Determination of Early-Late Discriminator Errors on Filtered BPSK Waveforms

Blair Fonville, *U.S. Naval Observatory*
Edward Powers, *U.S. Naval Observatory*
Demetrios Matsakis, *U.S. Naval Observatory*

ABSTRACT

GPS receivers commonly use Early-Late discriminators to estimate the position and rate of the internal Prompt codes with respect to the received waveforms. For ideal (non-filtered) input waveforms, the Early-Late discriminator steers the Prompt codes into perfect alignment. However, when tracking filtered BPSK waveforms, a misalignment of the Prompt code occurs, as a function of correlator spacing for a given filter transfer function. The misalignments will cause errors in the final pseudorange measurements, as discussed in previous publications authored by USNO and MITRE Corporation, and may introduce time biases between the different GPS code families, even when they are on a common carrier frequency. The receiver's pseudorange measurement errors are dependent on the degree of front-end filtering and on the tracking technique that is implemented.

In this paper we will focus on the pseudorange errors for BPSK signals. We will show how filtering and correlator spacing affects the pseudorange measurement, and we will show a way to measure and compute the resulting errors so that they could be corrected in receiver software.

INTRODUCTION

In modern GPS receivers, digital correlators are used to estimate the time-of-arrival of a received GPS signal. However, between the phase-center of the antenna and the receiver's analog-to-digital (ADC) converter, the signal passes through various components used for signal reception and signal conditioning. Each of these front-end components contribute to the overall group delay of the signal and, perhaps more importantly, each component has the potential to introduce timing biases among the signal's constituent code families (such between C/A and P(Y), or between P(Y) and M-code, etc).

As discussed in a previous publication [1] authored by USNO and MITRE Corporation, time biases will be introduced between the different GPS code families as they pass through the various stages of front-end filtering, even when they are on a common carrier frequency. The receiver pseudorange measurements of legacy GPS BPSK signals through a typical front-end filter will differ from the measurements of the modernized signals through the same filter. These pseudorange deviations are dependent on the degree of front-end filtering and on the tracking technique that is implemented.

In this paper we will focus of these pseudorange deviations. We will show how filtering affects the pseudorange measurement, and we will show a method to adjust for the resultant errors.

CAUSE

It is well known that the edges of codes which pass through limited-bandwidth filters will distort in various ways, defined by the filter's step response. Figure 1 demonstrates the group delay and amplitude differences for several ideal Butterworth filters.

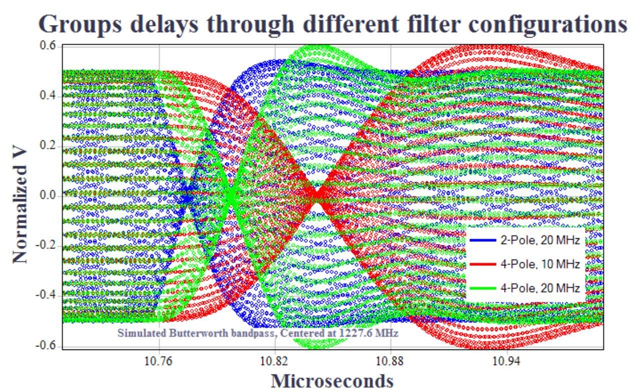

Figure 1. BPSK edges through various filters

Although these signal distortions are well understood, they give rise to errors of prompt-code alignment in Early minus Late correlators, and should not be dismissed for any high-precision GPS signal tracking. As an example, Figure 2 shows the values of a cross-correlation between an ideal "prompt" code and a filtered code (using a 5-pole 6 MHz two-sided Butterworth).

Proceedings of the 26th International Technical Meeting of the ION Satellite Division, ION GNSS+ 2013, Nashville, Tennessee, September 16-20, 2013

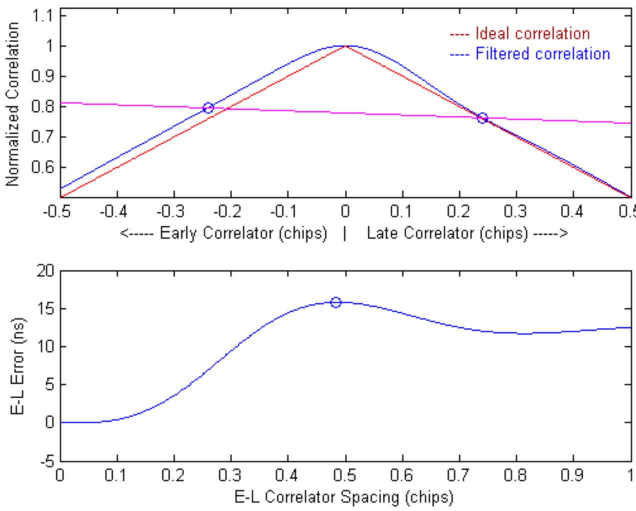

Figure 2. Effects of pulse filtering on correlator tracking

The top plot shows the normalized correlation values of both the early correlator (left half) and late correlator (right half) for an ideal correlation, or auto-correlation, and the cross-correlation. Two points are marked on the top of Figure 2, identifying ¼ chip spacings for a ½-chip E-L solution. The line connecting these points has a non-zero slope which results in a timing error of about 15 nanoseconds (ns), as shown in the bottom plot. Note that as the two points of the top plot converge to the middle, i.e. as the E-L spacing goes to zero, the slope of the connecting line approaches zero. The bottom plot shows this clearly.

ALGORITHM

To describe the method of estimating the correlation time errors resulting from waveform distortions of pseudorandom noise code sequences through filters, it is sufficient to simplify the code to three chips (Figure 3): a high chip surrounded by two low chips. This sequence is represented by Equation 1.

$$x(t) = u(t - T_C) - u(t - 2 \times T_C) \tag{1}$$

where T_C is the period of the code, and $u(t-a)$ is the Heaviside unit function defined by Equation 2.

$$u(\mathrm{t} - \mathrm{a}) = \begin{cases} 0 & \mathrm{t} < a \\ 1 & \mathrm{t} > a \end{cases} \tag{2}$$

Given a filter's impulse response, $h(t)$, or transfer function, $H(f)$, we can apply the filter to Equation 1. The system response of the signal $x(t)$ through the filter is

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau \\ &= \int_{-\infty}^{\infty} X(f)H(f)e^{j2\pi ft}df \end{aligned} \tag{3}$$

where $X(f)$ is the Fourier Transform of Equation 1:

$$X(f) \equiv F\{x(t)\} = \frac{e^{(-j2\pi fT_C)} - e^{(-j4\pi fT_C)}}{j2\pi f} \tag{4}$$

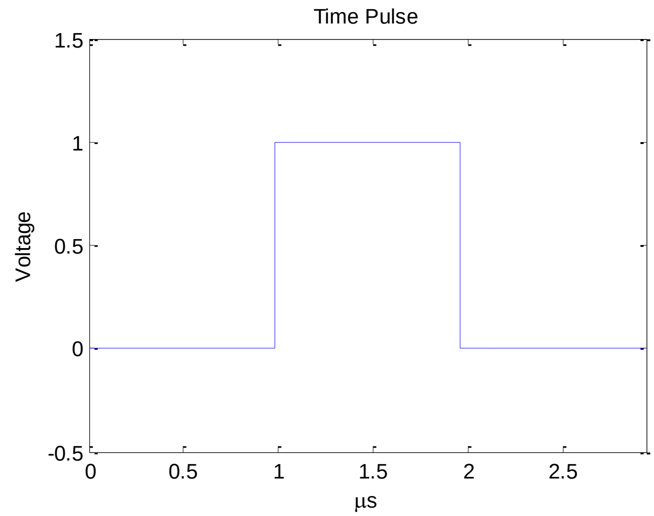

Figure 3. Single PRN pulse

When compared to the pulse in $x(t)$, the pulse in a filtered signal $y(t)$ will be shifted and distorted. The pulse distortion causes a non-zero $(E-L)/d$ slope, as shown in Figure 2. If the shift is denoted t_d, and we treat the distortion as a superimposed error source $e(t)$, then

$$y(t) = x(t - t_d) + e(t) \tag{5}$$

$$e(t) = \mathrm{y(t)} - x(t - t_d) \tag{6}$$

To illustrate, consider the transfer function $H(f)$ for a 6-pole, 4 MHz low-pass Butterworth filter, which is given as

$$H(f)_{butter} = \frac{1}{\left(-\frac{f^2}{16} + 0.12941jf + 1\right)\left(-\frac{f^2}{16} + 0.35355jf + 1\right)\left(-\frac{f^2}{16} + 0.48296jf + 1\right)} \tag{7}$$

The red waveform at the top of Figure 4 represents the Matlab computation of Equation 3.

The blue waveform at the top of Figure 4 is $x(t-t_d)$, which is $x(t)$ shifted by the delay of the filter, t_d. The shift, t_d, can be found by solving for the peak of the correlation between the original pulse and the filtered pulse.

$$f(t_d) = \left| \int_{-\infty}^{\infty} X(\acute{f})X(f)H(f)\, e^{j2\pi ft}df \right|_{max} \tag{8}$$

If we subtract the shifted version of $x(t)$, or $x(t-t_d)$, from $y(t)$ we get the distortion, $e(t)$. This is shown at the bottom of Figure 4.

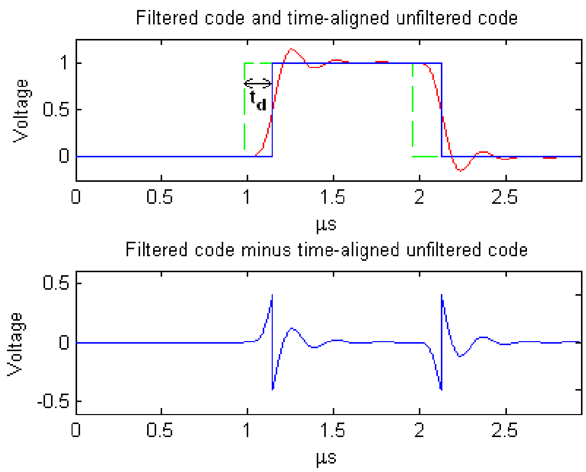

Figure 4. Top – Filtered pulse with time aligned pulse
Bottom – Filtered pulse minus time aligned pulse, e(t)

The distortion, $e(t)$, shown at the bottom of Figure 4 causes errors in Early-Late correlator tracking. Therefore, to estimate the tracking errors, we integrate $e(t)$ over the length of the Early and Late chips, and difference the two. An example is shown in Figure 5.

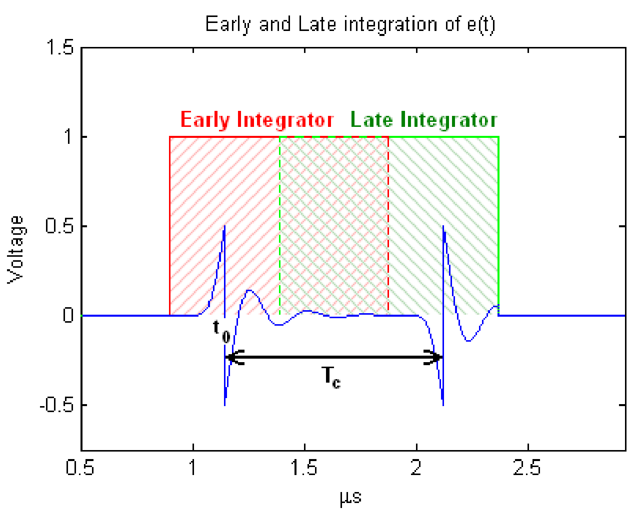

Figure 5. Integration of e(t)

Because the overlapping section will be cancelled during the Early minus Late operation, we can remove it from the integrations (Figure 6).

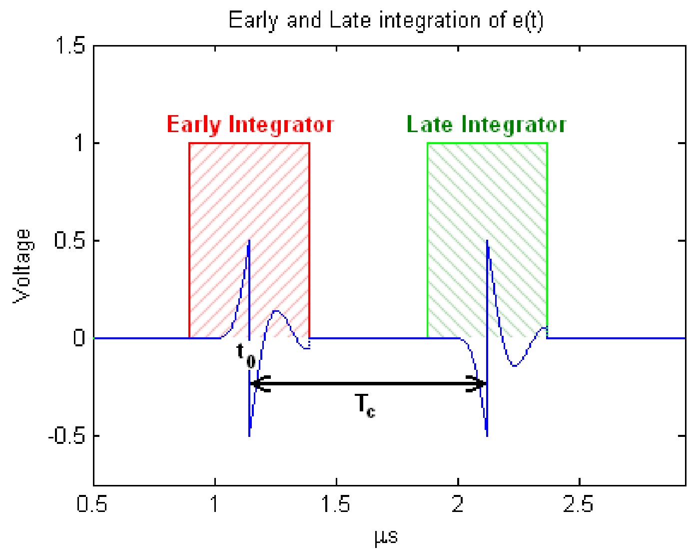

Figure 6. Reduced Integration of e(t)

The following equations result from these operations:

$$E = \int_{t_0 - T_c\frac{d}{2}}^{t_0 + T_c\frac{d}{2}} e(t)dt \qquad (9)$$

$$L = \int_{t_0 + T_c\left(1-\frac{d}{2}\right)}^{t_0 + T_c\left(1+\frac{d}{2}\right)} e(t)dt \qquad (10)$$

$$\varepsilon(d) = -\frac{1}{2}(\mathrm{E} - \mathrm{L}) \qquad (11)$$

where T_C, as before, is the period of the code; in our example $t_0 = T_c + t_d$, since the original pulse described by Equation 3 begins at T_C and the delay between the filtered pulse and the original pulse is t_d; E is the integration over the early block; L is the integration over the late block; and $\varepsilon(d)$ is the estimated tracking (pseudorange) error in seconds, given an Early-Late spacing d (in chips). The negative sign puts the error in terms of pseudorange as opposed to prompt code offset. In other words, a negative Early-Late correlator signal means that the receiver's prompt code is to the left of the correlation peak, which equates to a positive pseudorange value, and vice-versa.

Note that if this algorithm were performed with a filter of infinite bandwidth, the transfer function $H(s)$ of Equation 6 would be 1 for all frequencies, and t_d would be 0 for all time t. This would equate to correlating $x(t)$ with itself, and $e(t)$ would consequently equal 0 for all t. Therefore, the early and late integrations, and $\varepsilon(d)$, would also yield 0.

If Equation 11 is computed over all spacings d, from 0 to 1 (chip), for the example Butterworth filter, we obtain the curve shown in Figure 7.

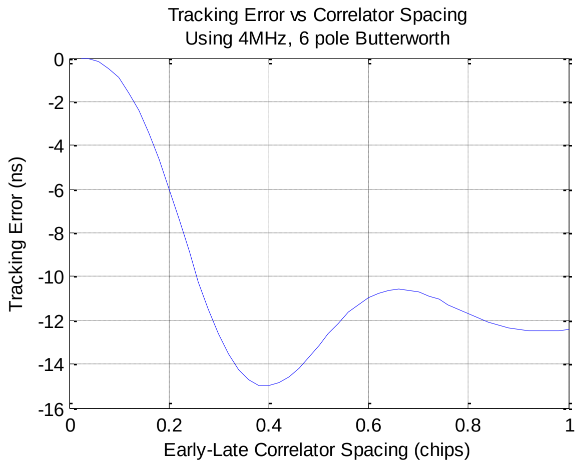

Figure 7. Evaluation of $\boldsymbol{\varepsilon}(\boldsymbol{d})$, for $\{\boldsymbol{d}: \mathbf{0} > \boldsymbol{d} \leq \mathbf{1}\}$

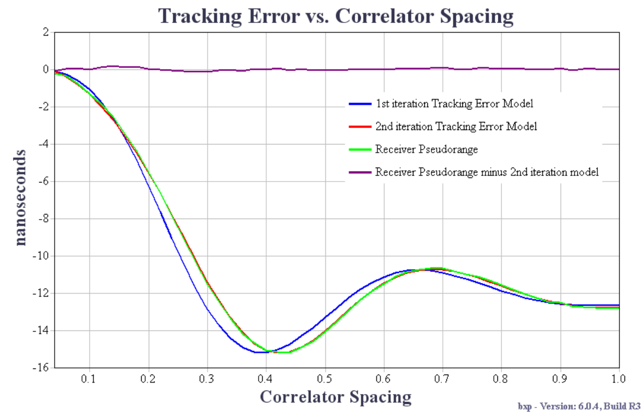

Figure 8. Error model compared to software receiver tracking

This curve, however, only represents a first approximation to the solution. Because the accuracy of the E-L discriminator is affected by code filtering, the solution may be improved upon by iteration using the results of Equation 11 to update the alignment of the filtered code in the evaluation of $e(t)$. For example, Equation 6 becomes

$$e(t)_{\mathrm{i}+1} = \mathrm{y}(\mathrm{t} + \varepsilon(d)_{\mathrm{i}}) - x(t - t_d) \tag{12}$$

for use in calculations of the updated $\mathrm{E}_{\mathrm{i}+1}$ and $\mathrm{L}_{\mathrm{i}+1}$. Then the new estimated tracking error, $\varepsilon(d)_{i+1}$, equals the previous estimated error, $\varepsilon(d)_i$, plus the new correction:

$$\varepsilon(d)_{i+1} = -\frac{1}{2}(\mathrm{E}_{\mathrm{i}+1} - \mathrm{L}_{\mathrm{i}+1}) + \varepsilon(d)_i \tag{13}$$

To illustrate, a GPS code (PRN 32) was simulated, filtered by the Butterworth filter given by Equation 7, and modulated onto a digital L1 carrier at a 500 MHz sample rate (which equates to an intermediate frequency of 75.42 MHz). A custom software-defined GPS receiver was written to track the signal over the 2.5 seconds of simulated data. The receiver was set up to track the same signal across 50 channels each using a unique E-L correlator spacing, ranging from 0.04 chips to 1 chip. The pseudoranges computed by the receiver for each of the 50 channels are plotted in Figure 8 along with the 1st and 2nd iterations of the modeled errors. For illustration purposes, a constant was removed from the receiver's pseudoranges to shift the tracking results to the error model traces, in order to allow us to more easily view the results on the same scale. It is the changes in the pseudoranges across correlator spacing that is important; the actual position along the y-axis is uninteresting. This plot demonstrates that the 2nd iteration model solution very well matches the results of a receiver's actual pseudorange tracking. When the 2nd iteration error model values are removed from the receiver pseudoranges, we see that the curve flattens, indicating that we have successfully removed the pseudorange's dependence on the implementation of the discriminator.

In other words, if this filter, or its equivalent 8MHz RF bandpass, appropriately described a receiver's front end filter, the receiver could employ this error model curve to remove the pseudorange error, for a given correlator spacing.

USING REAL FILTERS

The discussions above have demonstrated the correlation error effect, calling upon examples of ideal Butterworth filters, or, in general, any mathematical IIR/FIR filter representation. However, while these theoretical filters are useful to describe the problem, they do not provide a means to correct actual errors encountered using real hardware. It is generally difficult or impossible to model real hardware filters mathematically with enough accuracy to produce sub-nanosecond corrections. However, following the logic described in Section 0, we may import a filter's profile from actual measurements recorded from a network analyzer (NA). These NA measurements may then be used to find the estimated pseudorange errors when tracking through the characterized filter.

The process of properly utilizing S-parameter measurements from a network analyzer is slightly more involved than using theoretical IIR filters. In particular, because the theoretical transfer functions are normally ideally complex-conjugate symmetric, we are able to safely ignore the carrier and operate purely at baseband. With real-world filters characterized by network analyzers, we instead do our analysis at passband from 0-Hz to f_{span}, where f_{span} is the frequency span recorded by the network analyzer.

To find the system response (Eq. 3), the discrete baseband chip must be frequency shifted to the filter's passband-mapped center frequency, digitally filtered by the NA measurements, and shifted back to baseband.

While there are undoubtedly numerous approaches to employing the NA measurements, an outline of the method used for the analyses presented in this paper is as follows:

- Carefully calibrate the network analyzer, and then characterize the filter under test. For this research, we used an Agilent PNA N5222A network analyzer, and set it to record polar form characterizations – which stores a data file of frequency, real, imaginary values.

- Generate the baseband digital chip, using a sample rate equal to the frequency span used for the NA measurement. Typically the NA frequency span should be set to be much larger than the bandwidth of the filter under test, but not so large as to obscure visibility of the characteristics of the filter. For example, the N5222A permits a maximum of 20,000 points per recording. We used a frequency span of 200 MHz, which provides a sample every 10 kHz across the sweep.

- Generate a complex digital carrier, $e^{j\omega_c t}$. The frequency $\omega_c = 2\pi(f_{RF} - f_0)$, where f_{RF} is RF carrier frequency (generally the GPS L-Band frequency, in this paper), and f_0 is the first frequency value in the network analyzer collection.

- The NA does not provide complex conjugate values to 0 Hz, since it produces data only between the start and stop frequencies, in the positive frequency axis. We treat the filter as a positive-pass filter, and $x(t)e^{j\omega_c t}$ as an analytic signal. To compute $y(t)$ from Eq. 3, we can multiply the FFT of $x(t)e^{j\omega_c t}$ by the complex NA filter data, H, then multiply the inverse FFT of the product by $e^{-j\omega_c t}$.

$$y(t)' = real[F^{-1}\{F\{x(t)e^{j\omega_c t}\} * H\} * e^{-j\omega_c t}] \qquad (14)$$

However, because H is generally not complex conjugate symmetric, our results were made more consistent by replacing our down-conversion carrier $e^{-j\omega_c t}$ in Equation 14, with a filtered image of the up-conversion carrier $e^{j\omega_c t}$. Doing so helped ensure that most of the power in $y(t)$ was in the real component. Equation 14 then becomes

$$y(t) = real\left[F^{-1}\{F\{x(t)e^{j\omega_c t}\} * H\} * conj[F^{-1}\{F\{e^{j\omega_c t}\} * H\}]\right] \qquad (15)$$

The amplitude of the filtered code $y(t)$ is then normalized by the gain of the filter at ω_c:

$$C = \frac{1}{|H(\omega_c)|^2} \qquad (16)$$

HARDWARE MEASUREMENTS

For the purpose of this research, we procured a set of RF filters from Lark Engineering. A plot of the set of L1 filter gains is shown in Figure 9. The set includes filters of 50, 24, 8, and 3 MHz bandwidths, all centered at 1575.42 MHz.

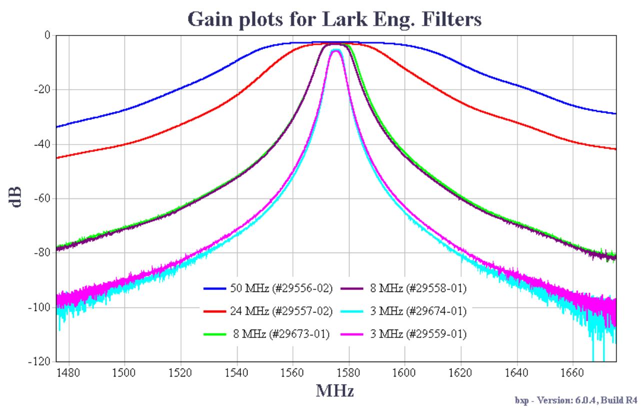

Figure 9. Gain plots for L1 test filters

Using the technique discussed in Section 0, we applied the algorithm to the filters to compute the error models shown in Figure 10.

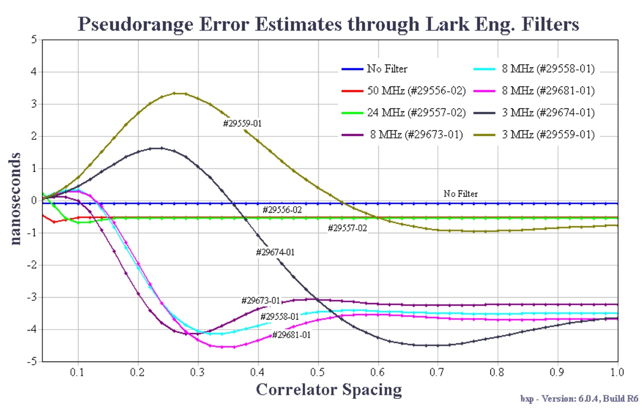

Figure 10. Error models computed for the filters under test

It is interesting to note that even filters which were constructed to be identical can have error models with relatively large differences. For example, the two 3-MHz filters, #29674-01 and #29559-01, were both manufactured under the same specifications. From Figure 10, we see that these two filters would produce tracking differences from up to almost 4 ns. This demonstrates that replacing filters blindly, for example to remedy a component failure, could produce unexpected results.

The trace identified as "No Filter" constitutes a benchmark test, in which data were collected from the network analyzer with the filter replaced by a simple barrel connector.

An 8-MHz IF filter (70-MHz center frequency), #29681-01, is also shown in Figure 10. This test was included to confirm that the model works independently of the carrier frequency.

VERIFICATION OF ERROR CURVES

The error curves shown in Figure 10 are the tracking errors estimated for a receiver using Early-minus-late correlators, for a given correlator spacing. To test these models, we generated a static zero-doppler GPS C/A code RF signals (data-less) at 1575.42 MHz, using a vector signal generator (Agilent E4438C). The signal generator output was passed through the filter under test (FUT) and then to an oscilloscope (Agilent DSO90254A) for binary collection. The oscilloscope and vector signal generator were both frequency locked to the same external 10 MHz reference signal to ensure phase coherency between data collections. The oscilloscope was set to trigger each data collection on an external 1 pulse-per-second (PPS) signal, derived from the reference clock, so that each binary collection began at the same location within the millisecond-long GPS C/A signal. The binary samples were then offloaded from the oscilloscope, and tracked using our software GPS receiver.

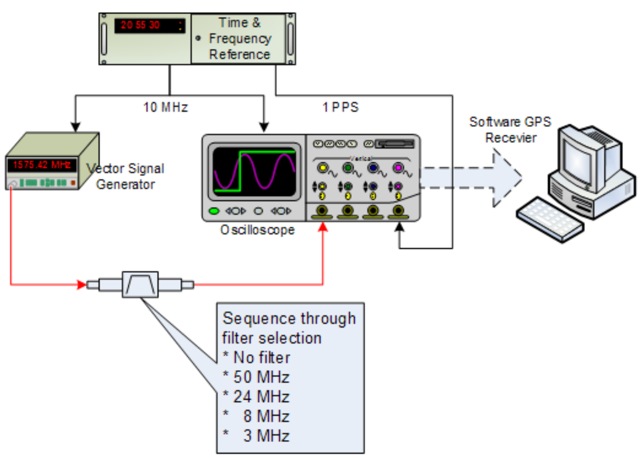

Figure 11. Setup for the ADC data collection through the test filters

The oscilloscope used has a memory depth of 205 million points. If set to its full sample rate, 20 GSa/s, we would only have a record 10 milliseconds long. Because our GPS receiver requires a few hundred milliseconds to lock-on, pull-in, and converge to a stable solution, we set the oscilloscope to sub-sample the L1 band at 500 MSa/s instead. This increases the record length to 410 milliseconds and still leaves about 150 MHz of unfiltered bandwidth. The sub-sampled center frequency is then 75.42 MHz.

The software GPS receiver is a custom receiver written specifically for this research. As such, certain considerations were made during its development. For instance, the number of receiver channels, the Early-minus-Late correlator spacing for each channel, the DLL bandwidths, etc, are all configurable by XML files. For the validation phase, we assigned 50 software channels each tracking the same PRN (in fact, the only PRN present in the binary file), and set each channel to use a different Early-minus-Late correlator spacing. Because each of the 50 channels was reading the same binary file, the signals going into each channel were identical; the only difference in each channel was the separation between the Early and Late correlators.

If we plot the pseudorange measurements through each filter, versus the separation between the Early and Late correlators, we get the results shown in Figure 12.

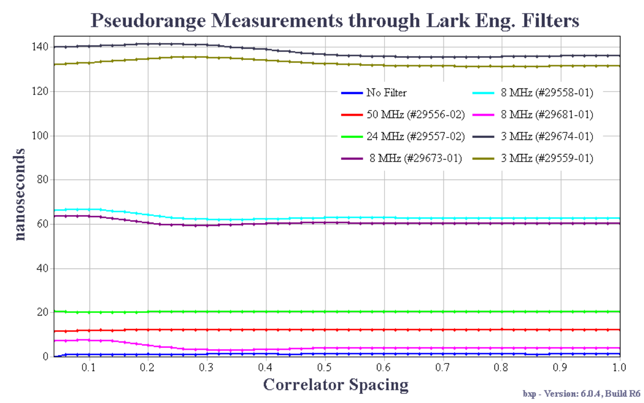

Figure 12. Pseudoranges vs. correlator spacing for each filter

Figure 12 show the raw pseudorange tracking measurements through each filter. For clarity, a constant bias was removed from the curves to center the control data set (i.e. "No Filter") around zero. Therefore, the vertical position for each of the curves represents the group delay of the given filter. Ideally, the curves shown would all be perfectly flat, since we know that the true range between the receiver and the transmitter is not a function of correlator spacing. However, the plots show that the group delays for many of the curves are not flat, when the pseudoranges are solved using variable correlator spacings. This is more obvious in Figure 13, where the pseudoranges have all been centered around zero, in order to put them all on the same scale.

Using the modeled errors (Figure 10) for these particular filters, we can correct the receiver's pseudorange estimates. Figure 14 shows the corrected pseudorange measurements, centered about zero. While the uncorrected data shows a peak-peak scale of about 8 nanoseconds, the corrected figure shows a scale of 0.5 nanoseconds.

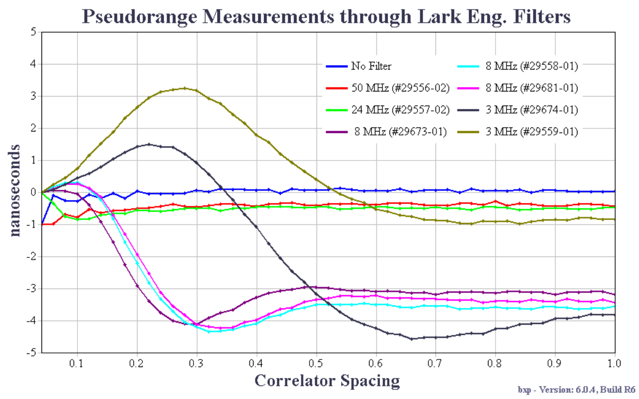

Figure 13. Pseudoranges centered about zero

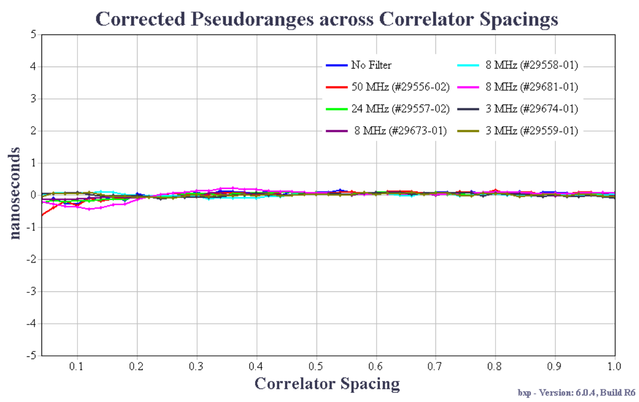

Figure 14. Pseudoranges corrected using modeled errors

SUMMARY AND CONCLUSIONS

We have shown that a receiver's pseudorange estimation can have a dependency on its tracking implementation – in particular, its filtering and correlator spacing. This is shown conclusively in Figure 13. Ideally, when range estimations are plotted against variable correlator spacings, the curves should be flat. We know that there is only one true range at a particular instance of time, between the receiver and the transmitter, irrespective of receiver tracking implementation. Because of this dependency, it may not be good enough to calibrate an antenna, conditioning filter, or receiver without also taking the receiver's tracking technique into consideration. For example, the effective delay of a filter within an RF distributor would depend of the tracking architecture of the receiver it feeds.

In this paper we have proposed a basic methodology which could be employed to largely remove the errors associated with tracking filtered GPS signals. While the success of applying these techniques depends on the ability to characterize all of the signals' filtering stages, ultimately it is the most restrictive filters in the chain which shape the errors. Luckily, the most restrictive filters in the chain are usually located near or within the GPS receiver itself, and not on the satellite, thus making them accessible for measurement.

DISCLAIMER

Although some manufacturers are identified for the purpose of scientific clarity, USNO does not endorse any commercial product, nor does USNO permit any use of this document for marketing or advertising. We further caution the reader that the equipment quality described here may not be characteristic of similar equipment maintained at other laboratories, nor of equipment currently marketed by any commercial vendor.

REFERENCES

[1] C. Hegarty, E. Powers, and B. Fonville, ION GNSS 2005 "*Accounting for Timing Biases between GPS, Modernized GPS, and GALILEO Signals"*