Adaptive Detection of Code Delay and Multipath in a Simplified GPS Signal Model*

Lisa M. Nelson
National Institute of Standards and Technology
Time and Frequency Division
Boulder, CO 80303-3328

Penina Axelrad
University of Colorado
Department of Aerospace Engineering
Boulder, CO 80309

Delores M. Etter
University of Colorado
Department of Electrical and Computer Engineering
Boulder, CO 80309

BIOGRAPHY

Lisa M. Nelson is an Electronics Engineer in the Time and Frequency Division at the National Institute of Standards and Technology in Boulder, Colorado. She received her B.S. and M.S. degrees in Electrical Engineering from the University of Colorado at Boulder.

Penina Axelrad is an Assistant Professor of Aerospace Engineering Sciences at the University of Colorado at Boulder. Her research and teaching are focused on GPS technology and applications. Dr. Axelrad received her Ph.D. in Aeronautics and Astronautics from Stanford University and her S.M. and S.B. from M.I.T.

Delores M. Etter is a Professor in the Electrical/Computer Engineering Department at the University of Colorado at Boulder. Her research is primarily in the area of adaptive digital signal processing. She received her B.S. and M.S. degrees in mathematics from Wright State University and her Ph.D. degree in Electrical Engineering from The University of New Mexico.

ABSTRACT

This paper introduces an application of adaptive filtering in a noise-canceling configuration that detects code delay and multipath in GPS using a simplified GPS signal model. An adaptive filtering process which allows a receiver to accommodate for changes in the environment surrounding the antenna is used. In this algorithm, filter weights from an LMS adaptive noise-canceling algorithm enable the receiver to determine the code delay from the direct satellite signal as well as the delay and attenuation of the multipath. A simplified signal model which includes idealized GPS C/A-code processing, multipath signals of varying delay and amplitude, interference of other C/A-codes, and quantization noise is used for the preliminary demonstration. Plans for application of this technique to a more realistic GPS signal model and assessment of the computational requirements are also discussed.

INTRODUCTION

Correlation permits the receiver to lock onto the PRN code and use it for measuring pseudorange to the satellite. The autocorrelation function for a short PN sequence where s(t)=±1 is [Navigation, 1980]

$$R(i) = \left(\frac{1}{\left(2^n - 1\right)}\right) \int_0^{\left(2^n-1\right)} s\left(t\right) \bullet s\left(t+i\right) dt \,. \qquad (1)$$

The symmetry of the peaks in the function helps the receiver to find the true code delay. This peak occurs every millisecond (1023 chips transmitted at 1.023 MHz) which simplifies acquisition so we can use the waveform for accurate time and consequently pseudorange measurements. The autocorrelation spectra for a typical Gold Code like the GPS C/A-code is shown in Figure 1. [Spilker(a), 1994, p. 99].

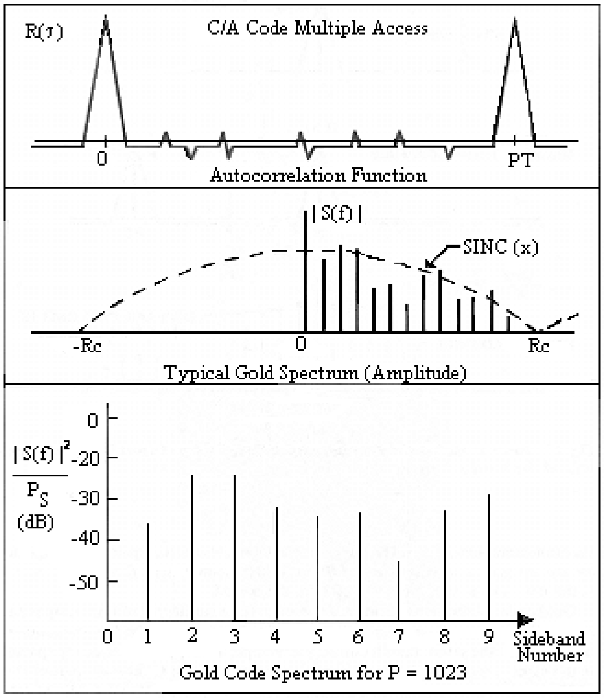

Figure 1: Gold Code Spectra for C/A-code [Spilker(a), 1994].

Figure 2 shows an ideal correlation peak when multipath is not present. Here the spacing of the correlator is not critical because the receiver can use the midpoint of the peak as the true delay of the signal. Figure 3 shows the correlation peak with multipath present and shows the location of the early and late samples of the correlator. In this case, the spacing of the correlator is directly involved with the determination of the signal delay because the midpoint of the peak is no longer the true occurrence of the code epoch.

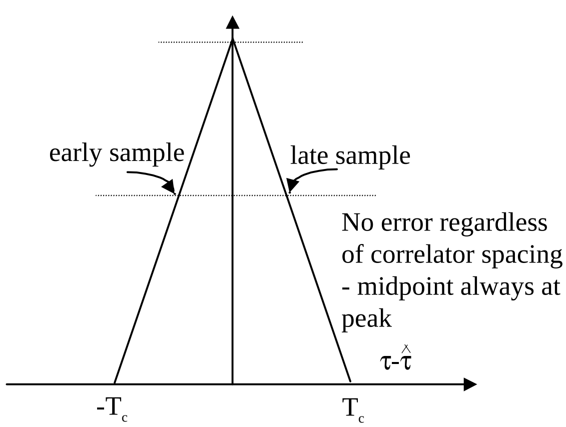

Figure 2: Correlation peak when multipath not present.

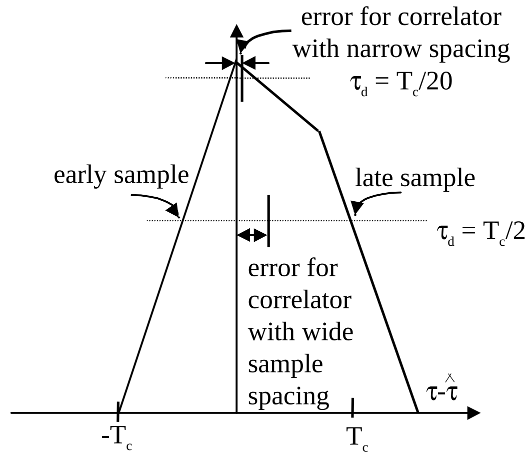

Figure 3: Correlation peak when distorted by multipath. The figure also shows the location of the early and late samples for a receiver with wide sample spacing and with narrow sample spacing [Enge, 1994].

One of the factors limiting performance of code-based differential GPS is multipath - the interference of reflected signals on the receiver tracking of the direct GPS signal. A number of techniques have been developed for reducing multipath including modifications to the code tracking loops, antenna designs, and carrier smoothing of code observations. Currently receivers can suppress the corruption of the direct signal by multipath at delays greater 1.5 chips [Braasch, 1995; Enge, 1994]. However, if the delay of the multipath signal is less than 1.5 chips then the receiver must use carrier aiding and/or narrow correlator spacing to reduce the influence of multipath on receiver tracking.

Modern receivers use both coherent and noncoherent DLL (delay locked loop) operations to track the peak of the autocorrelation function. Average multipath error is usually smaller with the coherent DLL; however, the coherent DLL can be disrupted by cycle slips, whereas the noncoherent DLL continues to function regardless of the operation of the carrier phase-tracking loop. By using narrow correlation with the noncoherent DLL, one can reduce the maximum multipath error by a factor of 10, and eliminate multipath with relative delays of 1 chip or greater when a 0.1 chip correlator spacing is used [Braasch, 1995].

It is also possible to use antennas which mask out lower elevation angles to reduce multipath from surfaces below the antenna. However, this becomes a problem if the user is mobile and the antenna is changing its position relative to the satellites, like an antenna on an airplane that is banking. Calibrating the multipath for static receiver locations is possible, but may require large amounts of storage and computer power [Kee, 1994]. Van Nee [Van

Nee, 1994] developed a technique called the Multipath Estimating Delay Lock Loop (MEDLL) that greatly reduces the multipath error by using multi-correlators for each channel. However, this technique requires more hardware and computations within the receiver.

Recently Breivik et al. developed an SNR-based approach to multipath correction in pseudorange measurements. The results show up to 50% improvement in differential GPS but have some limitation due to a bias which must be determined separately [Breivik, 1997].

In this paper we explore a way to detect and possibly reduce the effect of multipath in GPS pseudorange measurements based on an adaptive filtering technique. This approach should be able to adjust to changing multipath characteristics and thus could be applied to stationary or mobile receivers.

ADAPTIVE FILTERING

In this section the fundamentals of adaptive filtering and various ways to implement it are discussed. A review of the properties and behavior of the traditional adaptive filter is presented to illustrate how to detect code delay and multipath in the GPS signal.

An adaptive filter is a self-optimizing, self-designing system that changes its effects on a signal according to its environment. It does this according to a specified performance measure. In other words, it is a trained system that repairs itself according to the situations encountered over time. Figure 4 shows the adaptive transversal filter with an output that can be written as [Widrow, 1985]

$$y_k = \sum_{l=0}^{L} w_{lk} x_{k-l} = \mathbf{X}_\mathbf{k}^\mathbf{T}\mathbf{W}_\mathbf{k} \,, \qquad (2)$$

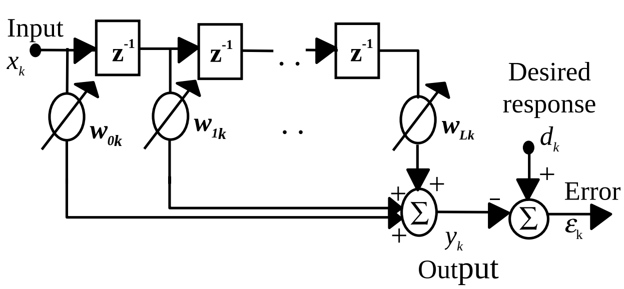

Figure 4: The adaptive linear combiner as a transversal filter. [Widrow, 1985]

where $\mathbf{X_k} = [x_k \; x_{k-1} \; . \; . \; . \; x_{k-L+1}]^T$ is the vector of current and previous L-1 input values, and $\mathbf{W_k} = [w_{0k} \; w_{1k} \; . \; . \; . \; w_{Lk}]^T$ is the weight vector at time k. It is the weight vector in this equation that is used to adapt the filter so that the output y_k agrees as closely as possible with the desired response.

The adaptation of $\mathbf{W_k}$ is accomplished by comparing the output signal y_k with a desired "training" signal d_k and then using the difference in this comparison to adjust or optimize the weight vector and minimize this "error." The error signal is written as [Widrow, 1985]

$$\varepsilon_k = d_k - y_k = d_k - \mathbf{X}_\mathbf{k}^\mathbf{T}\mathbf{W}_\mathbf{k} \,. \qquad (3)$$

Most adaptive processes are aimed at minimizing the mean squared value or the average power of the error signal. Using Equation (3) to obtain the instantaneous squared error, we have

$$\varepsilon_k^2 = d_k^2 + \mathbf{W}_\mathbf{k}^\mathbf{T}\mathbf{X}_\mathbf{k}\mathbf{X}_\mathbf{k}^\mathbf{T}\mathbf{W}_\mathbf{k} - 2d_k\mathbf{X}_\mathbf{k}^\mathbf{T}\mathbf{W}_\mathbf{k} \,. \qquad (4)$$

Now assuming that ε_k, d_k and $\mathbf{X_k}$ are statistically stationary we take the expected value of Equation (4) over k and find:

$$E\left[\varepsilon_k^2\right] = E\left[d_k^2\right] + \mathbf{W}_\mathbf{k}^\mathbf{T} E\left[\mathbf{X}_\mathbf{k}\mathbf{X}_\mathbf{k}^\mathbf{T}\right]\mathbf{W}_\mathbf{k}^\mathbf{T} - 2E\left[d_k\mathbf{X}_\mathbf{k}^\mathbf{T}\right]\mathbf{W}_\mathbf{k}^\mathbf{T} \,. \qquad (5)$$

Next define the input correlation matrix,

$$\mathbf{R} = E\left[\mathbf{X}_\mathbf{k}\mathbf{X}_\mathbf{k}^\mathbf{T}\right], \qquad (6)$$

and $\mathbf{P}$ as

$$\mathbf{P} = E\left[d_k\mathbf{X}_\mathbf{k}\right] = E\left[d_k x_{0k} \quad d_k x_{1k} \quad . \; . \; . \quad d_k x_{Lk}\right] , \qquad (7)$$

which is the vector of cross-correlations between the desired response and the input components. Now the mean squared error (MSE) may be written as

$$\begin{aligned} \xi &= E[\varepsilon_k^2] = E[(d_k - y_k)^2] \\ &= E[d_k^2] + \mathbf{W}_\mathbf{k}^\mathbf{T}\mathbf{R}\mathbf{W}_\mathbf{k} - 2\mathbf{P}^\mathbf{T}\mathbf{W}_\mathbf{k}^\mathbf{T} \,. \end{aligned} \qquad (8)$$

Assuming that the input components and the desired response input are stationary stochastic variables results in an MSE that is a quadratic function of the components of the weight vector. The resulting "bowl-shaped" paraboloid is called the performance surface for these weights. The surface is always concave upward, because the values of ξ are squared and therefore always positive values. The point at the bottom of the bowl is the point of the minimum mean squared error, where we find the optimal weight vector. This is important because a quadratic performance function will always give a single global optimum; no local minima exist [Widrow,1985].

WEIGHT UPDATE ALGORITHMS

Graphical representation of the performance surface shows that the minimum error is attained by seeking the minimum of the performance surface. Using a weight update algorithm that moves in the opposite direction of the gradient should seek the minimum of the performance surface. For instance, $w_{k+1} = w_k - \mu\nabla_k$ is the steepest descent algorithm and adapts the weights, according to the adaptation constant μ, in a direction opposite to the gradient ∇_k.

The gradient of the performance surface, specified by $\nabla(\xi)$, can be obtained by differentiating Equation (8) to obtain the column vector

$$\nabla \equiv \frac{\partial\xi}{\partial\mathbf{W}} = \left[\frac{\partial\xi}{\partial w_o}\ \frac{\partial\xi}{\partial w_1} \cdots \frac{\partial\xi}{\partial w_L}\right] = 2\mathbf{RW} - 2\mathbf{P}\,. \quad (9)$$

Then to obtain the minimum error the weight vector is set to its optimal value, called $\mathbf{W}^*$, where the gradient is zero:

$$\nabla = \mathbf{0} = 2\mathbf{RW}^* - 2\mathbf{P}\,. \quad (10)$$

If we assume that $\mathbf{R}$ is nonsingular and solve for $\mathbf{W}^*$, we find that

$$\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P}\,. \quad (11)$$

This is an expression of the Wiener-Hopf equation in matrix form [Widrow, 1985]. Substituting this into Equation (8), we can obtain the expression for the minimum mean-square error as

$$\xi_{min} = E\left[d_k{}^2\right] - \mathbf{P}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{P} = E\left[d_k{}^2\right] - \mathbf{P}^{\mathbf{T}}\mathbf{W}^*\,. \quad (12)$$

Unfortunately, the statistical properties of the desired signal and the input signal are not always known, so we can not directly compute the optimal weight vector for the transversal filter. This means that we must form an estimate for the correlation matrix $\mathbf{R}$ and the cross-correlation vector $\mathbf{P}$ to obtain an approximation to the optimal Wiener solution. We attempt to do this by using an iterative process that updates the optimal weight vector according to the currently available input data and desired signal data. This is acceptable because many applications have slowly changing statistics that make it possible to estimate these parameters in real time. This is why using a gradient method is well suited for finding the minimum of our performance surface. The "bowl-shaped" surface gives the filter a way to slide its way to the bottom and use gradient measurements as its guide. The gradient also eliminates the need for algorithms to estimate the correlation matrix in order to estimate this minimum correctly. To illustrate the evolution of the gradient estimation we show algorithms that use signal correlation information as well as those that do not.

Initially, if we have information about the input signal correlation, we may make use of Newton's algorithm [Widrow, 1985], where the weights are updated according to

$$w_{k+1} = w_k - \frac{1}{2}\mathbf{R}^{-1}\nabla_k\,. \quad (13)$$

This algorithm may be generalized to include a step-size parameter μ for controlling the convergence rate. The modified Newton's algorithm [Widrow, 1985] is then

$$w_{k+1} = w_k - \mu\mathbf{R}^{-1}\nabla_k\,. \quad (14)$$

However, if there is no input correlation matrix knowledge the search for the minimum of the performance surface can still be accomplished by using gradient information. This is done by using the steepest descent algorithm [Widrow, 1985]

$$w_{k+1} = w_k - \mu\nabla_k\,, \quad (15)$$

which systematically converges to the minimum, assuming that μ is carefully chosen and if the statistical characteristics of the desired and input signals are constant or slowly changing. Therefore**,** we need to find an algorithm that finds an efficient estimate of the gradient, which we call $\hat{\nabla}_k$.

THE LMS ALGORITHM

The LMS, least mean square, algorithm is an approach that uses an estimate for the gradient to descend on the performance surface [Widrow, 1985]. It is generally the best choice for systems where the input vector and the desired response are available at each iteration. LMS estimates $\hat{\nabla}_k$ by using ε_k^2 itself, so at each iteration in the adaptive process the gradient estimate is

$$\hat{\nabla}_k = \begin{bmatrix} \frac{\partial\varepsilon_k^2}{\partial w_0} \\ \vdots \\ \frac{\partial\varepsilon_k^2}{\partial w_L} \end{bmatrix} = 2\varepsilon_k \begin{bmatrix} \frac{\partial\varepsilon_k}{\partial w_0} \\ \vdots \\ \frac{\partial\varepsilon_k}{\partial w_L} \end{bmatrix} = -2\varepsilon_k\mathbf{X}_k\,. \quad (16)$$

With this simple estimate of the gradient we can use a modified version of the method of steepest descent to give the LMS algorithm [Widrow, 1985]:

$$\begin{aligned}\mathbf{W_{k+1}} &= \mathbf{W_k} - \mu\hat{\nabla}_k \\ &= \mathbf{W_k} + 2\mu\varepsilon_k \mathbf{X_k}\,,\end{aligned} \tag{17}$$

where μ is the gain constant that regulates the speed and stability of adaptation. Theoretically the convergence of the weight vector is assured in the transversal filter case when

$$0 < \mu < \frac{1}{\mathbf{tr}[\mathbf{R}]} \quad \text{[Widrow, 1985]}, \tag{18}$$

where the eigenvalues, i.e., diagonal elements, of **R** must be known. For practical applications we use,

$$0 < \mu < \frac{1}{(L+1)(V)}\,, \tag{19}$$

where L+1 is the number of weights and V is the signal power [Widrow, 1985]. This is useful because the term (L+1)(V) estimates the trace of **R**. Typically values of μ on the order of a tenth of the upper bound given in Equation (19) are used. The LMS algorithm introduces a large component of noise into the system because the gradient components are not averaged and therefore do not form a smooth transition to the minimum of the performance surface. However, the noise is attenuated by the adaptive process because as time passes the adaptation serves as its own low-pass filter. The value of the step size μ can also be chosen to help reduce the overall noise that is introduced into the gradient estimation.

If the input signals are non-stationary and have varying power, it changes the value of the correlation matrix **R** and consequently also changes the value for μ. In this case we can use the normalized LMS (NLMS) technique [Haykin, 1996]. In this technique, the LMS value of μ is updated along with the weight equation, using

$$\mu = \frac{\tilde{\mu}}{(a+V)}\,, \tag{20}$$

where 0< $\tilde{\mu}$ <2, a is a small positive constant to keep the denominator greater than 0, and the power of the signal is calculated over a small moving window of the input signal data. This maintains a mean of the signal near 0, because over the length of the windowed data the change in the mean is small enough to not change the signal power significantly. If the length of the window is not small enough, a running mean needs to be computed along with the adaptation.

The next section will discuss the simulated code delay and multipath GPS signal and the results of using an adaptive process to detect these delays.

GPS CODE SIMULATION

The GPS 1023 chip C/A-code was simulated using a high-level computer language on a workstation [Krauss, 1994]. It is assumed that the carrier had been removed by frequency locking so that only the code was present. Each bit in the code was initially sampled 10 times and the full code was repeated 3 times resulting in a total of 30690 samples. The PRN code chosen was for SV#1 [ICD, 1993, p. 8, 30].

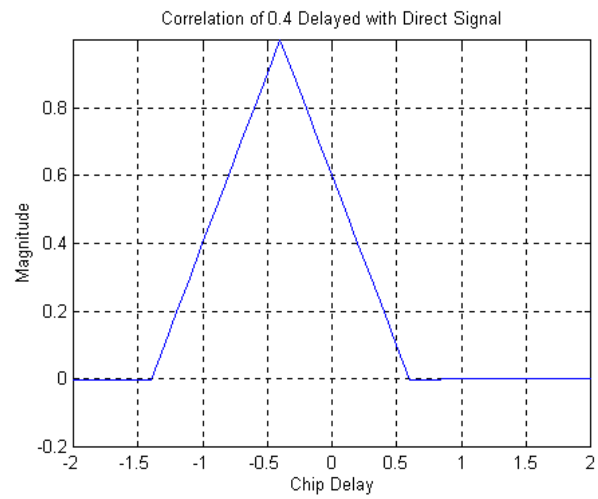

Figure 5: Cross-correlation of 0.4 delayed signal with direct signal.

Figure 5 shows the cross correlation of a direct signal and a signal delayed by 0.4 chip with no multipath introduced. We see the shift to the left by 0.4 chips accordingly in the correlation function. The correlation is based upon a single simulation. The function which estimates the cross correlation of the two sequences is set to normalize the output sequence so that the autocorrelation at zero lag is identically 1, corresponding to 0 dB. In Figure 6 zero mean Gaussian noise is added to the signal (SNR = 0.4) We still see a symmetric peak but the amplitude is 3 dB lower and the side lobes are noisy.

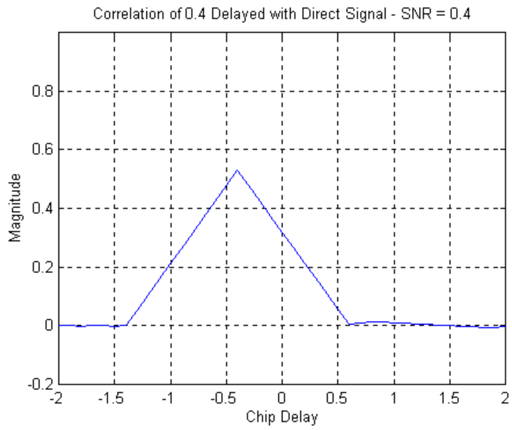

Figure 6: Cross-correlation of 0.4 delayed signal with direct signal - SNR = 0.4.

Multipath is then added to the signal without noise and the cross correlation is shown in Figure 7. The multipath-corrupted signal is created from the delayed direct signal by attenuating the signal from an amplitude of 1 to an amplitude of 0.2 and delaying the attenuated signal by an additional 0.3 chip, and finally adding it back to the original delayed-direct signal. We see a slight change in the symmetry of the correlation function shown in Figure 6 as compared to Figure 4. In Figure 8 noise (SNR = 1) was added to the multipath signal to see how it affects the correlation peak. In this case we see a slight shift in the amplitude of the peak to the left and a decrease in amplitude of the peak. This will lead to larger errors in tracking when half chip correlation spacing is used. Tracking error using a narrow correlator with half chip spacing is shown in Figures 7 and 8. With 0.1 chip error this makes the estimated correlation error comparable to that of the standard correlator errors shown earlier in Figures 2 and 3.

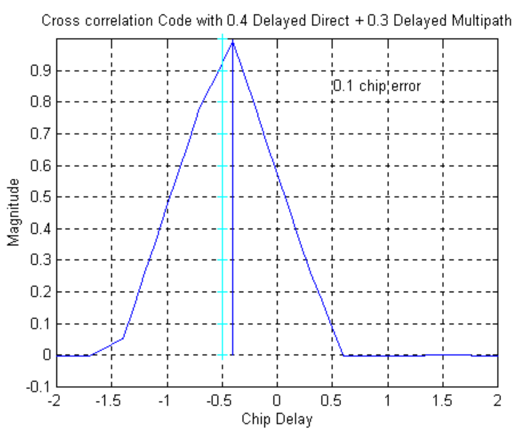

Figure 7: Cross-correlation of 0.4 delayed direct with 0.3 delayed multipath signal without noise.

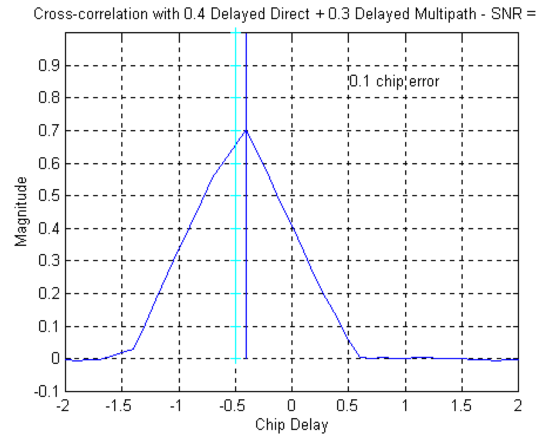

Figure 8: Cross-correlation of 0.4 delayed direct with 0.3 delayed multipath signal with an SNR of 1.

USING ADAPTIVE ALGORITHM TO IDENTIFY DELAY AND MULTIPATH - DISCUSSION OF TEST CASES

Figure 9 shows the structure for the LMS algorithm and how it is applied to determine the delay of the C/A-code between the satellite and the receiver. First we will develop the case where no noise n_k is added to the received satellite signal d_k. In this case x_k is the receiver-generated C/A-code, the simulated code. The other input to the filter s_k is a delayed version of the receiver generated C/A-code x_k. A delay of 0.3 chip (z^{-ma}, ma=3) was introduced into the signal s_k to simulate the offset of the code in transmission from the satellite to the receiver. Therefore, d_k is the simulated satellite signal at the receiver. This offset is used in calculating the range from the receiver to the satellite. Of course this offset must be adjusted by the number of complete C/A-code repeats between the satellite time of transmission and the receiver apparent time of reception.

Figure 10 shows the results of the adaptation. The weight vector of the filter modifies the signal x_k to make it look like the signal d_k. By using the output of the weight vector we can determine the 0.3 chip delay of the direct satellite signal from the receiver generated signal. The small bumps past 1 chip delay appear to be artifacts of the number of samples per bit of the 1023 bit length of the PRN code.

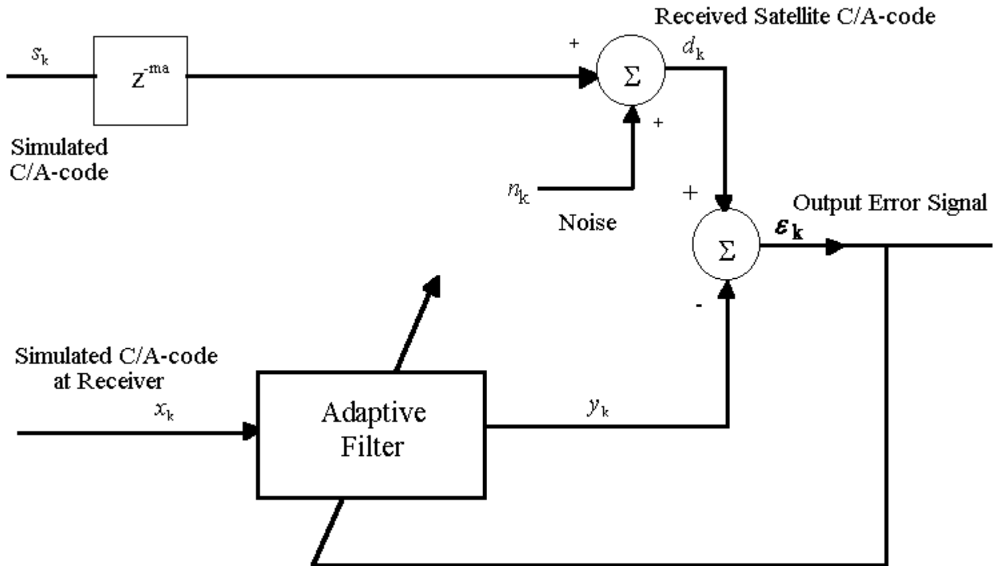

Figure 9: Block diagram of adaptive filter for GPS C/A-code delay detection.

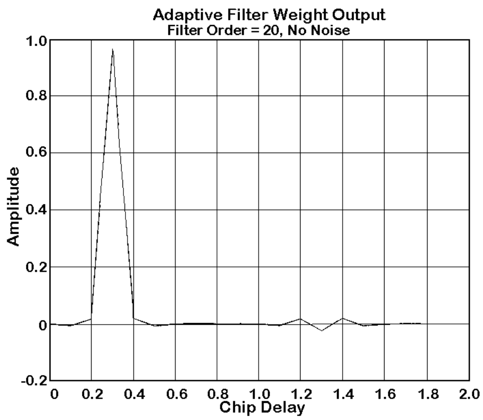

Figure 10: Adaptive filter weight output values for filter order = 20.

CODE DELAY DETECTION - WITH NOISE

For the next simulation, noise n_k is added to the signal to create a representation for a more realistic received satellite signal d_k. The noise n_k is modeled as Gaussian with zero mean and represents the noise introduced to the signal up to the low-noise amplifier in the receiver. Figure 11 shows the results of the adaptation. Simulations dependably determined the code delay peak down to a SNR of 0.004, that is, -23.9 dB.

Now we add multipath to the incoming signal and see how the adaptation performs. The filter block diagram for this case is shown in Figure 12. We will disregard the noise component n_k for this adaptation. The multipath is created by delaying and attenuating the incoming delayed satellite signal. Once again the receiver generated C/A-code is the x_k signal into the adaptive filter. The desired signal d_k is the delayed version of the direct signal plus the multipath signal, delayed from the tardy direct signal. For this adaptation we use a direct signal delayed by 0.3 chip (z^{-ma}, ma=3) and the multipath delayed by 0.2 chip (z^{-mb}, mb=2) and attenuated by 0.2 (gain) from the delayed direct signal. The adaptation should occur such that the weights of the filter indicate the delay of the direct signal from the receiver generated signal, as well as the multipath delay from the direct signal. Figure 13 shows the weight output vector for this adaptation. The peaks determine the correct delays and maintain an amplitude ratio of 0.2/1 indicating the ratio of amplitudes between the original multipath and direct signals.

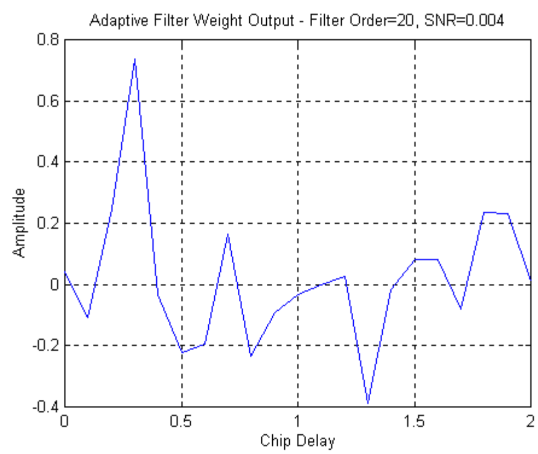

Figure 11: Adaptive filter weight output values for SNR = 0.004 (-23.9 dB) and filter order = 20.

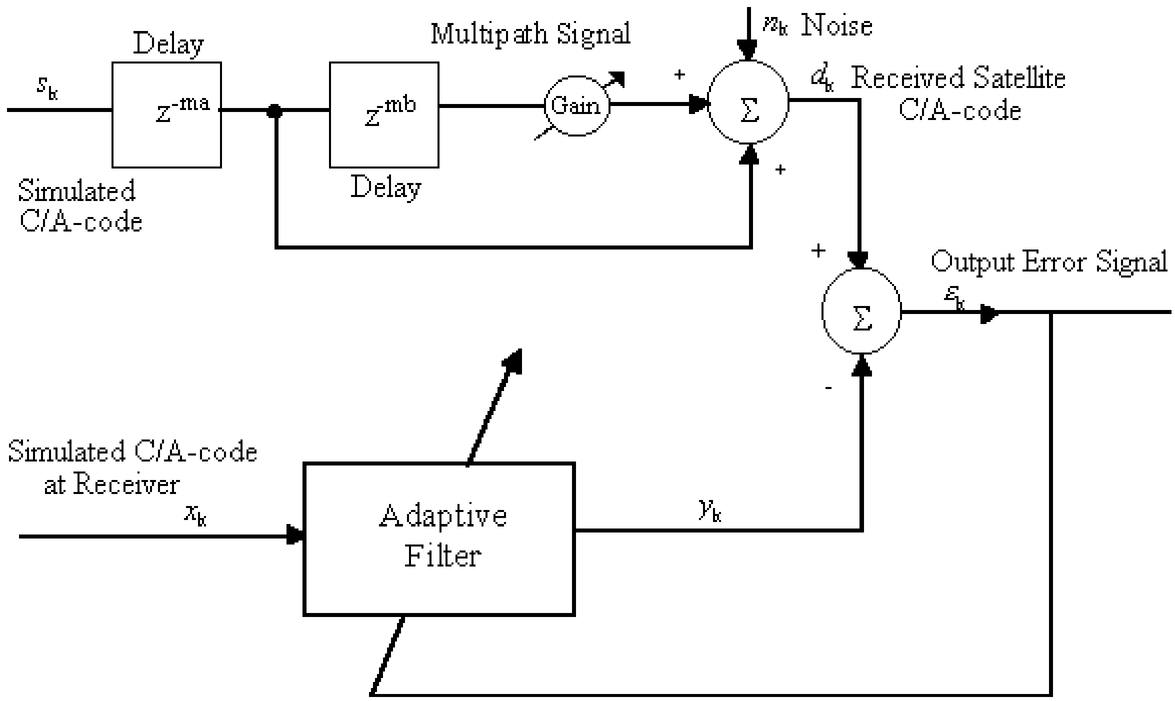

Figure 12: Block diagram of adaptive filter for GPS C/A-code delay and multipath detection.

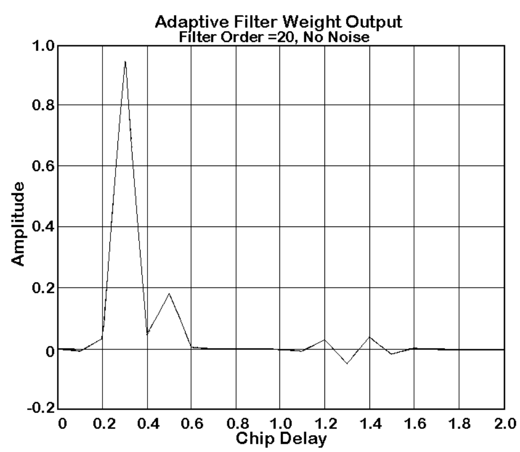

Figure 13: Adaptive filter weight output values with multipath for filter order = 20.

MULTIPATH DETECTION - WITH NOISE

Once again, we add the noise n_k back into received satellite signal simulation d_k and repeat the adaptation. Figure 14 shows the process with an SNR of 1. The adaptation was able to reliably determine the multipath delay down to an SNR of 0.4, shown in Figure 15. Once the SNR was lower than 0.4 it became difficult to locate the multipath peak from the noise in the adaptation. As SNR decreases, so does the value of μ. This is because μ is dependent upon the average signal power. This means that it will take more signal samples for the adaptation process to converge, which could account for the loss of detection with SNR values less than 0.4. For all of these simulations 30690 samples were used (3 repeats of the full C/A-code) for the adaptation.

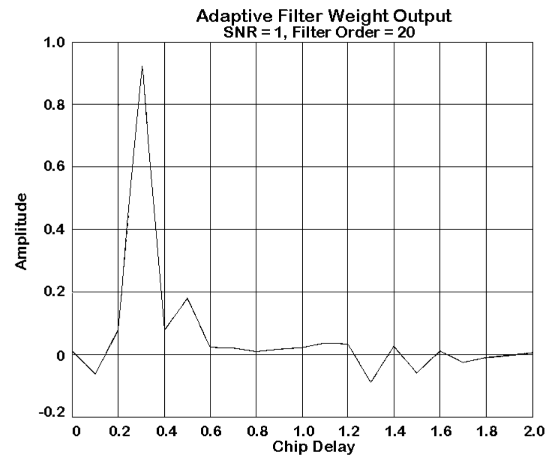

Figure 14 : Adaptive filter weight output values for SNR = 1 and filter order = 20.

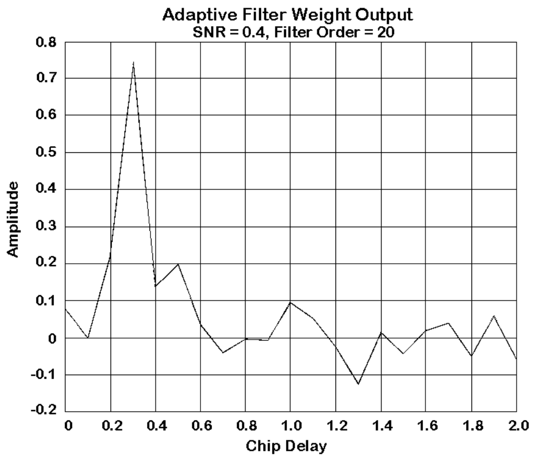

Figure 15: Adaptive filter weight output values for SNR = 0.4 and filter order 20.

MULTIPATH DETECTION - QUANTIZED d_k

Simulations were also performed with a 9-level quantizer on d_k, the incoming satellite signal. This simulates noise that is added by the A/D converter in the receiver. When the quantizer was used, the filter was able to pick out the multipath only down to an SNR of 0.6 (see Figure 16). This is expected because the quantizer introduces more error into the system and therefore degrades the performance of the filter. Figure 17 shows the quantizer with an SNR of 1 to show the effect of noise on the output in comparison to Figure 14 where a quantizer is not used.

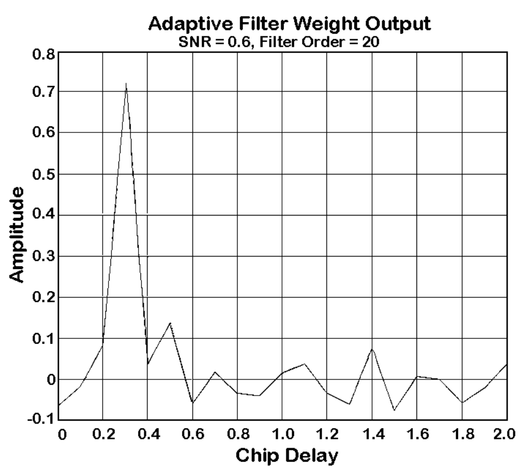

Figure 16: Adaptive filter weight output for 9-level quantizer on d_k with a SNR of 0.6 and filter order = 20.

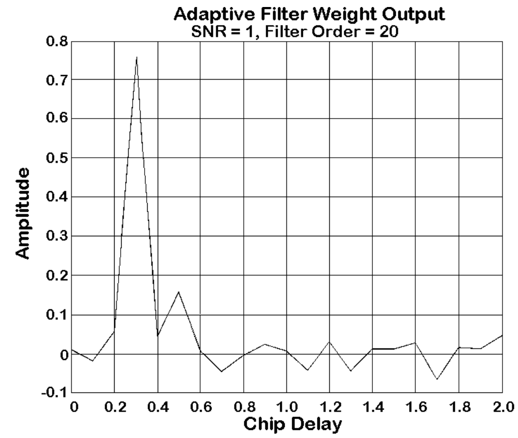

Figure 17: Adaptive filter weight output for 9-level quantizer on d_k with a SNR of 1 and filter order = 20.

MULTIPATH DETECTION - INTERPOLATION

The resolution of the peak in the signal chip delay adaptation can be increased with more repeats of each code bit in the simulation code. Initially we had 10 points per chip setting the delay resolution limit at 0.1 chip. The increased resolution is shown in Figures 20-22, for various levels of SNR, where the number of C/A-code chip repeats was increased from 10 to 100 using interpolation (see Figure 19). Interpolation is used, which increases the number of samples per bit by inserting zeroes between the original data values and then low pass filtering the data [Krauss, 1994]. This essentially performs an exponential interpolation between data points. Figure 18 shows the adaptation using 100 points per chip with no noise introduced to the received satellite signal d_k. By increasing the number of points we are able to see a more narrow, defined peak at the delay locations. Once again the bump in the data beyond 1 chip delay is somehow related to the number of samples per bit of the C/A-code in the simulation.

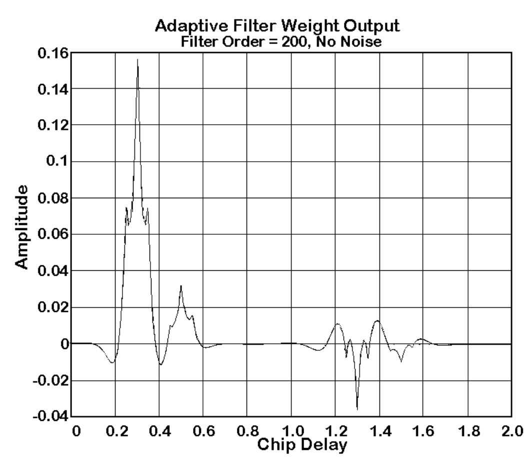

Figure 18: Adaptive filter weight output values for filter order = 200, no noise present.

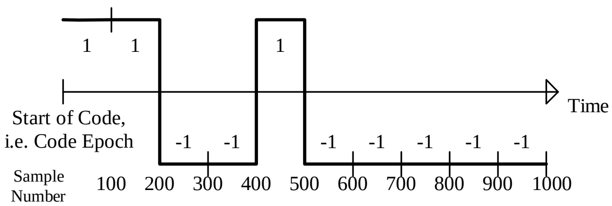

Figure 19: First 10 chips of C/A-code for PRN signal, 100 repeats per bit [ICD,1993].

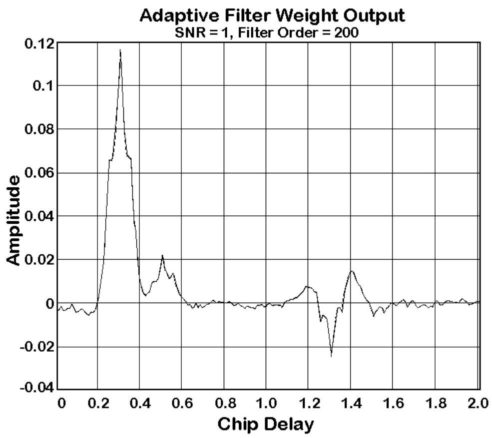

Figure 20: Adaptive filter weight output with an SNR of 1 and filter order =200, using interpolated data.

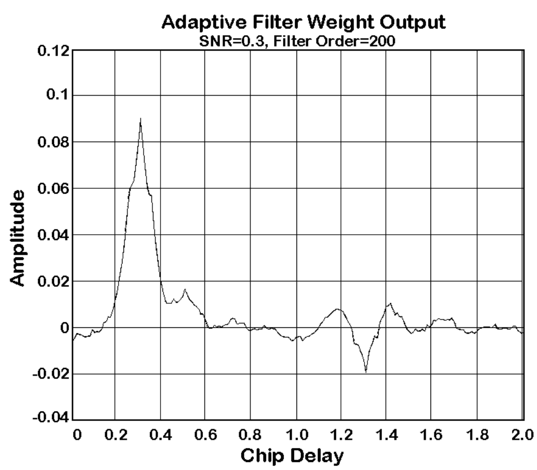

Figure 21: Adaptive filter weight output with an SNR of 0.3 and filter order =200, using interpolated data.

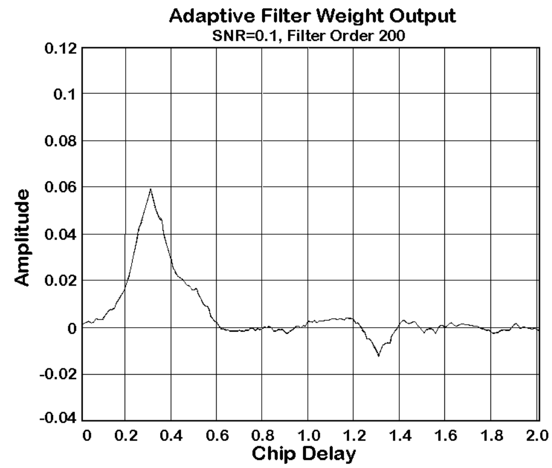

Figure 22: Adaptive filter weight output with an SNR of 0.1 and filter order =200, using interpolated data.

USING TWO DIFFERENT PRN CODES

The following plots show the results of using two different PRN codes in the adaptation. There should be no correlation between the two signals and therefore no peak in the correlation or adaptation of the two signals. Figure 23 shows that the correlation of the two codes gives small noise-like bumps, but no large peaks. The next figure shows the noisy output given by the adaptive filter. This is good because it shows that the filter will not lock onto the wrong satellite signal if the desired signal is not present.

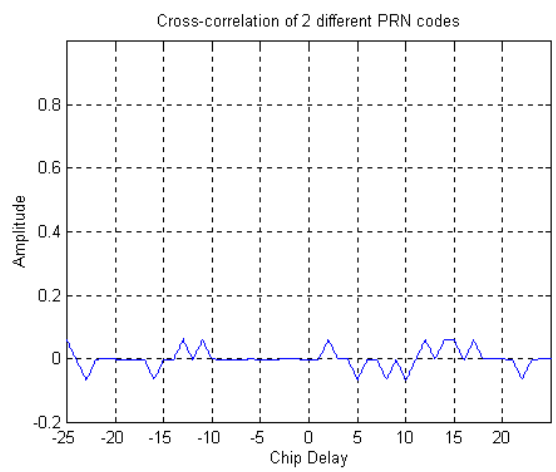

Figure 23: Correlation of two different PRN codes. The codes used were for SV#1 and SV#2 in this figure.

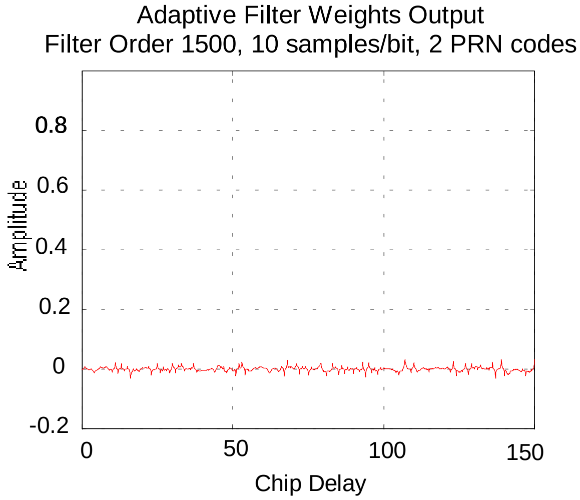

Figure 24: Adaptive filter weight output for two different PRN codes (SV#1 and SV#2).

COMPUTATION TIME AND CONVERGENCE RATE

The computation time for these simulations through the filter with 21 weights and 10 samples/bit takes about 1 minute on the workstation. With 21 weights and 100 samples/bit, it takes about 14 minutes. When the size of the weight vector is increased to 1500, the simulation takes approximately 37 minutes with 100 samples/bit and 3 minutes with 10 samples/bit. It appears that increasing the number of samples per bit increases the time by about the same factor - 10 times. Increasing the number of weights in the weight vector from 20 to 2000 increases the time through each iteration of the loop by over 3 times, going from about 3 ms per iteration to about 9 ms per iteration through the loop. However the algorithm processes the 10 sample/bit code just as quickly as it does the 100 sample/bit code with the same number of weights. In other words, we do not gain or lose anything by increasing the samples per bit using this algorithm. Furthermore, the value of μ does have an effect on the convergence time of the filter and is directly related to the SNR value used for each simulation.

We also tested how long the adaptation took as a function of SNR to converge to a threshold value on the peak amplitude value of the chip delay in the weight vector. As the SNR increased (as well as μ) it took fewer iterations through the filter to reach the threshold. As the threshold amplitude was increased performance became limited by the number of samples available for processing (30690 samples = 3 repeats of the 1023 bit code sampled 10 times per bit) as we went to lower SNR values. The smallest SNR we were able to achieve was 0.18 (-7.44 dB) with a threshold amplitude of 0.75 on a scale of 0 to 1. SNR values were calculated using the first 500 samples of the simulated multipath and noise code.

ANALYSIS OF RESULTS

The LMS algorithm was able to determine the code delay between the incoming signal and the received signal with SNR levels of 0.004 (-23.98 dB) or greater. When multipath is present the code delay can be found down to an SNR of 0.4 (-3.98 dB), using 1/10 chip resolution. Using 100 points per chip we can find the multipath peak with an SNR of 0.6 (-2.21 dB). These results suggest that the adaptive algorithm can accurately locate the code delay even at small signal-to-noise ratios. However, when multipath is present the algorithm is increasingly influenced by the SNR. This problem is attributable to the time, or number of signal samples, needed for the adaptation to converge. This is because the SNR determines the value of μ which controls the rate of adaptation for the algorithm.

Modern GPS receivers are able to determine the code delay to better than 0.01 chip [Van Dierendonck, 1995]. Using interpolation this software adaptation is comparable to the performance of current receivers that use DLLs, usually in hardware, to track the code delay.

FUTURE EXPLORATION

It is possible than an adaptive process could be implemented in lieu of the DLL in an all software receiver. It would provide a way to determine the delay of the code and also provide more information about the multipath at the receiver antenna.

Another area of investigation would be the difference in the amount of time used in the adaptive algorithm versus that required of the DLL. On the workstations each simulation took approximately 1 minute to complete using a smaller weight vector, but the code processing time could be reduced by eliminating unneeded variables. When interpolation was used the storage of y_k and ε_k led to much longer processing times than when we only stored the updated weight values for the adaptation.

To further evaluate the performance of this approach we need a more realistic signal model. One area to research would be with uneven sampling of the code stream. Currently the simulation assumes that the frequencies of the received signal and the receiver signal are syntonized. However in real applications the received signal frequency has been Doppler shifted and therefore “slides” the code, forcing the receiver to sample the code at uneven intervals. Implementing this could make the simulation more realistic and make it possible to achieve detection at even lower SNR levels.

ACKNOWLEDGMENTS

This work was performed while the first author was a NIST Graduate fellow at the University of Colorado at Boulder. She was sponsored by the NIST Time and Frequency Division and extends her gratitude for their support.

REFERENCES

[Braasch, 1995] Braasch, M.S. " Multipath Effects," in Parkinson, B.W. et al. (ed.), *Global Positioning System : Theory and Applications, Progress in Astronautics and Aeronautics*, Vol. 163, pp. 547-568, AIAA, 1996.

[Breivik, 1997] Breivik, K., B. Forssell, C. Kee, P. Enge, T. Walter, "Estimation of Multipath Error in GPS Pseudorange Measurements," *Navigation: Journal of the Institute of Navigation*, Vol. 44, No. 1, Spring 1997.

[Enge, 1994] Enge, P.K. "The Global Positioning System: Signal, Measurements and Performance," International Journal of Wireless Information Networks, Vol. 1, No. 2, pp. 83-105, Plenum Publishing Corporation, 1994.

[Etter, 1991] Etter, D.M. "Adaptive Signal Processing," tutorial paper in IASTED 91, 1991.

[Etter, 1993] Etter, D.M. *Engineering Problem Solving with MATLAB.* Prentice Hall, Inc., N.J., 1993.

[GPS NAVSTAR, 1991] *GPS NAVSTAR: Global Positioning System - User's Overview.* NAVSTAR Global Positioning System Joint Program Office and ARINC Research Corporation, 5th Edition, CA, March 1991.

[Gross, 1992] Gross, J.H. *Multirate Adaptive Filtering in Subbands, Masters Thesis*, University of Colorado, Electrical Engineering Department 1992.

[Haykin, 1996] Haykin, S. *Adaptive Filter Theory. Third Edition.* Prentice Hall. New Jersey, 1996.

[Hofmann-Wellenhof, 1994] Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J., *GPS: Theory and Practice. Third, revised edition*. Springer-Verlag, Vienna, 1994.

[ICD, 1993] *ICD-GPS-200*, NAVSTAR GPS Joint Program Office and ARINC Research Corporation, CA, 1993.

[Kee, 1994] Kee, C. and Parkinson, B.W., "Calibration of Multipath Errors on GPS Pseudorange Measurements," Proceedings of the ION-GPS-94, 1994.

[Klobuchar, 1996] Klobuchar, J.A., "Ionospheric Effects on GPS," in Parkinson, B.W. et al. (ed.), *Global Positioning System : Theory and Applications, Progress in Astronautics and Aeronautics*, Vol. 163, pp. 485-515, AIAA, 1996.

[Krauss, 1994] Krauss, T.P., Shure, L., and Little, J.H. *Signal Processing Toolbox: For Use with MATLAB.* The Mathworks, Inc., 1994. Simulations were done using an SGI Workstation and MATLAB software. *Certain commercial software and equipment are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purposes.

[Navigation, 1980] Janiczek, P.M., ed. *Global Positioning System - Papers published in Navigation*. Institute of Navigation, 1980.

[NAVSTAR, 1991] *NAVSTAR GPS User Equipment - Introduction*. NATO, NAVSTAR-GPS Joint Program Office, Los Angeles Air Force Base, CA, February 1991.

[Spilker(a), 1996] Spilker, Jr., J.J., "GPS Signal Structure and Theoretical Performance," in Parkinson, B.W. et al. (ed.), *Global Positioning System : Theory and Applications, Progress in Astronautics and Aeronautics*, Vol. 163, pp. 57-119, AIAA, 1996.

[Spilker(b), 1996] Spilker, Jr., J.J., "Tropospheric Effects on GPS," in Parkinson, B.W. et al. (ed.), *Global Positioning System : Theory and Applications, Progress in Astronautics and Aeronautics*, Vol. 163, pp. 517-546, AIAA, 1996.

[Spilker(c), 1996] Spilker, Jr., J.J. and Parkinson, B.W., "Overview of GPS Operation and Design," in Parkinson, B.W. et al. (ed.), *Global Positioning System : Theory and Applications, Progress in Astronautics and Aeronautics*, Vol. 163, pp. 29-55, AIAA, 1996.

[Van Dierendonck, 1992] Van Dierendonck, A.J., Fenton, P. and Ford, T., "Theory and Performance of Narrow Correlator Spacing in a GPS Receiver," *Navigation*. Journal of the Institute of Navigation, Vol. 39, No. 3, pp. 265-283, Fall 1992.

[Van Dierendonck, 1995] Van Dierendonck, A.J., “GPS Receivers,” in Parkinson, B.W. et al. (ed.), *Global Positioning System : Theory and Applications, Progress in Astronautics and Aeronautics*, Vol. 163, pp. 329-407, AIAA, 1996.

[Van Nee, 1992] Van Nee, R.D.J., “GPS Multipath and Satellite Interference,” Proceedings of ION 48th Annual Meeting, August 1992.

[Van Nee, 1994] Van Nee, R.D.J., Siereveld, J., Fenton, P.C., and Townsend, B.R., “The Multipath Estimating Delay Lock Loop: Approaching Theoretical Accuracy Limits,” IEEE 1994 Position Location and Navigation Symposium, Las Vegas, Nevada, April 1994, pp. 246-251.

[Widrow, 1985] Widrow, B. and Stearns, S.D. *Adaptive Signal Processing*. Prentice Hall, Inc., N.J., 1985.