c) Accuracy Measures

The best accuracy measure is the covariance matrix of the estimated states itself. This matrix contains all the information available about the covariance of the state vector and can be used to generate the error-ellipse (in 2D) or ellipsoid (in 3D) etc. In the 2-D case, the 2x2 covariance matrix contains 3 independent elements, a variance n and e (or y and x) and a covariance. The error-ellipse can be constructed by finding the eigen-values and eigen vectors of the covariance matrix. The eigen vectors will be orthogonal and represent the directions of the semi-major and semi-minor axes of the covariance ellipse. The eigen values are the magnitudes of each semi-axis. This can be generalized to an unlimited number of dimensions. For the 2-D case

$$a = \left[\frac{1}{2}\left(\sigma_e^2 + \sigma_n^2\right) + \left[\frac{1}{4}\left(\sigma_e^2 - \sigma_n^2\right)^2 + \sigma_{en}^2 \right]^{\frac{1}{2}} \right]^{\frac{1}{2}}$$

$$b = \left[\frac{1}{2}\left(\sigma_e^2 + \sigma_n^2\right) - \left[\frac{1}{4}\left(\sigma_e^2 - \sigma_n^2\right)^2 + \sigma_{en}^2 \right]^{\frac{1}{2}} \right]^{\frac{1}{2}}$$

$$\theta = \frac{1}{2}\arctan\left[\frac{2\sigma_{en}}{\sigma_e^2 - \sigma_n^2} \right]$$

where θ is the azimuth of the semi-major axis.

A 2-D error ellipse only encloses 39% of the probability density of the solution. In order to have a higher probability region, the semi-axes must be multiplied by scale factors. For 90%, the factor is 2.15. For 95%, the factor is 2.45, and for 99% the factor is 3.03.

DRMS and CEP

Two other measures are commonly used are the “Distance Root Mean Squared” (DRMS) and “Circular Error Probable” (CEP). Both of these were developed to simply the math in the era when calculations were done by hand. DRMS is the square root of the trace of the 2-D covariance matrix. It is an important measure in GNSS accuracy because of it’s relationship with Horizontal Dilution of Precision (HDOP). CEP is an even more simplified measure of accuracy and is defined as the the radius of a circle containing half of the probability density function of the position estimate. It originated as a measure of accuracy for ballistics, specifically if you fire n shots with a cannon, what is the radius that contains n/2 of the hits. Derivations of both will be presented in class.

d) Sequential Least Squares and the Linear Kalman filter.

There is a third LS formulation called “Sequential Least Squares” that once again rearranges the batch LS solution. The sequential solution is recursive, that is it is expressed in terms of the answer from the first batch and the change to that answer resulting from adding new observations. The standard solution for Sequential LS is given by

$$\hat{\delta}^{(+)} = \hat{\delta}^{(-)} - K[w_2 + A_2\hat{\delta}^{(-)}]$$
$$C_{\hat{\delta}^{(+)}} = N_1^{-1} - KA_2N_1^{-1}]$$

where K is called the gain matrix and is given by

$$K = N_1^{-1}A_2^T[C_l + A_2N_1^{-1}A_2^T]^{-1}$$

What the sequential LS solution represents is a weighted average of the new observations and the previous parameter estimates in a way that produces new parameter estimates that are equivalent to the parameter estimates that would be obtained had all the observations been used together. The biggest advantage of the sequential solution is that it is recursive. Only the previous estimate and its covariance matrix need be stored from epoch to epoch.

It should be noted that sequential LS can process one observation at a time as long as that observation is independent of the other observations and a previous solution exists. In this case the misclosure vector becomes a scalar, the design matrix has only one row, and the inverse in the gain matrix becomes a scalar inverse. The quality is very useful in indoor navigation and/or integrated systems where you might only have one observation available at one time (and not enough for a full solution). This one observation is still better than no observations and can be used in the solution.

The Kalman Filter is an extension of sequential least squares to the case where the parameters being estimated vary from one epoch to the next. By this we mean that they physically vary, not just that their estimates vary. In other words, sequential least squares can only be used to estimate the value of a stationary process, while Kalman filtering can be used on time varying processes.

i. A comparison of common Kalman Filter and Least Squares notation

Most Kalman filter literature uses different notation from what we have seen for Least squares. The matrix formulations are equivalent, but different letters are used to represent each matrix. The equivalent expressions are shown in the table below. For simplicity the linear LS case is shown. Note there are some minus signs missing.

Least Squares	Kalman Filtering
Parametric Batch	
$l = Ax + r$	$z = Hx + v$
C_l	R
$\hat{x} = (A^T C_l^{-1} A)^{-1} A^T C_l^{-1} l$	$\hat{x} = (H^T R^{-1} H)^{-1} A^T R^{-1} z$
$C_{\hat{x}} = (A^T C_l^{-1} A)^{-1}$	$P = (H^T R^{-1} H)^{-1}$
x is called the parameter vector	x is called the state vector
Sequential	
$\hat{x}^{(+)} = \hat{x}^{(-)} - K[w_2 + A_2 \hat{x}^{(-)}]$	$\hat{x}^{(+)} = \hat{x}^{(-)} + K[z - H\hat{x}^{(-)}]$
$C_{\hat{\delta}^{(+)}} = N_1^{-1} - KA_2 N_1^{-1}]$	$P^{(+)} = P^{(-)} - KHP^{(-)} = (I - KH)P^{(-)}$
$K = N_1^{-1} A_2^T [C_l + A_2 N_1^{-1} A_2^T]^{-1}$	$K = P^{(-)} H^T [HP^{(-)} H^T + R]^{-1}$

ii. The Kalman Filter and accounting for changes in the parameters

The reason for the (-) and (+) in the sequential formulas is that (-) represents the state and its covariance before being updated with new observations and the (+) quantities represent the states after they have been updated. The Kalman filter adds one more element to this procedure, the idea of prediction. If there is a model for the behaviour of the state vector as a function of time, it can be used to predict the state vector forward in time to the time of the next observations. The next observations can then be used to update the state vector. The newly updated state vector can then be predicted to the time of the next observations and so on.

The behaviour of the state vector over time can be described by a dynamics model. In continuous time, a dynamics model can be given by a system of first order linear differential equations

$$\dot{x} = Fx$$

That is each element of the first time derivative of the state vector equals some linear combination of the state vector with the linear combination given by the dynamics matrix F. In discrete time, this same relationship can be expressed as

$$x_k = \Phi x_{k-1}$$

where Φ is the transition matrix that represents the linear combination that predicts the current epoch as a function of the state vector from the last epoch.

The covariance matrix can also be propagated forward remembering the error propagation rule that when a linear transformation is applied to a vector, the same linear transformation must be applied to both sides of the corresponding covariance matrix.

$$P_k^{(-)} = \Phi P_{k-1}^{(+)} \Phi^T + Q$$

The additional matrix Q is called process noise and is normally added to represent the fact that predicting the state vector forward in time is based on a less than perfect model. Often the process noise is time dependent to represent the fact that as more time elapses, our knowledge of the state vector decreases. The only way to reduce the covariance is through adding new observations with an sequential update step. Thus the standard procedure for Kalman filtering is, starting with an initial value, predict the state and covariance forward to the time of the observations, then update the state and covariance with the observations and then predict forward to the next observation interval.

This type of filter is very useful since it can incorporate individual observations that occur at different times. It can also output a state value at any prediction point, which means that a Kalman filter can provide the best available state estimate at regular intervals even though observations may be irregularly spaced. The choice of transition matrix and process noise depends on the type of dynamic system that is being modeled by the state vector. The simplest possible case involves an identity transition matrix and zero process noise. In this case the KF reduces back to sequential least squares.

A simple example of a transition matrix is the constant velocity model. In this model, position and velocity are being estimated and the dynamic model assumes constant velocity over time, and a position that depends on previous position and velocity. If the state vector is

$$x = [x, y, z, v_x, v_y, v_z]^T$$

then the transition matrix is

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

assuming that the time interval between the two epochs is Δt . If you multiply this matrix by the state vector, you get three equations each saying position = position + velocity x time and three equations saying velocity = velocity, hence the name “constant velocity model”.

State vectors and transition functions can be chosen to suit the estimation task at hand. Kalman filters are very common in carrier phase GPS, inertial navigation, and sensor

integration, especially when many sensors are being used and these sensors have biases or systematic error that need to be estimated during each run.

The process noise added to the state covariance serves to make the filter slowly forget earlier observations. This is very useful if the state vector is expected to change over time, but in an unpredictable way. Without process noise, the state covariance will get smaller with every observation update, meaning that eventually the state estimate will be so well known that new observations will have no additional effect on the solution, in order to avoid this, process noise is added. Choosing appropriate process noise is not easy and balancing the process noise with the observation noise is known as filter tuning.