Homework 3

The purpose of this problem set is to get hands-on experience implementing the following: discrete-time Wiener filtering, and Kalman filter (KF) smoothing.

P3.1 – (adapted from Si.3.14.) **Discrete-time Wiener filters.** In this problem, we reconsider the simple examples given in class (way back, around Lecture 7), where we solved for optimal parameterized, non-causal, and causal Wiener filters for a Gauss-Markov signal corrupted by white noise. In class, we discussed the continuous-time Wiener filter. Here, you will compare the results from class (and also in the text) with your own implementation of a "discrete Wiener filter", where we only have access to samples from the actual (continuous) process. **(50 pts)**

For this problem, we will assume we are measuring the output of a continuous-time first-order Gauss-Markov noise process with the following power spectral density (PSD) function:

$$S_x(\omega) = \frac{2\sigma^2\beta}{\omega^2+\beta^2} = \frac{2}{\omega^2+1}$$

This PSD intentionally matches the CT examples for Wiener filtering from class (Lecture 7), from B&H (p. 159-183), and also from Simon (p. 94-101), i.e., where $\sigma^2 = \beta = 1$. Data are collected with a sampling rate $\Delta t = .02$, and a total of $n = 101$ sequential measurements, y_k, are to be processed. Each discrete sensor measurement is corrupted by unity white noise: $v_k \sim (0, ?)$.

Your tasks are to find the causal and non-causal discrete-time (DT) Wiener filters for this dynamic system and to compare these results to the optimal continuous-time (CT) Wiener filters derived in class.

a) Causal filter. Here, our goal is to find the set of weighting coefficients for a finite set of sampled (past) data, that produces the optimal (MMSE) estimate for x_n:

$$\hat{x}_n = k_1 y_1 + k_2 y_2 + \cdots + k_n y_n$$

From Section 4.7 in Brown and Hwang (pp. 181-183), we know the optimal causal Wiener filter weights, k_n, for discretely sampled data will solve the following (B&H Eq. 4.7.4):

$$\begin{bmatrix} E(y_1^2) & E(y_1y_2) & \cdots & \cdots \\ E(y_2y_1) & \ddots & & \\ \vdots & & \ddots & \\ E(y_ny_1) & E(y_ny_2) & \cdots & E(y_n^2) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} E(y_1x_n) \\ E(y_2x_n) \\ \vdots \\ E(y_nx_n) \end{bmatrix}$$

(Variable names from 4.7.4 are modified here to match class notation…) We can abbreviate this matrix equation as: $Mk_{opt} = b$. In MATLAB, we can solve this using: `kopt = M\b`

i) Provide analytic expressions for M(i,j) (in matrix M) and b(i) for our particular DT system.
ii) Use MATLAB to solve for kopt, and plot the result versus an appropriately-scaled CT Wiener filter. Recall for the causal case, the optimal continuous-time Wiener filter was:

$$g(t) = \left(\sqrt{3}-1\right)e^{-\sqrt{3}t},\ t \geq 0 \quad ; \quad g(t) = 0,\ t < 0$$

(Hint: In the limit, for what scaling factor, f, would kopt and be expected to approach f*g(t)? See also the APPENDIX on last page, which included a different problem statement from Simon…)

b) Repeat part **a)** for the non-causal filtering case. Here, we wish to estimate what the value of x at the "mid-point" of the 101 data points (i.e., at time step 51). You will need to rewrite Eq. B&H 4.7.4 appropriately. Recall for the non-causal case, the optimal continuous-time filter was:

$$g(t) = \frac{1}{\sqrt{3}} e^{-\sqrt{3}t},\ t \geq 0 \quad ; \quad g(t) = \frac{1}{\sqrt{3}} e^{+\sqrt{3}t},\ t < 0$$

Some additional notes on the problem follow. You shouldn't need these details to solve the problem. (Think about the known autocorrelation for the CT process…) However, it is important and quite useful to understand how to model a continuous-time process that is sampled discretely.

For our example, the CT equation of motion for the process is:

$$\dot{x}(t) = Ax(t) + Bu(t) = -\beta x(t) + \sqrt{2\sigma^2\beta}u(t)$$

where $u(t)$ is unity Gaussian white noise, and so: $A = -\beta = -1$, $B = \sqrt{Q_c} = \sqrt{2\sigma^2\beta} = \sqrt{2}$.

Our measurements, however, are taken only at discrete intervals, with sampling time $\Delta t = .02$ seconds. The following equations allow us to simulate this CONTINUOUS process as a DISCRETE one:

$$x_k = Fx_{k-1} + (w_{DT})_{k-1}$$
$$y_k = x_k + v_k$$

where (as is standard when converting a constant-coefficient CT "*A*" matrix to a sampled DT "*F*" matrix) $F = e^{A\Delta t}$, and w_{DT} is white, Gaussian noise: $(w_{DT})_k \sim (0, \sigma^2(1 - e^{-2\beta\Delta t}))$, which for small "$\beta\Delta t$" is approximated accurately as $(w_{DT})_k \sim (0, \sigma^2(2\beta\Delta t))$. As previously mentioned, assume unity Gaussian measurement noise for the discrete samples: $v_k \sim (0,1)$.

P3.2 – KF Smoothing – the Rauch, Tung, and Striebel (RTS) algorithm. (50 pts) Here, we will model the dynamics of a second-order spring-mass-damper system with natural frequency $\omega_o = 1$ (rad/s), and a damping ratio $\zeta = 0.1$; the mass is 1 kg. Both states (position and velocity) are to be estimated, only process noise drives the system (no "u"), and we have an independent (but noisy) measurement of each state. Using $\Delta t = 0.1$ seconds, create a reasonable discrete-time version of this system (for instance, using the c2d command in MATLAB).

a) What are the F (dynamics) and H (measurement) matrices you are using for the DT system?

Our end goal is to find the optimal (MMSE: minimum mean squared error) estimate of the state (position and velocity; this is a second-order system) at a fixed lag: that is, we want the estimate, $\hat{x}_k$, for a fixed number of time steps, *N*, before our last measurement was taken. Given there are $n + N$ total measurements, that means we want $\hat{x}_k$ for $k = n$.

To solve this, use the RTS algorithm described in class and also outlined in Simon (pp. 279-294) and Brown and Hwang (pp. 312-322). This involves a forward pass which is IDENTICAL to the standard KF algorithm, and a backward pass that efficiently combines the usual forward dynamics estimate (forward pass) with a "backward dynamics" estimate.

Say we are interested in the state at time step *n*, and the last available measurement is at time step *n+N* (*N*>0). Initialize for k=0 (a posteriori), and then for k=1:1:(n+N), perform the usual KF steps, which can be written (as always) as:

1. $\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+$
2. $P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1}$
3. $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
4. $P_k^+ = (I - K_k H_k)P_k^-(I - K_k H_k)^T + K_k R_k K_k^T$
5. $\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)$

Then, perform a loop backward in time for the RTS estimate. Initialize the RTS estimate by using the state estimate and covariance calculations from the final forward step (k=n+N).

$$(\hat{x}_{RTS})_{n+N} = \left(\hat{x}_f^+\right)_{n+N}$$
$$(P_{RTS})_{n+N} = \left(P_f^+\right)_{n+N}$$

For clarity, we will use the subscript "f" for values from the forward (normal KF) pass and the subscript "RTS" for the new values in the loop below. Then, for k=(n+N-1):-1:n,

1. $(K_{RTS})_k = \left(P_f^+\right)_k (F^T)_k \left(P_f^-\right)_{k+1}^{-1}$
2. $(P_{RTS})_k = \left(P_f^+\right)_k + (K_{RTS})_k \left((P_{RTS})_{k+1} - \left(P_f^-\right)_{k+1}\right)(K_{RTS}^T)_k$
3. $(\hat{x}_{RTS})_k = \left(\hat{x}_f^+\right)_k + (K_{RTS})_k \left((\hat{x}_{RTS})_{k+1} - \left(\hat{x}_f^-\right)_{k+1}\right)$

b) Write a MATLAB script to simulate the noisy dynamic process, and also to generate noisy measurement data. To determine Q and R for your DT system, assume an impulse occurs at each time step, and that its magnitude is selected from a unity variance Gaussian distribution. (Thus, the process noise acts directly on the velocity, to change its value at each time step. Q will then be a 2x2 matrix with only one non-zero element.) Assume R=diag([10,10]).

c) Implement the RTS smoothing algorithm. Simulate $n = 200$ data points. Create two plots: one for position over time, and one for velocity. On each plot, include the following:

i) The actual state
ii) The measurement data
iii) The forward Kalman estimate
iv) The RTS Kalman estimate (plot all 4 data sets for the entire period, k=1 to 200).

d) Now, consider the estimate at a particular time step, k=180. (This is a lag of N=20, wrt n=200 total points.) What is the a posteriori covariance of the standard (forward) KF estimate at this point, $\left(P_f^+\right)_{180}$? And what is the covariance matrix for the smoothed RTS estimate, $(P_{RTS}^+)_{180}$?

e) Recall from lecture that the RTS algorithms is equivalent to producing two, independent estimates of the state at any given time step, k – one forward (subscripted f), and one running the dynamics and data backward in time (subscripted b) – and then combining these two estimates. Given this, what must the mathematical relationship be among $\left(P_f^+\right)_{n-N}$, $(P_{RTS}^+)_{n-N}$, and $(P_b^-)_{n-N}$? Why does this relationship hold for $(P_b^-)_{n-N}$ rather than for $(P_b^+)_{n-N}$? Again considering k=180, and using your answers from part d, what would the a priori covariance be for a "backward only" KF filter for time step k=180?

References

[BH] Brown, R.G., and Hwang, P.Y.C. Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises and Solutions, 3rd ed. Wiley, 1997.
[Si] Simon, D. Optimal State Estimation: Kalman, H-inf, and Nonlinear Approaches. Wiley, 2006.

In particular, see pp. 181-182 in B&H for Problem 3.1. (Distributed in class, Monday Nov. 22.)

APPENDIX:

Also of use may be this Problem statement from Simon (for Problem 3.1):

3.14 Implement the Wiener filters for the three examples given in Section 3.4 and verify the results shown in Section 3.4.5. Hint: Example 8.6 shows that if $\dot{x} = -x + w$ where $w(t)$ is white noise with a variance of $Q_c = 2$, then

$$S_x(\omega) = \frac{2}{\omega^2 + 1}$$

From Sections 1.4 and 8.1 we see that this system can be simulated as

$$\begin{aligned} x(t + \Delta t) &= e^{-\Delta t} x(t) + w(t)\sqrt{Q_c \Delta t} \\ y(t) &= x(t) + v(t)\sqrt{R_c / \Delta t} \end{aligned}$$

where $w(t)$ and $v(t)$ are independent zero-mean, unity variance random variables.