Ben Gurion University of the Negev
Faculty of Engineering Sciences

Intelligent Robotics System

Exercise 1

Nir Dgani

Lecturer: Dr. Armin Biz

Date: 23/04/2017

1. Calculation of Kalman filter (KF) steps.

The motion model of 1D system:

$$x_t = x_{t-1} + \epsilon_t \quad \text{; with process noise: } \epsilon_t = \mathcal{N}(0, R_t) = \mathcal{N}(0, r^2) = \mathcal{N}(0,1)$$

The measurement model:

$$z_t = x_t + \delta_t \quad \text{; with measurment noise: } \delta_t = \mathcal{N}(0, Q_t) = \mathcal{N}(0,2)$$

Where $\mathcal{N}(\text{mean, variance})$ is the Gaussian random variable.

The following measurement are taken:

$$z_1 = 2, \;\; z_2 = 3$$

The initial state is given by:

$$\mu_0 = 1$$
$$\Sigma_0 = \sigma_0^2 = 10$$

The initial belief is:

$$bel(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0) = \mathcal{N}(1,10)$$

a. Calculation of two KF steps (t=1,2).

The algorithm of Kalman filter:

$$\bar{\boldsymbol{\mu}}_t = \boldsymbol{A}_t \boldsymbol{\mu}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t$$
$$\bar{\boldsymbol{\Sigma}}_t = \boldsymbol{A}_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{A}_t^T + \boldsymbol{R}_t$$
$$\boldsymbol{K}_t = \bar{\boldsymbol{\Sigma}}_t \boldsymbol{C}_t^T (\boldsymbol{C}_t \bar{\boldsymbol{\Sigma}}_t \boldsymbol{C}_t^T + \boldsymbol{Q}_t)^{-1}$$
$$\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \boldsymbol{K}_t (\boldsymbol{z}_t - \boldsymbol{C}_t \bar{\boldsymbol{\mu}}_t)$$
$$\boldsymbol{\Sigma}_t = (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{C}_t) \bar{\boldsymbol{\Sigma}}_t$$

Where:

$$\boldsymbol{A}_t = 1;\; \boldsymbol{B}_t = 0;\; \boldsymbol{C}_t = 1;\; \boldsymbol{R}_t = 1;\; \boldsymbol{Q}_t = 2$$

The algorithm is:

$$\bar{\mu}_t = \mu_{t-1}$$
$$\bar{\Sigma}_t = \Sigma_{t-1} + R_t = \Sigma_{t-1} + 1$$

$$K_t = \frac{\overline{\Sigma_t}}{\overline{\Sigma_t} + Q_t} = \frac{\overline{\Sigma_t}}{\overline{\Sigma_t} + 2}$$

$$\mu_t = \overline{\mu_t} + K_t(z_t - \overline{\mu_t})$$

$$\Sigma_t = (1 - K_t)\overline{\Sigma_t}$$

<u>For t=1:</u>

$$\overline{\mu_1} = \mu_0 = 1$$

$$\overline{\Sigma_1} = \Sigma_0 + 1 = 10 + 1 = 11$$

$$K_1 = \frac{\overline{\Sigma_1}}{\overline{\Sigma_1} + 2} = \frac{11}{11 + 2} = \frac{11}{13}$$

$$\mu_1 = \overline{\mu_1} + K_1(z_1 - \overline{\mu_1}) = 1 + \frac{11}{13}(2 - 1) = \frac{24}{13} = 1.846$$

$$\Sigma_1 = (1 - K_1)\overline{\Sigma_1} = \left(1 - \frac{11}{13}\right) \cdot 11 = \frac{22}{13} = 1.692$$

<u>For t=2:</u>

$$\overline{\mu_2} = \mu_1 = \frac{24}{13} = 1.846$$

$$\overline{\Sigma_2} = \Sigma_1 + 1 = \frac{22}{13} + 1 = \frac{35}{13} = 2.692$$

$$K_2 = \frac{\overline{\Sigma_2}}{\overline{\Sigma_2} + 2} = \frac{35/13}{35/13 + 2} = \frac{35}{61} = 0.573$$

$$\mu_2 = \overline{\mu_2} + K_2(z_2 - \overline{\mu_2}) = \frac{24}{13} + \frac{35}{61}\left(3 - \frac{24}{13}\right) = \frac{153}{61} = 2.508$$

$$\Sigma_2 = (1 - K_2)\overline{\Sigma_2} = \left(1 - \frac{35}{61}\right) \cdot \frac{35}{13} = \frac{70}{61} = 1.147$$

t	$\overline{\boldsymbol{\mu}_t}$	$\overline{\boldsymbol{\Sigma}}_t$	$\boldsymbol{K_t}$	$\boldsymbol{\mu_t}$	$\boldsymbol{\Sigma_t}$
1	1	11	11/13	24/13	22/13
2	24/13	35/13	35/61	153/61	70/61

b. Explanation of the parameters:

- $\boldsymbol{A}_t$ is the state transition matrix. This matrix describes how the state evolves for one time step without controls or noise.
- $\boldsymbol{B}_t$ is the control input matrix. This matrix describes how the control, u_t, changes the state for one time step.
- $\boldsymbol{C}_t$ is the measurement matrix. This matrix describes how to map the state, x_t, to an observation, z_t.
- $\boldsymbol{R}_t$ is the covariance matrix of process noise. This matrix describes the noise of the motion.
- $\boldsymbol{Q}_t$ is the covariance matrix of measurement noise. This matrix describes the noise of the measurement.
- ϵ_t and δ_t is random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance $\boldsymbol{R}_t$ and $\boldsymbol{Q}_t$ respevtively.
- $\overline{\mu_k}$ and $\overline{\Sigma_t}$ is the prediction. They describe the belief without measurement.
- K_t is the Kalman gain. The Kalman gain balances between the prediction and the measurement.
- μ_k and Σ_t is the measurement update. They describe the belief after the measurement taken into account (using the Kalman gain).

c. The calculation of steady state covariance matrix:

$$\Sigma_t = (1-K_t)\bar{\Sigma}_t = \left(1-\frac{\bar{\Sigma}_t}{\bar{\Sigma}_t+2}\right)\bar{\Sigma}_t = \left(\frac{\bar{\Sigma}_t+2-\bar{\Sigma}_t}{\bar{\Sigma}_t+2}\right)\bar{\Sigma}_t = \frac{2\bar{\Sigma}_t}{\bar{\Sigma}_t+2}$$

$$= \frac{2(\Sigma_{t-1}+1)}{\left((\Sigma_{t-1}+1)\right)+2} = 2\frac{\Sigma_{t-1}+1}{\Sigma_{t-1}+3} = 2\left(\frac{\Sigma_{t-1}+3}{\Sigma_{t-1}+3}-\frac{2}{\Sigma_{t-1}+3}\right)$$

$$= 2\left(1-\frac{2}{\Sigma_{t-1}+3}\right)$$

- $\Sigma_0 = 10$
- $\Sigma_1 = 2\left(1-\frac{2}{\Sigma_0+3}\right) = 2\left(1-\frac{2}{13}\right) = \frac{22}{13} = 1.6923$
- $\Sigma_2 = 2\left(1-\frac{2}{\Sigma_1+3}\right) = 2\left(1-\frac{2}{\frac{22}{13}+3}\right) = \frac{70}{61} = 1.1475$
- $\Sigma_3 = 2\left(1-\frac{2}{\Sigma_2+3}\right) = 2\left(1-\frac{2}{\frac{70}{61}+3}\right) = \frac{262}{253} = 1.0355$
- $\Sigma_4 = 2\left(1-\frac{2}{\Sigma_3+3}\right) = 2\left(1-\frac{2}{\frac{262}{253}+3}\right) = \frac{1030}{1021} = 1.0088$
- $\Sigma_5 = 2\left(1-\frac{2}{\Sigma_4+3}\right) = 2\left(1-\frac{2}{\frac{1030}{1021}+3}\right) = \frac{4102}{4093} = 1.0022$

$$\Sigma_\infty = \lim_{t\to\infty}\Sigma_t = 2\lim_{t\to\infty}\left(\frac{\Sigma_{t-1}+1}{\Sigma_{t-1}+3}\right) = 2\frac{\Sigma_\infty+1}{\Sigma_\infty+3}$$

$$\Sigma_\infty^2 + 3\Sigma_\infty = 2\Sigma_\infty + 2$$

$$\Sigma_\infty^2 + \Sigma_\infty - 2 = (\Sigma_\infty+2)(\Sigma_\infty-1) = 0$$

The roots of the equation are:

$$\Sigma_\infty = 1 \quad or \quad \Sigma_\infty = -2$$

Because the covariance always positive, $\Sigma_t > 0$, the result is:

$$\boxed{\Sigma_\infty = 1}$$

The calculation of steady state covariance matrix using MATLAB code:

```
t = 0:49;
E(1) = 10;

for j=2:50
    E(j) = 2*(1-2/(E(j-1)+3));
end

plot (t,E,'.b','MarkerSize',15)
grid on
xlabel('t','FontWeight','bold','FontSize',16);
ylabel('\Sigma_t','FontWeight','bold','FontSize',16);
axis([0 49 0 11]);
```

The plot:

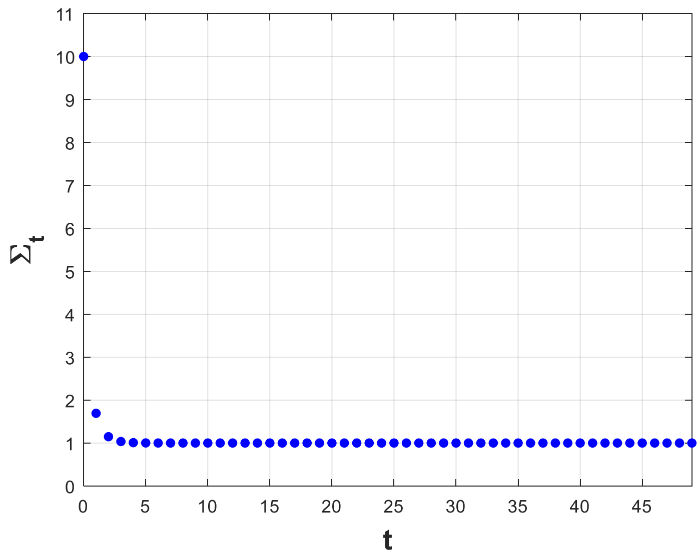

Figure 1 – The covariance for each step – t=1,…, 50. In the steady state the covariance equal to 1.

2. State estimation using a KF.

The car is moving with constant acceleration $a = 1.5\ \mathrm{m/s^2}$. The noise is: $\epsilon_a = \mathcal{N}(0, \sigma_a^2)$.

z is the measurement of the position x. The measurement noise is: $\delta_t = \mathcal{N}(0, \sigma^2)$.

The total time of the simulation is: $T = 10\mathrm{s}$ with a time step of $\Delta t = 0.1\mathrm{s}$ (100 steps).

a. The state vector :

$$\boldsymbol{x}_t = \begin{Bmatrix} x_t \\ \dot{x}_t \end{Bmatrix}$$

By develop x by Taylor series (second order):

$$x_{t+1} = x_t + \dot{x}_t \Delta t + \frac{1}{2} \ddot{x}_t \Delta t^2$$

Another way to calculate the position is:

$$a_x = const. = 2$$

$$v_x = \int a_x \, dt = a_x \int dt = a_x t + v_{x_0}$$

$$x = \int v_x \, dt = \int a_x t + v_{x_0} \, dt = \frac{1}{2} a_x t^2 + v_{x_0} t + x_0$$

For discrete time, Δt, the position is calculated numerically:

$$x_{t+1} = x_t + \dot{x}_t \Delta t + \frac{1}{2} \ddot{x}_t \Delta t^2 = x_t + v_t \Delta t + \frac{1}{2} a \Delta t^2$$

The acceleration can be thought of as the second derivative of x with respect to t:

$$\boldsymbol{a} = < a_x, a_y, a_z > = \frac{d^2 \boldsymbol{x}}{dt^2} = \ddot{\boldsymbol{x}} = < \ddot{x}, \ddot{y}, \ddot{z} > \ \rightarrow \ \ a_x = a = \frac{d^2 x}{dt^2} = \ddot{x}$$

The velocity, $v_x = \dot{x}$, is defined by:

$$\dot{x}_{t+1} = \dot{x}_t + \ddot{x}_t\Delta t = \dot{x}_t + a\Delta t$$

The motion model:

$$\boldsymbol{x}_{t+1} = \boldsymbol{A_t}\boldsymbol{x}_t + \boldsymbol{B_t}\boldsymbol{u}_t + \boldsymbol{\epsilon}_t$$

$$\underbrace{\begin{Bmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{Bmatrix}}_{\boldsymbol{x}_{t+1}} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{A_t}} \underbrace{\begin{Bmatrix} x_t \\ \dot{x}_t \end{Bmatrix}}_{\boldsymbol{x}_t} + \underbrace{\begin{Bmatrix} \Delta t^2/2 \\ \Delta t \end{Bmatrix}}_{\boldsymbol{B_t}} \underbrace{a}_{\boldsymbol{u}_t} + \boldsymbol{\epsilon}_t$$

Where the process noise:

$$\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{R}_t)$$

$$\boldsymbol{\epsilon}_t = \boldsymbol{B}_t \sigma_a \eta = \begin{Bmatrix} \Delta t^2/2 \\ \Delta t \end{Bmatrix} \sigma_a \cdot \mathcal{N}(0,1)$$

Covariance matrix of process noise:

$$\boldsymbol{R}_t = E[\boldsymbol{\epsilon}_t \cdot \boldsymbol{\epsilon}_t^T] = \boldsymbol{B}_t \boldsymbol{B}_t^T E[\sigma_a^2 \eta^2] = \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix} \sigma_a^2 \underbrace{E[\eta^2]}_{=1}$$

$$= \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix} \sigma_a^2$$

The measurement, z, is measured the location x only:

$$z_t = x_t$$

The measurement model:

$$\boldsymbol{z}_t = \boldsymbol{C}_t \boldsymbol{x}_t + \boldsymbol{\delta}_t$$

$$\boldsymbol{z}_t = \underbrace{\{1 \quad 0\}}_{\boldsymbol{C}_t} \cdot \boldsymbol{x}_t + \boldsymbol{\delta}_t = x_t + \boldsymbol{\delta}_t$$

Where the measurement noise:

$$\boldsymbol{\delta}_t \sim \mathcal{N}(0, \boldsymbol{Q}_t) = \mathcal{N}(0, \sigma^2).$$

Covariance matrix of measurement noise:

$$\boldsymbol{Q}_t = \sigma^2$$

b. Reading of the position measurement of the sensor from the 'Data_Ex2.txt' file:

```
data = dlmread('Data_EX2.txt'); % Reading of the data file
t = data(:,1); % Time
pos = data(:,2); % The position measurment of the sensor
```

The estimate of the car position provided using **smooth** function:

```
pos_smth = smooth(pos); % An estimate of the car position
```

The code of the graph of the measurement obtained using the code:

```
plot(t,pos,'.b',t,pos_smth,'.r','MarkerSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('Position [m]','FontWeight','bold','FontSize',16);
title('x Position Measurment','FontWeight','bold','FontSize',16);
legend('Position Measurment','Position Measurment -
Estimate','FontWeight','bold','FontSize',16);
axis([0, 10, -40, 100]);
```

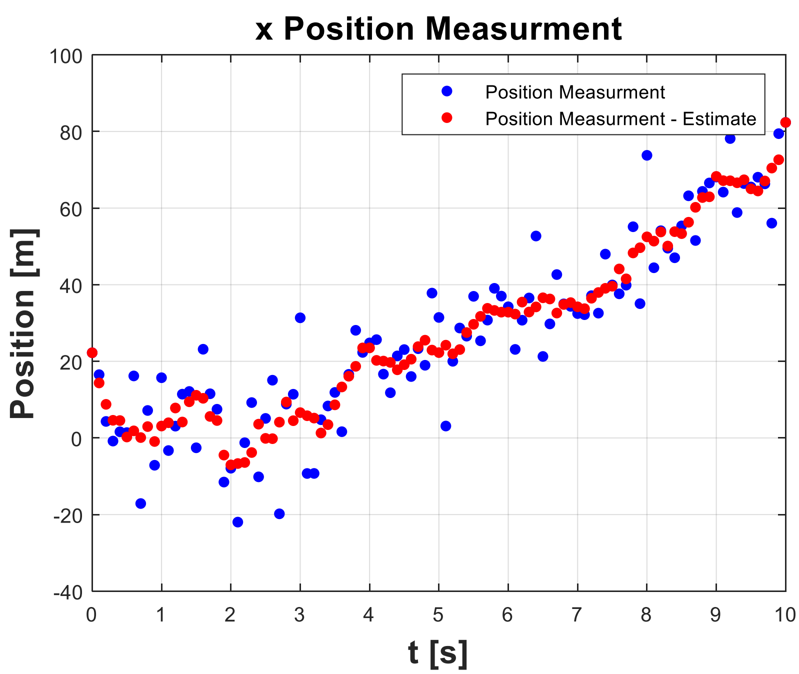

Figure 2 – The position measurement depend on time. The blue markers is the original measurement data of the position. The red markers is the estimate of the car position (using smooth function).

The car velocity is calculated numerically:

$$v_t = \dot{x}_t = \frac{x_{t+1} - x_t}{\Delta t}$$

```
dt = 0.1;
for j = 1:100
   v(j) = (pos(j+1)-pos(j))/dt; % Velocity
   v_smth(j) = (pos_smth(j+1)-pos_smth(j))/dt; % Estimate velocity
end
```

The code of the graph of the estimate of the car velocity obtained using the code:

```
plot(t(1:100),v,'.b',t(1:100),v_smth,'.r','MarkerSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('v [m/s]','FontWeight','bold','FontSize',16);
title('Velocity','FontWeight','bold','FontSize',16);
legend('Velocity','Estimate velocity');
axis([0, 10, -450, 450]);
```

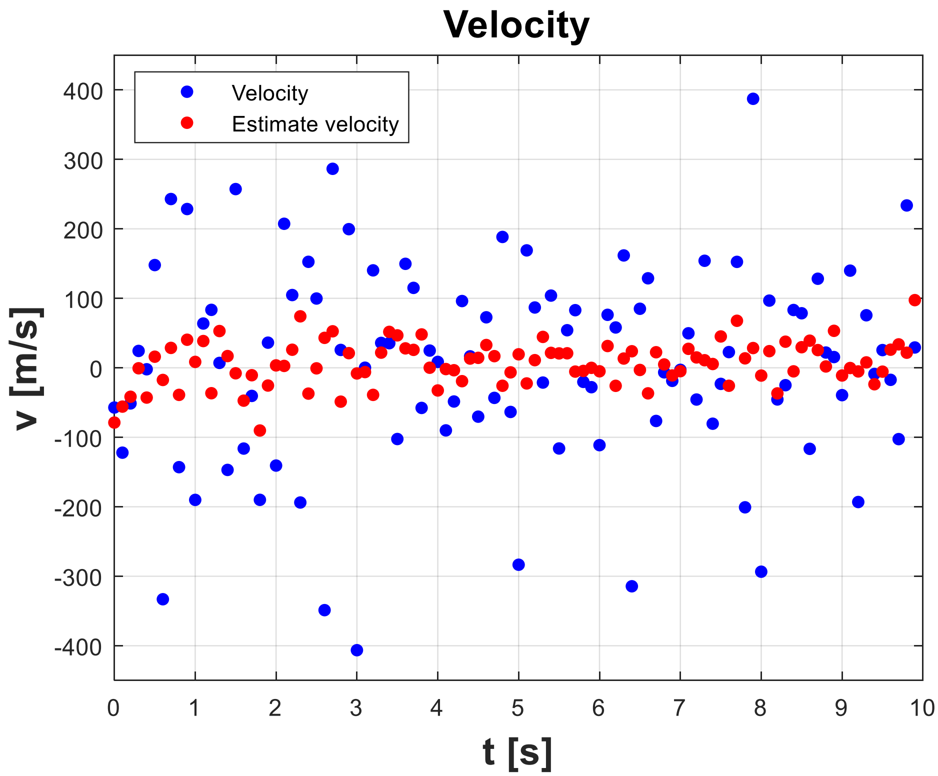

Figure 3 - The car velocity depend on time. The blue markers is the velocity obtained from the original measurement data of the position. The red markers is the estimate of the car velocity.

c. The exact position and velocity:

The process noise is: $\sigma_a^2 = 0.25 \quad \rightarrow \quad \sigma_a = 0.5$.

The initial state is:

$$\boldsymbol{x}_0 = \boldsymbol{\mu}_0 = \begin{Bmatrix} x_0 \\ \dot{x}_0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

The motion model:

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{\epsilon}_t$$

$$\begin{Bmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{Bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} x_t \\ \dot{x}_t \end{Bmatrix} + \begin{Bmatrix} \Delta t^2/2 \\ \Delta t \end{Bmatrix} a + \begin{Bmatrix} \Delta t^2/2 \\ \Delta t \end{Bmatrix} \sigma_a \eta$$

The covariance matrix of process noise:

$$\boldsymbol{R}_t = \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix} \sigma_a^2$$

The exact position and the exact velocity is given using the code:

```
x0 = 0; v0 = 0; % The initial state
Xc(1,1) = x0;
Xc(2,1) = v0;
Sa = 0.5; % The process noise: Sa^2=0.25

A = [1 dt; 0 1]; % State transition matrix
B = [0.5*dt^2; dt]; % Control input matrix:
u = 1.5; % u=a=1.5 m/s^2

for j = 1:100;
    etta = normrnd(0,1);
    ea = B*Sa*etta; % The process noise
    Xc(:,j+1) = A*Xc(:,j) + B.*u + ea; % The motion model
end
```

The figure of the position depend on time including the exact position using the dynamic (part **a**):

```
plot(t,pos,'.-b',t,pos_smth,'.-r',t,X(1,:),'.-g','MarkerSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('Position [m]','FontWeight','bold','FontSize',16);
title('x Position','FontWeight','bold','FontSize',16);
legend('Position Measurment','Position Measurment -
Estimate','Exact Position');
axis([0, 10, -40, 100]);
```

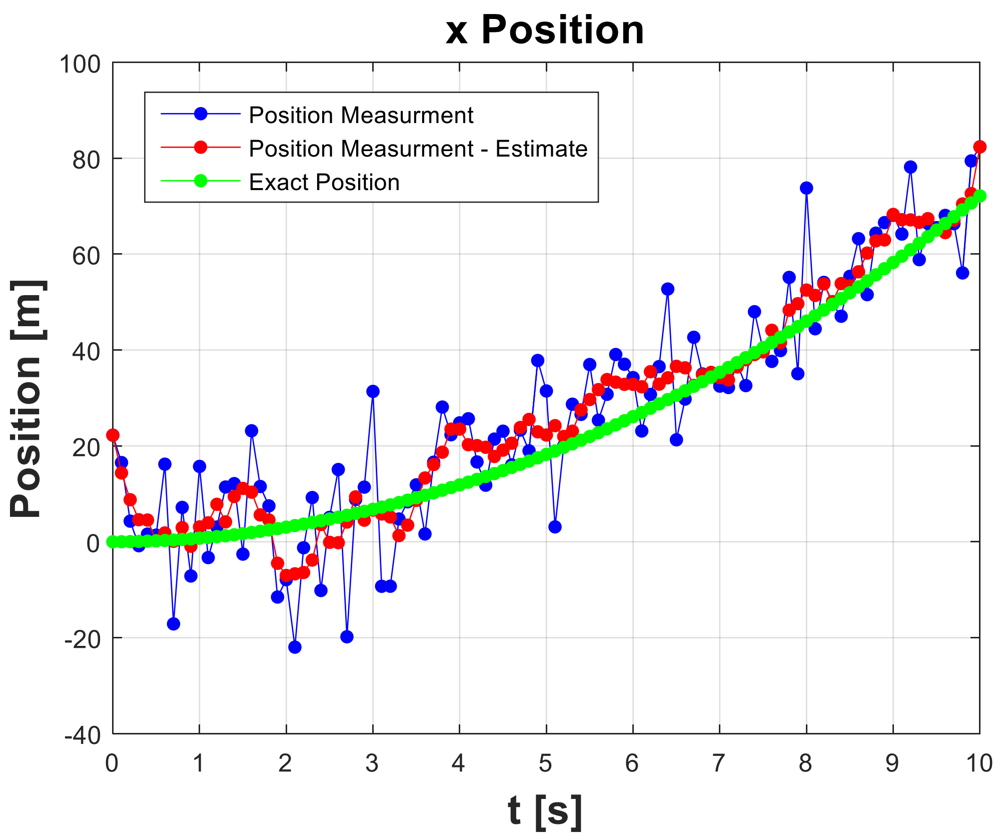

Figure 4 - The position measurement depend on time. The blue markers is the original measurement data of the position. The red markers is the estimate of the car position (using smooth function). The green markers is the exact position velocity is obtained using the dynamic of the system.

The figure of the velocity depend on time including the exact velocity using the dynamic (part **a**):

```
plot(t(1:100),v,'.b',t(1:100),v_smth,'.r',t,X(2,:),'.g','MarkerSiz
e',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('v [m/s]','FontWeight','bold','FontSize',16);
title('Velocity','FontWeight','bold','FontSize',16);
legend('Velocity Mesurment','Estimate Velocity','Exact Velocity');
```

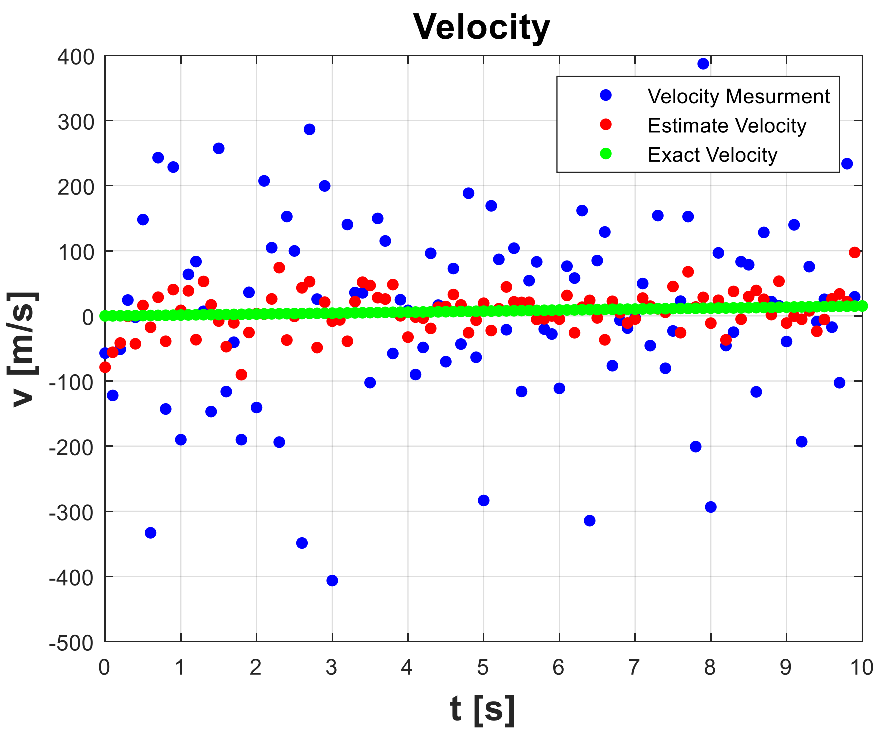

Figure 5 - The car velocity depend on time. The blue markers is the velocity obtained from the original measurement data of the position. The red markers is the estimate of the car velocity. The green markers is the exact velocity is obtained using the dynamic of the system.

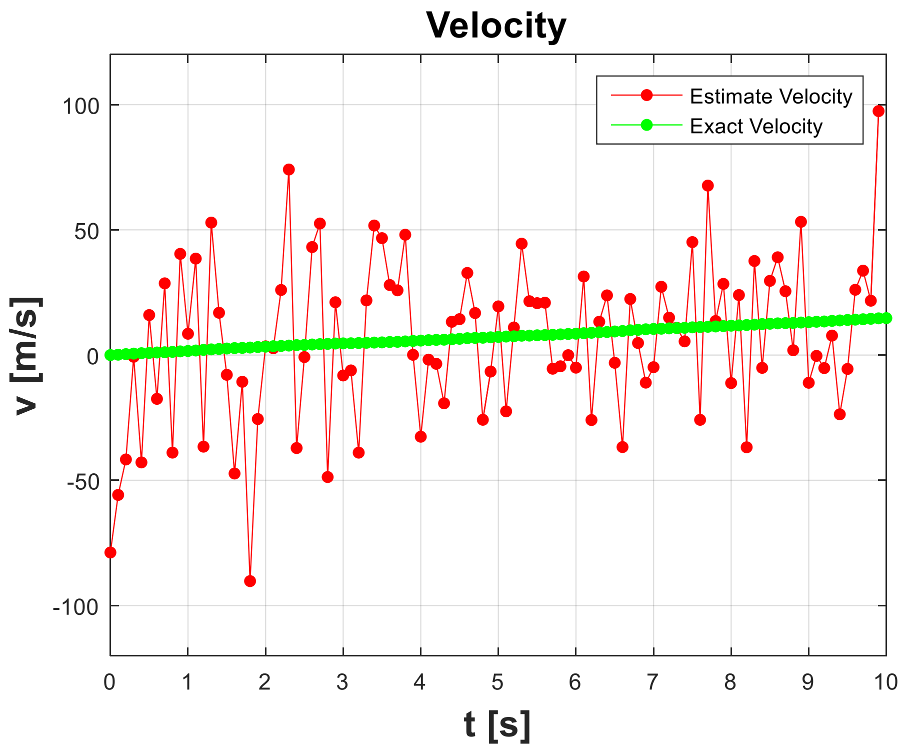

Figure 6 – An enlargement graph of the car velocity depend on time. The red markers is the estimate of the car velocity. The green markers is the exact velocity is obtained using the dynamic of the system.

d. The implement of KF:

The process noise is: $\sigma_a^2 = 0.25 \quad \rightarrow \quad \sigma_a = 0.05.$

The measurement noise is: $\sigma^2 = 100 \quad \rightarrow \quad \sigma = 10.$

The initial state is:

$$\boldsymbol{x}_0 = \boldsymbol{\mu}_0 = \begin{Bmatrix} x_0 \\ \dot{x}_0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The algorithm of Kalman filter:

$$\overline{\boldsymbol{\mu}}_t = \boldsymbol{A}_t \boldsymbol{\mu}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{\mu}_{t-1} + \begin{Bmatrix} \Delta t^2/2 \\ \Delta t \end{Bmatrix} a$$

$$\overline{\boldsymbol{\Sigma}}_t = \boldsymbol{A}_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{A}_t^T + \boldsymbol{R}_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{\Sigma}_{t-1} \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} + \begin{bmatrix} \Delta t^4/4 & \Delta t^3/2 \\ \Delta t^3/2 & \Delta t^2 \end{bmatrix} \sigma_a^2$$

$$\boldsymbol{K}_t = \overline{\boldsymbol{\Sigma}}_t \boldsymbol{C}_t^T (\boldsymbol{C}_t \overline{\boldsymbol{\Sigma}}_t \boldsymbol{C}_t^T + \boldsymbol{Q}_t)^{-1}$$

$$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - \boldsymbol{C}_t\overline{\boldsymbol{\mu}}_t)$$

$$\boldsymbol{\Sigma}_t = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{C}_t)\overline{\boldsymbol{\Sigma}}_t$$

The algorithm of Kalman filter using MATLAB code:

```
x0 = 0; v0 = 0; % The initial state
E0 = zeros(2);
Xd(1,1) = x0;
Xd(2,1) = v0;
E(:,:,1) = E0;
mu = Xd;
mu_bar(:,1)=[0;0];
dt = 0.1; % Time step [s]

Sa = 0.5; % The process noise: Sa^2=0.25
S = 0.1; % The measurment noise: S^2=1

A = [1 dt; 0 1]; % State transition matrix
B = [0.5*dt^2; dt]; % Control input matrix
C = [1 0]; % Measurement matrix
R = [dt^4/4 dt^3/2; dt^3/2 dt^2]*Sa^2; % Covariance matrix of
process noise
Q = S^2; % Covariance matrix of measurement noise
u = 1.5; % u=a=1.5 m/s^2
z = pos; % The position measurment of the sensor from
'Data_EX2.txt'

for k = 2:100
    mu_bar(:,k) = A*mu(:,k-1)+B*u; % Prediction (move)
    E_bar(:,:,k) = A.*E(:,:,k-1).*A'+R; % Prediction (move)
    K(:,k) = E_bar(:,:,k)*C'/(C*E_bar(:,:,k)*C'+Q); % Kalman gain
    mu(:,k) = mu_bar(:,k)+K(:,k).*(z(k)-C*mu_bar(:,k)); %
Measurment upadat (sense)
    E(:,:,k) = (eye(2)-K(:,k)*C).*E_bar(:,:,k); % Measurment
upadat (sense)
end
```

The figure of the position depend on time including the KF estimate:

```
plot(t,pos,'.b',t,pos_smth,'.r',t,X(1,:),'.g',t,mu(1,:),'.c','Mark
erSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('Position [m]','FontWeight','bold','FontSize',16);
title('x Position','FontWeight','bold','FontSize',16);
legend('Position Measurment','Position Measurment -
Estimate','Exact Position','Position KF');
axis([0, 10, -40, 100]);
```

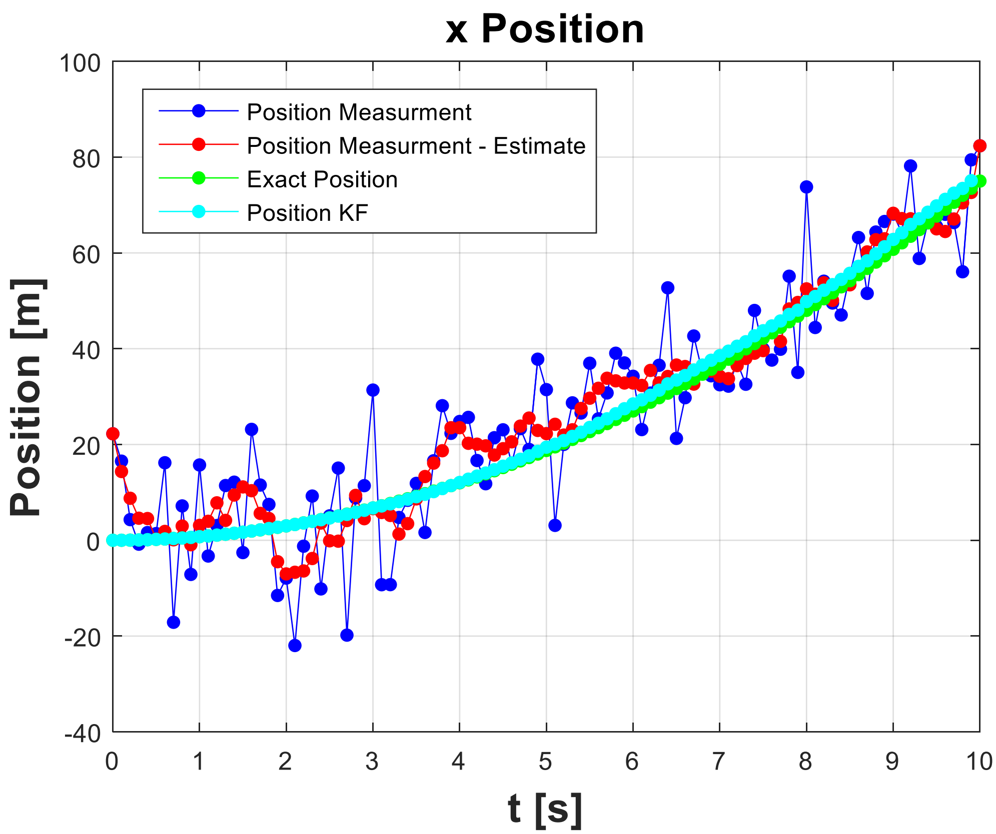

Figure 7 - The position measurement depend on time for σ=10. The blue markers is the original measurement data of the position. The red markers is the estimate of the car position (using smooth function). The green markers is the exact position velocity is obtained using the dynamic of the system. The cyan markers is the KF estimate for position.

The figure of the velocity depend on time including the KF estimate:

```
plot(t,pos,'.b',t,pos_smth,'.r',t,X(1,:),'.g',t,mu(1,:),'.c','Mark
erSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('Position [m]','FontWeight','bold','FontSize',16);
title('x Position','FontWeight','bold','FontSize',16);
legend('Position Measurment','Position Measurment -
Estimate','Exact Position','Position KF');
axis([0, 10, -40, 100]);
```

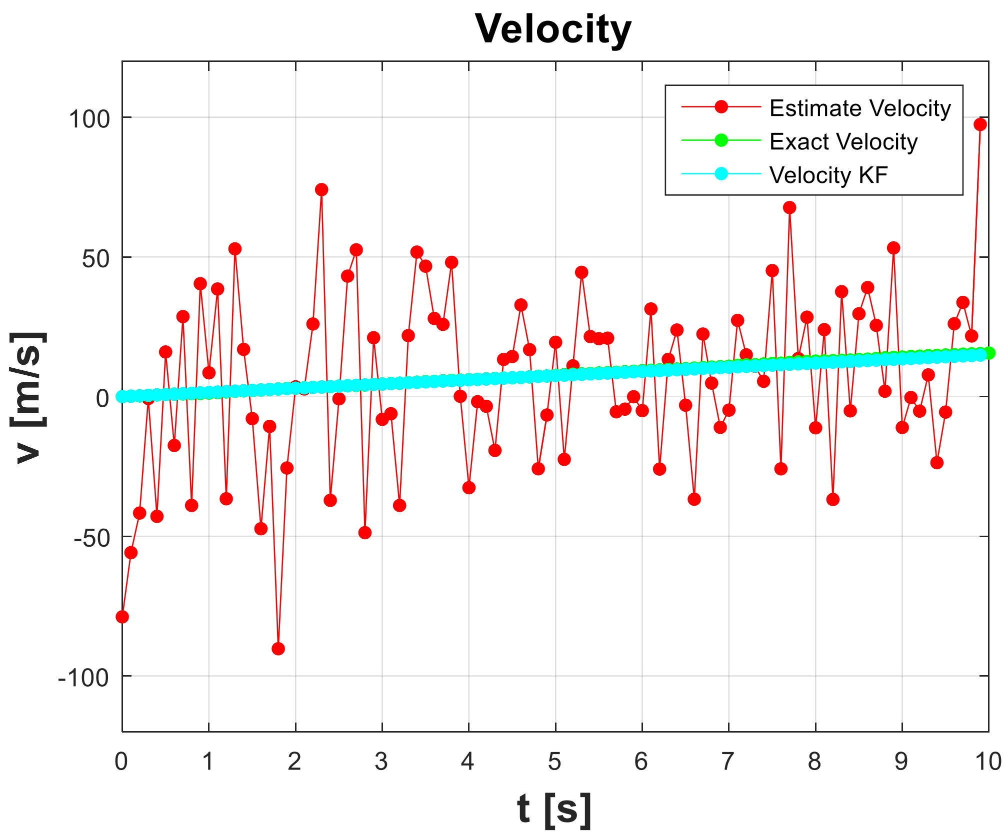

Figure 8 - The car velocity depend on time for σ=10. The red markers is the estimate of the car velocity. The green markers is the exact velocity is obtained using the dynamic of the system. The cyan markers is the KF estimate for velocity.

As can see in Figs. 6 and 7, the Kalman estimate get good accurate to the real position and the real velocity.

In the case of the measurement noise is: $\sigma^2 = 1 \ll 100$. We expect to see that Kalman gain is increased because the relation between K_t and $Q_t = \sigma^2$ is inverse. The Kalman gain for the cases of $\sigma^2 = 100$ and $\sigma^2 = 1$ where calculated using MATLAB code:

```
A = [1 dt; 0 1]; % State transition matrix
B = [0.5*dt^2; dt]; % Control input matrix
C = [1 0]; % Measurement matrix
R = [dt^4/4 dt^3/2; dt^3/2 dt^2]*Sa^2; % Covariance matrix of
process noise
Q100 = 100; % Covariance matrix of measurement noise
Q1 = 1 ; % Covariance matrix of measurement noise
u = 1.5; % u=a=1.5 m/s^2
z = pos; % The position measurment of the sensor from
'Data_EX2.txt'

for k = 2:100
    mu_bar100(:,k) = A*mu100(:,k-1)+B*u; % Prediction (move)
    E_bar100(:,:,k) = A*E100(:,:,k-1)*A'+R; % Prediction (move)
    K100(:,k) = E_bar100(:,:,k)*C'*inv(C*E_bar100(:,:,k)*C'+Q100);
% Kalman gain
    mu100(:,k) = mu_bar100(:,k)+K100(:,k)*(z(k)-C*mu_bar100(:,k));
% Measurment upadat (sense)
    E100(:,:,k) = (eye(2)-K100(:,k)*C)*E_bar100(:,:,k); %
Measurment upadat (sense)

    mu_bar1(:,k) = A*mu1(:,k-1)+B*u; % Prediction (move)
    E_bar1(:,:,k) = A*E1(:,:,k-1)*A'+R; % Prediction (move)
    K1(:,k) = E_bar1(:,:,k)*C'*inv(C*E_bar1(:,:,k)*C'+Q1); % Kalman
gain
    mu1(:,k) = mu_bar1(:,k)+K1(:,k)*(z(k)-C*mu_bar1(:,k)); %
Measurment upadat (sense)
    E1(:,:,k) = (eye(2)-K1(:,k)*C)*E_bar1(:,:,k); % Measurment
upadat (sense)
end
```

The Kalman gain, $\boldsymbol{K}_t$, is a vector:

$$\boldsymbol{K}_t = \begin{Bmatrix} k_{position_t} \\ k_{velocity_t} \end{Bmatrix} = \begin{Bmatrix} k_{x_t} \\ k_{\dot{x}_t} \end{Bmatrix}$$

The plots of the Kalman gain K100 and K1 is obtained by:

```
figure(10)
plot(t(1:100),K1(1,:),'.b',t(1:100),K100(1,:),'.r','MarkerSize',10);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('K(pos)','FontWeight','bold','FontSize',16);
legend('K1(position)','K100(position)');

figure(11)
plot (t(1:100),K1(2,:),'.b',t(1:100),K100(2,:),'.r','MarkerSize',10);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('K(velocity)','FontWeight','bold','FontSize',16);
legend('K1(velocity)','K100(velocity)');
axis([0, 10, 0, 1.5*10^-4]);
```

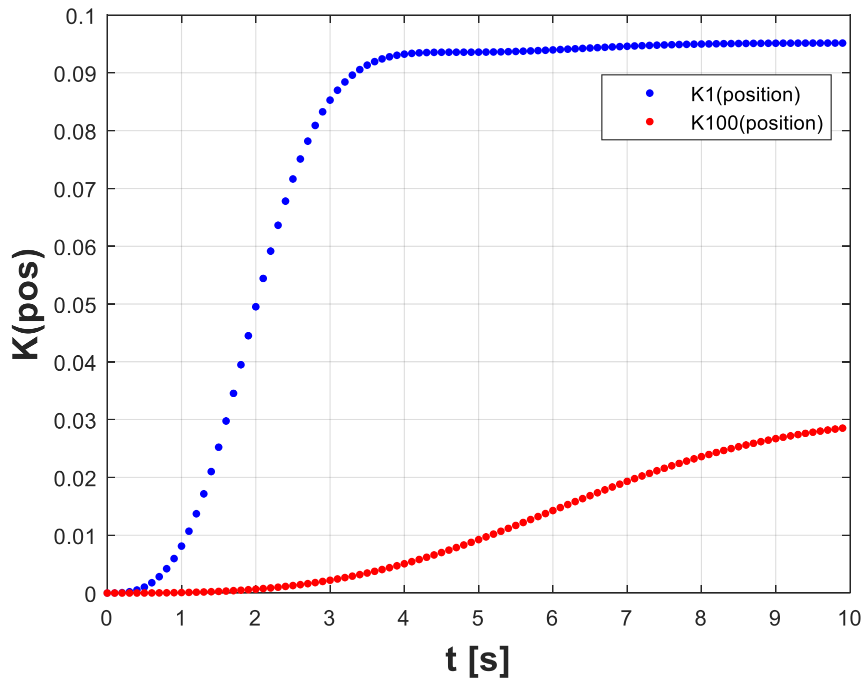

*Figure 9 – The Kalman gain of the **position** of the car. The blue markers is for the case the measurement noise is $\sigma^2 = 1$ and the red markers is for the case the measurement noise is $\sigma^2 = 100$.*

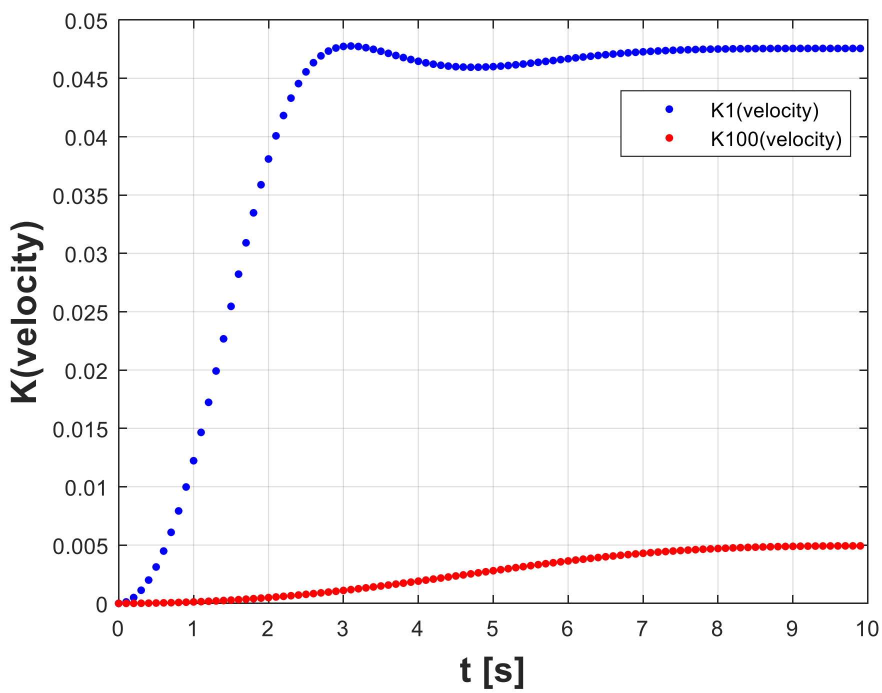

*Figure 10 - The Kalman gain of the **velocity** of the car. The blue markers is for the case the measurement noise is $\sigma^2 = 1$ and the red markers is for the case the measurement noise is $\sigma^2 = 100$*

As can see in Figs. 8 and 9, the Kalman gain in the case of $\sigma^2 = 1$ is larger than one of $\sigma^2 = 100$. This fact occur because the relation between $\boldsymbol{K}_t$ and $\boldsymbol{Q}_t = \sigma^2$ is inverse. If the measurement noise $\boldsymbol{Q}_t$ is large, then the Kalman gain $\boldsymbol{K}_t$ is small. This fact leading to:

$$\begin{cases} \mu_t \cong \bar{\mu}_t \\ \Sigma_t \cong \bar{\Sigma}_t \end{cases}$$

To compare between the results of two cases, the plot shown below:

```
figure(15)
plot (t,pos,'.-b',t,pos_smth,'.-r',t,Xc(1,:),'.-
g',t(1:100),mu1(1,:),'.-k',t(1:100),mu100(1,:),'.-
c','MarkerSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('Position [m]','FontWeight','bold','FontSize',16);
title('x Position','FontWeight','bold','FontSize',16);
legend('Position Measurment','Position Measurment - Estimate','Exact
Position','Position KF \sigma^2=1','Position KF \sigma^2=10');
axis([0, 10, -40, 100]);

figure(16)
plot (t(1:100),v_smth,'.-r',t,Xc(2,:),'.-g',t(1:100),mu1(2,:),'.-
k',t(1:100),mu100(2,:),'.-c','MarkerSize',15);
grid on;
xlabel('t [s]','FontWeight','bold','FontSize',16);
ylabel('v [m/s]','FontWeight','bold','FontSize',16);
title('Velocity','FontWeight','bold','FontSize',16);
legend('Estimate Velocity','Exact Velocity','Velocity KF
\sigma^2=1','Velocity KF \sigma^2=10');
axis([0, 10, -120, 120]);
```

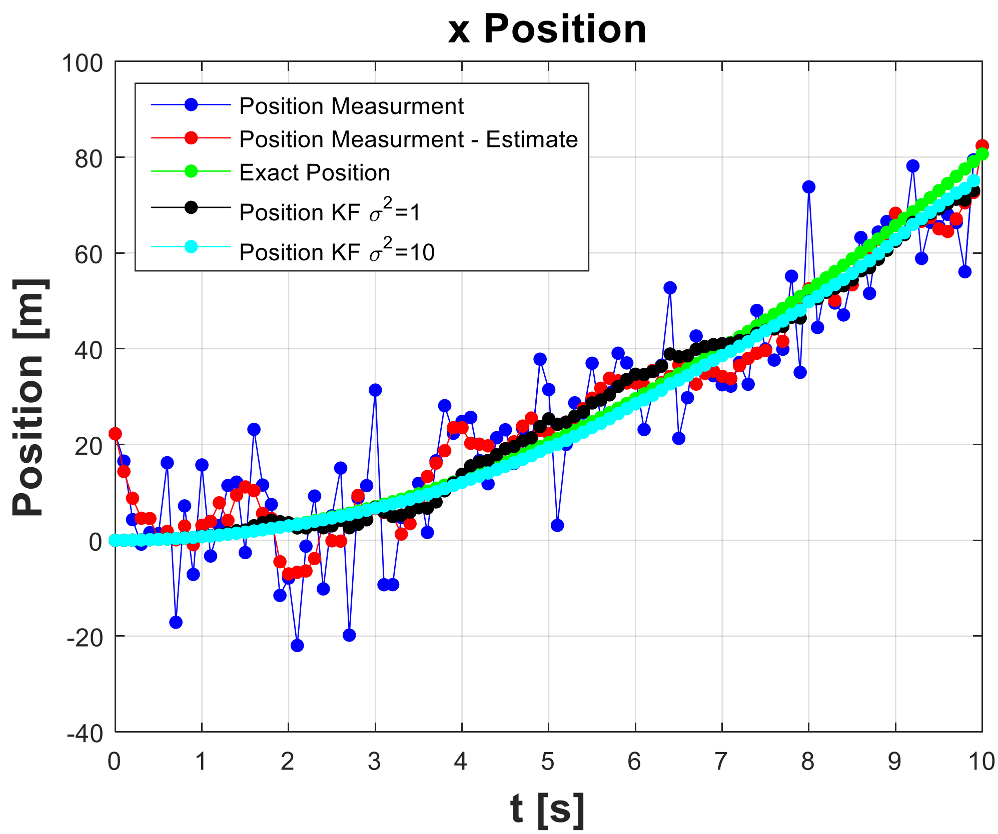

Figure 11 - The position measurement depend on time. The blue markers is the original measurement data of the position. The red markers is the estimate of the car position (using smooth *function). The green markers is the exact position velocity is obtained using the dynamic of the system. The* ***black*** *markers is the KF estimate for position where σ=1. The cyan markers is the KF estimate for position where σ=10.*

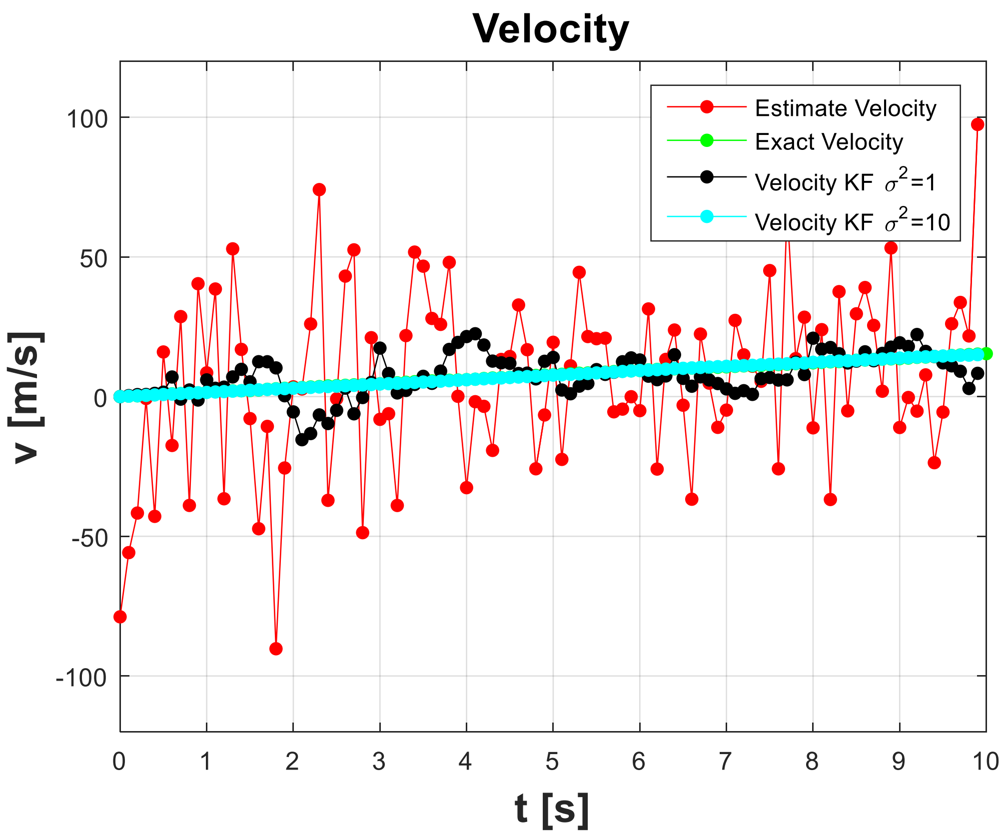

*Figure 12 - The car velocity depend on time for σ=10. The red markers is the estimate of the car velocity. The green markers is the exact velocity is obtained using the dynamic of the system. The **black** markers is the KF estimate for velocity where σ=1. The cyan markers is the KF estimate for velocity.*

In the results of the run of the KF in the second case of $\sigma^2 = 1$ have little difference and it can be seen that the result is less accurate and do not corresponding with the exact velocity. The filtered results will be more accurate as Kalman gain will be larger (the measurement noise $\boldsymbol{Q}_t$ is smaller).

e. The probability density functions (PDFs):

- State transition probability density: motion under Gaussian noise leads to:

$$p(x_t|x_{t-1},u_t) = \mathcal{N}(Ax_{t-1} + B_t u_t,\ R_t) =$$
$$= \det(2\pi R_t)^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\big(x_t - (Ax_{t-1} + B_t u_t)\big)^T R_t^{-1}\big(x_t - (Ax_{t-1} + B_t u_t)\big)\right]$$

The covariance $R_t = E[\epsilon_T \cdot \epsilon_t^T]$ describes noise of motion.

- Measurement probability density: measurement under Gaussian noise leads to:

$$p(z_t|x_t) = \mathcal{N}(C_t x_t,\ Q_t) =$$
$$= \det(2\pi Q_t)^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right]$$

The covariance $Q_t = E[\delta_t \cdot \delta_t^T]$ describes noise of measurement.

- The initial belief:

$$bel(x_0) = \mathcal{N}(\mu_0,\ \Sigma_0) =$$
$$= \det(2\pi\Sigma_0)^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right]$$

<u>For t=0:</u>

$$p_0(x) = \mathcal{N}(\mu_0,\ \sigma_0) = \frac{1}{2\pi\sigma_0}\exp\left[-\frac{(x-\mu_0)^2}{2\sigma_0}\right]$$

```
T = [0, 5, 10]; % time
X = 0:0.1:100;

% The prediction
MuBar = [mu_bar(1,1), mu_bar(1,50), mu_bar(1,99)];
EBar = [eig(E_bar(:,:,1)), eig(E_bar(:,:,50)), eig(E_bar(:,:,99))];

% The measurement update model
Position_smth = [pos_smth(1), pos_smth(50), pos_smth(99)];
Sigma = 1; % The measurement noise is Sigma^2 = 100

% The Kalman Position Esti
Mu_Kalman = [mu(1,1), mu(1,50), mu(1,99)];
E_Kalman = [eig(E(:,:,1)), eig(E(:,:,50)), eig(E(:,:,99))];

% The exact location
X_Exact = [Xc(1,1), Xc(1,51), Xc(1,100)];


for k = 1:3
%PDF\\ normpdf(x,mu,sigma):
% computes the pdf at each of the values in x using the normal
distribution.
% with mean mu and standard deviation sigma.
Prediction = normpdf(mu_bar(1,:), MuBar(k), EBar(1,k)); % (i) the
prediction step before measurement update
Measurement = normpdf(X, Position_smth(k),Sigma^2); % (ii) the
measurement model
Kalman = normpdf(mu(1,:), Mu_Kalman(k), E_Kalman(1,k)); % (iii) the
Kalman position estimate

Maximum_Value = max([max(Prediction),max(Measurement),max(Kalman)]);

    figure(k)
    plot(mu_bar(1,:), Prediction,'-g', X, Measurement,'-b', mu(1,:),
Kalman,'-r');
    hold on;
    plot([X_Exact(k) X_Exact(k)], [0 Maximum_Value], '.-k');
    grid on;
    xlabel('x [m]','FontWeight','bold','FontSize',16);
    ylabel('PDF','FontWeight','bold','FontSize',16);
    title(sprintf('PDF For
t=%d',T(k)),'FontWeight','bold','FontSize',16);
    legend('Prediction','Measurement Model','Kalman Position
Estimate','Exact Location');
    hold off;
end
```

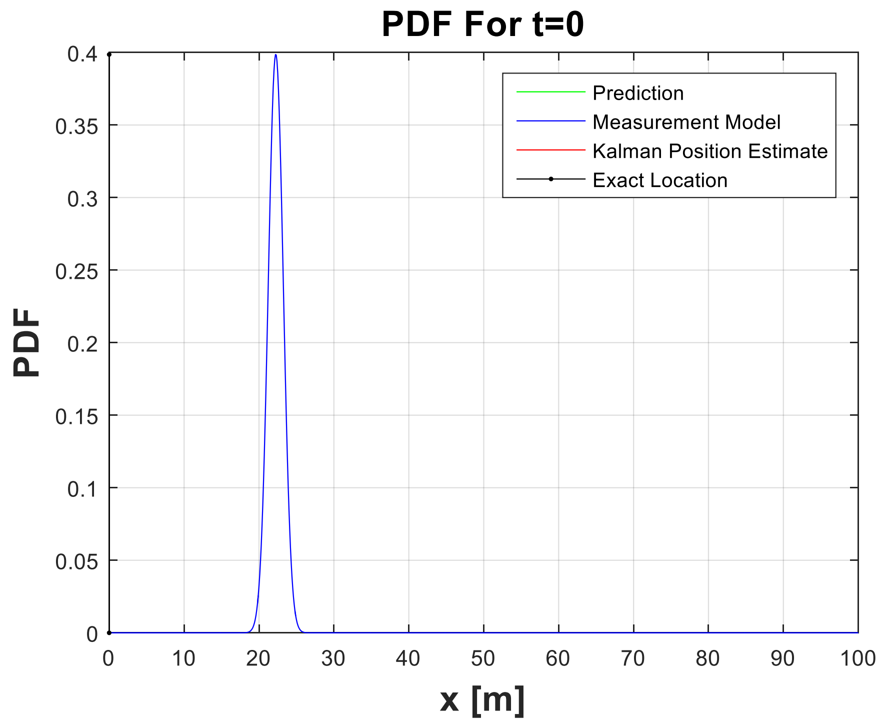

*Figure 13 – The probability density functions for time step t=0: **green**- the prediction step before measurement update. **Blue**- the measurement model. **Red**- the Kalman position estimate. **Black-** exact location of the car.*

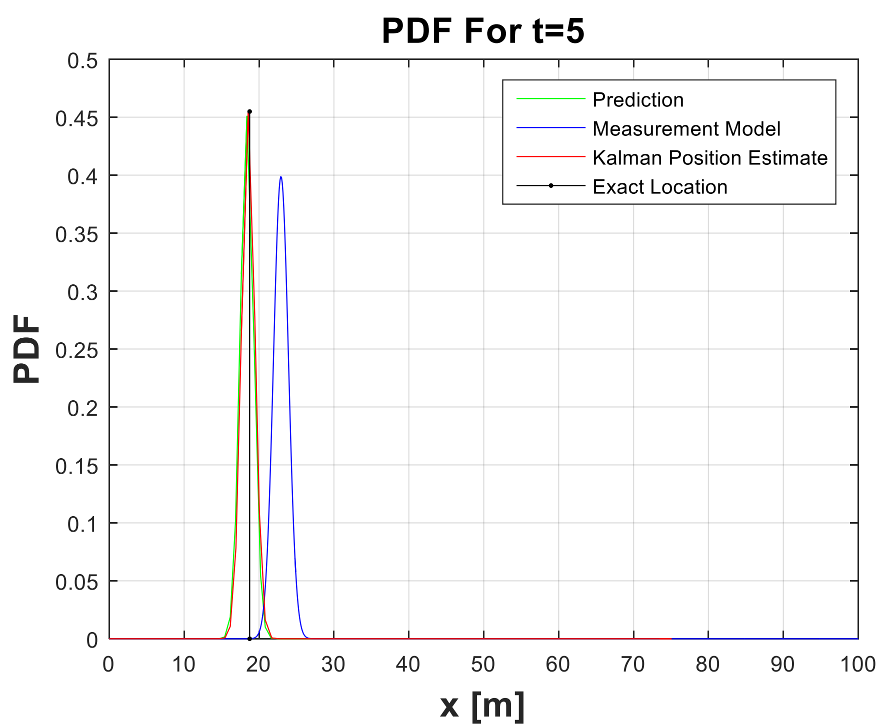

*Figure 14 - The probability density functions for time step t=5 sec: **green**- the prediction step before measurement update. **Blue**- the measurement model. **Red**- the Kalman position estimate. **Black-** exact location of the car.*

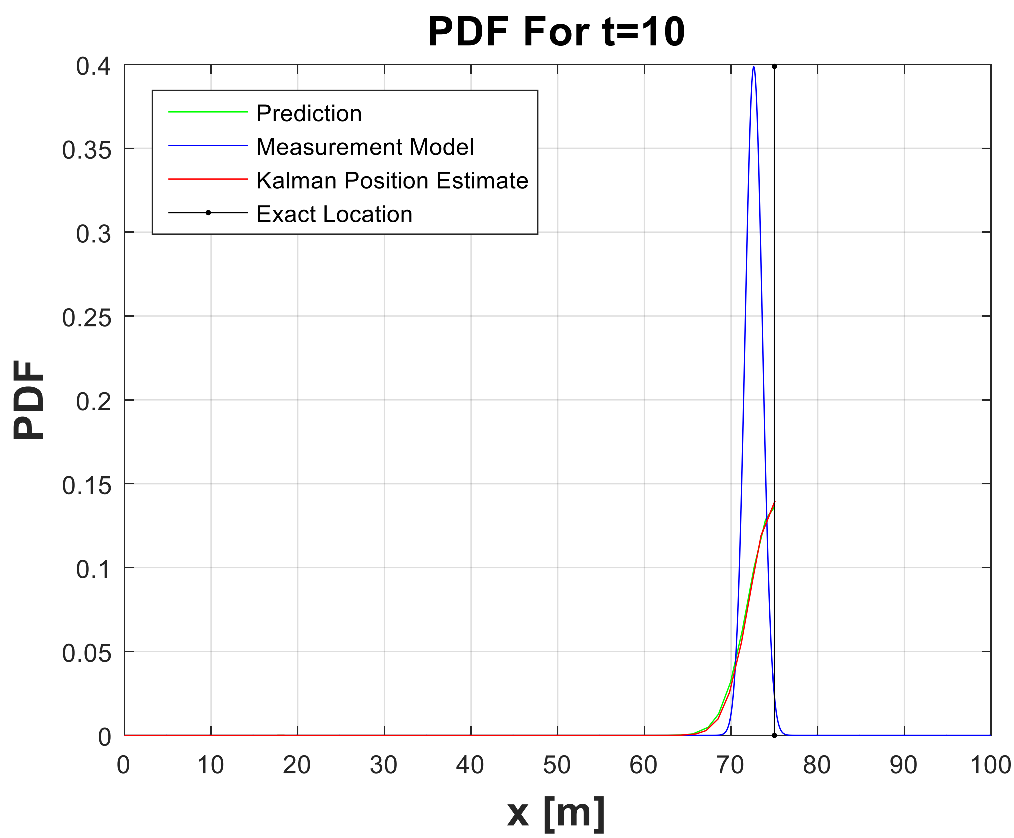

*Figure 15 - The probability density functions for time step t=10 sec: **green**- the prediction step before measurement update. **Blue**- the measurement model. **Red-** the Kalman position estimate. **Black-** exact location of the car.*

In figures, 13 – 15 can see the probability density functions. It can be seen that the prediction step loses information and measurement update gains information. The corrections results (after KF) is between the measurement and the prediction and it strives to reach the exact location (depend on the noises)