Kalman Filter and Least Squares

by Davide Micheli

The Kalman filter

The Kalman filter is a multiple-input multiple output digital filter that can optimally estimates, in real time, the values of variables describing the state of a system from a multidimensional signal contaminated by noise.

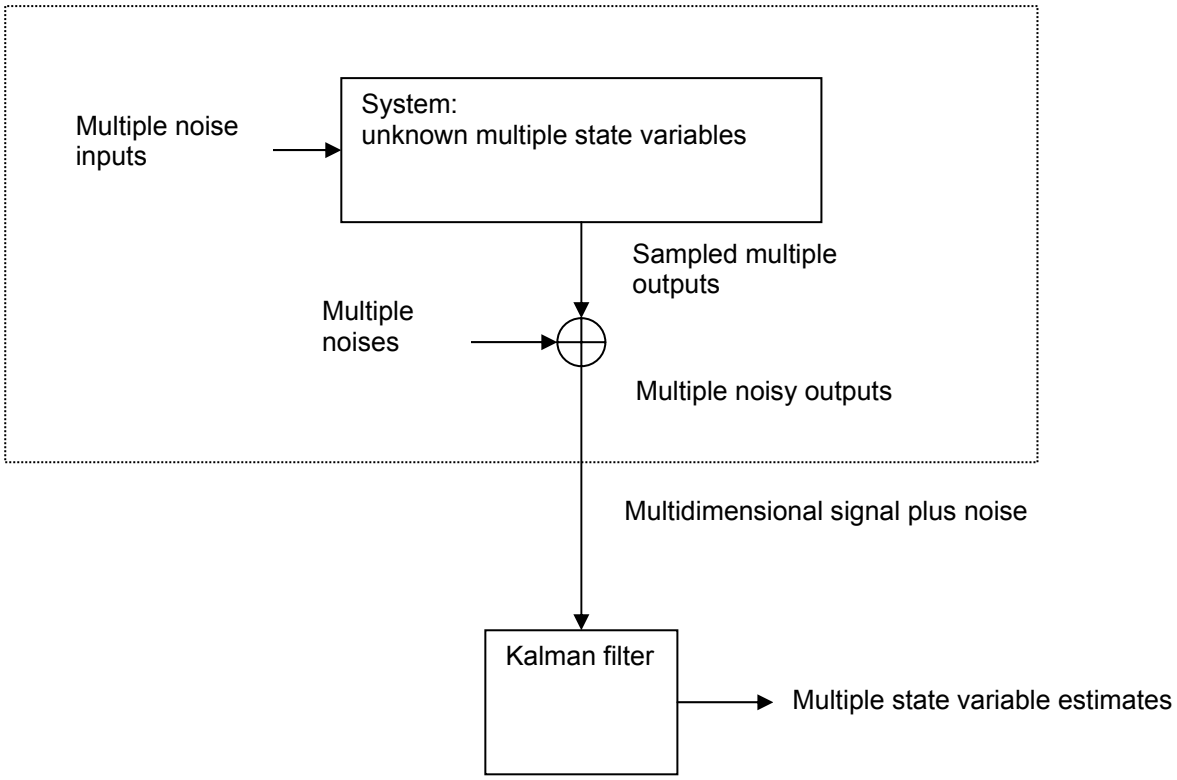

Figure 1

These ***states*** are all variable needed to completely describe the system behaviour as a function of time (such as position, velocity, and so forth); ***states are the unknown signal to be estimate***.
To provide current estimate of the system variable, the filter uses ***statistical models*** to properly weight each new measurement relative to past information. It also determines up-to-date uncertainties of the estimates for real time quality assessments.

Because the states (or signal) is typically a vector of scalar random variables, the *state uncertainty estimates* is a ***covariance matrix***. Each diagonal term of the matrix is the variance of scalar random variable that describes the *mean squared deviation* from its mean. The other terms (matrix's off-diagonal terms), are the covariance that describe any correlation between pair of variables.

The multiple measurements (at each time point/epoch) are also vectors that a recursive algorithm processes sequentially in time. This means that the algorithm iteratively repeats itself for each new measurement vector, using only values stored from previous cycle. This procedure distinguishes itself from batch-processing algorithms, which must save all past measurements
The Kalman filter is usually referred to a Discrete Kalman filter and an Extended Kalman Filter definition is used in the case of non linear process.

Discrete Kalman filter

The Kalman filter addresses the general problem of trying to ***estimate the state x of a discrete-time controlled process.***

In figure 2 we reported the scheme of process:

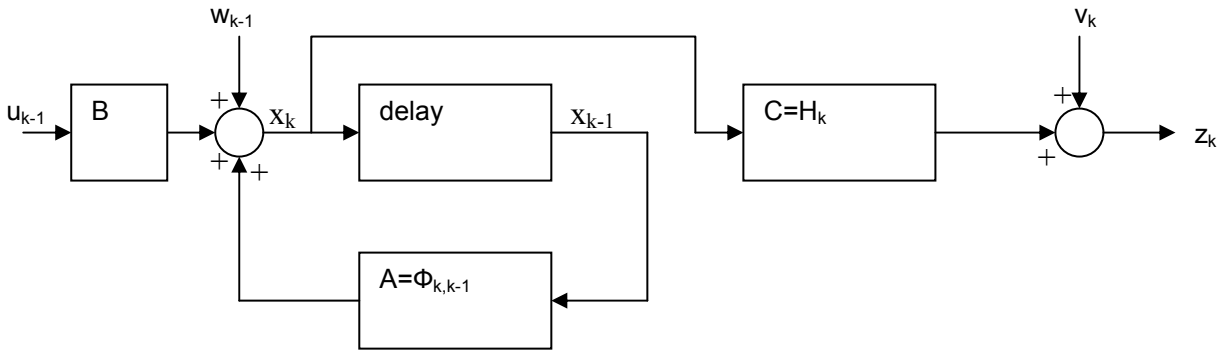

Figure 2

where

$x \in R^n$ is the state
$u \in R^l$ is the control input

time discrete controlled process is governed by linear difference equation:

1. the state ***x*** at actual time step ***k*** is

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \qquad \text{Equation 1}$$

2 the measurement ***z*** at actual time step ***k*** is

$$z_k = H_k x_k + v_k \qquad \text{Equation 2}$$

where:

- *n x n* Matrix ***A*** is the transition matrix and relates the state at the previous time step ***K-1*** to the current state ***k*** in absence of either driving function or process noise. Note that ***A*** might change with each time step, but here we assume it is constant.
- *n x l* Matrix ***B*** relates the optional control input ***u*** to the state ***x***
- *m x n* Matrix ***H*** characterizes the sensitivity of measurements to each of the state component relating the current state to the current measurement z_k

the random variables w_k and v_k represent the process and measurement noise respectively.
They are assumed to be independent (of each other), white, and, with normal probability distributions:

$$p(w) \approx N(0, Q) \qquad \text{Equation 3}$$
$$p(v) \approx N(0, R) \qquad \text{Equation 4}$$

The ***process noise <u>covariance</u> Q*** and ***measurement noise <u>covariance</u> R*** matrices might change with each time step or measurement, here we assume they are constant.

State estimate

We define $\hat{x}_k^- \in R^n$ to be our a ***priori*** *state estimate* at step ***k*** before incorporating current measurements $\boldsymbol{z_k}$ at the same step ***k***, given the knowledge of the process prior to step ***k***.

We define $\hat{x}_k \in R^n$ to be our a ***posteriori*** *state estimate* at step k after incorporating measurements $\boldsymbol{z_k}$ at the same step k.

We can then define a ***priori*** and a ***posteriori*** estimate errors as:

$$e_k^- \equiv x_k - \hat{x}_k^- \qquad \text{Equation 5}$$

$$e_k \equiv x_k - \hat{x}_k \qquad \text{Equation 6}$$

The ***a priori*** estimate error covariance is then

$$P_k^- = E\left[e_k^- e_k^{-T}\right] = E\left[\left(x_k - \hat{x}_k^-\right)\left(x_k - \hat{x}_k^-\right)^T\right] \qquad \text{Equation 7}$$

the ***a posteriori*** estimate error covariance is

$$P_k = E\left[e_k e_k^T\right] = E\left[\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right] \qquad \text{Equation 8}$$

In deriving the equations for the Kalman filter, we begin with the goal of finding an equation that computes an a posteriori state estimate $\hat{x}_k \in R^n$ as a linear combination of a priori state estimate $\hat{x}_k^- \in R^n$ and a *weighted difference* between an actual measurement $\boldsymbol{z_k}$ and a measurement prediction $\hat{z}_k^- = H\hat{x}_k^-$:

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H\hat{x}_k^-\right) \qquad \text{Equation 9}$$

The difference $\left(z_k - H\hat{x}_k^-\right)$ is called ***measurement innovation*** or ***residual***. The residual reflects the discrepancy between the predicted measurement $\hat{z}_k^- = H\hat{x}_k^-$ and the actual measurement $\boldsymbol{z_k}$
A residual of zero means that the two are in complete agreement.

The *n x m matrix* ***K*** is chosen to be the *gain or blending factor* that *minimizes* the a posteriori error covariance $\boldsymbol{P_k}$
It's possible to determine it and one form of the resulting $\boldsymbol{K_k}$ is:

$$K_k = P_k^- H^T\left(HP_k^- H^T + R\right)^{-1} = \frac{P_k^- H^T}{\left(HP_k^- H^T + R\right)} \qquad \text{Equation 10}$$

- Observing the equation of K_k we see that as the <u>*measurement error covariance R*</u> approaches zero, the gain K_k weights the residual more heavily (i.e. the measurement's weight will be high and the predicted state estimate's weight will be low):

  $if \qquad R_k \to 0 \qquad K \to H^{-1} \quad \text{then} \qquad \hat{x}_k \to H^{-1} z_k$ because

  $$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + H^{-1}\left(z_k - H\hat{x}_k^-\right) = \hat{x}_k^- + H^{-1}z_k - H^{-1}H\hat{x}_k^- \\ &= \hat{x}_k^- + H^{-1}z_k - I_n\hat{x}_k^- = \hat{x}_k^- + H^{-1}z_k - \hat{x}_k^- = H^{-1}z_k \\ \Rightarrow z_k &= H\hat{x}_k \end{aligned}$$

- On the other hand, as the <u>*a priori state estimate error covariance*</u> P_k^- approaches zero, the gain K_k weights the residual less heavily (readily visible states or easily "observable" states will receive the higher weigh):

  $if \qquad P_k^- \to 0 \qquad K \to 0 \quad \text{then} \qquad \hat{x} \to \hat{x}^-$

The probabilistic Origin of the filter

The justification for $\hat{x}_k = \hat{x}_k^- + K\left(z_k - H\hat{x}_k^-\right)$ is rooted in the probability of the ***a priori*** estimate $\hat{x}_k^-$ conditioned on all priori measurement $\hat{z}_k^- = H\hat{x}_k^-$ (Bayes' rule). For now let it suffice to point out that the Kalman filter maintains the first two moments of the state distribution,

- The ***a posteriori state estimate*** reflects the ***mean*** (the first moment) of the state distribution, it is normally distributed if the condition $p(w) \approx N(0, Q)$ and $p(v) \approx N(0, R)$ are met

  $$E[x_k] = \hat{x}_k \qquad \text{Equation 11}$$

- The a ***Posteriori estimate error covariance*** reflects the variance of the state distribution (the second non central moment) i.e. the distance root mean square from the mean value.

  $$E\left[\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right] = P_k \qquad \text{Equation 12}$$

  in other word

  $$p(x_k \mid z_k) \approx N\left(E[x_k], E\left[\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right]\right) = N\left(\hat{x}_k, P_k\right) \qquad \text{Equation 13}$$

Algorithm

the Kalman filter estimates a process by using a form of ***feedback control***: the filter estimates the process state at some time and then obtains feedback in the form of (*noisy*) measurements

The equations for the Kalman filter fall into two groups:

1. ***Time Update equations (predictor equations)*** are responsible for projecting forward (in time), from step ***k-1*** to step ***k*** the state and the error covariance estimates to obtain the a *priori estimate* for the next time step ***k***
2. ***Measurement update equations (correct equations)*** are responsible for the *feedback* -- i.e. for incorporating a new measurement at step ***k*** into the a *priori estimate* of the same step ***k*** to obtain an improved a *posteriori estimate of the current state k*

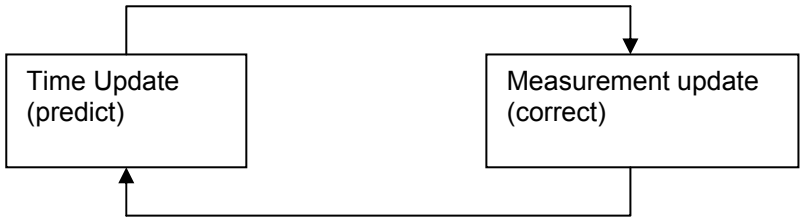

Figure 3

1. ***Time update prediction equation*** are:

- $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$ Equation 14
- $P_k^- = AP_{k-1}A^T + Q$ Equation 15

2. ***Measurement update correction equations*** are:

- $K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1}$ Equation 16
- $\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - H\hat{x}_k^-\right)$ Equation 17
- $P_k = \left(I - K_k H\right)P_k^-$ Equation 18

with z_k

$z_k = H_k x_k + v_k$ are the measurements

schematically the algorithm can be viewed as shown below:

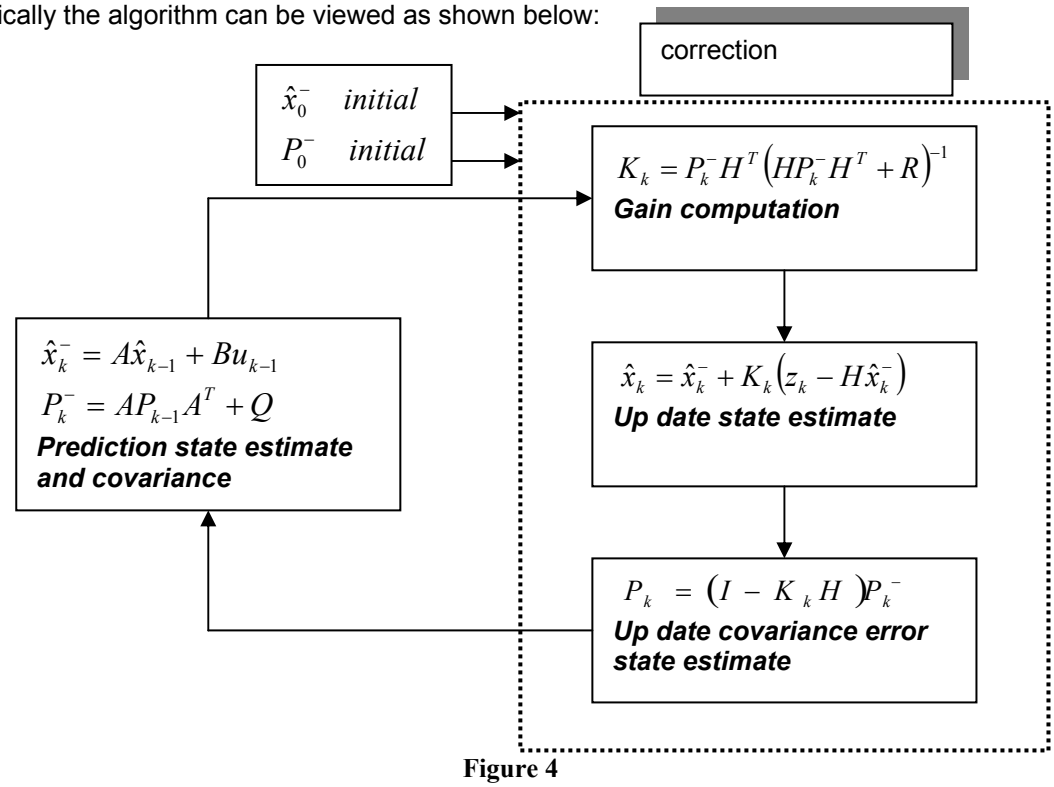

Figure 4

- The first task during the measurement update is to compute the Kalman gain $\boldsymbol{K_k}$.
- The next step is to actually measure the process to obtain $\boldsymbol{z_k}$ and then to generate an a posteriori state estimate $\hat{x}_k$ by incorporating the measurement as in equation 17.
- The final step is to obtain an a posteriori error covariance estimate via equation 18.
- The process is repeated by prediction of a new a priori state estimate $\hat{x}_{k+1}^-$ and covariance error estimate P_{k+1}^- for the next time step ***k+1*** this is done by using $\hat{x}_{k+1}^- = A\hat{x}_k + Bu_k$ and $P_{k+1}^- = AP_k A^T + Q$ just computed in the actual state. The system ***state*** vector is assumed to change with time (projecting) according to a deterministic linear transformation (***model state A***) plus an independent random noise. *Because the actual noise value is unknown, the predicted state estimate follow only the deterministic linear transformation*. On the contrary the covariance prediction ac-counts of random noise ***Q*** too because the random noise uncertainty is known.

- After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project o to predict the new a priori estimates, this recursive nature is one of the very appealing features of the Kalman filter.

We can see that as the measurement vectors are recursively processed, the state estimate's uncertainty should generally decrease (if all states are observable) because of the accumulated information measurements.
However, in the prediction step some information is lost, so uncertainty will increase. The uncertainty will reach a steady state when the amount of uncertainty accumulated in the prediction step will be balanced by the uncertainty decreasing due to update correction step.
If no random noise exist in the actual model when the state evolves to the next step, then the uncertainty will eventually approach zero.
Because the state uncertainty change with time so to will the weight. Generally speaking, the Kalman filter is a digital filter with time-varying gains.

If the state of a system is constant, the Kalman filter reduces to a sequential form of deterministic, classical least squares with a weight matrix equal to the inverse of the measurement noise covariance matrix.

Mathematically speaking we have:

Substituting equation 2,11,13 into equation 14 we obtain for ***update state estimate***

$$\begin{aligned}\hat{x}_k &= \hat{x}_k^- + K_k\left(z_k - H\hat{x}_k^-\right) = A\hat{x}_{k-1}^- + Bu_{k-1} + K_k\left(H_k x_k + v_k - H\left(A\hat{x}_{k-1}^- + Bu_{k-1}\right)\right) \\ & A\hat{x}_{k-1}^- + Bu_{k-1} + K_k H_k x_k + K_k v_k + K_k HA\hat{x}_{k-1}^- + K_k HABu_{k-1} = \\ & \left(1 + K_k H\right)A\hat{x}_{k-1}^- + \left(1 + K_k HA\right)Bu_{k-1} + K_k H_k x_k + K_k v_k\end{aligned}$$

Equation 19

Substituting equation 12 into equation 15 we obtain for ***update covariance error state estimate***

$$\begin{aligned}P_k &= \left(I - K_k H\right)P_k^- = \left(I - K_k H\right)\left(AP_{k-1}A^T + Q\right) = AP_{k-1}A^T + Q - K_k HAP_{k-1}A^T - K_k HQ = \\ &= \left(AP_{k-1} - K_k HAP_{k-1}\right)A^T + \left(1 - K_k H\right)Q\end{aligned}$$

Equation 20

the final block schematic presentation of equation 11—14 relative to the single step state estimate is shown below:

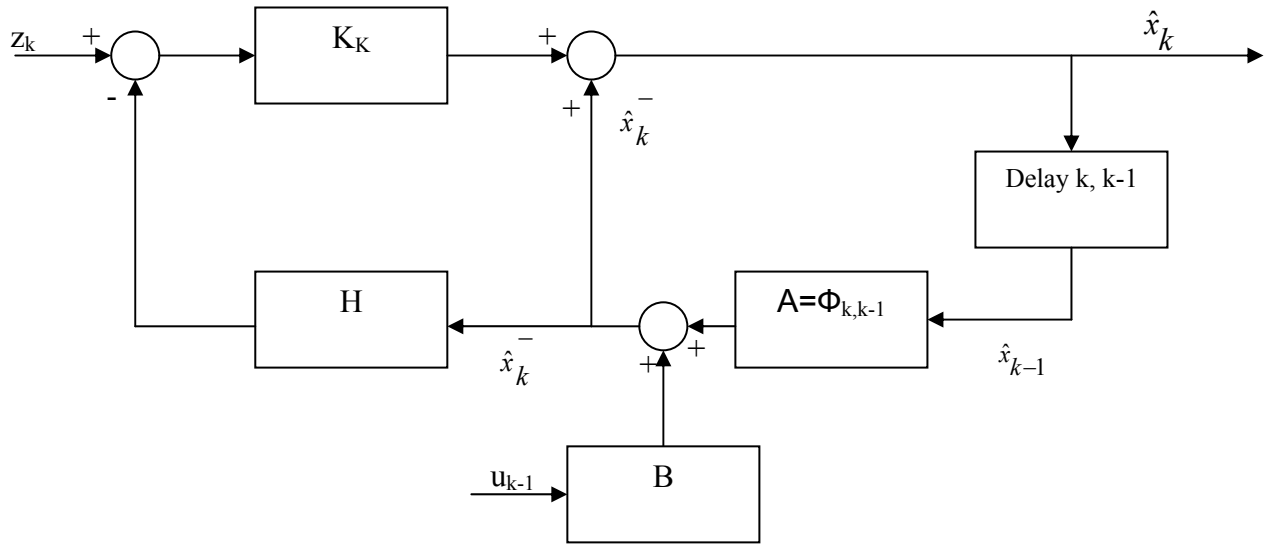

Figure 5

The extended Kalman Filter

If the process to be estimated and (or) the measurement relationship to the process is ***non-linear*** we have to use an EKF (extended Kalman Filter)

EKF operate a linearization of the estimation around the current estimate using the partial derivatives of the process and measurement function when these are non linear relationship.

Let us assume that the process is now governed by the non linear difference equation:

1. the state ***x*** at actual time step ***k*** is

$$x_k = f\left(x_{k-1}, u_{k-1}, w_{k-1}\right) \qquad \text{Equation 21}$$

2. the measurement ***z*** at actual time step ***k*** is

$$z_k = h\left(x_k, v_k\right) \qquad \text{Equation 22}$$

where:

- random variables w_k and v_k again represent the process and measurement noise as in the preceding case
- ***f*** is the non-linear function relate the state at the previous time step ***(k-1)*** to the state at the current time step ***(k).***
- u_{k-1} is the driving function
- the non linear function ***h*** in the measurement equation relates the state x_k to the measurement z_k

In practice of course one does not know the individual values of the noise $\boldsymbol{w_k}$ *and* $\boldsymbol{v_k}$ *at each time step*. However, one can approximate the state and measurement vector without them as:

$$\tilde{x}_k = f\left(\hat{x}_{k-1}, u_{k-1}, 0\right) \qquad \text{Equation 23}$$

$$\tilde{z}_k = h\left(\tilde{x}_k, 0\right) \qquad \text{Equation 24}$$

where $\hat{x}_k \in R^n$ is some a ***posteriori*** estimate of the state (from the previous time step k).

It is important to note that a fundamental flaw of the EKF is that the distribution (or density function in the continuous case) of the various random variables are no longer normal after undergoing their respective non linear transformation. The EKF is simply an ad hoc estimator that only approximates the optimally of Bayes rule by linearization:

If U=[A1……An] is a partition of S with A_i =event of U and B is an arbitrary event then from total probability and Bayes' theorem follow:

$$P(B) = P(B \mid A_1)P(A_1) + \ldots\ldots\ldots\ldots + P(B \mid A_n)P(A_n)$$

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{P(B \mid A_1)P(A_1) + \ldots\ldots\ldots\ldots + P(B \mid A_n)P(A_n)} \qquad \text{Equation 25}$$

where the terms ***a priori*** is used for $\boldsymbol{P(A_i)}$ and ***a posteriori*** is used for $\boldsymbol{P(A_i \mid B)}$

it is possible to extend this result to random variables:

if $B = \{X \le x\}$ we obtain

$$P(X \le x) = P(X \le x \mid A_1)P(A_1) + \ldots\ldots\ldots\ldots + P(X \le x \mid A_n)P(A_n)$$

$$P(A_i \mid X \le x) = \frac{P(X \le x \mid A_i)P(A_i)}{P(X \le x \mid A_1)P(A_1) + \ldots\ldots\ldots\ldots + P(X \le x \mid A_n)P(A_n)} \qquad \text{Equation 26}$$

for the ***conditional distribution F(x)*** follows that:

$$F(x) = F(x \mid A_1)P(A_1) + \ldots\ldots\ldots\ldots + F(x \mid A_n)P(A_n) \qquad \text{Equation 27}$$

for the ***conditional density f(x)*** follows that:

$$f(x) = f(x \mid A_1)P(A_1) + \ldots\ldots\ldots\ldots + f(x \mid A_n)P(A_n) \qquad \text{Equation 28}$$

The computational Origins of the filter

To estimate a process with non-linear difference and measurement relationships, we begin by writing new governing equations that ***linearize*** an estimate about $\tilde{x}_k = f\left(\hat{x}_{k-1}, u_{k-1}, 0\right)$ and $\tilde{z}_k = h\left(\tilde{x}_k, 0\right)$:

$$x_k = \tilde{x}_k + A\left(x_{k-1} - \hat{x}_{k-1}\right) + Ww_{k-1} \qquad \text{Equation 29}$$

$$z_k = \tilde{z}_k + H\left(x_k - \tilde{x}_k\right) + Vv_k \qquad \text{Equation 30}$$

where:

- x_k and z_k are the actual state and measurement vectors,
- $\tilde{x}_k$ and $\tilde{z}_k$ are the ***approximate*** state and measurement vectors,
- $\hat{x}_k$ is ***a posteriori*** estimate of the state at step ***K,***
- random variables w_k and v_k again represent the process and measurement noise as in the preceding case
- ***A*** is the Jacobian matrix of partial derivatives of ***f*** with respect to ***x*** that is:

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}\left(\hat{x}_{k-1}, u_{k-1}, 0\right) \qquad \text{Equation 31}$$

where the Jacobian ***j(x)*** di ***f*** is defined for ***i=1….m*** and ***j=1….n*** as :

$$j(x) \equiv \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_m(x)}{\partial x_1} & & \frac{\partial f_m(x)}{\partial x_n} \end{pmatrix} \qquad \text{Equation 32}$$

- ***W*** is the jacobian matrix of partial derivatives of ***f*** with respect to ***w***,

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}\left(\hat{x}_{k-1}, u_{k-1}, 0\right) \qquad \text{Equation 33}$$

- ***H*** is the jacobian matrix of partial derivatives of ***h*** with respect to ***x***

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}\left(\tilde{x}_k, 0\right)$$

Equation 34

- ***V*** is the jacobian matrix of partial derivatives of ***h*** with respect to ***v***

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}\left(\tilde{x}_k, 0\right) \qquad \text{Equation 35}$$

Note that for simplicity in the notation we do not use the time step subscript ***k*** with the Jacobians ***A, W, H,*** and ***V***, even though they are in fact different at each time step.

Now we define a new notation for the ***prediction error*** as the approximate state error as:

$$\tilde{e}_{x_k} = x_k - \tilde{x}_k$$

Equation 36 and the

measurement residual as the approximate measurement error as:

$$\tilde{e}_{z_k} = z_k - \tilde{z}_k \qquad \text{Equation 37}$$

Remember that in practice one does not have access to x_k, in fact it is the ***actual state vector,*** i.e. the quantity one is trying to estimate. On the other hand, one does have access to z_k, in fact it is the actual measurement that one is using to estimate x_k.

Using the preceding equations we can write governing equations for an *error process* as:

$$\tilde{e}_{x_k} = x_k - \tilde{x}_k = A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1} = A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k \quad \text{Equation 38}$$

$$\tilde{e}_{z_k} = z_k - \tilde{z}_k = H(x_k - \tilde{x}_k) + Vv_k = H(x_k - \tilde{x}_k) + \eta_k = H\tilde{e}_{x_k} + \eta_k \quad \text{Equation 39}$$

where:

- ε_k represent a new independent random variable having zero mean and covariance matrices ***WQW^T***
- η_k represent a new independent random variable having zero mean and covariance matrices ***VRV^T***

with Q and R as $p(w) \approx N(0, Q)$; $p(v) \approx N(0, R)$

Notice that the Equation 40 and Equation 41 are linear, and that they closely resemble the difference and measurement equations $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$ and $z_k = H_k x_k + v_k$ from the discrete Kalman filter.

*This motivate us to use the **actual measurement residual** $\tilde{e}_{z_k} = z_k - \tilde{z}_k$ and a second (hypothetical) Kalman filter to estimate the prediction error $\tilde{e}_{x_k} = x_k - \tilde{x}_k$. This estimate called $\hat{e}_k$, could than be used along with $\tilde{e}_{x_k} = x_k - \tilde{x}_k$ to obtain the a **posteriori state estimates** for the original non-linear process as:*

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \quad \text{Equation 42}$$

the random variables $\tilde{e}_{x_k} = x_k - \tilde{x}_k$ and $\tilde{e}_{z_k} = z_k - \tilde{z}_k$ have approximately the following ***probability distributions***:

$$p(\tilde{e}_{x_k}) \approx N(0, E[\tilde{e}_{x_k} \tilde{e}_{x_k}^T]) \quad \text{Equation 43}$$

$$p(\varepsilon_k) \approx N(0, WQ_k W^T) \quad \text{Equation 44}$$

$$p(\eta_k) \approx N(0, VR_k V^T) \quad \text{Equation 45}$$

Given these approximations and letting the predicted value of $\hat{e}_k$ to be zero, the Kalman filter equation used to estimate $\hat{e}_k$ is :

$$\hat{e}_k = K_k \tilde{e}_{z_k} \quad \text{Equation 46}$$

by substituting $\hat{e}_k = K_k \tilde{e}_{z_k}$ in equation $\hat{x}_k = \tilde{x}_k + \hat{e}_k$ and using $\tilde{e}_{z_k} = z_k - \tilde{z}_k$ we obtain:

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k = \tilde{x}_k + K_k \tilde{e}_{z_k} = \tilde{x}_k + K_k(z_k - \tilde{z}_k) \quad \text{Equation 47}$$

this equation can now be used for measurement update in the extended Kalman filter, with

- $\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0)$
- $\tilde{z}_k = h(\tilde{x}_k, 0)$
- $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$ this last with the appropriate substitution for the measurement error covariance.

The complete set of EKF equations is shown below. Note that we have substituting $\hat{x}_k^-$ for $\tilde{x}_k$ to remain consistent with the earlier "super minus" ***a priori*** notation, and that we now attach the subscript ***k*** to the

Jacobians ***A,W,H***, and ***V*** to reinforce the notation that they are different at (and therefore must be recomputed at) each time step.

- ***Time update (prediction equations)***

$$x_k^- = f\left(\hat{x}_{k-1}, u_{k-1}, 0\right)$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

- ***Measurement update (correction) equations***

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + V_k R_k V_k^T\right)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h\left(\hat{x}_k^-, 0\right)\right)$$

$$P_k = \left(I - K_k H_k\right) P_k^-$$

with z_k

$$z_k = \tilde{z}_k + H\left(x_k - \tilde{x}_k\right) + V v_k$$

GPS Position and velocity determination using Kalman Filter

As stated above the Kalman filter is a recursive algorithm that provides optimum estimates of user PVT (position velocity timing) based on noise statistics and current measurements.
The filter contain a ***dynamical model*** of the ***GPS receiver platform motion*** and outputs of user receiver PVT state estimate as well as associated error variances.
The ***dynamical model*** can be derived by a ***Taylor series expansion*** about the *true position of the receiver*.
*Let **u(t)** represent the true position of the receiver at time **t**. Then at time **t**, shortly after time **t_0** the receiver will be at position:*

$$u(t) = u(t_0) + \frac{du(t)}{dt}_{t=t_0}(t - t_0) + \frac{1}{2!}\frac{d^2u(t)}{dt^2}_{t=t_0}(t - t_0)^2 + \frac{1}{3!}\frac{d^3u(t)}{dt^3}_{t=t_0}(t - t_0)^3 \ldots\ldots \frac{1}{(n-1)!}\frac{d^{(n-1)}u(t)}{dt^{(n-1)}}_{t=t_0}(t - t_0)^{(n-1)} \ldots\ldots$$

where:

$$\frac{du(t)}{dt}_{t=t_0} = velocity$$

$$\frac{1}{2!}\frac{d^2u(t)}{dt^2}_{t=t_0} = acceleration$$

$$\frac{1}{3!}\frac{d^3u(t)}{dt^3}_{t=t_0} = jerk$$

terms following the third derivative expression representing jerk are usually considered negligible.

Position and velocity for most vehicles *change relatively slowly* over the time period of the measurements used to estimate them, so the *estimates are reasonably close*.
The same may not be true for ***acceleration or jerk*** in some vehicles. If these terms are not negligible, such as perhaps in some aircraft, they are often modelled as random quantities (i.e. white noise), in order to take their effects into account.

Filter designated for ***PVT*** determination typically estimate eight user states: ***position (x_u,y_u,z_u), velocity (dx_u/dt, dy_u/dt,dz_u/dt), receiver clock offset (t_u), and receiver clock drift (dt_u/dt)***

A flow diagram using Kalman filter is shown below:

Time update <u>prediction</u> equation

- The filter is first initialized with approximate value for each user state. This initial user state estimate values are input to the ***dynamical model (A).*** Assume that the platform is moving with constant velocity therefore rectilinear dynamical model motion (A) will be expressed as:

$$u(t) = u(t_0) + \frac{du(t)}{dt}_{t=t_0} (t - t_0)$$

as part of recursive algorithm, the ***dynamical model*** propagates the platform position from one time position ***k-1*** to the next ***k***

The relative prediction state and covariance equation for Kaman filter are:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$
$$P_k^- = AP_{k-1}A^T + Q$$

Measurement update <u>correction</u> equations

- After propagating the platform position to the next epoch ***k***, the *ephemeredes* for each satellite in view is extracted from the navigation message. Using the propagated ***k*** *state* (i.e. *user position* and *velocity* estimates), the receiver calculate the a ***priori pseudorange*** and a ***priori delta pseudornage*** (i.e. change in pseudorage per epoch) for each satellite in view .

$$\hat{z}_k^- = H\hat{x}_k^-$$

- Next, *pseudorange* and *delta pseudorange* are than measured $\boldsymbol{z_k}$ and the difference is taken between the estimate values $\hat{z}_k^- = H\hat{x}_k^-$. These differences are referred to as ***residual***.

$$\left(z_k - H\hat{x}_k^-\right)$$

- If the a priori <u>*estimated*</u> *pseudorange end delta pseudorange values* exactly match the <u>*measured*</u> *pseudorange and delta pseudorange values* the ***residual*** would be zero. No zero ***residual*** indicate errors in the user PVT estimates. By the Kalman gain $\boldsymbol{k_k}$***:***

$$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1}$$

The filter then adjust the *user state estimate* to minimize the residuals according to a minimum mean squared error criterion. These adjusted *user state estimate* are output to the user and also feedback to the dynamical model to repeat the recursive estimation process:

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - H\hat{x}_k^-\right)$$
$$P_k = \left(I - K_k H\right)P_k^-$$

figure below show the process:

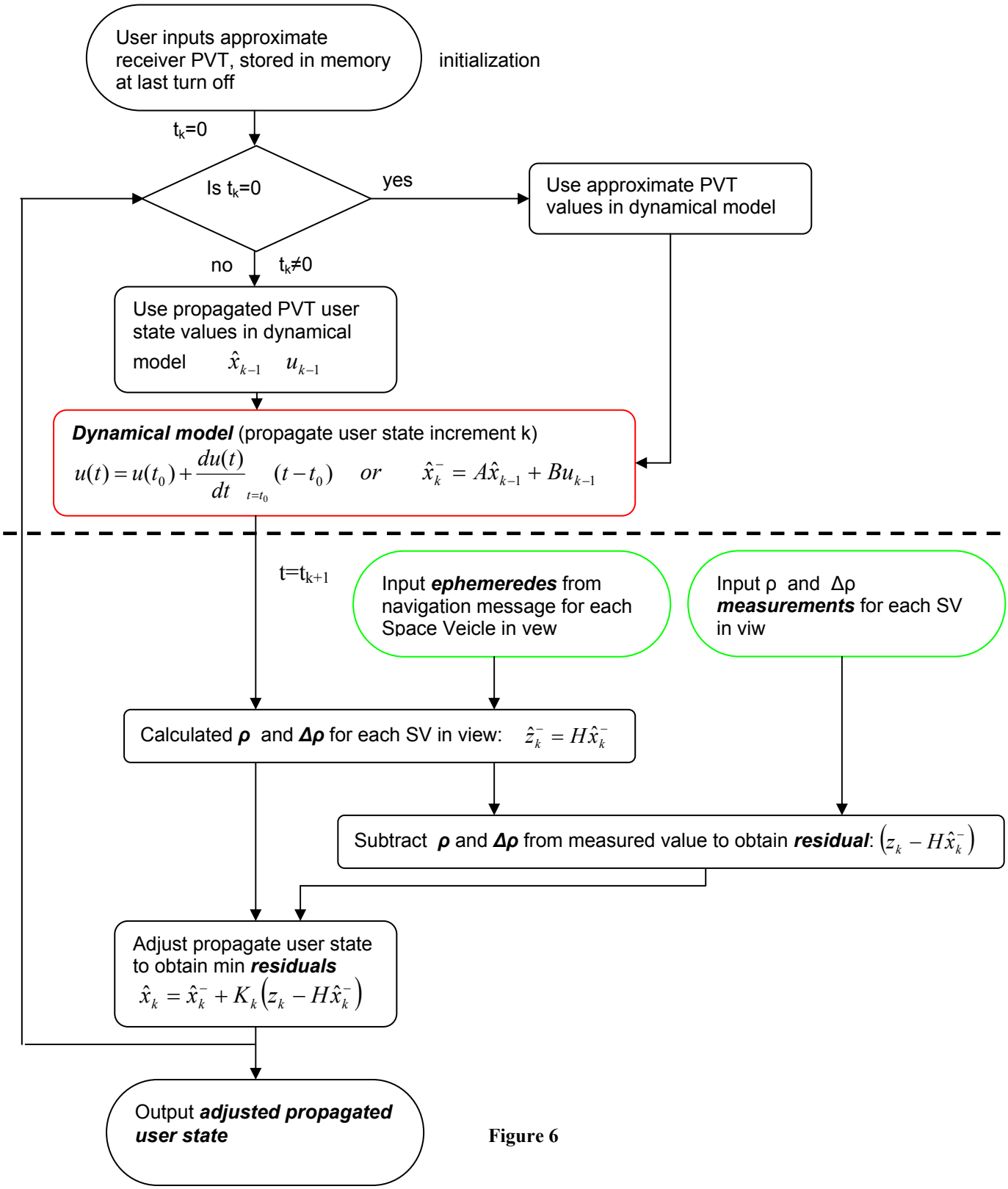

Figure 6

The filter provides an estimated solution using a partial measurement set until other satellites can be acquired and tracked.

As the measurement noise increases (including Selective availability effects for SPS user), the filter decreases the weights of the measurement information while relying more on the user state estimate. When the noise variance decreases, the filter utilizes the measurement information more and relies on the estimates less.

Bibliography

[1] The Kalman filter: Navigation's integration Workhorse by Larry j.Levy [www.cs.edu]
[2] An Introduction to the Kalman Filter by Greg Welch and Gary Bishop [www.cs.unc.edu]
[3] Understanding GPS principles and Navigations [Kaplan]
[4] Probability Random Variables and Stochastic Porcess [A. Papoulis]
[5] Corso di analisi matematica [S.M. Nikolskij]

Least Squares

The linearized peseudorange equations have the matrix form:

$$H \cdot \Delta X = \Delta\rho \qquad \textbf{Equation 48}$$

If *N> 4* this system has more equations than unknowns.
Generally the system will be inconsistent in fact the value of *Δρ* will preclude any value of *ΔX* from exactly solving the system.

The *Least Square* method can be used to obtain an estimate of *ΔX* (i.e. the user position and time offset)

Assuming that *H* and *Δρ* are available, the vector quantity :

$$\boxed{r = H \cdot \Delta X - \Delta\rho} \qquad \text{Equation 49}$$

is called the *RESIDUAL*.

We seek ΔX so that the RESIDUAL is small.

The ordinary least square solution is defined as the value of *ΔX* that minimizes the square of the residual called R_{SE} :

$$R_{SE}(\Delta X) = (H \cdot \Delta X - \Delta\rho)^2 \qquad \textbf{Equation 50}$$

The basic idea is to differentiate Equation 51 with respect to *ΔX* in order to obtain the gradient of R_{SE} . The minimum for R_{SE} must occur at a value of *ΔX* that gives a zero fore the gradient

$$R_{SE}(\Delta X) = (H\Delta X - \Delta\rho)^T (H\Delta X - \Delta\rho) = (\Delta X^T H^T - \Delta\rho^T)(H\Delta X - \Delta\rho) =$$

$$= \Delta X^T H^T H\Delta X - \Delta X^T H^T \Delta\rho - \Delta\rho^T H\Delta X - \Delta\rho^T \Delta\rho$$

differentiating we obtain:

$$\nabla R_{SE} = \frac{\partial R_{SE}}{\partial X} = \Delta X^T H^T H - \Delta\rho^T H$$

$$\nabla R_{SE} = 0 \quad \Rightarrow \quad \Delta X^T H^T H - \Delta\rho^T H = 0$$

In order to get ΔX we take the transpose of: ∇R_{SE}

$$(\Delta X^T H^T H - \Delta\rho^T H)^T = (H^T H\Delta X - H^T \Delta\rho) = 0$$

$$H^T H\Delta X = H^T \Delta\rho$$

provided that $H^T H$ is non singular we solve for ΔX obtaining:

$$\boxed{\Delta X = (H^T H)^{-1} H^T \Delta\rho}$$

the condition that $H^T H$ is non singular is equivalent with the condition that the columns of H are independent (RANK=4). These statements are also equivalent with the condition that the tips of the unit vectors from the linearization point to the satellites do not all lie in a common plane. If this condition are violated the least square problem dose not have a unique solution.