Practical 1 - Estimation

Task 1: System Identification

This week's laboratory exercise will require students to investigate a recursive parameter estimator.

Assume that a constant vector $C = [1.3, 5]'$ is observed in Gaussian noise with variance $R = 0.1$. That is, a sequence of vector measurements $y_k \in R^2$ is described by the process:

$$y_k = C + w_k,$$

where $w_k \in R^2$ is zero mean Gaussian noise with covariance matrix $R \times I_2$, for each k. For example, $y_1 = C + w_1$, $y_2 = C + w_2$, etc. In this question, you must generate suitable data, and then construct a recursive estimate $\hat{C}_k \in R^2$ for the constant $C \in R^2$.

From the lecture material, given $\hat{C}_1 \in R^2$, a suitable recursive estimator is:

$$\hat{C}_k = \frac{k-1}{k}\hat{C}_{k-1} + \frac{1}{k}y_k$$

for $k = 2, \ldots$. At the time k, the estimate $\hat{C}_k \in R^2$ is recursive because the new estimate is constructed when a new measurement is received. Note that the new estimate is based on both the previous estimate $\hat{C}_{k-1} \in R^2$ and the new measurement $y_k \in R^2$.

Now assume that you begin with an initial estimate $\hat{C}_1 = [0.5, 1]'$, and then sequentially receive 99 noisy vector measurements $\{y_2, \ldots, y_k, \ldots, y_{100}\}$.

Tasks

The following tasks are to be completed in MATLAB:

1. Data generation: write MATLAB code that generates a noisy measurement sequence y_k.
   (a) For $k = 2, \ldots$, on the same graph, plot both the vector y_k and a vector of constant values C.
2. Recursive estimation: write MATLAB code that applies the above recursive estimator to the generated measurement data.
   (a) More plots: For $k = 1, \ldots$, on the same graph, plot both the vector of both elements of $\hat{C}_k$ and a vector of constant value C. Then plot the elements of the estimation error squares (of each element). That is, plot $(\hat{C}_k^i - C^i)^2$, for $i = 1, 2$.
   (b) Calculate the RMS value of each estimate, for $i = 1, 2$:

$$RMS^i = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\hat{C}_k^i - C^i)^2}, \quad \text{where } T = 100.$$

   Note that this RMS value is a scalar, and provides a measurement of how quickly the estimate how each element of the sequence $\{\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_k, \ldots, \hat{C}_{100}\}$ approaches the true value C.
3. Conduct a parametric study:

(a) Repeat the data generation and estimation steps for different values of R (at least 2 other values). Hint: try factors of 10. Also, try different initializations for $\hat{C}_1$.

(b) Record the results of your study in a table of RMS values.

4. Present your table, and answer the following questions:

(a) What is the role of R and/or $\hat{C}_1$on estimation error?

(b) What might happen if you had a longer/shorter data sequence?

(c) Summarise what you discovered (i.e. compare and contrast the impact of $\hat{C}_1$, R and data length on estimation error).

5. Explaining the use and importance of parameter estimation in engineering (in 300 words or less, dot point preferred)

(a) How would you choose to initialise your estimator when/if not provided an initial estimate? Why? What is important in your choice?

(b) Give an example from previous units that you have studies where you used (or could have used) parameter estimation. Why useful?

(c) Give an example related to the context of this unit (i.e. related to autonomous systems) where the need for parameter estimation might arise? Why important?

Task 2: Kalman Filter

In this question, we consider the 2D navigation problem of estimating a vehicle's cartesian (x^c, y^c) location and $(\dot{x}^c, \dot{y}^c)$ velocity from noisy GPS location information. To avoid notation confusion, we have used the superscript c to remind us that these quantities are cartesian coordinates, not our state/measurement vectors. The navigation problem considers the state-space equations:

$$\begin{aligned} x_k &= A_k x_{k-1} + v_k \\ y_k &= H_k x_k + w_k \end{aligned}$$

where

$$x_k = \begin{bmatrix} x_k^c & (m) \\ y_k^c & (m) \\ \dot{x}_k^c & (m/s) \\ \dot{y}_k^c & (m/s) \end{bmatrix} \qquad A_k = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Q_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 \end{bmatrix}$$

$$y_k = \begin{bmatrix} x_k^{gps} & (m) \\ y_k^{gps} & (m) \end{bmatrix} \qquad H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad R_k = R \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

with Q_k and R_k being co-variance matrices for v_k and w_k respectively, and parameter values:

$$\Delta T = 1\ s \quad \sigma_x^2 = \sigma_y^2 = 0\ m^2 \quad \sigma_{\dot{x}}^2 = \sigma_{\dot{y}}^2 = 0.01\ m^2/s^2 \quad R = 10\ m^2.$$

Initially, the target state is $x_0 = [10000\ m, 10000\ m, 1\ m/s, 0\ m/s]'$. Using the Kalman filter, you must recursively estimate x_k from the measurements, for $1000s$, beginning with an initial state estimate $\hat{x}_0 = x_0 + [100\ m, 100\ m, 0.1\ m/s, -0.1\ m/s]'$ and initial error covariance $P_0 = 1000 I_{4\times4}$, where $I_{4\times4}$ is the 4×4 identity matrix. Note: P_0 involves units of m^2, $(m/s)^2$ and m^2/s.

Tasks:

The following tasks are to be completed in MATLAB:

1. Data generation: write MATLAB script code that generates both the state process and measurement data for 1000 time instants.
   (a) For $k = 1, \ldots,$ on the same graph, report a plot of the element components of the vector y_k and components of the vector x_k versus k. Hint: sometimes plotting individual vector components (such as x_k^{gps} and x_k^c versus k) is quite informative.
2. Kalman filter: write MATLAB script code that applies a Kalman filter to the generated measurement data.
   (a) More plots to report: For $k = 1, \ldots,$ on the same graph, plot element components of the vector $\hat{x}_k$ and components of the vector x_k. Also, on the same graph, plot components of the vector $(\hat{x}_k - x_k)^2$ and the diagonal components of P_k. Finally, plot components of K_k.
   (b) Calculate and report the RMS value:

$$RMS = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(\hat{x}_k - x_k)(\hat{x}_k - x_k)}, \qquad \text{where } T = 1000.$$

   In our Kalman filtering problem, this RMS value provides a measurement of average filtering performance.

3. Conduct a parametric study (you may wish to "seed" your random number generator so that you have the same noise sequence for each case):
   (a) Repeat the data generation and filtering steps for different values of parameters R, Q and P_0. Hint: try factors of 10. You should examine at least 3 values of each parameter (that is, at least 12 in total).
   (b) Record and report the results of your study in a table of the RMS values, P_{1000} values and K_{1000} values for the different parameter choices.

4. Report what you discovered, present your table, and answer the following questions:
   (a) In terms of RMS, P_{1000}, and K_{1000}, describe what happens for different choices of P_0, R and Q? In your answer, include a description of the impact on each of RMS, P_{1000}, and K_{1000} if P_0 increases. Similar, if R increases. Similar, if Q increases.
   (b) Describe what happens if incorrect parameters are used in the KF equations. That is, data is generated with (slightly?) different model matrices than those used in the filter implementation. This situation is called **model-mismatch** and very commonly occurs in realistic problems. Hint: You may need to run a number of extra cases having a model mismatch.

5. Explain the use of Kalman filters by answering the following (in 600 words or less, dot point preferred)
   (a) Give an example from previous units that you have studied where you could have used a Kalman filter. Why would this have been useful/better than what you did?
   (b) Consider a practical situation that has the possibility of model-mismatch (see the previous task for definition). Report what strategies might you use to manage, minimise or avoid the performance impact of model mismatch (this can include the addition of extra processing stages or implementation steps).

Task 3: Extended Kalman Filter

Now consider the navigation problem with the same linear state equation:

$$x_k = A_k x_{k-1} + v_k$$

where

$$x_k = \begin{bmatrix} x_k^c & (m) \\ y_k^c & (m) \\ \dot{x}_k^c & (m/s) \\ \dot{y}_k^c & (m/s) \end{bmatrix} \qquad A_k = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Q_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 \end{bmatrix}$$

with Q_k being co-variance matrix for v_k, and parameter values:

$$\Delta T = 1\ s \quad \sigma_x^2 = \sigma_y^2 = 0\ m^2 \quad \sigma_{\dot{x}}^2 = \sigma_{\dot{y}}^2 = 0.01\ m^2/s^2.$$

However, we will assume that we have non-linear radar range and bearing measurements described as:

$$y_k = h(x_k, k) + w_k = \begin{bmatrix} \sqrt{(x_k^c)^2 + (y_k^c)^2} \\ \tan^{-1}\left(\frac{y_k^c}{x_k^c}\right) \end{bmatrix} + w_k$$

with R_k being co-variance matrix for w_k respectively, and parameter values: $R = 10\ m^2$. Initially, the target state is $x_0 = [10000\ m, 10000\ m, 1\ m/s, 0\ m/s]'$.

Tasks:

The following tasks are to be completed in MATLAB:

1. Data generation: write MATLAB script code that generates both the state process and measurement data for 1000 time instants.
2. Extended Kalman filter: write MATLAB script code that applies an Extended Kalman filter to the generated measurement data.
3. Parameterise study. Design and conduct a parameter study that investigates the impact of initial values of $\hat{x}_0$ and P_0. This study should find cases where the filter is stable and cases where the filter diverges.

Task 4: Nonlinear Filter

For the signal model provided in Task 3, implement an alternative non-linear filter and compare the performance and behaviour with the EKF of the previous section. Unscented Kalman filter or particle filter are suggestions.