EE 661 *Modern Control*

Project #5: ***The Kalman Observer***

Due Date: Wednesday, December 17, 2014

In this project, you will design a Kalman observer to estimate the state-vector of a given state-space model during a simulation, based on a given set of input and noise signals. The given data include noise inputs, which we generally think of as unknowable, but we must supply the noise signals in order to compute a simulation of a noisy system.

So, even though we actually know the noise signals, the Kalman observer state-estimate must be computed as if we don't know them. That accords with reality in a situation where we have a noisy system generating the output, instead of a *simulator* generating the output.

What we do know about the noise signals is their statistics, *i.e.*, their means and covariances. We need to know these statistics (or estimates of them) in order to compute the Kalman filter equations.

You should download the zip-file, **`proj5_abcd_ts.zip`**, which contains a mat-file with the same name. In that mat-file is a structure variable, again with the same name. The structure variable has five fields defining a 3-input, 3-output, 10th-order, discrete-time state-space model with 20-kHz sample rate. The fields are:

`proj5_abcd_ts.a`
`proj5_abcd_ts.b`
`proj5_abcd_ts.c`
`proj5_abcd_ts.d` (note that *D* is a 3x3 matrix of zeros)
`proj5_abcd_ts.ts`

You should also download the zip-file **`proj5_noise_sequences.zip`**, which contains the three noise signals. Note that each of these noise signals is multi-dimensional**:**

$$\text{output noise } v_k = \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix}, \text{ input noise } w_k = \begin{bmatrix} w_1(k) \\ w_2(k) \\ w_3(k) \end{bmatrix}, \text{ and state noise } \Xi_k = \begin{bmatrix} \xi_1(k) \\ \vdots \\ \xi_{10}(k) \end{bmatrix}$$

These noise sequences are provided in matrix form, where the k^{th} column of the matrix is the value of the noise vector at time $(k-1)$. So, in the mat-file,

$$\text{3-by-}M \text{ matrix} \quad \mathbf{V} = [\, v_0 \;\; v_1 \;\; v_2 \;\; \cdots \;\; v_{M-1} \,]$$

$$\text{3-by-}M \text{ matrix} \quad \mathbf{W} = [\, w_0 \;\; w_1 \;\; w_2 \;\; \cdots \;\; w_{M-1} \,]$$

$$\text{10-by-}M \text{ matrix} \quad \mathbf{Xi} = [\, \xi_0 \;\; \xi_1 \;\; \xi_2 \;\; \cdots \;\; \xi_{M-1} \,]$$

Your simulation should run for the total number of time-samples in the data file, *i.e.*, *M* = **20000**.

All noise sequences are stationary zero-mean gaussian, and uncorrelated (white) with themselves and with the other sequences. That means the covariance matrices are all diagonal. In this case,

$$R = E[v_k v_k^T] = 0.003\, I_{3\times 3}, \quad Q = E[w_k w_k^T] = 0.0025\, I_{3\times 3}, \text{ and } \quad \Sigma = E[\Xi_k \Xi_k^T] = 10^{-5}\, I_{10\times 10}.$$

Finally, you should also download the zip-file **`proj5_input_sequence.zip`**, which contains a mat-file by the same name. The mat-file contains a single matrix U, the input signal for the simulation, $u_k = \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$, stored as a matrix, similar to those for the noise signals. So the matrix **U** is 3x20000.

Here are the quantities you must retain during the simulation, for plotting purposes afterward:

(1) the "true" (unknowable) internal state: $x_{k+1} = Ax_k + B(u_k + w_k) + \Xi_k$
Initialize the true state to a nonzero condition by: **`randn(10,1);`**

(2) the "true" (unknowable) output: $y_k = Cx_k$ (note that $D = 0$)

(3) the measured (noisy) output: $z_k = Cx_k + v_k$

(4) the Kalman optimal gain, $\overline{K}_k$
Here is the update equation for the Kalman gain:

$$\overline{K}_k = AP_kC^T(CP_kC^T + R)^{-1}$$

(5) the Kalman error-covariance matrix, P_k
Initialize the error-covariance according to $P_0 = 100\, I_{10\times10}$
Here is the update equation for the error-covariance matrix:

$$P_{k+1} = (A - \overline{K}_kC)P_k(A - \overline{K}_kC)^T + \overline{K}_kR(\overline{K}_k)^T + BQB^T + \Sigma$$

(6) the Kalman estimate of the internal state: $\hat{x}_{k+1} = (A - \overline{K}_kC)\hat{x}_k + Bu_k + \overline{K}_kz_k$
Choose a vector of zeros as the initial value of the estimated state.

(7) the Kalman estimate of the output: $\hat{y}_k = C\hat{x}_k$ (note that $D = 0$)

Note that most of these saved quantities are column vectors, so their histories can be stored column-wise in two-dimensional matrices. But matrices P_k and $\overline{K}_k$ have more than one column, so their histories should be stored as "slices" of three-dimensional matrices.

After your simulation has finished, use those histories to compute and plot the following:

(a) Plot the ten "true" state-vectors and the ten estimated state-vectors, versus time. For the most useful display of data, create five different plots, each showing only two of the states, and the corresponding state-estimates (*i.e.*, plot states #1 and #2 together, states #3 and #4 together, *etc*.). Be sure to include the appropriate state numbers in each plot title.

(b) Plot the three **"true" outputs**, the three **measured outputs**, and the three **estimated outputs**, versus time. Verify that the Kalman estimates of the outputs are generally closer to the "true" outputs than they are to the measured outputs, by zooming-in to a representative area of the plot that clearly demonstrates this trend.

(c) Plot the norms of the error covariance matrix P_k and the Kalman gain matrix $\overline{K}_k$, versus time. Use Matlab's "**`norm`**" function to compute these norms, slice-by-slice. On this same plot, show the computed covariance of the state-estimate error, based on the data <u>after</u> the filter has reached steady-state (*i.e.*, don't include the first 30 milliseconds of data in the computation). Your plot should show that the norm of P_k closely approaches the computed steady-state covariance, as time goes on (actually, it should reach steady-state in about 30 ms).

Your project should be submitted primarily as a Matlab m-file that performs all the steps necessary to do the things listed above. The m-file should be well-commented.

Comments:

From your simulation, you will have history matrices with the following sizes:

- Error-covariance history is 10 x 10 x 20000
- Kalman gain history is 10 x 3 x 20000
- "True" internal state history is 10 x 20000
- Kalman estimated state history is 10 x 20000
- "True" output history is 3 x 20000
- Measured-output history is 3 x 20000
- Kalman estimated output history is 3 x 20000

These matrices should be pre-allocated with those sizes, to greatly improve execution speed of the simulation. If your 20000-iteration simulation takes much more than about five seconds to complete, then you are doing *something* inefficiently.

For part **(c)**, approximate the norm of the steady-state state-estimate error-covariance by the following computations, where **`X`** is the true state history and **`Xhat`** is the Kalman state-estimate history. Note that the computation calls Matlab's variance function "**`var`**", and specifies that the variance is to be computed along the rows, rather than the default columns.

```
stdystate_ndx = round( 0.030/proj5_abcd_ts.ts );

stdystate_cov_est = ...
   var( Xhat(:,stdystate_ndx:end)-X(:,stdystate_ndx:end), 0, 2 );

approx_norm_stdystate_cov = max( stdystate_cov_est );
```