University of Connecticut **Fall 2008**
ECE Dept. **KRP**

Problem Set #8
MORE ON LEAST SQUARES
(Due October 29, 2008)
(Do Problems 1, 2, 3, 4 and 7)

1. From *m* independent measurements $x = b_i$ of the pulse rate with different variances $\sigma_1^{\;2}, \sigma_2^{\;2}, ..., \sigma_m^{\;2}$, what is the best estimate $\hat{x}$? What is the variance *P* in the best estimate?

2. Consider two measurements having error variances $R_{11} = 3, R_{22} = 2$, and covariance $R_{12} = R_{21} = -1$. If the measurements of the parameter *x* are $b_1 = 0$ and $b_2 = 5$, what is the best estimate $\hat{x}$? What is the error variance?

3. A child's age is estimated as b_1 by his mother with error probabilities $e_{-1} = 0.1, e_0 = 0.8, e_1 = 0.1$, and b_2 by his father with error probabilities $e_{-1} = 0.2, e_0 = 0.6, e_1 = 0.2$.
   (a) Find the best linear estimate (Hint: find the variance of errors and use least squares).
   (b) Find the minimum mean squared estimate (i.e. conditional mean).

4. Suppose we have the QR factorization for an *m* by *n* matrix *A* and now wish to solve the least squares problem $\min \left\| \left(A + \underline{u}\underline{v}^T \right) \underline{x} - \underline{b} \right\|_2$ where $\underline{u}$ and $\underline{b}$ are m vectors, and $\underline{v}$ is an n vector. Give an algorithm for solving this problem in *O*(*mn)* operations. Assume that *Q* must be updated.

5. Give an algorithm for updating the QR factorization of a matrix *A* when the k^{th} row is deleted from *A*.

6. A practical application of recursive least squares and Kalman filtering occurs in applications where periodic measurements of a changing system are made in order to determine the parameters of the system. A typical example is the track-while-scan operation of a radar in which the position of a target is measured each time the antenna rotates past the target position. If it is assumed that the target is moving in a straight line at a constant velocity, then these measurements can be used to determine both the target position and velocity. In any practical case, the measurements are corrupted by noise so they are not precise, but have errors associated with them. This means that the tracking operation must provide some filtering or smoothing in the process of estimating the system parameters.

A simple system that is capable of generating the track of a target from a sequence of discrete measurements is the $\alpha - \beta$ tracker (a more sophisticated tracker is the Kalman filter). In this system, current and past measurements are used to predict the next measured value. The difference between the actual measured values and predicted value (called the residual) is used to correct the predictor parameters. The fractions of the error between the measured and predicted positions used to correct the prediction of position and velocity estimates are the constants α and β, respectively. The system is easily implemented in two dimensions (x-y) or three dimensions. However, in the interests of simplicity, we will consider the problem of tracking in only one dimension (x). That is, we want to estimate the target's position and its velocity along the x-axis based on inaccurate measurements.

Let:

$T =$ time interval between two successive measurements (i.e. period of measurement = 1 s)

$x(n) =$ true position of target along the *x*-coordinate at time step *n*

$v(n) =$ true velocity of the target along the *x*-coordinate at time step *n*

$w(n) =$ acceleration of the target modeled as random noise at time step *n*

$e(n) =$ measurement noise at time step *n*

$z(n) =$ measured target position along *x*-axis at time step *n*

$\hat{X}(n) =$ estimate of the target position along the *x*-coordinate at time step *n*

$\hat{V}(n) =$ estimate of the target velocity along the *x*-coordinate at time step *n*

$x_p(n) =$ predicted target position at time step *n*

$v_p(n) =$ predicted target velocity at time step *n*

The motion of the target is modeled by the difference equation:

$$\begin{bmatrix} x(n+1) \\ v(n+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(n) \\ v(n) \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} w(n),$$

with the initial conditions: $x(0) = 0$ and $v(0) = 1000$ *ft/sec*. At each time step *n*, the process noise is assumed to be a random variable drawn from a Gaussian distribution with zero mean and standard deviation of 30 ft/sec^2.

The measurement equation is given by $z(n) = x(n) + e(n)$, where $e(n)$ is the measurement noise. At each time step *n*, the measurement noise is assumed to be a random variable drawn from a Gaussian distribution with zero mean and standard deviation of 300 *ft.*

The $\alpha - \beta$ filter operation can be described by the following difference equations:

(1) $\hat{x}(n) = x_p(n) + \alpha\left(z(n) - x_p(n)\right)$

(2) $\hat{v}(n) = v_p(n) + \dfrac{\beta}{T}\left(z(n) - x_p(n)\right)$

(3) $x_p(n+1) = \hat{x}(n) + T\hat{v}(n)$

(4) $v_p(n+1) = \hat{v}(n),$ where $\alpha, \beta =$ filter parameters.

(a) Using Eqs. (1-3), show that the $\alpha - \beta$ filter can be written in state-space form:

$$\begin{bmatrix} x_p(n+1) \\ v_p(n+1) \end{bmatrix} = \begin{bmatrix} 1-\alpha-\beta & T \\ -\beta/T & 1 \end{bmatrix} \begin{bmatrix} x_p(n) \\ v_p(n) \end{bmatrix} + \begin{bmatrix} \alpha+\beta \\ \beta/T \end{bmatrix} z(n),$$

and the output equation is given by:

$$\hat{x}(n) = (1-\alpha)x_p(n) + \alpha z(n).$$

The filter starts at n = 1 with initial conditions: $x_p(1) = \hat{X}(0) = z(0),$ and $v_p(1) = 0.$ For the target parameters selected, the values α and β are given by:
$\alpha = 0.36$ and $\beta = 0.08$.

(b) Simulate the target dynamics and measurement equations for $n \in (1, 200)$.

(c) Couple the simulation of Task 2 with the simulation of the $\alpha - \beta$ filter. Plot the prediction error (also called residual) ($z(n)$-$x_p(n)$) and the true tracking error $\left(x(n) - \hat{X}(n)\right)$. Also, plot the true target positions $x(n)$ and the estimated target positions $\hat{X}(n)$ as a function of n. Discuss how well your tracker tracks the target.

(d) Repeat (a) and (c) using the conventional Kalman filter.

(e) Repeat (a) and (c) using the square-root version of the Kalman filter.

7. Derive the square-root version of the probabilistic data association filter (PDAF). To learn about PDAF, read the paper by Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with a Probabilistic Data Association Filter," Automatica, Vol. 11, 1975, pp. 451-560.

8. Resolve problems 3-7 of problem set #7 using recursive least squares, and plot the residuals.