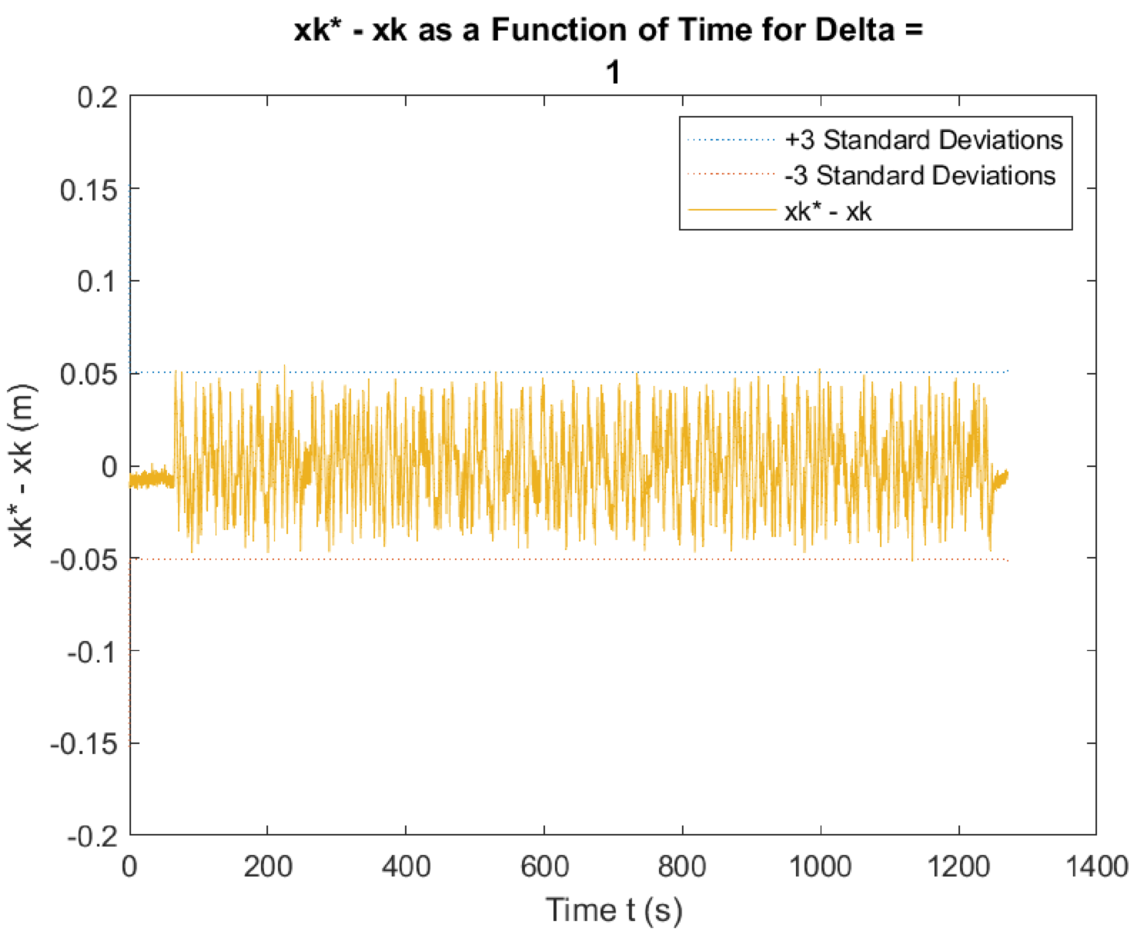

Figure 2: Difference Between Predicted State and True State Over Time for Delta = 1

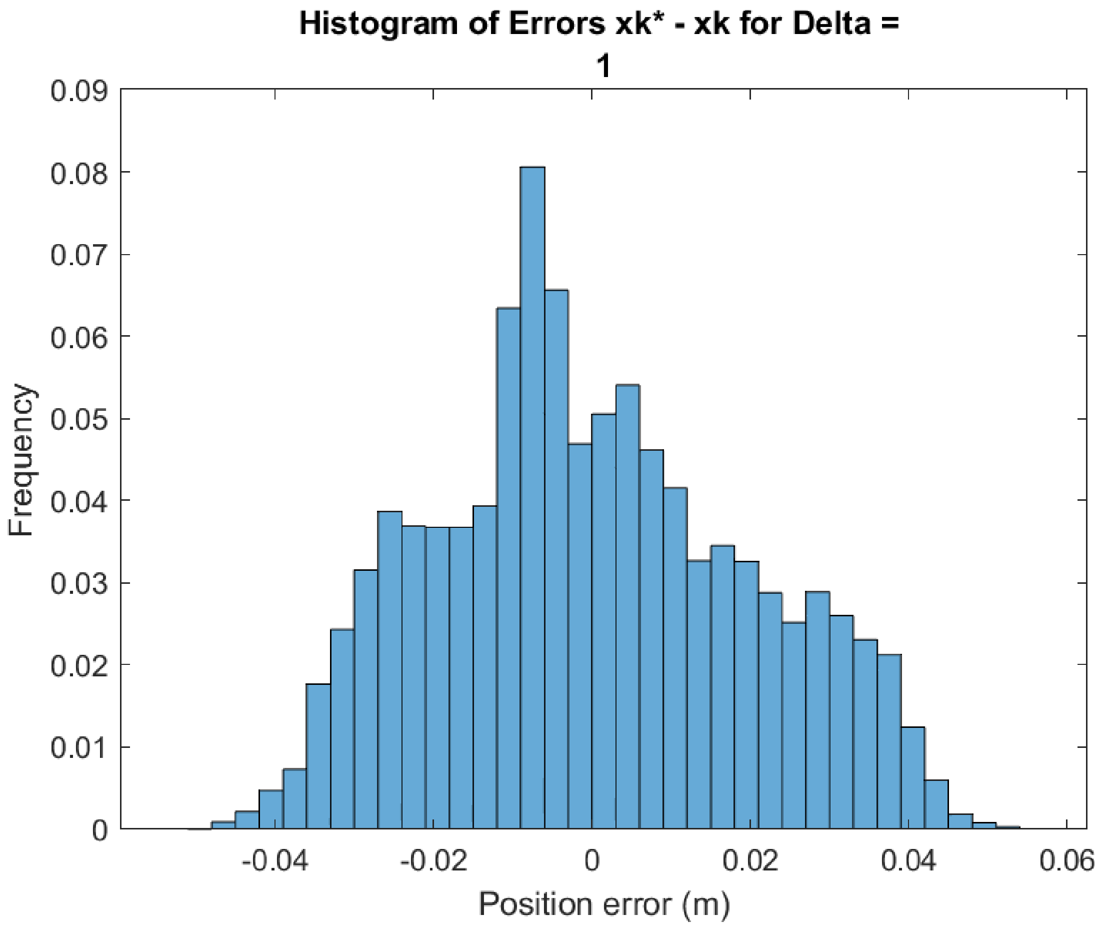

Figure 3: Histogram of Errors of the Predicted State Compared to the True State for Delta = 1

The histogram of errors and uncertainty envelope corresponds very well to the uncertainty given by the least squares formulation.

Delta = 10

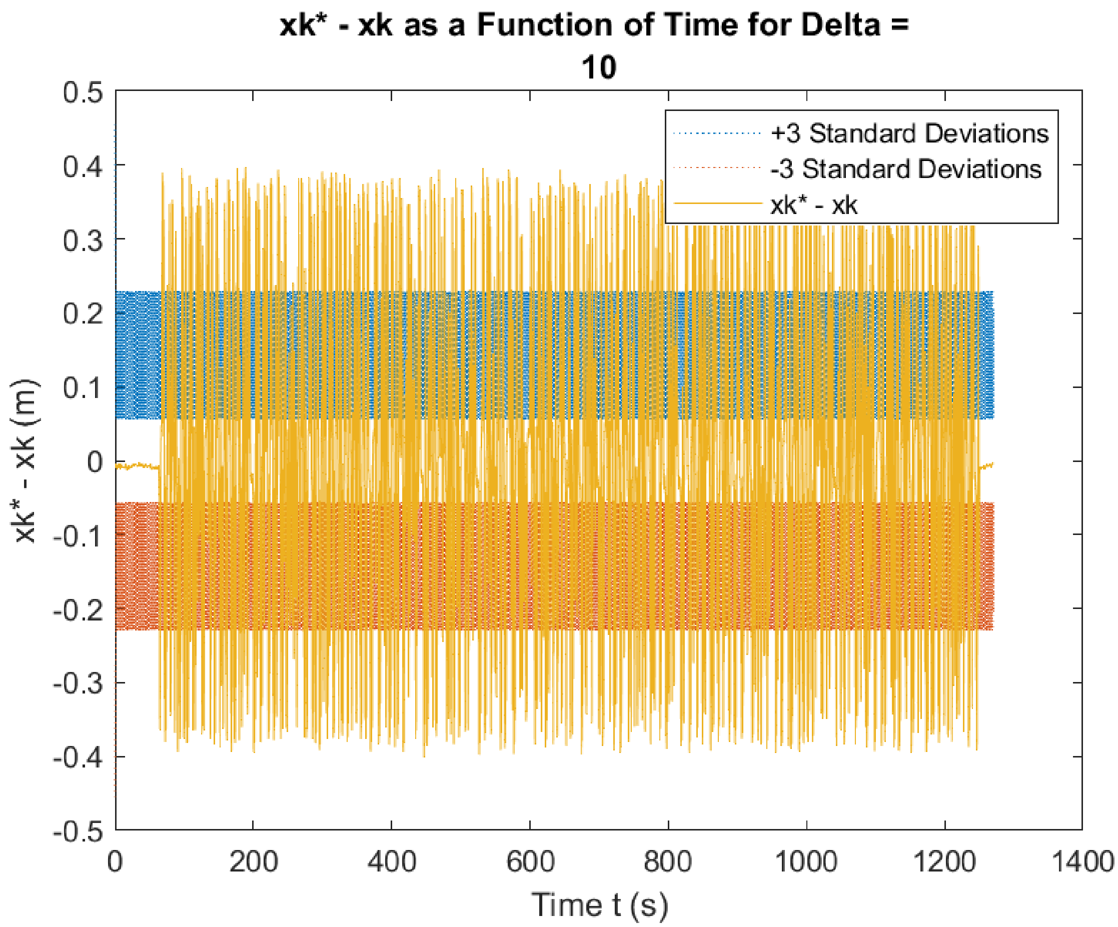

Figure 4: Difference Between Predicted State and True State Over Time for Delta = 10

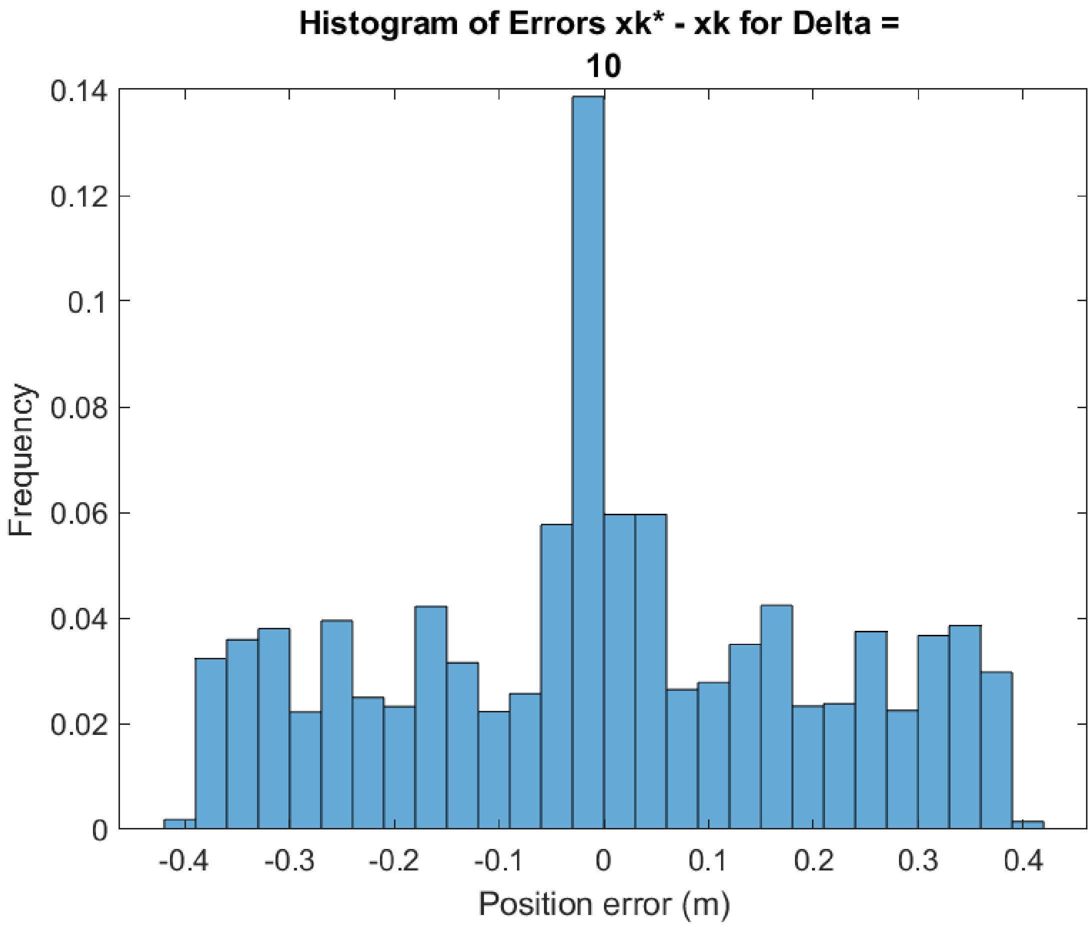

Figure 5: Histogram of Errors of the Predicted State Compared to the True State for Delta = 10

The histogram of errors and uncertainty enveloped slightly corresponds to the highest uncertainty value given by the least squares formulation. Over half the data is contained within the highest uncertainty value bound.

Delta = 100

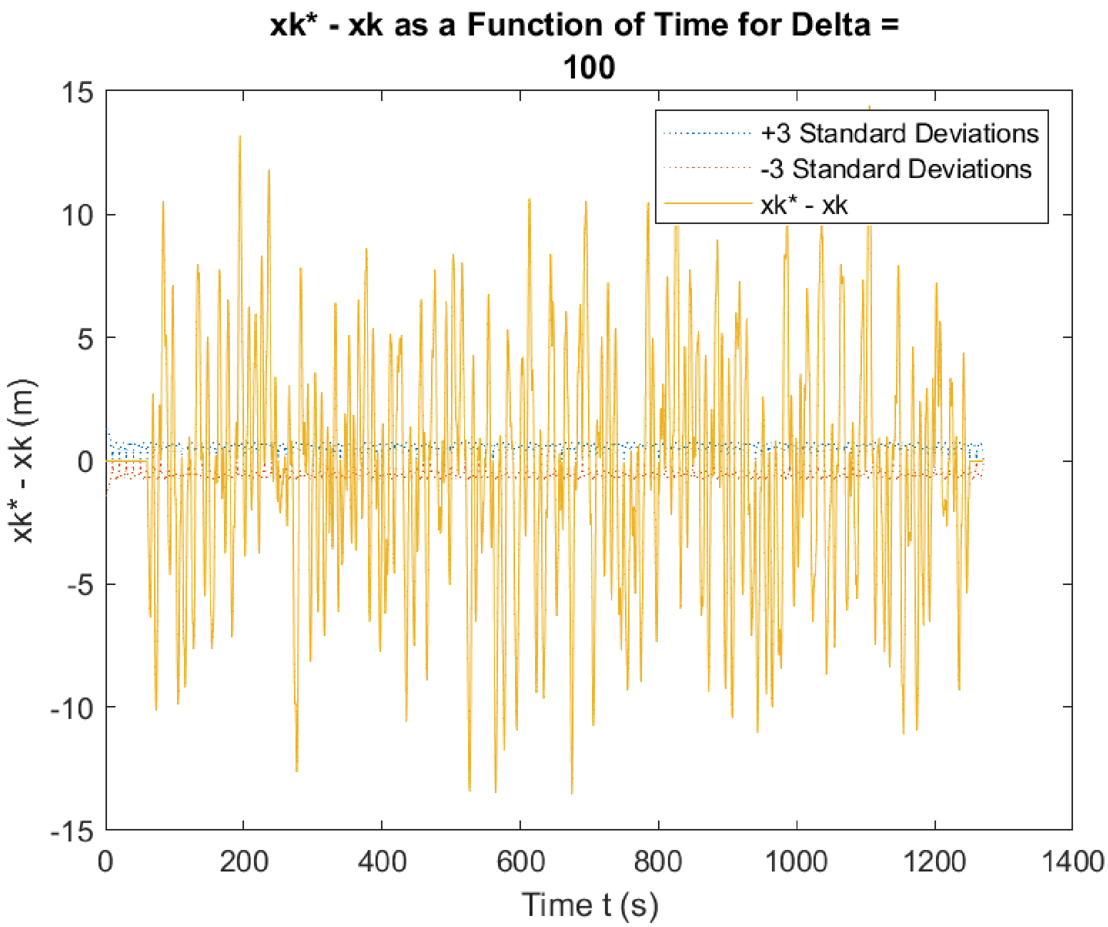

Figure 6: Difference Between Predicted State and True State Over Time for Delta = 100

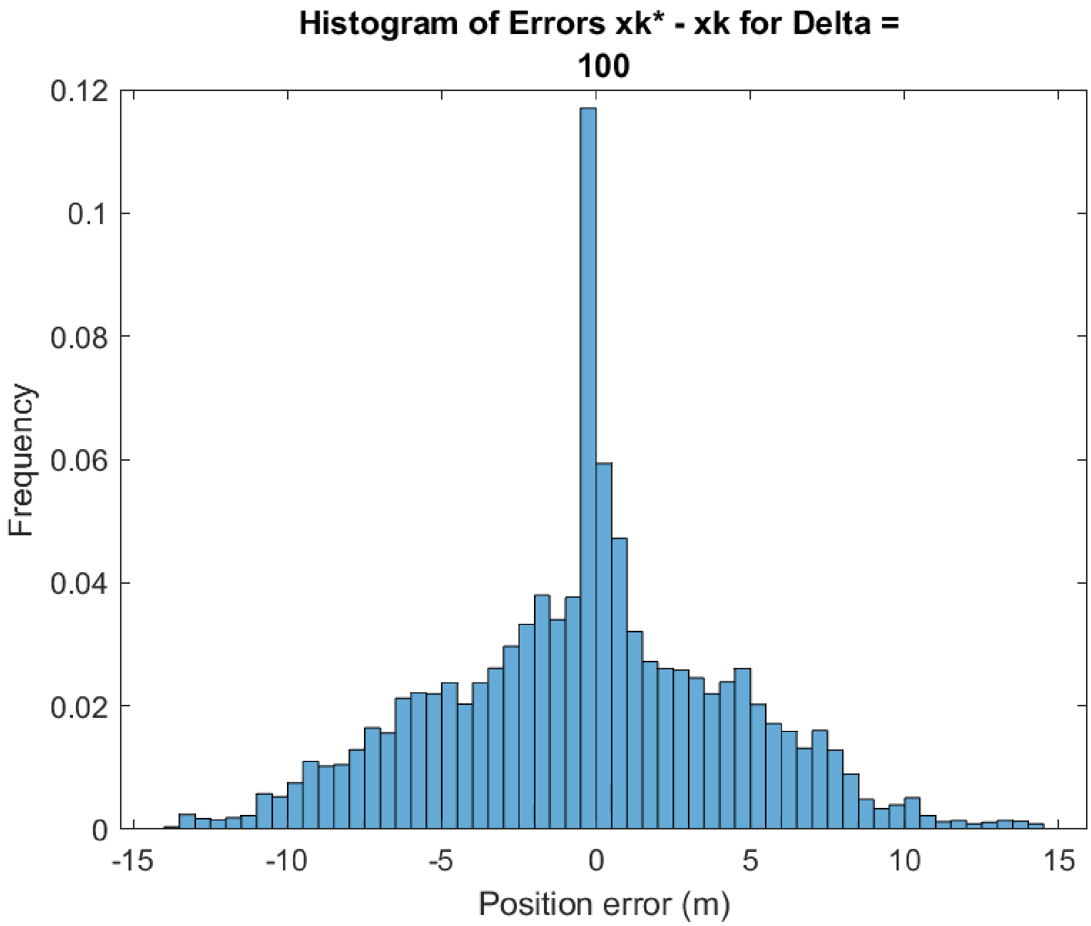

Figure 7: Histogram of Errors of the Predicted State Compared to the True State for Delta = 100

The histogram of errors and uncertainty envelope does not correspond well to the highest uncertainty value given by the least squares formulation.

Delta = 1000

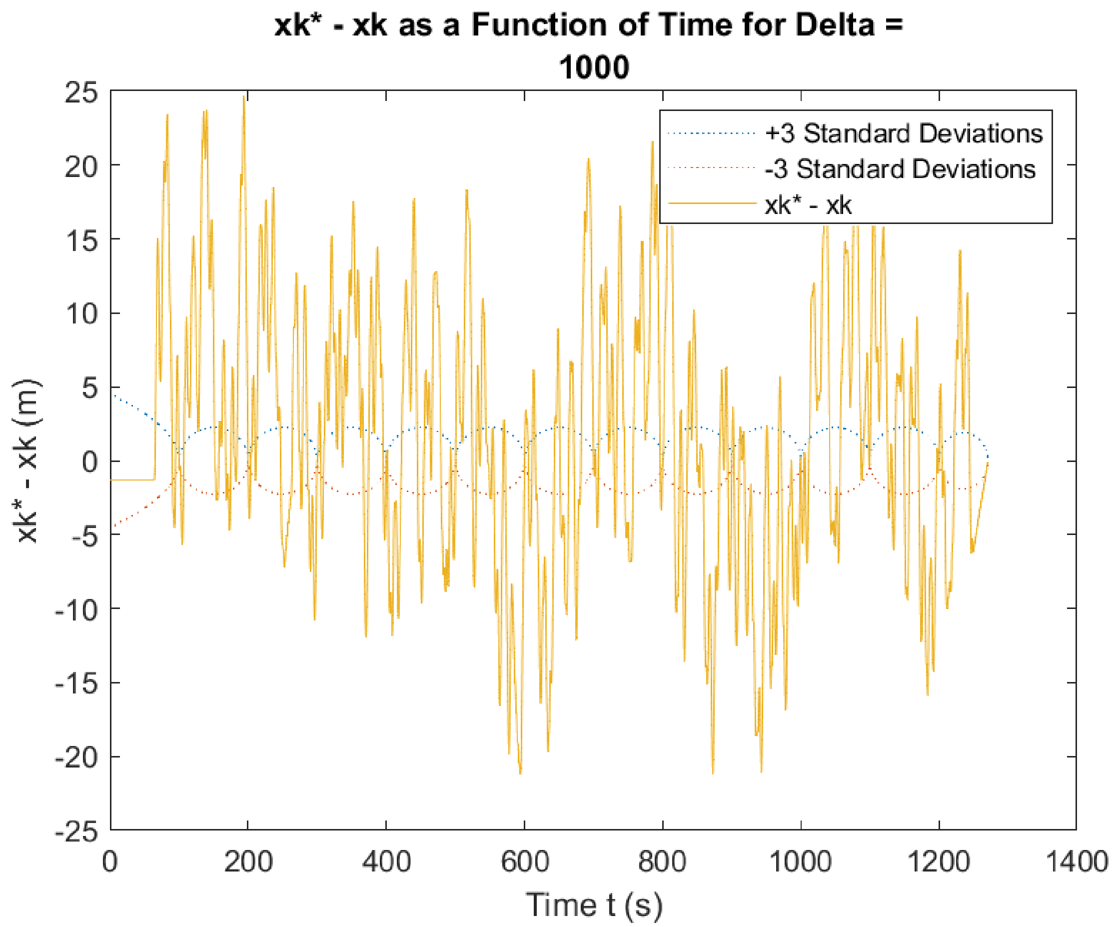

Figure 8: Difference Between Predicted State and True State Over Time for Delta = 1000

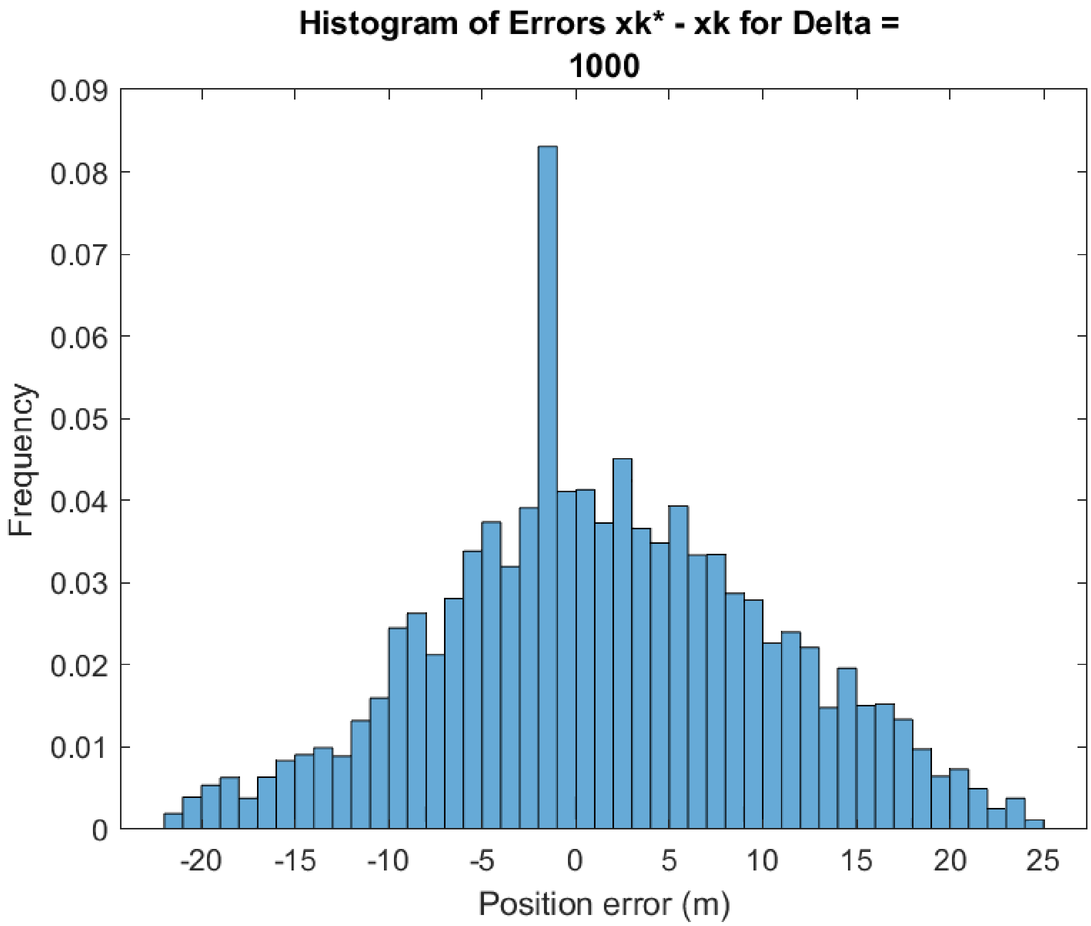

Figure 9: Histogram of Errors of the Predicted State Compared to the True State for Delta = 1000

The histogram of errors and uncertainty envelope does not correspond well to the highest uncertainty value given by the least squares formulation.

3 Appendix

```matlab
% xk = xk-1 + Tuk + wk
% yk = xc - rk = xk + nk

clear
%Load dataset
load('dataset1.mat')
deltas = [1,10,100,1000];
%Loop over deltas
for delta = deltas
    %Range readings used
    r_used = r(1+delta:delta:end);
    %Range reading indices used
    i_y = 10*t(1+delta:delta:end);

    if mod((length(r)-1),delta) ~= 0
        r_used = [r_used;r(end)];
        i_y = [i_y;10*t(end)];
    end

    %Measurements Used
    y_used = l - r_used;
    %Odometry Velocities Used
    v_used = v(2:end);
    %Z Vector
    z = [v_used;y_used];
    %Number of velocities used
    K = size(v_used,1);
    %Number of measurements used
    y_K = size(y_used,1);

    %Create A Inverse matrix
    I_K = eye(K);
    I_y = eye(y_K);
    O_K = zeros(K,1);
    pos = [O_K,I_K];
    neg = [-I_K,O_K];

    A_inv = pos + neg;

    %Create C Matrix with only taking a measurement every delta time steps
    C = zeros(y_K,K+1);
    acc = 1;
    for i = i_y'
        C(acc,uint32(i+1)) = 1;
        acc = acc + 1;
    end

    %Create H marix
    H = [A_inv;C];
```

```
%Create W Inverse matrix
Q_inv = (1/v_var)*I_K;
R_inv = (1/r_var)*I_y;
Os_K = zeros(K,y_K);
Os_yK = zeros(y_K,K);
W_inv = [Q_inv,Os_K;Os_yK,R_inv];

%%
% Left hand side of linear least squares problem
A = H'*W_inv*H;
% Right hand side of linear least squares problem
b = H'*W_inv*z;
%%

% Perform Cholesky factorization
[L,flag] = chol(A,"lower");

% Solve lower triangular linear system
opts.LT = true;
d = linsolve(L,b,opts);
opts.LT = false;

% Solve upper triangular linear system
opts.UT = true;
x_hat = linsolve(L',d,opts);
opts.UT = false;

% Calculate covariance by inverting left hand side of linear system
cov = inv(A);

% Take only the variance of each individual point
cov_entries = diag(cov);
% Get standard deviation
std_dev = sqrt(cov_entries);
% Difference between predicted and true state
x_diff = x_hat - x_true;

% Upper and lower bounds of standard deviation (+/- 3 std deviations)
x_upper = 3*std_dev;
x_lower = -3*std_dev;

% Plot difference between predicted and true state
figure
plot(t,x_upper,':',t,x_lower,':',t,x_diff,'-');
title(["xk* - xk as a Function of Time for Delta = ", num2str(delta)]);
xlabel("Time t (s)");
ylabel("xk* - xk (m)");
legend(["+3 Standard Deviations", "-3 Standard Deviations","xk* - xk"]);
fname = sprintf('Plot_%d.png', delta);
saveas(gcf,fname)
```

```
    % Plot histogram of errors between predicted and true state
    figure
    histogram(x_diff,'Normalization','probability');
    title(["Histogram of Errors xk* - xk for Delta = ", num2str(delta)]);
    xlabel("Position error (m)");
    ylabel("Frequency");
    fname = sprintf('Histogram_%d.png', delta);
    saveas(gcf,fname)

end
```