

# GNSS Solutions:

## GNSS Filtering Options

“GNSS Solutions” is a regular column featuring questions and answers about technical aspects of GNSS. Readers are invited to send their questions to the columnist, Dr. Mark Petovello, Department of Geomatics Engineering, University of Calgary, who will find experts to answer them. His e-mail address can be found with his biography at the conclusion of the column.

## What are the differences between least-squares and Kalman filtering?

Most, if not all, GNSS receivers compute their positions using Kalman filtering (more common) or least-squares (less common) estimation algorithms (“estimators”). Kalman filtering also finds application in a wide variety of integrated navigation systems (e.g., GNSS integrated with inertial navigation systems). However, the role of each of these estimators is not always well understood, especially for people new to the navigation and/or estimation field.

To fully explain their differences is both simple and complex, depending on the approach taken. Here, we consider the simpler option because of limited space and because the simpler explanation arguably offers more practical insight. Of course, the simple answer is not mathematically rigorous, and those interested in such rigor are encouraged to flush out the details provided in the myriad of textbooks available on the subject.

So, with this in mind, the differences between least-squares and Kalman are surprisingly minor. This is unintuitive, given the derivation of the different algorithms; least-squares is based on minimizing the measurement residuals (i.e., the difference between the actual and predicted measurements) whereas the Kalman filter is derived based on minimizing the mean-square error of the solution. Nevertheless, this will be borne out in subsequent discussion.

## Mathematical Overview

Least-squares and Kalman filtering employ the following measurement model:

$$\vec{z} = h(\vec{x}) + \vec{v} \quad (1)$$

where  $\vec{z}$  is the vector of measurements (e.g., GNSS pseudoranges),  $\vec{x}$  is the state vector containing the parameters to be estimated (e.g., position, velocity and time),  $h(\vec{x})$  relates the measurements to the states, and  $\vec{v}$  is the vector of measurement errors. For generality, we assume that  $h(\vec{x})$  is non-linear such that the following linearization can be applied

$$\vec{z} \approx h(\vec{x}_0) + \left. \frac{\partial h(\vec{x})}{\partial \vec{x}} \right|_{\vec{x}=\vec{x}_0} \cdot \delta\vec{x} + \vec{v} \quad (2)$$

$$\vec{z} - h(\vec{x}_0) = \left. \frac{\partial h(\vec{x})}{\partial \vec{x}} \right|_{\vec{x}=\vec{x}_0} \cdot \delta\vec{x} + \vec{v}$$
$$\delta\vec{z} = H \cdot \delta\vec{x} + \vec{v}$$

where  $\vec{x}_0$  is the current estimate of the state vector,  $\delta$  is an error in the following term, and  $H$  is the Jacobian of the measurement model. Based on equation (2), the least-squares solution for the error in  $\vec{x}_0$  — which is then applied to the original state vector to correct it to  $\vec{x}_1$  — is given by

$$\delta\vec{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} \delta\vec{z} \quad (3)$$

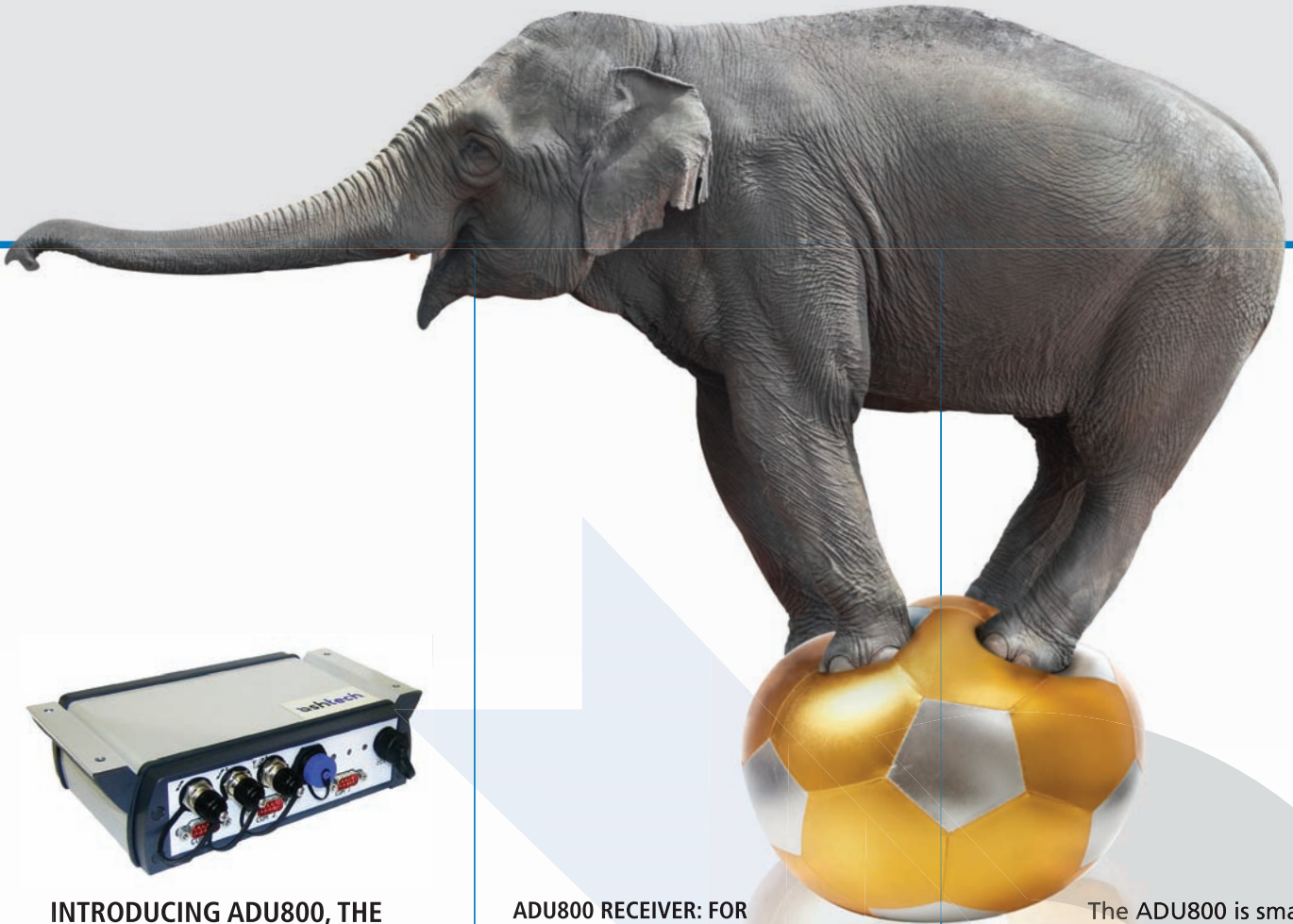
where  $R$  is the covariance matrix of the measurement errors,  $\vec{v}$ . If, in addition to the information contained in the measurements, some *a priori* information about the state is also available, equation (3) can be changed to

$$\delta\vec{x} = (H^T R^{-1} H + P_0^{-1})^{-1} H^T R^{-1} \delta\vec{z} \quad (4)$$

where  $P_0$  is the covariance matrix reflecting the uncertainty of the *a priori* state information. Of course, equation (4) degenerates to equation (3) as the uncertainty in the *a priori* state information increases (i.e., as  $P_0$  tends to infinity). Finally, the covariance matrix of the estimated parameters,  $P$ , is given by

$$P = (H^T R^{-1} H + P_0^{-1})^{-1} \quad (5)$$

# NEED TO CAREFULLY MONITOR YOUR ATTITUDE?



## INTRODUCING ADU800, THE LATEST ASHTECH REAL-TIME, INSTANTANEOUS 6D GNSS POSITIONING SYSTEM

Providing real-time heading, pitch and roll measurements, with accurate position and velocity, at a standard update rate of up to 20 Hz, for static and dynamic platforms.

### ADU800 RECEIVER: FOR REAL-TIME, HIGH-ACCURACY, ABSOLUTE 6D POSITIONING

- GPS/GLONASS/SBAS/QZSS with embedded Z-Blade™ technology
- Patented, multi-constellation signal processing
- Advanced multi-path mitigation and robust signal tracking
- Fast initialization and antenna set-up auto-calibration
- GNSS-centric operation.

The ADU800 is small, weatherproof, and rugged to allow operation in harsh environments. It is ideal for open-pit mining, seismic exploration, oceanographic research, and airborne applications. For a better attitude, explore the full scope of Ashtech's GNSS sensors at [www.ashtech-oem.com](http://www.ashtech-oem.com)

**ashtech**  
A TRIMBLE COMPANY

[www.ashtech-oem.com](http://www.ashtech-oem.com)

[oemsales@ashtech.com](mailto:oemsales@ashtech.com)

Collectively, equations (4) and (5) represent the least-squares solution. Note that, in general, the non-linear least-squares solution is iterated until the corrections are sufficiently small. However, if the initial estimate of the state vector is sufficiently accurate, this is not necessary.

In contrast to least-squares, Kalman filtering always assumes some *a priori* knowledge of the state, usually obtained from some past estimates (more on this later). With this in mind, the update equation is usually written as

$$\begin{aligned}\delta\bar{x}_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \delta\bar{z}_k \\ &= K_k \delta\bar{z}_k\end{aligned}\quad (6)$$

where subscript  $k$  represents the  $k$ -th epoch, the subscript  $k|k-1$  is the estimate at the  $k$ -th epoch based on measurements up to epoch  $k-1$  (i.e., the *a priori* information), and  $K$  is the Kalman gain matrix which effectively weights the information in the observations against the *a priori* knowledge of the states. It is precisely this weighting that “filters” the measurements in order to yield a better solution.

The covariance matrix of the state vector after incorporating the measurements is given by

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (7)$$

where subscript  $k|k$  denotes the estimate at the  $k$ -th epoch based on measurements up to epoch  $k$ .

At face value, the two preceding equations show no obvious resemblance to those for least-squares. However, we can show that the Kalman filter equations can be re-written as

$$\delta\bar{x} = (H_k^T R_k^{-1} H_k + P_{k|k-1}^{-1})^{-1} H_k^T R_k^{-1} \delta\bar{z}_k \quad (8)$$

$$P_{k|k} = (H_k^T R_k^{-1} H_k + P_{k|k-1}^{-1})^{-1} \quad (9)$$

These newer equations are the exact parallels of equations (4) and (5) with only a change in notation for the subscripts of  $P$ . In other words, given a set of measurements, least-squares and Kalman filtering incorporate the information from the measurements in

the same manner and will generate the exact same answer.

## So Why Use a Kalman Filter?

As shown previously, the key difference between least-squares and Kalman filtering has nothing to do with how measurements are processed. Rather, it has to do with how the two estimators obtain their *a priori* information. Whereas least-squares usually obtains this information from external means (e.g., by occupying a known point), Kalman filtering predicts the *a priori* information using the most recent estimate of the state vector.

This predicted state vector is based on some *assumed* model for how the state vector changes/evolves in time; this is usually called the system model. This is expressed as

$$\bar{x}_{k+1|k} = \Phi_{k,k+1} \bar{x}_{k|k} \quad (10)$$

where  $\Phi_{k,k+1}$  is the transition matrix that propagates the state at epoch  $k$  to epoch  $k+1$ . The transition matrix is usually based on the physics of the systems (e.g., velocity multiplied by time gives change in position). Although not discussed here, the discrete-time formula in equation (10) is often — though not always (refer to the September 2010 issue of this column by Lo Presti *et alia* for an example) — derived from a continuous-time state space model.

The covariance matrix of the predicted state is given by

$$P_{k+1|k} = \Phi_{k,k+1} P_{k|k} \Phi_{k,k+1}^T + Q_{k+1} \quad (11)$$

where  $Q$  is the process noise matrix that ultimately accounts for the uncertainty/errors in the assumptions used to derive the transition matrix.

To illustrate, let's consider a one-dimensional example of trying to position a train. We define our states to be the position,  $p$ , and velocity,  $v$ , of the train, namely

$$\bar{x}^T = [p \quad v]$$

Since trains are large and heavy, they do not accelerate or decelerate

very quickly; so, we can assume that the train is moving with approximately constant velocity. The transition matrix can thus be written as

$$\Phi_{k,k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

where  $\Delta t$  is the time between epochs  $k$  and  $k+1$ . Substituting this term into the first term on the right hand side of equation (11) propagates the uncertainty of the states forward in time. This is necessary because, if the states are not known perfectly at some time in the past, it follows that the prediction of those states will introduce larger uncertainties.

To determine  $Q$ , we need to revisit our assumption that the train is moving with constant velocity. Although this may be a reasonable approximation over short time periods, it is not always true (i.e., the train will eventually stop). This means that, in addition to the propagation of the errors from the previous epoch, we also need to add some uncertainty to describe the fact that our assumption is not perfect.

For the case at hand, the uncertainty may be small, precisely because the train's momentum is so high. However, if we were positioning race cars instead of trains, the value of  $Q$  would have to be larger to accommodate the fact that the race car can change its speed much more quickly.

Although the objective of this article is not to give a detailed description of how to select  $Q$  (this is often covered in several chapters of textbooks and even then requires practical experience to fully appreciate), the key point is that selecting  $Q$  directly affects the performance of the filter. In the extreme, if  $Q$  is zero, the filter essentially averages information over time, that is, the filter becomes a cumulative averaging algorithm. Conversely, if  $Q$  is selected to be infinite,  $P_{k|k-1}^{-1}$  in equations (8) and



# GPS NETWORKING

- Military Solutions
- Survey and Mapping
- Timing and Synchronization
- Fire and Rescue, E911

## Solutions customized to your needs

GPS Networking has the experience and technical expertise to provide reliable GPS Signal Distribution components to solve any GPS application challenge. Our proven record of reliability is unsurpassed in the GPS Industry today with over 20 years experience.

800.463.3063 • [www.gpsnetworking.com](http://www.gpsnetworking.com)



## The Institute of Navigation

Your source for 24/7, anytime, anywhere access to objective, credible and trusted technical information!



Positioning | Navigation | Timing

The world's premier professional organization for the advancement of positioning, navigation and timing.

# JOIN TODAY!

[www.ion.org](http://www.ion.org)

## GNSS SOLUTIONS

(9) becomes zero and the Kalman filter degenerates to the least-squares solution in equation (3).

So, the real benefit of a Kalman filter relative to least-squares is that it provides additional information to the system based on assumed knowledge of how the states (e.g., position, velocity, etc.) change with time. If the assumptions regarding the evolution of the states are correct and/or if deviations from the assumed behaviour can be accommodated through some form of adaptation, the Kalman filter will typically provide temporally smoother and more accurate solutions.

Two other benefits are also realized by Kalman filters, relative to least-squares. First, if the uncertainty in the *a priori* estimate is not infinite, we can update the state vector using fewer measurements than there are states. For example, for GNSS positioning, we can update a Kalman filter using pseudoranges from one or two satellites, whereas a least-squares update in this case would be impossible because of insufficient measurements (assuming no *a priori* information is available, which is likely for least-squares except in specific cases such as static positioning).

Second, we can estimate parameters in a Kalman filter that may not be completely observable using least-squares. A good example of this is the ability to use GNSS pseudoranges to estimate position and velocity in a Kalman filter, whereas least-squares could only estimate position using the same data. Intuitively, this is possible in a Kalman filter because successive position estimates can be used to infer velocity.

### No Silver Bullets

Despite the potential benefits of Kalman filtering, we need to exercise caution before blindly applying the algorithm in all scenarios. The reason is that if the assumed behaviour of the states is incorrect, the filter will still try to make the results fit the assumed behavior anyway — after all, that is


what we are instructing it to do! This “self-fulfilling prophecy” behavior can mask certain effects (both good and bad) and potentially degrade performance.

A good example of this is in urban canyons where, if a receiver can only track two satellites, the Kalman filter will usually predict forward motion. However, if the receiver turns a corner, it may take several seconds (or longer) for the filter to identify this condition resulting in an “over shoot” of the trajectory at the corner.

By extension, computing a least-squares estimate at every epoch — although much noisier and less accurate — does have the benefit of providing insight into the “raw” data quality, that is, in the absence of any filtering. Often, this information can then be used to better determine the parameters for a suitable Kalman filter.

### Summary

In this article we showed how least-squares and Kalman filtering estimators handle measurements the same way and that the main difference between them is that Kalman filters provide information about how the states change with time. This additional information, if correct, will indeed improve estimation accuracy.

Nevertheless, an epoch-by-epoch least-squares solution still has an important role in assessing the raw quality of the data that is not usually possible using a Kalman filter. Depending on the application, one or both approaches may prove useful, or even complementary. 



**Mark Petovello** is a Professor in the Department of Geomatics Engineering at the University of Calgary. He has been actively involved in many aspects of positioning and navigation since 1997 including GNSS algorithm development, inertial navigation, sensor integration, and software development.

Email: [mark.petovello@ucalgary.ca](mailto:mark.petovello@ucalgary.ca)



# ION GNSS+ 2013

The 26th International Technical Meeting of the Satellite Division of The Institute of Navigation



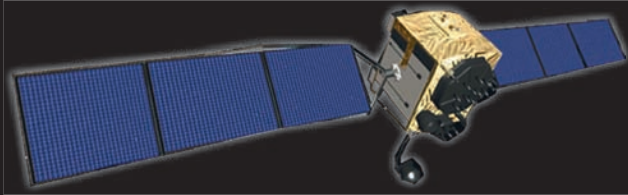
September 16–20, 2013

Show Dates: Sept. 18 – 19

Tutorials: Sept. 16 – 17

Nashville Convention Center

Nashville, Tennessee



**The world's largest technical meeting and showcase of GNSS technology, products and services.**

- Advanced Inertial Sensing and Applications **A**
- Advances in Military GNSS Systems and Applications **A**
- Algorithms and Methods
- Alternatives and Backups to GNSS
- Aviation Applications **A**
- Clock Technologies
- Emerging GNSS (Galileo, COMPASS, QZSS, IRNSS)
- Future PNT and Its Applications **A**
- Geodesy, Surveying and RTK for Civil Applications **C**
- GNSS Algorithms and Methods
- GNSS and the Atmosphere
- GNSS Compatibility, Interoperability, and Interchangeability
- GNSS Ground Based Augmentation Systems (GBAS)
- GNSS Simulation and Testing
- GNSS Space Based Augmentation Systems (SBAS)
- GNSS-MEMS Integration
- GPS and GLONASS Modernization
- Indoor Navigation and Timing
- Interference and Spectrum Issues
- Land Based Applications **A**
- Marine Navigation and Applications **A**
- Multi-Constellation User Receivers **C**
- Multi-Sensor and Integrated Navigation in GNSS-Challenged Environments
- New Products and Commercial Services **C**
- Next Generation GNSS Integrity
- Non Traditional PNT Applications **A**
- Portable Navigation Devices **C**
- Precise Point Positioning
- Receiver/Antenna Technology
- Remote Sensing with GNSS and Integrated Systems
- Safety Critical Applications **A**
- Software Receivers
- Space Applications **A**
- Standalone GNSS Services in Challenging Environments
- Timing and Scientific Applications **A**
- Urban Navigation Technology

**C** Commercial Session  
**A** Application Session

**Exhibitors — Reserve your booth today. Space is limited.**

# NASHVILLE

**REGISTER TODAY! [www.ion.org](http://www.ion.org)**