GNSS Solutions:

Calculating Time Offsets

“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.

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

How do you deal with timing differences between GNSSs?

All GNSSs inherently depend on precise timekeeping to measure the satellite/receiver time of flight of signal propagation with sufficient accuracy to compute ranges/distances for multilateration calculations. Each GNSS ground segment therefore dedicates considerable effort to maintaining a highly stable atomic time scale as well as the corresponding offset to global standards such as UTC (Coordinated Universal Time).

Nevertheless, even after accounting for gross differences between the different GNSS time scales — for example, GPS is approximately synchronized to UTC modulo one second whereas GLONASS is synchronized with UTC less three hours — differences still exist among the time scales at the 10s or 100s of nanosecond level. When scaled by the speed of light, these seemingly small time differences manifest as very large ranging errors that degrade positioning accuracy.

The various GNSSs already plan to broadcast the offset between their system time and the other system times. Unfortunately, especially for the newly launched systems, this information is either not available or is not being transmitted in the navigation message. (GLONASS does broadcast their offset relative to GPS, but currently not on all satellites. On May 3, the European Space Agency announced that the four Galileo in-orbit validation satellites had begun broadcasting the European GNSS system’s time offset from GPS.)

Consequently, until such information is readily available on all signals of all satellites in all systems, users are left to handle the time differences on their own. This article addresses how they can do this.

Before dealing with the offsets between the GNSS time scales, let’s briefly review the pseudorange measurement equations. For the purpose of this article, we can write the pseudorange measurement, *P*, to the *i*-th satellite for a single system (denoted with subscript “*sys*”), as

$$P_{sys}^i = \rho^i + b_{sys} + \varepsilon^i \qquad (1)$$

where ρ is the geometric distance between the receiver and the satellite, b_{sys} is the receiver clock *bias* for the subscripted system, and ε is the combined effect of all measurement errors. Equation (1) implicitly assumes that each satellite’s clock is corrected to its system’s time scale using data broadcast in the navigation message.

A more specific definition for the receiver clock bias is: the difference between the receiver’s estimate of time and the time maintained by the particular GNSS to which the pseudorange is measured. By extension, because every GNSS has a slightly different time scale, each system will have different receiver clock biases even if all of the measurements are made by a single receiver. This article looks at how to estimate the different clock biases (or related parameters).

Assuming measurements from two different GNSSs (denoted with subscripts 1 and 2) are available, equation (1) can be used to write

$$\begin{aligned} P_1^i &= \rho^i + b_1 + \varepsilon^i \\ P_2^j &= \rho^j + b_2 + \varepsilon^j \end{aligned} \qquad (2)$$

However, we can express the clock bias for the second system in terms of the bias of the first system as follows:

$$b_2 = b_1 + \Delta b_{12} \qquad (3)$$

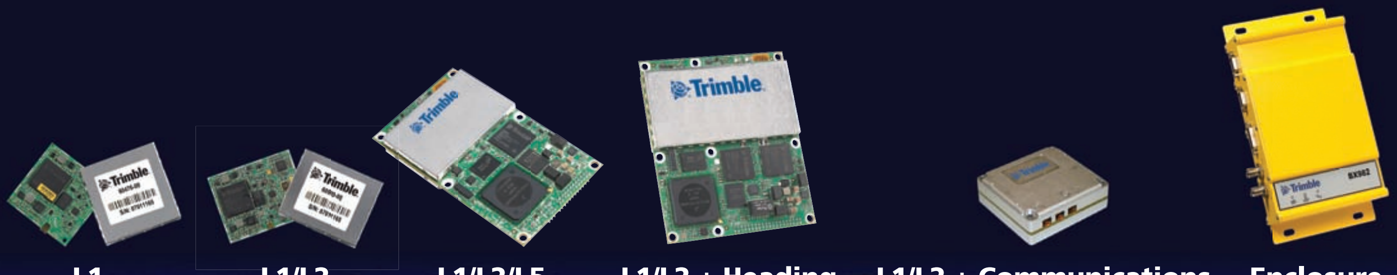
where Δb_{12} is the time offset (or *inter-system offset* or, simply, *offset*) between the two systems.

In this scenario, b_1 can be interpreted as the “reference” system bias and the

GNSS spoken here

[GPS/GLONASS/BeiDou/Galileo]

One family, many solutions. Trust Trimble GNSS OEM for all your high precision applications.



L1	L1/L2	L1/L2/L5	L1/L2 + Heading	L1/L2 + Communications	Enclosure
BD910	BD920	BD970	BD982	BD920-W3G	BX982
GPS, GLONASS, BeiDou, Galileo	GPS, GLONASS	GPS, GLONASS, BeiDou, Galileo	GPS, GLONASS, OmniSTAR, Heading	GPS, GLONASS, WiFi, 3G	GPS, GLONASS, OmniSTAR, Heading

Trimble®

www.trimble.com/gnss-inertial

offset to each other system is computed relative to this. Note that because each GNSS maintains a highly stable time scale, the time offset between any two systems changes vary slowly, typically at a few nanoseconds per day — or on the order of 10 fs/s (i.e., femtoseconds per second; femto = 10^{-15}). Obviously, equation (3) can be substituted into equation (2). The question is whether this is worth doing. The answer to this question is addressed next in the context of least-squares and Kalman filtering.

Least-Squares Estimation

If a receiver uses epoch-to-epoch least-squares to estimate its navigation solution (i.e., the position and receiver clock biases), there is no numerical difference between using equation (2) as is, or substituting equation (3). Rather, the difference depends entirely on how the manufacturer/programmer wishes to implement the navigation solution.

If equation (3) is used, measurements from the reference system are only related to a single clock bias, as in the single-system case. However, measurements to non-reference system satellites are related to the clock bias for the reference system and the corresponding inter-system offset. This increases the complexity of the software. Therefore I believe that using equation (2) directly is slightly easier to implement, because a measurement to a satellite in a particular system is only related to the clock bias for that particular system.

Kalman Filtering

As a quick review, development of Kalman filters often starts with the continuous-time state-space system model given by

$$\dot{\vec{x}} = F\vec{x} + G\vec{w} \tag{4}$$

where $\vec{x}$ is the state vector of parameters to be estimated and $\dot{\vec{x}}$ is its time derivative, F is the dynamics matrix, G is the shaping matrix, and $\vec{w}$ is the process noise vector consisting of Gaussian white noise and having a spectral density matrix given by Q_c. This can then be converted to the discrete-time equivalent which is given by

$$\vec{x}_k = \Phi_{k,k-1}\vec{x}_{k-1} \tag{5}$$

$$P_k = \Phi_{k,k-1}P_{k-1}\Phi^T_{k,k-1} + Q_k \tag{6}$$

where the subscript k represents a value at the k-th epoch, $\Phi_{k,k-1}$ is the transition matrix that converts the state from epoch k−1 to k, P is the covariance matrix of $\vec{x}$ and Q_k is the discrete-time process noise covariance matrix.

For details about the meaning/interpretation of the above terms and equations and how they are obtained from the continuous-time equations, the reader is referred to one of the many textbooks available on Kalman filtering. Here, the focus is on how to derive models to handle the different GNSS time scales.

With this in mind, a very common clock model used for single-system GNSS is

$$\begin{pmatrix} \dot{b} \\ \dot{d} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} b \\ d \end{pmatrix} + \begin{pmatrix} w_b \\ w_d \end{pmatrix} \tag{7}$$

where *d* is the receiver clock *drift* and, w_b and w_d are the process noise values for the clock bias and clock drift states, respectively. The spectral densities of the latter two terms are defined based on the quality of the receiver's oscillator. For simplicity, we assume herein that the two process noise terms are uncorrelated, which is not true in general. However, accounting for the correlation between the two processes is well understood and can be easily added.

Starting Simple. As a starting point for the multi-system case we will, for the time being, use the substitution in equation (3) such that the state vector consists of the receiver clock bias for system one (b_1), the inter-system offset between the two systems (Δb_{12}), and a *single* clock drift.

Two things are worth noting at this point. First, for simplicity we are excluding any position or velocity states from the equations. From the perspective of the system model (i.e., equation (4)), the position and velocity states are unrelated with the clock states and thus can be added or removed as necessary. Of course, any practical system will include such states when processing real data, but these will only serve to complicate the discussion that follows here.

Second, by only including a single clock drift state, the model implicitly assumes that both GNSSs have the exact same frequency, which, as mentioned in the opening paragraphs, is generally not the case. We will revisit this assumption again later. Mathematically, the system model for the Kalman filter under consideration can be written as

$$\begin{pmatrix} \dot{b}_1 \\ \Delta\dot{b}_{12} \\ \dot{d} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ \Delta b_{12} \\ d \end{pmatrix} + \begin{pmatrix} w_b \\ 0 \\ w_d \end{pmatrix}, \qquad (8)$$

and the transition and process noise matrices are respectively given by

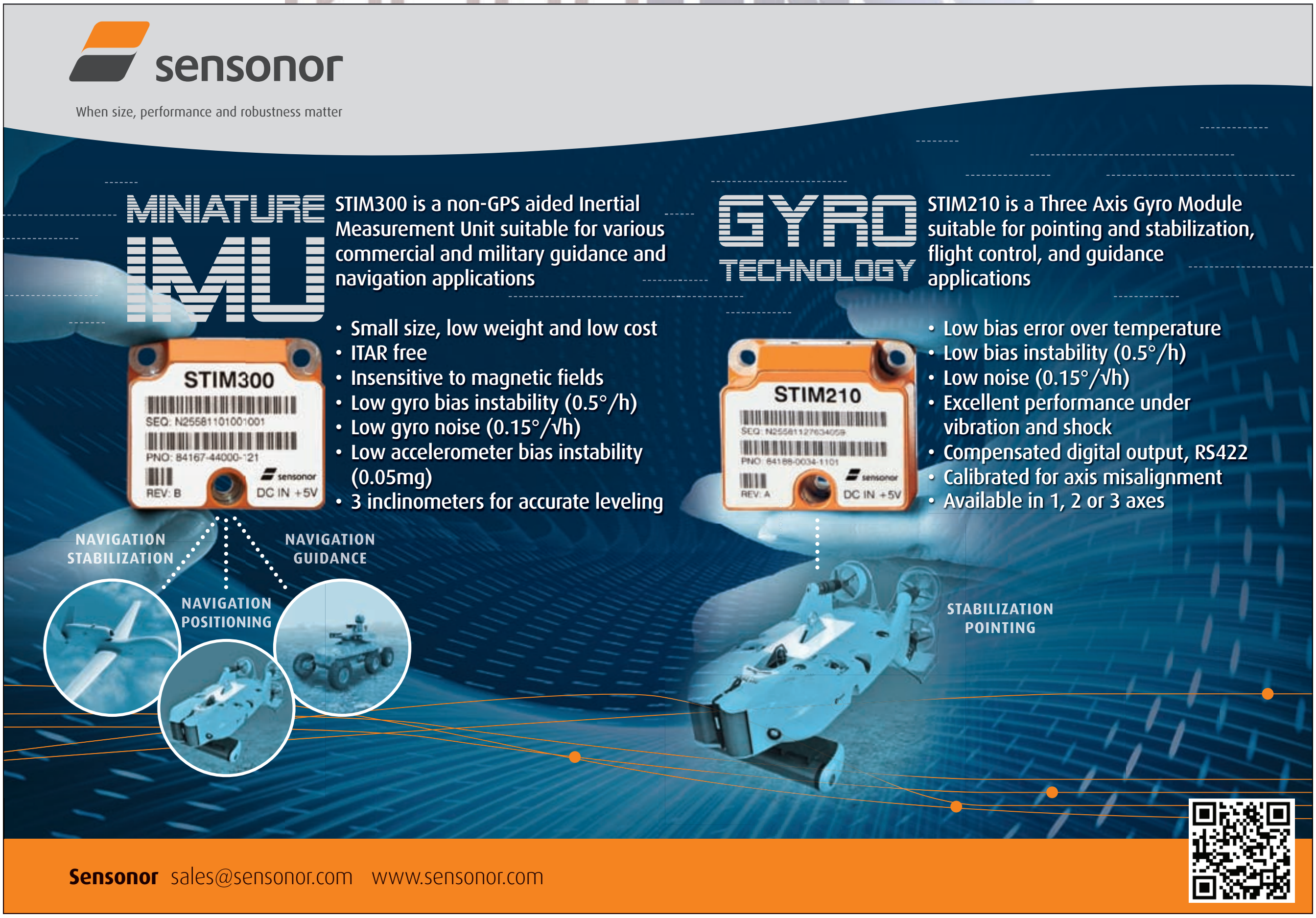
$$\Phi = \begin{pmatrix} 1 & 0 & \Delta t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (9)$$

$$Q = \left(\begin{array}{c|c|c} q_b \Delta t + q_d \frac{\Delta t^3}{3} & 0 & q_d \frac{\Delta t^2}{2} \\ \hline 0 & 0 & 0 \\ \hline q_d \frac{\Delta t^2}{2} & 0 & q_d \Delta t \end{array}\right) \qquad (10)$$

where q_b is the spectral density of the clock bias of the reference system, q_d is the spectral density of the clock drift, and Δt is the time difference between epochs *k* and *k*−1.

In other words, we treat the inter-system offset as a random constant. Adding extra systems to this model is therefore straightforward because one need only add an extra constant to the state vector (and corresponding rows and columns of zero to the process noise matrix).

One Clock Bias Per System. As with the least-squares case, a bit more housekeeping is associated with this model because we need to first define a reference system and then keep track of what inter-system offsets are being estimated. Consequently, we prefer to estimate a "separate" clock bias for each



system. To do this, we transform the state vector in equation (8) as follows:

$$\vec{x}' = T\vec{x} \qquad (11)$$

$$\begin{pmatrix} b_1 \\ b_2 \\ d \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ \Delta b_{12} \\ d \end{pmatrix} \qquad (12)$$

where the second state, b_2, is obtained directly from equation (3).

From equation (11), we can write the following two equations:

$$\dot{\vec{x}}' = \dot{T}\vec{x} + T\dot{\vec{x}} \qquad (13)$$

$$\vec{x} = T^{-1}\vec{x}' \qquad (14)$$

Substituting equations (4) and (14) into (13) and simplifying gives

$$\dot{\vec{x}}' = \underbrace{\left(\dot{T}T^{-1} + TFT^{-1}\right)}_{F'}\vec{x}' + \underbrace{\left(TG\right)}_{G'}\vec{w} \qquad (15)$$

which is of the same form as equation (4) and can thus be used to derive an equivalent discrete-time Kalman filter. Doing so for the state vector in equation (12) gives

$$\begin{pmatrix} \dot{b}_1 \\ \dot{b}_2 \\ \dot{d} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ d \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} w_b \\ w_d \end{pmatrix} \qquad (16)$$

and, by extension,

$$\Phi = \begin{pmatrix} 1 & 0 & \Delta t \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \qquad (17)$$

$$Q = \left(\begin{array}{c|c|c} q_b\Delta t + q_d\frac{\Delta t^3}{3} & q_b\Delta t + q_d\frac{\Delta t^3}{3} & q_d\frac{\Delta t^2}{2} \\ \hline q_b\Delta t + q_d\frac{\Delta t^3}{3} & q_b\Delta t + q_d\frac{\Delta t^3}{3} & q_d\frac{\Delta t^2}{2} \\ \hline q_d\frac{\Delta t^2}{2} & q_d\frac{\Delta t^2}{2} & q_d\Delta t \end{array}\right). \qquad (18)$$

The peculiar thing about equation (18) is that the process noise for clock bias states is *perfectly* correlated. In other words, the matrix is rank-deficient. Filter designers would normally shudder at this thought since perfect correlation amongst state estimates — that is, perfect correlation within P — normally leads to numerical instabilities.

In this case, however, we are only dealing with the uncertainty added in the Kalman filter's prediction step, not the uncertainty of the states themselves (i.e., P is still full rank because the measurements from each system still only update the clock bias for that system). In fact, the perfect correlation between the clock biases makes sense because this model was derived from the (albeit inaccurate) assumption that all system time scales have the same frequency, and thus predictions of their respective clock biases are equally uncertain.

Equations (16) to (18) are generally easier to program than equations (8) to (10) because — as with the least-squares case — a measurement from a particular system only relates to the clock bias for that system.

Back to Reality. We now return to the inaccurate assumption that all GNSS time scales have the same frequency. Fortunately, since the GNSS time scales drift with respect to each other at a very low rate — again, on the order of nanoseconds per day — the above models would likely be sufficient for relatively short processing periods.

For example, if an application processes 24 hours of data at a time, the accumulated drift between systems would be on the order of a few meters (i.e., a few nanoseconds scaled by the speed of light). If this is less than the magnitude of other errors — and the application can tolerate this level of error — then using the foregoing models probably will not adversely affect results. By extension, shorter processing runs will be even less affected.

However, what if an application requires a more rigorous model? In that case, the best approach would be to estimate a separate clock drift for each system. For the two-system case discussed earlier, the system model might look as follows:

$$\begin{pmatrix}\dot{b}_1\\ \dot{d}_1\\ \dot{b}_2\\ \dot{d}_2\end{pmatrix}=\begin{pmatrix}0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{pmatrix}\begin{pmatrix}\dot{b}_1\\ \dot{d}_1\\ \dot{b}_2\\ \dot{d}_2\end{pmatrix}+\begin{pmatrix}1&0&0&0\\0&1&1&0\\1&0&0&0\\0&1&0&1\end{pmatrix}\begin{pmatrix}w_b\\ w_d\\ w_{\Delta d1}\\ w_{\Delta d2}\end{pmatrix} \qquad (19)$$

where the subscript on the clock drift states denotes the system to which it applies. Also, note that each clock drift state is driven by the sum of a common process, w_d, and a system-specific process, $w_{\Delta di}$ (where *i* denotes the system). The common term models the receiver's oscillator while the system-specific term models the drift of each system's time scale (and is thus likely very small, by comparison).

The drawback of this model is that the size of the state vector increases by two for every new system used. An alternative but less rigorous approach would be to update equation (8) to model the inter-system offset (Δb_{12}), not as a random constant, but as a random-walk process driven by a very small spectral density.

Doing this and applying the transformation in equation (12) will yield the same transition matrix as in equation (17), but the process noise matrix in equation (18) would have to be increased by the following matrix

$$\Delta Q=\begin{pmatrix}0&0&0\\0&q_{\Delta b}\Delta t&0\\0&0&0\end{pmatrix} \qquad (20)$$

where $q_{\Delta b}$ is the spectral density used to drive the inter-system bias. The net effect of matrix (20) would be to ensure that the process noise for the different clock biases are not fully correlated, thus allowing them to change more independently of each other.

Summary

This article has looked at some approaches to estimating the clock biases for the different GNSSs, with particular focus on Kalman filter models. These models will be more important in the short term because the time offsets between the various GNSSs are not readily available.

As these offsets are included in the various navigation messages, the models presented here should — theoretically, at least — degenerate to the single-system case. Of course, this assumes that the accuracy of the broadcast offsets is sufficiently high. If not, the above models may still be needed to estimate the residual offsets. IG