ASE 381P Problem Set #5

Posting Date: October 24, 2019

You need not hand in anything. Instead, be prepared to answer any of these problems on an upcoming take-home exam. You may discuss your solutions with classmates up until the time that the exam becomes available, but do not swap work.

1. This problem will walk you through a derivation of the Kalman filter similar to the maximum *a posteriori* (MAP)-based derivation discussed in lecture but without explicitly minimizing the cost function with respect to $\boldsymbol{x}(k+1)$. Along the way, you'll get plenty of exercise manipulating the matrices involved in Kalman filtering.

   Suppose that one defines the following weighted least-squares cost function for purposes of developing a MAP implementation of the Kalman filter:

$$\begin{aligned} J_a[\boldsymbol{x}(k), \boldsymbol{v}(k), \boldsymbol{x}(k+1), k] = \tfrac{1}{2}\left[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)\right]^T P^{-1}(k)\left[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)\right] + \tfrac{1}{2}\boldsymbol{v}^T(k)Q^{-1}(k)\boldsymbol{v}(k) \\ + \tfrac{1}{2}\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}(k+1)\right]^T R^{-1}(k+1)\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}(k+1)\right] \end{aligned}$$

   Next, suppose one uses the dynamics model

$$\boldsymbol{x}(k+1) = F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) + \Gamma(k)\boldsymbol{v}(k)$$

   and the state estimate propagation equation

$$\bar{\boldsymbol{x}}(k+1) = F(k)\hat{\boldsymbol{x}}(k) + G(k)\boldsymbol{u}(k)$$

   to eliminate $\boldsymbol{x}(k)$ and $\hat{\boldsymbol{x}}(k)$ from the cost function and thereby define

$$\begin{aligned} J_b[\boldsymbol{v}(k), \boldsymbol{x}(k+1), k] =& J_a[F^{-1}(k)\{\boldsymbol{x}(k+1) - G(k)\boldsymbol{u}(k) - \Gamma(k)\boldsymbol{v}(k)\}, \boldsymbol{v}(k), \boldsymbol{x}(k+1), k] \\ =& \tfrac{1}{2}\left[F^{-1}(k)\{\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1) - \Gamma(k)\boldsymbol{v}(k)\}\right]^T P^{-1}(k)\left[\sim\right] \\ & + \tfrac{1}{2}\boldsymbol{v}^T(k)Q^{-1}(k)\boldsymbol{v}(k) \\ & + \tfrac{1}{2}\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}(k+1)\right]^T R^{-1}(k+1)\left[\sim\right] \end{aligned}$$

   where the abbreviation $[\boldsymbol{q}]^T A[\sim]$ is used to shorten the quadratic form $[\boldsymbol{q}]^T A[\boldsymbol{q}]$. Next, suppose that one partially optimizes $J_b[\cdot,\cdot,\cdot]$ with respect to $\boldsymbol{v}(k)$, making use of the covariance propagation equation

$$\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k)$$

   to arrive at the formula

$$\boldsymbol{v}(k) = Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)\left[\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)\right]$$

   Next, suppose that one uses this formula to eliminate $\boldsymbol{v}(k)$ from $J_b[\cdot,\cdot,\cdot]$, thereby effectively defining a new cost function

$$J_c[\boldsymbol{x}(k+1), k+1] = J_b[Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)\{\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)\}, \boldsymbol{x}(k+1), k]$$

Prove that $J_c[\boldsymbol{x}(k+1), k+1]$ can be expressed in the following form

$$\begin{aligned} J_c[\boldsymbol{x}(k+1), k+1] = & \tfrac{1}{2}\left[\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)\right]^T \bar{P}^{-1}(k+1)\,[\sim] \\ & + \tfrac{1}{2}\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}(k+1)\right]^T R^{-1}(k+1)\,[\sim] \end{aligned}$$

Offer an interpretation of the two terms in this form. Then, starting from this form of $J_c[\boldsymbol{x}(k+1), k+1]$, prove that

$$\begin{aligned} J_c[\boldsymbol{x}(k+1), k+1] = & \tfrac{1}{2}\left[\boldsymbol{x}(k+1) - \hat{\boldsymbol{x}}(k+1)\right]^T P^{-1}(k+1)\left[\boldsymbol{x}(k+1) - \hat{\boldsymbol{x}}(k+1)\right] \\ & + J_b[\hat{\boldsymbol{v}}(k), \hat{\boldsymbol{x}}(k+1), k] \end{aligned}$$

where

$$\hat{\boldsymbol{v}}(k) = Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)\left[\hat{\boldsymbol{x}}(k+1) - \bar{\boldsymbol{x}}(k+1)\right]$$

Offer an interpretation of the two terms in this form.

Assume that all undefined matrices and vectors in this problem formulation are the same as have been defined in lecture for the Kalman filter problem.

Hints: Both forms of $J_c[\boldsymbol{x}(k+1), k+1]$ are quadratic in the argument $\boldsymbol{x}(k+1)$. Equality of the two forms of can be proved by proving that the respective coefficients of the constant, linear, and quadratic $\boldsymbol{x}(k+1)$ terms are equal. To equate the constant and linear coefficients, leave $\hat{\boldsymbol{x}}(k+1)$ as a symbol. To show equivalence of the linear term, it's enough to show that the required $\hat{\boldsymbol{x}}(k+1)$ is equal to an expression for $\hat{\boldsymbol{x}}(k+1)$ derived in the MAP-based KF derivation lecture notes. You may need to try several of the alternate equivalent formulas for $P(k+1)$ in order to find the one that is most useful for purposes of this proof. Feel free to drop the k and $k+1$ indices when there is no danger of confusion. You may wish to write your solution on unlined paper in "landscape" orientation so that you have enough space to write lengthy expressions on one line, which helps readability.

2. Using the same notation and definitions as in Problem 1, prove that

$$J_c[\hat{\boldsymbol{x}}(k+1), k+1] = J_b[\hat{\boldsymbol{v}}(k), \hat{\boldsymbol{x}}(k+1), k] = \tfrac{1}{2}\boldsymbol{\nu}^T(k+1)S^{-1}(k+1)\boldsymbol{\nu}(k+1)$$

where $\boldsymbol{\nu}(k+1)$ is the Kalman filter innovation at time $k+1$ and $S(k+1)$ is its *a priori* covariance.

Hints: Rewrite the terms in $J_b[\hat{\boldsymbol{v}}(k), \hat{\boldsymbol{x}}(k+1), k]$ in terms of the innovation $\boldsymbol{\nu}(k+1)$. Note that the innovation is different from the process noise $\boldsymbol{v}(k)$. You should be able to reduce the cost expression to a quadratic form in $\boldsymbol{\nu}(k+1)$. It may help to use the definition of J_b in terms of J_a and to first express the terms $[\hat{\boldsymbol{x}}(k|k+1) - \hat{\boldsymbol{x}}(k)]$, $\hat{\boldsymbol{v}}(k)$, and $[\boldsymbol{z}(k+1) - H(k+1)\hat{\boldsymbol{x}}(k+1)]$ as simple linear functions of $\boldsymbol{\nu}(k+1)$. Note that $\hat{\boldsymbol{x}}(k|k{+}1) = F^{-1}(k)\{\hat{\boldsymbol{x}}(k{+}1) - G(k)\boldsymbol{u}(k) - \Gamma(k)\hat{\boldsymbol{v}}(k)\}$. After doing this, it will be necessary to use matrix manipulations in order to prove that the quadratic form's weighting matrix is $S^{-1}(k+1)$. Recall that $S(k+1)$ has been defined in lecture and in Bar-Shalom. The final matrix manipulations will be easier if you first simplify your expressions for $[\hat{\boldsymbol{x}}(k|k+1) - \hat{\boldsymbol{x}}(k)]$, $\hat{\boldsymbol{v}}(k)$, and $[\boldsymbol{z}(k+1) - H(k+1)\hat{\boldsymbol{x}}(k+1)]$ as much as possible.

3. Implement a Kalman filter for a stochastic linear time invariant (SLTI) system in the standard form used in class (with $\Gamma(k) \neq I$). The problem matrices and the measurement data, $z(k)$ for $k = 1, ..., 50$, can be loaded into your Matlab workspace by running the Matlab script `kf_example02a.m`. Hand in plots of the two elements of $\hat{\boldsymbol{x}}(k)$ vs. time and of the predicted standard deviations of $\hat{\boldsymbol{x}}(k)$ vs. time, i.e., of $\sqrt{[P(k)]_{11}}$ and $\sqrt{[P(k)]_{22}}$. Plot each element of $\hat{\boldsymbol{x}}(k)$ and its corresponding standard deviation together on the same graph. Use symbols on the plot at each of the 51 points and do not connect the symbols by lines (type "help plot" in order to learn how to do this). Also, hand in numerical values for the terminal values of $\hat{\boldsymbol{x}}(50)$ and $P(50)$.

4. Use `kalman.m`, as shown in the lecture notes, to calculate the steady-state Kalman filter gain, W_{ss}, and the steady-state *a posteriori* state estimation error covariance matrix, P_{ss}, for the system of Problem 3. Show that the time-varying *a posteriori* covariance matrix time history that you computed in Problem 3 converges to the steady-state covariance matrix that you computed in this part. Also, show that the steady-state value of the error transition matrix, $(I - W_{ss}H)F$, is stable, i.e., that all of its eigenvalues have complex magnitudes less than 1.

5. Repeat Problem 3, except use the problem matrices and measurement data that are defined by the Matlab script `kf_example02b.m`. Notice that the R and Q values are different for this problem and that there is a different measurement time history. Run your Kalman filter two additional times using the two alternate Q values that are mentioned in the comments in the file `kf_example02b.m`. It is uncertain which is the correct Q value. Decide which is the best value in the following way: Calculate $\epsilon_\nu(k)$ for $k = 1, 2, ..., 50$ for each of your runs. Compute the average of these 50 values. This average times 50, i.e., $\{\epsilon_\nu(1) + \epsilon_\nu(2) + ... + \epsilon_\nu(50)\}$, will be a sample of chi-square distribution of degree 50 if the filter model is correct. Develop upper and lower limits between which the average $\{\epsilon_\nu(1) + \epsilon_\nu(2) + ... + \epsilon_\nu(50)\}/50$ must lie 99% of the time if the Kalman filter model is correct, and test your averages for each of the three candidate Q values. Which is the most reasonable? Look at the state estimate differences between the best filter and the other two filters. Compute the RMS value of the difference time history for each state vector element. Do the averaging over the last 40 points. Are these differences significant compared to the computed state estimation error standard deviations for the best filter?

6. Develop a truth model simulation of a stochastic linear system and use it to test the consistency of a Kalman filter by doing Monte Carlo runs. Your truth model simulation should adhere to the following interface, which you can cut-and-paste from the pdf into Matlab:

```
function [xhist,zhist] = mcltisim(F,Gamma,H,Q,R,xbar0,P0,kmax)
% ltisim : Monte-Carlo simulation of a linear time invariant system.
%
%
% Performs a truth-model Monte-Carlo simulation for the discrete-time
% stochastic system model:
```

```
%
%     x(k+1) = F*x(k) + Gamma*v(k)
%     z(k)   = H*x(k) + w(k)
%
% Where v(k) and w(k) are uncorrelated, zero-mean, white-noise Gaussian random
% processes with covariances E[v(k)*v(k)'] = Q and E[w(k)*w(k)'] = R.  The
% simulation starts from an initial x(0) that is drawn from a Gaussian
% distribution with mean xbar0 and covariance P0.  The simulation starts at
% time k = 0 and runs until time k = kmax.
%
%
% INPUTS
%
% F ----------- nx-by-nx state transition matrix
%
% Gamma ------- nx-by-nv process noise gain matrix
%
% H ----------- nz-by-nx measurement sensitivity matrix
%
% Q ----------- nv-by-nv symmetric positive definite process noise covariance
%               matrix.
%
% R ----------- nz-by-nz symmetric positive definite measurement noise
%               covariance matrix.
%
% xbar0 ------- nx-by-1 mean of probability distribution for initial state
%
% P0 ---------- nx-by-nx symmetric positive definite covariance matrix
%               associated with the probability distribution of the initial
%               state.
%
% kmax -------- Maximum discrete-time index of the simulation
%
%
% OUTPUTS
%
% xhist ------- (kmax+1)-by-nx matrix whose kth row is equal to x(k-1)'. Thus,
%               xhist = [x(0), x(1), ..., x(kmax)]'.
%
% zhist ------- kmax-by-nz matrix whose kth row is equal to z(k)'. Thus, zhist
%               = [z(1), z(2), ..., z(kmax)]. Note that the state vector
%               xhist(k+1,:)' and the measurement vector zhist(k,:)'
%               correspond to the same time.
%
%+------------------------------------------------------------------------------+
% References:
%
%
% Author:
%+==============================================================================+
```

Use this truth model simulation and a Kalman filter to do a Monte Carlo study of the consistency of the Kalman filter. Test the Kalman filter's consistency on the Kalman filtering problem described in Problem 3. Use Monte Carlo techniques to calculate $E[\tilde{\boldsymbol{x}}(10)]$, $E[\tilde{\boldsymbol{x}}(10)\tilde{\boldsymbol{x}}^T(10)]$, $E[\tilde{\boldsymbol{x}}(35)]$, and $E[\tilde{\boldsymbol{x}}(35)\tilde{\boldsymbol{x}}^T(35)]$. Show that $E[\tilde{\boldsymbol{x}}(10)]$ and $E[\tilde{\boldsymbol{x}}(35)]$ approach zero as the number of Monte Carlo runs gets to be large. Recall that $\tilde{\boldsymbol{x}}(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)$. Show that $E[\tilde{\boldsymbol{x}}(10)\tilde{\boldsymbol{x}}^T(10)]$ approaches $P(10)$ and that $E[\tilde{\boldsymbol{x}}(35)\tilde{\boldsymbol{x}}^T(35)]$ approaches $P(35)$ as the number of Monte Carlo runs gets to be large. Do one Monte Carlo run that uses 50 simulations and do another that uses 1000 simulations and compare your results. As part of the record of your work include your calculated values for $E[\tilde{\boldsymbol{x}}(10)]$, $E[\tilde{\boldsymbol{x}}(10)\tilde{\boldsymbol{x}}^T(10)]$, $E[\tilde{\boldsymbol{x}}(35)]$, and $E[\tilde{\boldsymbol{x}}(35)\tilde{\boldsymbol{x}}^T(35)]$ and any code that you used to perform this study.

Hints: You will have to use the Cholesky factorization function `chol` to generate your initial truth state and the truth process and measurement noise vectors. You will also have to use the Gaussian random number generation function, `randn`, which samples a Gaussian distribution with a mean equal to zero and covariance equal to the identity matrix. This function's outputs are uncorrelated from call to call. You can check whether you have used `chol` and `randn` to correctly define the various error and noise vectors by re-computing their covariances based on your knowledge of how `chol` and `randn` work.

Also do the following problems from Bar Shalom: 5-1.1 (Hint: first calculate the steady-state *a priori* state error covariance), 5-5.1, 5-10, 5-11, 5-12 (Consider observability for part 2. A significant difference between problems 5-11 and 5-12 is that the bias in the measurements is assumed to be known in 5-11, but in 5-12 it is unknown and must be estimated.), and 5-16.