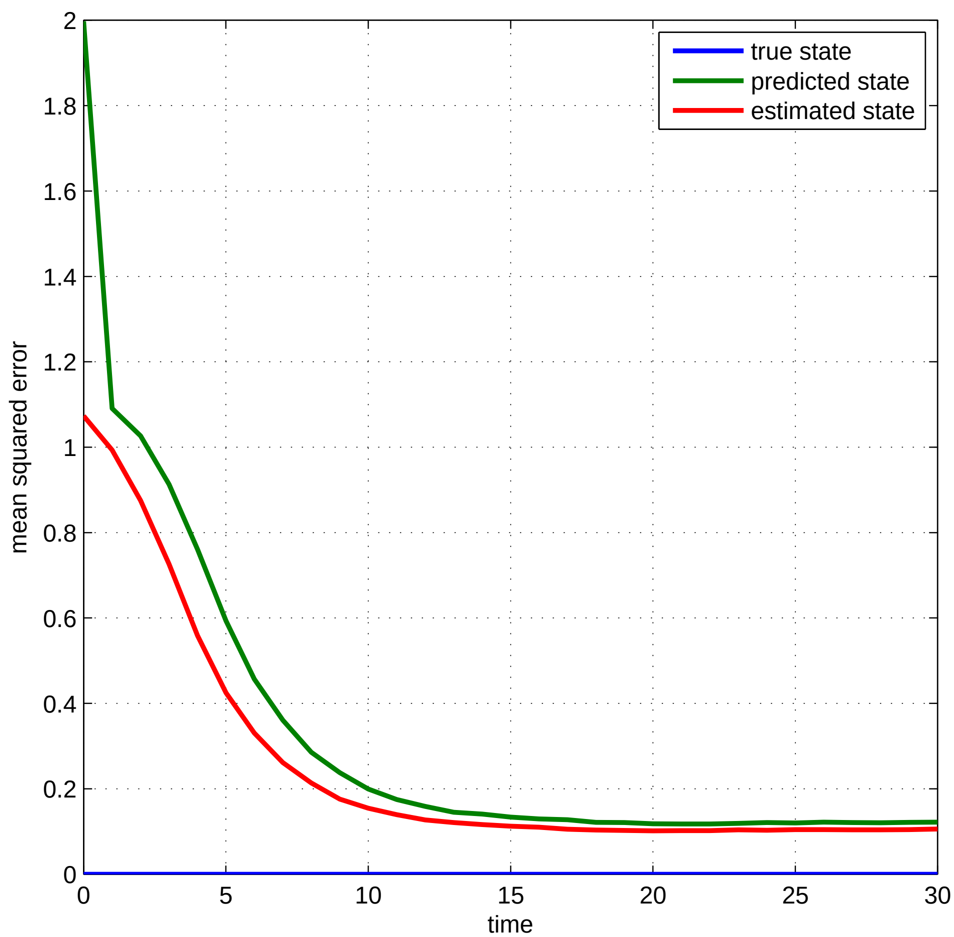
ECE531 Homework Assignment Number 10

Solution

Make sure your reasoning and work are clear to receive full credit for each problem.

1. 8 points. In Matlab, write a Kalman filter estimator of the one-dimensional motion state $X[n]$ given observations $Y[0], \dots, Y[n]$ under the same assumptions as homework assignment 9 except that $T = 0.1$ and you now have measurement noise $V[n] \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.1)$. Note that the variance of the measurement noise is equal to 0.1 here, not the standard deviation. Run several iterations of your Kalman filter and plot its mean-squared error performance as a function of n.

   Solution: The MSE plot for a simulation with 5000 runs is shown below.



The Matlab code used to obtain this plot is given below.

```
% Simulation to compute Mean Squared Error (MSE) performance of Kalman
% filter tracking the position and velocity of a particle moving in
% one-dimensional motion. See ECE531 lectures 10-11 slides for details on
% the dynamical model and performance plots. Note that this may take a
% minute to run on a slower computer.
%
% DRB 13-Apr-2009
%
% ---------------------------------------------------------------------
% USER PARAMETERS BELOW
% ---------------------------------------------------------------------
iterations = 5000;
N = 31;                              % number of states to generate (n=0,dots,N-1)
m0 = zeros(2,1);                     % mean of initial state
S0 = eye(2);                         % covariance matrix of initial state
T = 0.1;                             % sampling time
Q = 1;                               % covariance of input sequence
R = 0.1;                             % covariance of measurement noise
% ---------------------------------------------------------------------

% One dimensional motion dynamical model
F = [1,T;0,1];                       % state update matrix
G = [0 ; T];                         % input matrix
H = [1 0];                           % output matrix
Ep = zeros(iterations,N);            % squared prediction error
Ee = zeros(iterations,N);            % squared estimation error

for i = 1:iterations

    U = chol(Q)*randn(size(G,2),N);  % generate random input sequence
    V = chol(R)*randn(size(H,1),N);  % generate random measurement noise

    % allocate space for states and outputs
    X = zeros(size(F,1),N);
    Y = zeros(size(H,1),N);

    % allocate space for predictions and estimates
    Xp = zeros(size(F,1),N);         % predictions
    Xe = zeros(size(F,1),N);         % estimates

    % initial state (use Cholesky factorization to get desired covariance)
    X(:,1) = chol(S0)*randn(size(F,1),1)+m0;

    % initialize kalman filter
    Xp(:,1) = m0;
    Sp = S0;
    Ep(i,1) = (X(:,1)-Xp(:,1))'*(X(:,1)-Xp(:,1)); % squared prediction error

    for n=1:N-1,

        Y(:,n) = H*X(:,n) + V(:,n);  % current observation (output)

        % KALMAN FILTER (5 steps as shown in lecture)
        K = Sp*H'*inv(H*Sp*H'+R);    % Kalman gain
        Xe(:,n) = Xp(:,n) + K*(Y(:,n)-H*Xp(:,n)); % estimate
        Se = Sp - K*H*Sp;            % ECM of estimate
        Xp(:,n+1) = F*Xe(:,n);       % prediction
        Sp = F*Se*F'+G*Q*G';         % ECM of prediction

        Ee(i,n) = (X(:,n)-Xe(:,n))'*(X(:,n)-Xe(:,n)); % squared estimate error
```

```
            X(:,n+1) = F*X(:,n) + G*U(:,n);   % update state (dynamical model)

            Ep(i,n+1) = (X(:,n+1)-Xp(:,n+1))'*(X(:,n+1)-Xp(:,n+1)); % squared prediction error

        end

        % epilogue
        Y(:,N) = H*X(:,N) + V(:,N);             % current output
        K = Sp*H'*inv(H*Sp*H'+R);
        Xe(:,N) = Xp(:,N) + K*(Y(:,N)-H*Xp(:,N));
        Ee(i,N) = (X(:,N)-Xe(:,N))'*(X(:,N)-Xe(:,N));

end

plot(0:N-1, zeros(1,N), 0:N-1, mean(Ep), 0:N-1, mean(Ee), 'Linewidth',2)
grid on
axis square
xlabel('time');
ylabel('mean squared error');
legend('true state','predicted state','estimated state','Location','Northeast')
```

2. 4 points. Poor textbook Chapter V, Problem 2.

   Solution: Since $\{s[k]\}_{k=0}^{\infty}$ and $\{\Gamma[k]\}_{k=0}^{\infty}$ are both known, you can follow the derivation in the lecture notes to show that they affect the state prediction equation as a constant, i.e.

$$\hat{X}[n+1|\ell] = \mathcal{F}_n^{\ell}\hat{X}[\ell|\ell] + \sum_{j=\ell}^{n} \mathcal{F}_n^{j+1}\Gamma[j]s[j].$$

   Thus the one-step predictor is

$$\hat{X}[\ell+1|\ell] = \mathcal{F}[\ell]\hat{X}[\ell|\ell] + \Gamma[\ell]s[\ell].$$

   The ECM for the one-step predictor remains unchanged since the covariance of a constant is zero. Hence,

$$\Sigma[\ell+1|\ell] = F[\ell]\Sigma[\ell|\ell]F^T[\ell] + G[\ell]Q[\ell]G^T[\ell]$$

   The rest of the Kalman-Bucy recursion steps (computing the Kalman gain, computing the state estimate, and computing the ECM of the state estimate) are also unchanged.

3. 4 points. Poor textbook Chapter V, Problem 3.

   Solution Since $s[k]$ is a deterministic function of $\mathcal{Y}_0^k$, it is a constant when we condition on $\mathcal{Y}_0^k$. The only point at which this affects the derivation of the Kalman-Bucy filter is in the state prediction equations, which now become:

$$\hat{X}[\ell+1|\ell] = F[\ell]\hat{X}[\ell|\ell] + \mathrm{E}\{\Gamma[\ell]s[\ell] \,|\, \mathcal{Y}_0^{\ell}\} = F[\ell]\hat{X}[\ell|\ell] + \Gamma[\ell]s[\ell],$$

   and

$$\begin{aligned}\Sigma[\ell+1|\ell] &= F[\ell]\Sigma[\ell|\ell]F[\ell]^T + G[\ell]Q[\ell]G[\ell]^T + \mathbf{cov}\left(\Gamma[\ell]s[\ell] \,|\, \mathcal{Y}_0^{\ell}\right) \\ &= F[\ell]\Sigma[\ell|\ell]\mathbf{F}[\ell]^T + G[\ell]Q[\ell]G[\ell]^T\end{aligned}$$

   because of the conditioning on $\mathcal{Y}_0^{\ell}$. Note that the latter equation is the same as the "normal" case when there is no measurement feedback.

The measurement feedback has no effect on the remaining equations in the Kalman-Bucy recursion. Note that the presence of the output feedback may cause the state to be non-Gaussian, the joint *conditional* statistics of $X[n]$ and $Y[n]$ given $\mathcal{Y}_0^{n-1}$ are still Gaussian. Thus, the estimation update equation is unchanged from the case of no measurement feedback since it depends only on this joint Gaussian property and the linearity of measurement equation.

4. 4 points. Poor textbook Chapter V, Problem 4.

   Solution: There are several ways of approaching this problem. An interesting one is to note that, although $U[n]$ and $V[n]$ are dependent, the Gaussian vector

$$U'[n] \quad := \quad U[n] - C[n]R[n]^{-1}V[n]$$

   is independent of $V[n]$. Note that $Y[n] = H[n]X[n] + V[n]$. To take advantage of this, we may add the zero quantity $C[n]R[n]^{-1}\left[Y[n] - H[n]X[n] - V[n]\right]$ to the n^{th} state input, which yields the equivalent state equation

$$X[n+1] = F[n]X[n] + G[n]U'[n] + G[n]C[n]R[n]^{-1}\left(Y[n] - H[n]X[n]\right).$$

   So, we have an equivalent problem with independent state and measurement noises, but with the measurement feedback term

$$G[n]C[n]R[n]^{-1}Y[n],$$

   and with the modified state update matrix

$$F[n] - G[n]C[n]R[n]^{-1}H[n].$$

   We also have a different covariance matrix for the state input, since

$$\mathbf{cov}(U'[n]) = Q[n] - C[n]R[n]^{-1}C[n]^T.$$

   We can apply the results of the previous problem and eliminate the measurement update equations to arrive at the desired result.