EE363 homework 5

1. *One-step ahead prediction of an autoregressive time series.* We consider the following autoregressive (AR) system

$$p_{t+1} = \alpha p_t + \beta p_{t-1} + \gamma p_{t-2} + w_t, \qquad y_t = p_t + v_t.$$

Here p is the (scalar) time series we are interested in, and y is the scalar measurement available to us. The process noise w is IID zero mean Gaussian, with variance 1. The sensor noise v is IID Gaussian with zero mean and variance 0.01. Our job is to estimate p_{t+1}, based on knowledge of $y_0, \ldots, y_t$. We will use the parameter values

$$\alpha = 2.4, \qquad \beta = -2.17, \qquad \gamma = 0.712.$$

(a) Find the steady state covariance matrix Σ_x of the state

$$x_t = \begin{bmatrix} p_t \\ p_{t-1} \\ p_{t-2} \end{bmatrix}.$$

(b) Run three simulations of the system, starting from statistical steady state. Plot p_t for each of your three simulations.

(c) Find the steady-state Kalman filter for the estimation problem, and simulate it with the three realizations found in part (b). Plot the one-step ahead prediction error for the three realizations.

(d) Find the variance of the prediction error, *i.e.*, $\mathbf{E}(\hat{p}_t - p_t)^2$. Verify that this is consistent with the performance you observed in part (c).

2. *Performance of Kalman filter when the system dynamics change.* We consider the Gauss-Markov system

$$x_{t+1} = Ax_t + w_t, \qquad y_t = Cx_t + v_t, \tag{1}$$

with v and w are zero mean, with covariance matrices $V > 0$ and $W \geq 0$, respectively. We'll call this system the *nominal system*.

We'll consider another Gauss-Markov sysem, which we call the *perturbed system*:

$$x_{t+1} = (A + \delta A)x_t + w_t, \qquad y_t = Cx_t + v_t, \tag{2}$$

where $\delta A \in \mathbf{R}^{n \times n}$. Here (for simplicity) C, V, and W are the same as for the nominal system; the only difference between the perturbed system and the nominal system is that the dynamics matrix is $A + \delta A$ instead of A.

In this problem we examine what happens when you design a Kalman filter for the nominal system (1), and use it for the perturbed system (2).

Let L denote the steady-state Kalman filter gain for the nominal system (1), *i.e.*, the steady-state Kalman filter for the nominal system is

$$\hat{x}_{t+1} = A\hat{x}_t + L(y_t - \hat{y}_t), \qquad \hat{y}_t = C\hat{x}_t. \tag{3}$$

(We'll assume that (C, A) is observable and (A, W) is controllable, so the steady-state Kalman filter gain L exists, is unique, and $A - LC$ is stable.)

Now suppose we use the filter (3) on the perturbed system (2).

We will consider the specific case

$$A = \begin{bmatrix} 1.8 & -1.4 & 0.5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \delta A = \begin{bmatrix} 0.1 & -0.2 & 0.1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$C = [1 \ 0 \ 0], \qquad W = I, \qquad V = 0.01.$$

(a) Find the steady-state value of $\mathbf{E}\,\|x_t\|^2$, for the nominal system, and also for the perturbed system.

(b) Find the steady-state value of $\mathbf{E}\,\|\hat{x}_t - x_t\|^2$, where x is the state of the perturbed system, and $\hat{x}$ is the state of the Kalman filter (designed for the nominal system). (In other words, find the steady-state mean square value of the one step ahead prediction error, using the Kalman filter designed for the nominal system, but with the perturbed system.)

Compare this to $\mathbf{Tr}\,\hat{\Sigma}$, where $\hat{\Sigma}$ is the steady-state one step ahead prediction error covariance, when the Kalman filter is run with the nominal system. ($\mathbf{Tr}\,\hat{\Sigma}$ gives the steady-state value of $\mathbf{E}\,\|\hat{x}_t - x_t\|^2$, when x evolves according to the nominal system.)

3. *Open-loop control.* We consider a linear dynamical system with n states and m inputs,

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad t = 0, 1, \ldots,$$

where w_t are IID $\mathcal{N}(0, \Sigma_w)$, and $x_0 \sim \mathcal{N}(0, \Sigma_0)$ is independent of all w_t. The objective is

$$J = \mathbf{E}\left(\sum_{t=0}^{N-1} \left(x_t^T Q x_t + u_t^T R u_t \right) + x_N^T Q_f x_N \right)$$

where $Q \geq 0$, $Q_f \geq 0$, and $R > 0$.

In the standard stochastic control setup, we choose u_t as a function of the current state x_t, so we have $u_t = \phi_t(x_t)$, $t = 0, \ldots, N-1$. In *open-loop control*, we choose u_t as a function of the initial state x_0 only, so we have $u_t = \psi_t(x_0)$, $t = 0, \ldots, N-1$. Thus, in open-loop control, we must commit to an input sequence at time $t = 0$, based only on knowledge of the initial state x_0; in particular, there is no opportunity for recourse or changes in the input due to new observations. The open loop control problem is to choose the control functions $\psi_0, \ldots, \psi_{N-1}$ that minimize the objective J.

In this exercise, you will derive explicit expressions for the optimal control functions $\psi_0^\star, \ldots, \psi_{N-1}^\star$, for the open-loop control problem. The problem data are A, B, Σ_w, Q, Q_f, and R, and N.

Show that the optimal control functions are $\psi_0^\star(x_0) = K_0 x_0$, and

$$\psi_t^\star(x_0) = K_t(A + BK_{t-1}) \cdots (A + BK_0)x_0, \quad t = 1, \ldots, N-1,$$

where

$$K_t = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A, \quad t = 0, \ldots, N-1,$$

and

$$P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A, \quad t = N-1, \ldots, 0,$$

with $P_N = Q_f$. In other words, we can solve the open-loop control problem by solving the deterministic LQR problem obtained by taking $w_0 = w_1 = \cdots = w_{N-1} = 0$.

4. *Simulation of a Gauss-Markov system from statistical steady-state.* We consider a Gauss-Markov system,

$$x_{t+1} = Ax_t + w_t,$$

where $A \in \mathbf{R}^{n \times n}$ is stable (*i.e.*, its eigenvalues all have magnitude less than one), w_t are IID with $w_t \sim \mathcal{N}(0, W)$, and $x_0 \sim \mathcal{N}(0, \Sigma_0)$, independent of all w_t. Let Σ_x denote the asymptotic value of the state covariance. If $x_0 \sim \mathcal{N}(0, \Sigma_x)$ (*i.e.*, $\Sigma_0 = \Sigma_x$), then we have $\mathbf{E}\, x_t = 0$ and $\mathbf{E}\, x_t x_t^T = \Sigma_t$ for all t. We refer to this as statistical equilibrium, or statistical steady-state.

Generate a random $A \in \mathbf{R}^{10 \times 10}$ in Matlab using `A = randn(n)`, then scaling it so its spectral radius (maximum magnitude of all eigenvalues) is 0.99. Choose W to be a random positive semidefinite matrix, for example using `W = randn(n); W = W'*W;`. Create two sets of 50 trajectories for 100 time steps; in one set, initialize with $x_0 = 0$, in the other, with $x_0 \sim \mathcal{N}(0, \Sigma_x)$.

Create two plots, overlaying the trajectories of $(x_t)_1$ within each set. Comment briefly on what you see.

5. *Implementing a Kalman filter.* In this problem you will implement a simple Kalman filter for a linear Gauss-Markov system

$$x_{t+1} = Ax_t + w_t, \quad y_t = Cx_t + v_t$$

with $x_0 \sim \mathcal{N}(0, I)$, $w_t \sim \mathcal{N}(0, W)$ and $v_t \sim \mathcal{N}(0, V)$.

Generate a system in Matlab by randomly generating a matrix $A \in \mathbf{R}^{10\times 10}$ and scaling it so its spectral radius is 0.95, a matrix $C \in \mathbf{R}^{3\times 10}$, and positive definite matrices W and V. Find the Kalman filter for this system.

Plot $\sqrt{\mathbf{E}\,\|x_t\|^2}$ and $\sqrt{\mathbf{E}\,\|x_t - \hat{x}_t\|^2}$, for $t = 1, \ldots, 50$. Then, simulate the system for 50 time steps, plotting $\|x_t\|_2$ and $\|x_t - \hat{x}_t\|_2$.

6. *Simultaneous sensor selection and state estimation.* We consider a standard state estimation setup:

$$x_{t+1} = Ax_t + w_t, \qquad y_t = C_t x_t + v_t,$$

where $A \in \mathbf{R}^{n\times n}$ is constant, but C_t can vary with time. The process and measurement noise are independent of each other and the initial state $x(0)$, with

$$x(0) \sim \mathcal{N}(0, \Sigma_0), \qquad w_t \sim \mathcal{N}(0, W), \qquad v_t \sim \mathcal{N}(0, V).$$

The standard formulas for the Kalman filter allow you to compute the next state prediction $\hat{x}_{t|t-1}$, current state prediction $\hat{x}_{t|t}$, and the associated prediction error covariances $\Sigma_{t|t-1}$ and $\Sigma_{t|t}$.

Now we are going to introduce a twist. The measurement matrix C_t is one of K possible values, *i.e.*, $C_t \in \{C_1, \ldots, C_K\}$. In other words, at each time t, we have $C_t = C_{i_t}$. The sequence i_t specifies which of the K possible measurements is taken at time t. For example, the sequence $2, 2, \ldots$ means that $C_t = C_2$ for all t; the sequence

$$1, 2, \ldots, K,\ 1, 2 \ldots, K,\ \ldots$$

is called *round-robbin*: we cycle through the possible measurements, in order, over and over again.

Here's the interesting part: *you* get to choose the measurement sequence $i_0, i_1, \ldots,$. You will use the following greedy algorithm. You will choose the sequence in order; having chosen $i_0, \ldots, i_{t-1}$, you will choose i_t so as to minimize the mean-square prediction error associated with $\hat{x}_{t|t}$. This is the same as choosing i_t so that $\mathbf{Tr}\,\Sigma_{t|t}$ is minimized. Roughly speaking, at each step, you choose the sensor that results in the smallest mean-square state prediction error, given the sensor choices you've made so far, plus the one you're choosing.

Let's be very clear about this method for choosing i_t. The choice of $i_0, \ldots, i_{t-1}$ determines $\Sigma_{t|t-1}$; then, $\Sigma_{t|t}$ depends on i_t, *i.e.*, which of $C_1, \ldots, C_K$ is chosen as C_t. Among these K choices, you pick the one that minimizes $\mathbf{Tr}\,\Sigma_{t|t}$.

This method does not require knowledge of the actual measurements $y_0, y_1, \ldots$, so we can determine the sequence of measurements we are going to make *before any data have been received.* In particular, the sequence can be determined ahead of time (at least up to some large value of t), and stored in a file.

Now we get to the question. You will work with the specific system with

$$A = \begin{bmatrix} -0.6 & 0.8 & 0.5 \\ -0.1 & 1.5 & -1.1 \\ 1.1 & 0.4 & -0.2 \end{bmatrix}, \qquad W = I, \qquad V = 0.1^2, \qquad \Sigma_0 = I,$$

and $K = 3$ with

$$C_1 = \begin{bmatrix} 0.74 & -0.21 & -0.64 \end{bmatrix}, \qquad C_2 = \begin{bmatrix} 0.37 & 0.86 & 0.37 \end{bmatrix}, \qquad C_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

(a) *Using one sensor.* Plot the mean-square current state prediction error $\mathbf{Tr}\,\Sigma(t|t)$ versus t, for the three special cases when $C_t = C_1$ for all t, $C_t = C_2$ for all t, and $C_t = C_3$ for all t.

(b) *Round-robbin.* Plot the mean-square current state prediction error $\mathbf{Tr}\,\Sigma(t|t)$ versus t, using sensor sequence $1, 2, 3, 1, 2, 3, \ldots$

(c) *Greedy sensor selection.* Find the specific sensor sequence generated by the algorithm described above. Show us the sequence, by plotting i_t versus t. Plot the resulting mean-square estimation error, $\mathbf{Tr}\,\Sigma_{t|t}$, versus t. Briefly compare the results to what you found in parts (a) and (b).

In all three parts, you can show the plots over the interval $t = 0, \ldots, 50$.

To save you some time, we have created the file `sens_data.m`, which contains the problem data. The file also contains two lines, currently commented out, that implement a generic Kalman filter measurement and time update. You're welcome to use these, or to use or write your own.