Carleton UNIVERSITY

Department of
Systems and Computer Engineering

SYSC5702 W: Sensor Fusion for Autonomous Systems

Project 1: Kalman Filter Estimation of Linear Stochastic Dynamic Systems

Due: *Monday, 3 February 2025, 11:59 PM*

Problem Statement

We have seen in the lectures an example of a linear dynamic stochastic system of a single-dimensional straight-line motion with constant acceleration. Given initial conditions (initial position and initial velocity) + noisy position observations, we could estimate the acceleration and a smoothed version of the position using Kalman Filter (KF). To perform this KF-estimation, we defined the following system states:

$$\mathrm{x(t)} = \begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix} \tag{1}$$

and a state dynamic model as follows:

$$\dot{x}(t) = Fx(t) + Gw(t)$$
$$\dot{\begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix} + G \begin{bmatrix} w_p\ (t) \\ w_v\ (t) \\ w_a\ (t) \end{bmatrix} \tag{2}$$

And a measurement model defined by:

$$y(t) = Hx(t) + v(t)$$
$$\mathrm{y(t)} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix} + n(t) \tag{3}$$

To estimate this system states, in the lectures, we defined noise shaping matrix **G**, state error covariance matrix **P**, system noise matrix **Q**, and measurement noise matrix **R**. We also showed how to calculate the exact solution if we are given the noise-free acceleration a.

You are given a Matlab code in ***"single_dimensional_KF_example_code_file.m"*** file that uses KF to estimate the acceleration and smoothed positions given noisy position observations along time.

Note: In the code given to you, you will notice that we set initial value of the acceleration state to 0.0 (`x(3,1) = 0.0;`) which means that we do NOT know the acceleration and we want KF to estimate it. This is slightly different from the given code in the lectures where we set the set initial value of the acceleration state to the true-acceleration minus a bias and we want KF to estimate that bias.

Project Requirements

The project has THREE parts. Implement ALL three parts

Part 1: Familiarize yourself with the given code

In this part, go through the given code carefully and run the code and observe the generated figures. Put the generated figures in your report with a 1-2 sentences comment about what each plot/curve represents in each figure.

Part2: Change system noise parameters

In the beginning of the code, there are some variables set for you as follows:

```
% Original Parameters Values
observation_noise_std       =  1.0; %Used to add noise to observed y
system_noise_factor         = 0.01; %Used to scale Q matrix
measurement_noise_factor    = 0.05; %Used to scale R matrix
```

Implement the following tasks:

1- Add additional random noise to the observations (Observed y). To add additional random noise to the observed y you can use the following line of code:

```
%--> Add more noise to the observed y
y = y + observation_noise_std*randn(1,length(y));
```

In your testing, try the following values for the variable ***observation_noise_std*** : **{1, 25, and 50}** and in each case run the program and plot the figures, calculate the RMSE between the estimated position and the true position (true position is given in **y_true** vector in the code) and comment on the results.

To calculate the root mean square error (**RMSE**) between the estimated value and the true value of a vector, the **RMSE** formula for vectors of size N is given by:

$$rmse = \sqrt{\frac{1}{N}\sum_{i=0}^{N}[estimated\_value(i) - true\_value(i)]^2} \quad (4)$$

Modify the given code to calculate the required RMSE and include it in your report with each case.

2- Set ***observation_noise_std*** to **25** and change the value of **system_noise_factor** to the following values **{0.01, 10.0, 100}** and in each case plot the figures, calculate the RMSE between the estimated position and the true position (true position is given in **y_true** vector in the code) and comment on the results.

3- Set ***observation_noise_std*** to **25** and **system_noise_factor = 0.01** and change the value of **measurement_noise_factor** to the following values **{0.001, 100.0, 100000}** and in each case plot the figures, calculate the RMSE between the estimated position and the true position (true position is given in **y_true** vector in the code) and comment on the results.

4- In the following code line, we set the initial values of P matrix:

```
P(:,:,1) = diag([5; 5; 5;]);
```

Change these values to **diag([100; 100; 100;])** and **diag([0; 0; 0;])** and in each case plot the figures, calculate the RMSE between the estimated position and the true position (true position is given in **y_true** vector in the code) and comment on the results. In this test, set ***observation_noise_std*** to **1** and **system_noise_factor = 0.01** and **measurement_noise_factor** to **0.05.**

Part3: Two-dimensional motion

In this part, we need to modify our system to be two-dimensional as the following figure shows:

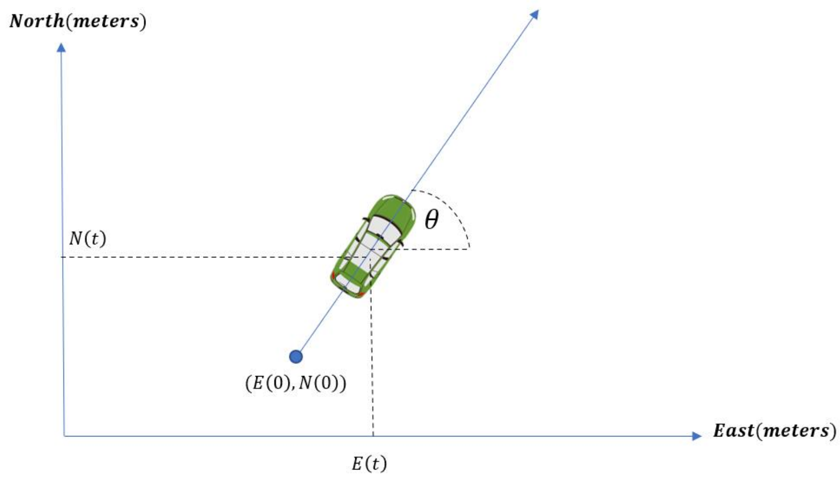

Figure 1. 2D Example Illustration

In this example, the same motions conditions apply so the object is moving in a straight line in the direction shown in the figure defined by ***constant angle*** $\boldsymbol{\theta}$ and at a *constant acceleration* $\boldsymbol{a}$ in the direction shown above defined by angle $\boldsymbol{\theta}$. The object's position is defined in 2D using east-north coordinates as shown. The position at any time t is defined by $E(t)$ and $N(t)$. Velocity is defined by $V_E(t)$ an $V_N(t)$. The system states are define as follows:

$$x(t) = \begin{bmatrix} E \\ N \\ V_E \\ V_N \\ a \end{bmatrix} \tag{5}$$

Given initial position $[E(0) = 4.3553, N(0) = 7.5435]$, initial velocity $[V_E(0) = 19.4374, V_N(0) = 33.6666]$, angle **θ = 60°**, and noisy observations of positions, we would like to 1) Estimate the value of the unknown constant acceleration ***a***. and 2) Estimate the smoothed positions. To achieve these objectives, follow the following steps and implement the following tasks:

1- Given the system states $x(t)$ defined above, write down the system dynamic equation in the form $\dot{x}(t) = Fx(t) + Gw(t)$
Where $w(t)$ is system noise of covariance ***Q*** matrix, **F** is the system dynamic matrix and ***G*** is the noise shaping matrix. Write the values of all matrices including ***F*** and for ***G*** assume that the acceleration a is the only source of

noise in the system and its noise shaping factor is **1m/s²** and for ***Q*** assume system noise covariance matrix is diag([0.001 0.001 0.001 0.001 0.001]).

Hint: Because acceleration a is constant and θ is constant, the states dynamic equations can be written as follows (note here the

$$\dot{E} = V_E$$
$$\dot{N} = V_N$$
$$\dot{V_E} = a\cos(\theta)$$
$$\dot{V_N} = a\sin(\theta)$$
$$\dot{a} = +n_a$$

where n_a is the acceleration random noise.

In writing ***G*** matrix, only the acceleration state ***a*** will have a noise shaping factor of 1 and all other sates will have zero noise shaping factor.

2- If the observed output $y(t)$ is the position vector $\begin{bmatrix} E(t) \\ N(t) \end{bmatrix}$, write down the measurement model in the form $y(t) = Hx(t) + v(t)$
Where $v(t)$ is a noise of covariance ***R*** matrix, ***H*** is the design matrix. Write the values of ***H*** matrix and for ***R*** matrix assume the measurement noise variance is **1m²** for east position and **1m²** for north position and ***R*** is diagonal (east and north position measurements noise are not corelated).

3- Write an initial value of the error covariance matrix ***P*** if your initial state is given from a probability distribution of covariance $[\mathbf{5}, \mathbf{5}, \mathbf{5}, \mathbf{5}, \mathbf{5}]^{\boldsymbol{T}}$.

4- Given that the acceleration is constant, and the motion is in straight line, write down the exact solution formula of the $E(t)$ and $N(t)$. In your equations, remember, θ is constant and the acceleration is a constant.

5- Write a Matlab software code like the one given to you to apply Kalman Filter to estimate the value of the constant acceleration ***a*** and to generate a smoothed positions estimate given the noisy position and time data. The time and noisy position data can be found in the file "**Project_1_observations_data.txt**". The file has the following format:

```
time,E_observed,N_observed
0.000000,6.083976,6.780164
0.100000,8.937036,10.860871
0.200000,9.111575,12.528999
0.300000,10.144184,14.355085
0.400000,14.917103,22.292575
0.500000,14.837196,20.900899
```

Read the data into Matlab to load the observed **y**. Note that observed **y** should be ***2xN*** matrix where ***N*** is the number of observations (***N*** is **100** in our case). In your code start from initial state **{4.3553, 7.5435, 19.4374, 33.6666, 0.0}** and $\boldsymbol{\theta} = \mathbf{60^o}$. The program must plot the following figures:

   a. Noisy and KF-estimated East position vs. time.
   b. Noisy and KF-estimated North position vs. time.
   c. Estimated acceleration (the fifth state in the state vector)

6- Using the exact solution formula you derived in previous task (4) and the estimated acceleration value you obtained from previous task (5), and the angle $\theta = 60^o$, calculate the exact east and north position solutions.

7- Using the exact solution calculated in previous step, calculate the RMSE error between the KF-estimated position and exact east and north position solutions.

8- Modify your program to perform KF-prediction only by commenting out the Kalman Filter update steps in your code and re-run your program. Include all figures in your report and comment on the plots.

Marking Rubric

1- **Report (50%)**:
   a. **Completeness** (Do ALL tasks in all parts),
   b. **Front Cover Page**: Project number, project title, name, ID, date,
   c. **Body**: write the following sections in the body of your report
      i. **Problem Statement**: maximum 300 words (summarize the problem and what you have done and the most important findings),
      ii. **Developed Solution** (describe your solution for each task as described in the tasks).
         1. If the task requires an equation or matrix writing, use professional equation writing tool like MS WORD "Equation", MathType, or Latex. Photo/Image equations are not allowed.
         2. Any equation must be numbered using (.) style
         3. If the task requires figures/plots, figures and plots have title, units on both axes and legends and different colors/styles if more than curve are plotted on the same figure.
         4. If the task requires comment, comments on the figures/plots must be precise and concise.

iii. **Conclusion**: maximum half a page (short summary about the most important findings)
iv. **Appendix**: If any instructions needed to run the associated code or load data or any required steps (if any) to run your code.

2- **Code (50%)**:
   a. Organized into clear sections,
   b. Well documented by descriptive comments,
   c. Proper variable names are used,
   d. Clear indent,
   e. Clear run/install instructions if needed,