GPS Spoofing Detection via Dual-Receiver Correlation of Military Signals

by Mark L. Psiaki
Sibley School of Mechanical & Aerospace Engineering, Cornell University, Ithaca, N.Y. 14853-7501

Brady W. O'Hanlon
School of Electrical and Computer Engineering, Cornell University, Ithaca, N.Y. 14853-7501

Jahshan A. Bhatti, Daniel P. Shepard, and Todd E. Humphreys
Department of Aerospace Engineering & Engineering Mechanics, The University of Texas at Austin, Austin, Texas 78712-0235

Abstract -- Cross-correlation of unknown encrypted signals between two Global Navigation Satellite System (GNSS) receivers is used for spoofing detection of publicly-known signals. This detection technique is one of the strongest known defenses against sophisticated spoofing attacks if the defended receiver has only one antenna. The attack strategy of concern overlays false GNSS radio-navigation signals on top of the true signals. The false signals increase in power, lift the receiver tracking loops off of the true signals, and drag the loops and the navigation solution to erroneous, but consistent results. This paper uses hypothesis testing theory to develop a codeless cross-correlation detection method for use in inexpensive, narrow-band civilian GNSS receivers. The detection method is instantiated by using the encrypted military GPS P(Y) code on the L1 frequency in order to defend the publicly-known civilian GPS C/A code. Successful detection of spoofing attacks is demonstrated by off-line processing of recorded RF data from narrow-band 2.5 MHz RF front-ends, which attenuate the wide-band P(Y) code by 5.5 dB. The new technique can detect attacks using correlation intervals of 1.2 sec or less.

Index terms -- GPS, Global Navigation Satellite System, spoofing detection, hypothesis testing.

I. INTRODUCTION

The vulnerability of unencrypted civilian GNSS signals to spoofing has long been known. The U.S. Department of Transportation has noted the vulnerability of GPS to spoofing [1]. Spoofing is the intentional broadcast of false signals that, in a user receiver, appear to be true signals. Spoofing of GNSS signals can cause a user receiver to determine a location that is far different from its true position, to compute erroneous corrections to its receiver clock, or to make both errors simultaneously [2,3,4,5,6,7].

The spoofing attack described in Refs. 5 and 6 is hard to detect. It synthesizes spoofing signals for multiple satellites in a way that initially overlays them on top of the true signals. Next, it slowly pulls the victim receiver away from truth time and location in a self-consistent way. Typical Receiver Autonomous Integrity Monitoring (RAIM) methods for spoofing detection [8] will fail to detect such an attack because they look for signal inconsistencies at the navigation level, which are not present in this scenario.

New RAIM methods are being developed to try to detect this type of attack at the tracking-loop/discriminator/correlator level [9,10,11]. These detection algorithms are complex and may be difficult to implement robustly. If such algorithms are to succeed, typically they must achieve detection at the moment of signal drag-off, which degrades their robustness.

Several other approaches have been proposed to detect this type of spoofing attack. These methods include cross-correlation of encrypted signals between secure and defended receivers [12,13,14], the use of multiple antennas [15], and methods that rely on inertial measuring devices and high-stability clocks. Other proposed methods would require changes to the navigation data message to provide Navigation Message Authentication (NMA) [3,16], or some sort of partial encryption of spreading codes [3,7]. NMA techniques may need to be implemented in conjunction with algorithms that detect dynamic estimation-and-replay spoofing of the NMA authentication bits [17].

The cross-correlation method of Refs. 12, 13, and 14 has one disadvantage compared to other spoofing detection methods: it requires a communication link between its secure and defended receivers so that parts of the two receivers' signals can be cross-correlated. For most applications, however, this disadvantage is outweighed by the method's several advantages: (1) it does not require an extra GPS antenna or an IMU; (2) it does not require alteration of the broadcast GPS signal, as do the techniques proposed in Refs. 3, 7, and 16; (3) it offers low-latency signal authentication -- one second or less as compared to 5 minutes per signal for the NMA-based technique proposed in Ref. 16; and (4) it is more robust than receiver-autonomous techniques that operate on the tracking-loop/discriminator/correlator level such as those considered in Refs. 9, 10, and 11 -- because it works even after initial signal drag-off and is not susceptible to multipath-induced false alarms. Because of these advantages, the remainder of this paper focuses on the cross-correlation spoofing detection method.

The cross-correlation method relies on encrypted signals that are broadcast on the same frequency as the publicly-known signal that is being tracked for navigation purposes. For example, a GPS civilian receiver might track and use the unencrypted civilian pseudo-random number (PRN) codes such as the C/A code on the L1 frequency or the new L2C code on the L2 frequency. These frequencies also carry the encrypted military P(Y) PRN codes and, on newer satellites, the encrypted military binary offset carrier (BOC) M-codes. The civilian PRN codes can be spoofed using the technique of Refs. 5 and 6 or related techniques because the spoofer has prior knowledge of the codes. The spoofing detection methods proposed in Refs. 12, 13, and 14 use the known carrier-phase and code-phase relationships between the tracked civilian codes and the encrypted military codes. These methods correlate the parts of the signal known to contain the encrypted military codes between two receivers. One receiver is presumed to reside in a secure location so that it has the correct encrypted code in the expected location. The spoofing detection algorithm correlates this part of the signal from the secure receiver with the same part of the signal from the other receiver, the potential spoofing victim. If the correlation is large enough, by an appropriate statistical measure, then the null-hypothesis of no spoofing is accepted. Otherwise, a spoofing alert is issued for the signal.

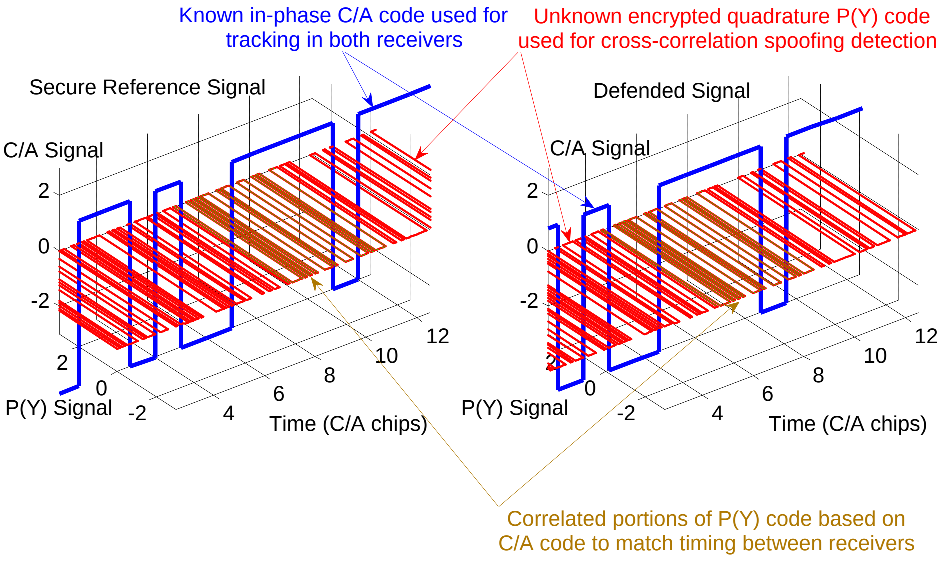

Fig. 1. Relationship of publicly-known C/A signal and encrypted P(Y) signal on two receivers.

This strategy and the relationship of the publicly-known and encrypted signals is illustrated in Fig. 1 for the C/A and P(Y) signals on the GPS L1 frequency. The signals in the secure reference receiver are depicted in the left-hand plot, with the vertical blue curve depicting the C/A PRN code signal and the horizontal red/green curve depicting the P(Y) PRN code. Time increases along the second horizontal axis. The right-hand plot shows the same sections of these two signals in the second receiver, the potential victim for which spoofing detection must be performed. The use of orthogonal axes represents the fact that the C/A and P(Y) codes are modulated onto the carrier signal in phase quadrature. The strategy of Refs. 12, 13, and 14 is to track the blue C/A signals in each receiver and to use the knowledge of these signals' phase and timing relationships to the P(Y) code in order to strip off the green part of the received P(Y) code in each receiver. Although this green signal is not known by either receiver a priori and although its received version is noisy, a correlation between these two green segments will produce a sufficiently large accumulated value only if the correct P(Y) code is present in both receivers. This will be true only if the defended receiver is not being spoofed.

The initial version of Ref. 14 represents the first known development of this technique. That reference is closely related to Ref. 12, which tested the un-spoofed case for this method. These tests demonstrated a significant inter-receiver correlation of the baseband-mixed signal that was in quadrature with the GPS L1 C/A code. Thus, Ref. 12 verified a lack of spoofing based on the encrypted L1 P(Y) signal. It did not perform a statistical analysis of the detection threshold for a spoofing alert, nor did it test the method under an actual spoofing attack. Its correlation calculations, which were based on batch laboratory data collection and analysis techniques, amounted to a proof-of-concept implementation. It required a somewhat expensive relative timing search between the quadrature signals of the two receivers. Therefore, it seems likely that further refinements could improve this method's efficiency and precision.

Reference 13 constitutes the initial publication of an on-going parallel effort to develop the needed refinements. It presents a statistical analysis of spoofing detection thresholds, and it reports on an attempt to develop a system that can function in real-time. Its approach to real-time detection is to stream raw RF samples directly from the secure receiver to the potential victim receiver via the Internet. The defended receiver, the potential victim of spoofing, is a software radio receiver. It has the real-time capacity to track signals both from its own antenna and in the streamed RF data that originated from the secure antenna. It also has the capacity to do the necessary correlation calculations of the quadrature baseband signals from the two data streams.

A significant contribution of Ref. 13 is an analysis which shows that the P(Y) code can be used for practical spoofing detection even in a narrow-band C/A-code receiver, i.e., one with an RF front-end bandwidth of only 1.9 MHz. Reference 12 implies the need for a wide-band RF front-end for this type of approach. A 1.9 MHz narrow-band receiver attenuates the P(Y) code by 6.9 dB and greatly distorts it, but there is still enough vestigial signal to achieve reasonable detection power for reasonable cross-correlation intervals. Unfortunately, Ref. 13 failed to achieve successful spoofing detection results due to software bugs in its real-time inter-receiver correlation calculations.

This paper makes two principal contributions. First, it provides a more complete explanation of the codeless spoofing detection test of Ref. 13. Second, it implements that method and provides the first demonstrations of its effectiveness in detecting a sophisticated spoofing attack as defined in Refs. 5 and 6. It does this using recorded RF front-end data from two receivers in off-line MATLAB calculations. The RF front-ends have bandwidths of only 2.4 and 2.6 MHz. Therefore, this demonstration confirms the hypothesis of Ref. 13 that narrow-band receivers have sufficient vestigial P(Y) code for purposes of spoofing detection.

This paper does not attempt to devise any strategy in the event that a spoofing attack has been detected. Rather, its only goal is to inform the defended receiver whether or not its tracked publicly-known signals are reliable.

The remainder of this paper consists of 4 sections plus conclusions. Section II presents a mathematical model of the L1 C/A and P(Y) signals and of quadrature baseband mixing. These two signals are, respectively, the example publicly-known and encrypted signals that are considered throughout this paper. Section III reviews, explains, and analyzes the codeless spoofing detection method. Section IV presents spoofing detection test results. Section V discusses the possibility that modified spoofing attack strategies might provide tougher challenges to this method, and it discusses possible responses to such challenges. Section VI presents this paper's conclusions.

II. MATHEMATICAL MODELS OF SIGNALS AND PRE-PROCESSING

A. Received Signal Models

The spoofing detection analysis starts with models of the received signals at the outputs of the RF front-ends of 2 receivers. These signals take the form:

$$\begin{aligned} y_{ai} &= A_{ca}C_f(t_{ai})D(t_{ai})cos[\omega_{IF}t_{ai} + \phi_a(t_{ai})] \\ &\quad + A_{pa}P_{Yf}(t_{ai})D(t_{ai})sin[\omega_{IF}t_{ai} + \phi_a(t_{ai})] + n_{ai} \quad (1a) \\ y_{bi} &= A_{cb}C_f(t_{bi})D(t_{bi})cos[\omega_{IF}t_{bi} + \phi_b(t_{bi})] \\ &\quad + A_{pb}P_{Yf}(t_{bi})D(t_{bi})sin[\omega_{IF}t_{bi} + \phi_b(t_{bi})] + n_{bi} \quad (1b) \end{aligned}$$

where y_{ai} is the sample output by Receiver A's RF front-end at Receiver Clock A sample time t_{ai} and where y_{bi} is the sample output by Receiver B's RF front-end at Receiver Clock B sample time t_{bi}. Receiver A is assumed to be the secure reference receiver. Receiver B is the potential victim of a spoofing attack, the receiver for which spoofing detection must be performed.

The function $C_f(t)$ is the C/A code as distorted and attenuated by the filter in the RF front-end. The function $P_{Yf}(t)$ is the distorted and attenuated received P(Y) code. The function $D(t)$ represents the 50 Hz navigation data bits. In the present analysis, these functions are presumed to be the same in both receivers. Nominally, these functions would be either +1 or -1 at all times due to the BPSK nature of the PRN codes and the navigation data, and their powers would equal 1. The RF front-end filters distort $C_f(t)$ and $P_{Yf}(t)$ so that they can take on different values than +/-1, and the filtering lowers their powers to values less than 1. Referring to Fig. 1, $C_f(t)$ is represented by the blue curves, and $P_{Yf}(t)$ is represented by the red/green curves, except that the figure does not depict distortion or attenuation. These functions' phase quadrature relationship in Eqs. (1a) and (1b) is illustrated in the figure by their being plotted along orthogonal axes.

The received C/A code amplitudes for the two receivers are, respectively, A_{ca} and A_{cb}. The corresponding received P(Y) amplitudes are A_{pa} and A_{pb}. Subsequent analyses in this paper assume that the P(Y) amplitudes can be deduced from the C/A amplitudes. This calculation takes the form:

$$A_p = A_c 10^{0.4/20}\sqrt{L_p} \quad (2)$$

where L_p is the power loss factor of the broadcast P(Y) code relative to the broadcast C/A code for the satellite in question. Typically $10log_{10}(L_p)$ equals approximately -3 dB [18]. The 0.4 dB term in the exponent of Eq. (2) compensates for the fact that L_p is defined in the +/-10.23 MHz bandwidth centered at L1, which contains only the main lobe of the P(Y) power spectral density but 18 additional side-lobes of the C/A spectral density. The "*a*" and "*b*" subscripts have been omitted from Eq. (2) because it applies to both pairs of amplitudes for both receivers using the identical loss factor L_p.

The frequency ω_{IF} is the nominal intermediate frequency. It is the frequency to which the nominal carrier at ω_{L1} = 2πx1575.42x10^6 rad/sec gets mixed by the RF front-end.

The functions $\phi_a(t)$ and $\phi_b(t)$ are the beat carrier phase time histories of the signals at Receivers A and B, respectively. They have the opposite sign to the usual definition of beat carrier phase in the GPS literature. Their time derivatives equal the received carrier Doppler shifts.

The quantities n_{ai} and n_{bi} are the receiver noise terms. They are assumed to be discrete-time Gaussian white-noise with statistics:

$$E\{n_{ai}\} = 0,\ E\{n_{ai}^2\} = \sigma_{RFa}^2,\ E\{n_{ai}n_{aj}\} = 0 \text{ for all } i \neq j \tag{3a}$$

$$E\{n_{bi}\} = 0,\ E\{n_{bi}^2\} = \sigma_{RFb}^2,\ E\{n_{bi}n_{bj}\} = 0 \text{ for all } i \neq j \tag{3b}$$

$$E\{n_{ai}n_{bj}\} = 0 \text{ for all } i, j \tag{3c}$$

B. C/A-Code and Carrier Tracking and Quadrature Baseband Mixing

The spoofing detection algorithms of this paper presume that the reference and defended receivers are able to acquire and track the C/A code signals in Eqs. (1a) and (1b). A Delay-Lock Loop (DLL) is presumed to track the C/A PRN code in order to determine the start/stop times of code periods in $C_f(t)$. Suppose that these times are τ_{ak} and τ_{bk} at the end of the (k-1)st C/A code period and the start of the kth C/A code period, as measured at Receivers A and B using their respective clocks. The tracking algorithms use a Phase-Lock Loop (PLL) in order to determine the estimated beat carrier phase time histories $\hat{\phi}_a(t)$ and $\hat{\phi}_b(t)$.

The PLL uses feedback from a carrier-phase discriminator. The discriminator is computed from the following prompt in-phase and quadrature accumulations for the kth code period:

$$I_k = \sum_{i=i_k}^{i_k+N_k-1} y_i C[(t_i - \tau_k)(1 + \hat{\omega}_{Dk}/\omega_{L1})] \times \cos[\omega_{IF}t_i + \hat{\phi}_k + \hat{\omega}_{Dk}(t_i - \tau_k)] \tag{4a}$$

$$Q_k = \sum_{i=i_k}^{i_k+N_k-1} y_i C[(t_i - \tau_k)(1 + \hat{\omega}_{Dk}/\omega_{L1})] \times \sin[\omega_{IF}t_i + \hat{\phi}_k + \hat{\omega}_{Dk}(t_i - \tau_k)] \tag{4b}$$

where the "a" and "b" subscripts have been omitted because the accumulation processing is similar in both receivers. The sample index i_k is the first sample of the kth code period, i.e., the first sample such that $\tau_k \leq t_i$. The number N_k is the total number of samples in the code period so that the terminal index i_k+N_k-1 is the last sample of the code period, that is, the last sample such that $t_i < \tau_{k+1}$. The function $C[t]$ is the +1/-1-valued C/A PRN code without RF filter effects. The frequency $\hat{\omega}_{Dk}$ is the PLL's carrier Doppler shift estimate for the kth code period, and the phase $\hat{\phi}_k$ is the estimated beat carrier phase at the code period start time τ_k.

Quadrature baseband mixing is used in order to isolate the P(Y)-code part of the signal. The quadrature baseband mixed signals for the kth C/A code period are computed as follows:

$$y_{qi} = y_i\{I_k \sin[\omega_{IF}t_i + \hat{\phi}_k + \hat{\omega}_{Dk}(t_i - \tau_k)] - Q_k \cos[\omega_{IF}t_i + \hat{\phi}_k + \hat{\omega}_{Dk}(t_i - \tau_k)]\} / \sqrt{I_k^2 + Q_k^2} \quad \text{for } i = i_k, \ldots, (i_k+N_k-1) \tag{5}$$

where y_{qi} is the quadrature baseband mixed signal that corresponds to the original sample y_i. This mixing formula uses both the estimated carrier-phase time history from the PLL and the in-phase and quadrature accumulations. If the PLL has settled, then the quadrature accumulation Q_k will nominally be zero, and this formula will approximate simple multiplication by the quadrature $sin[\omega_{IF}t_i$+...] signal. Equation (5) is used in place of this simple multiplication because it compensates for the effects of navigation data bit signs and for PLL tracking errors. The latter compensation assumes that the noise effects on I_k and Q_k are negligible.

Again, the "a" and "b" subscripts have been omitted from Eq. (5). In later analyses, the quadrature baseband-mixed samples of the two receivers must be distinguished from each other. They will be designated as y_{qai} and y_{qbi}. They are computed as in Eq. (5), except that the quantities y_i, I_k, Q_k, t_i, $\hat{\phi}_k$, $\hat{\omega}_{Dk}$, τ_k, i_k, and N_k are modified to include an "a" or "b" subscript, depending on whether y_{qai} or y_{qbi} is being calculated.

Equation (5) provides a recipe for computing the quadrature baseband-mixed signal in each receiver. It is helpful also to have a model of this signal for each receiver. A model can be derived by substitution of the signal model in Eq. (1a) or (1b) into Eq. (5) and by assuming that the true beat carrier phase time history is accurately represented by $\hat{\phi}_k + \hat{\omega}_{Dk}(t_i - \tau_k)$ -$atan2(Q_k, I_k)$. The function $atan2(,)$ is the usual 2-argument arctangent function. The resulting models for the two receivers take the form:

$$y_{qai} = \tfrac{1}{2}A_{pa}P_{Yf}(t_{ai}) + n_{qai} \tag{6a}$$

$$y_{qbi} = \tfrac{1}{2}A_{pb}P_{Yf}(t_{bi}) + n_{qbi} \tag{6b}$$

where the quadrature baseband noise terms n_{qai} and n_{qbi} have the statistics

$$E\{n_{qai}\} = 0,\ E\{n_{qai}^2\} = \tfrac{1}{2}\sigma_{RFa}^2,\ E\{n_{qai}n_{qaj}\} = 0 \text{ for all } i \neq j \tag{7a}$$

$$E\{n_{qbi}\} = 0,\ E\{n_{qbi}^2\} = \tfrac{1}{2}\sigma_{RFb}^2,\ E\{n_{qbi}n_{qbj}\} = 0 \text{ for all } i \neq j \tag{7b}$$

$$E\{n_{qai}n_{qbj}\} = 0 \text{ for all } i, j \tag{7c}$$

The models in Eqs. (6a) and (6b) ignore the parts of the signals in Eqs. (1a) and (1b) that get mixed to the vicinity the frequency $2\omega_{IF}$ by the operations in Eq. (5). This is reasonable because the neglected high-frequency signals will not affect the subsequent baseband processing.

C. Modeling W Encryption Chips and RF Filter Distortion of the P(Y) Code

The P(Y) code can be modeled as the product of the known P code [18] multiplied by unknown W encryption chips. This model takes the form

$$P_Y(t) = P(t)W(t) \tag{8}$$

where $P_Y(t)$ is the +/-1-valued encrypted P(Y) code, $P(t)$ is the +/-1-valued known P code, and $W(t)$ is the +/-1-valued unknown time history of encryption chips. The $W(t)$ encryption chips have an average chipping rate of 480 KHz.

The filtered version of the P(Y) code that appears in Eqs. (1a), (1b), (6a), and (6b) can be modeled as follows:

$$P_{Yf}(t) = \sum_{j=-\infty}^{\infty} w_j P_{fwj}(t) \tag{9}$$

where w_j is the j^{th} +/-1-valued W chip and where $P_{fwj}(t)$ is the attenuated and distorted version of the 20 or so P chips that correspond to the j^{th} W chip.

The w_j chip values cannot be known a priori in a civilian receiver, but the functions $P_{fwj}(t)$ can be determined based on the known P code, known W-chip timing, and the modeled effects of the RF front-end filter. Suppose that the unfiltered version of $P_{fwj}(t)$ takes the form:

$$P_{wj}(t) = \sum_{i=i_{wj}}^{i_{wj}+I_{wj}-1} p_i \Pi_{T_p}[t-(i-i_{wj})T_p - \tau_{wj}] \tag{10}$$

where p_i is the known +1/-1 value of the i^{th} P-code chip of the given GPS week, T_p is the P-code chip period, i_{wj} is the index of the initial P-code chip of the j^{th} W chip as measured from the start of the GPS week, I_{wj} is the total number of P-code chips in the j^{th} W chip, and τ_{wj} is the start time of the j^{th} W chip and of the $(i_{wj})^{th}$ P chip. The function $\Pi_T(t)$ is the usual rectangular support function, which is equal to one over the interval $0 \le t < T$ and zero elsewhere. The P-code chip period is nominally $T_p = 1/(10.23\text{x}10^6)$ sec, but it will vary if there is a non-zero code Doppler shift.

The filtered version of these same P-code chips takes the form

$$P_{fwj}(t) = \sum_{i=i_{wj}}^{i_{wj}+I_{wj}-1} p_i \Psi[t-(i-i_{wj})T_p - \tau_{wj}] \tag{11}$$

where $\Psi(t)$ is the filtered version of the rectangular support function $\Pi_T(t)$:

$$\Psi(t) = \begin{cases} 0 & t \le 0 \\ \int_{t-T_{hmax}}^{t} h_{RF}(t-\tau)\Pi_{T_p}(\tau)d\tau & 0 < t \le (T_p + T_{hmax}) \\ 0 & (T_p + T_{hmax}) < t \end{cases} \tag{12}$$

In this formula, $h_{RF}(t)$ is the real part of the envelope impulse response function of the receiver's RF filter. This function can be determined using off-line system identification techniques [19]. Equation (12) assumes that $h_{RF}(t)$ is a finite impulse response with zero response T_{hmax} sec after the impulse. This is a reasonable approximation for a large enough T_{hmax}, and it is consistent with the system identification assumptions of Ref. 19.

D. P(Y) Code and C/A Code Power Loss in the RF Front-End Filter

The filter impulse response function can be used to determine the P(Y) signal's power loss in the narrow-band RF front-end. This calculation starts by computing the envelope filter's frequency response

$$H_{RF}(j\omega) = \int_0^{T_{hmax}} h_{RF}(t)e^{-j\omega t}dt \tag{13}$$

where $j = (-1)^{1/2}$ in this formula. The square of the absolute value of this function multiplies the unfiltered P(Y) code's normalized power spectral density

$$S_{py}(\omega) = \left[\frac{sin(\omega T_p/2)}{(\omega T_p/2)}\right]^2 \tag{14}$$

in order to yield the corresponding filtered power spectral density. The ratio of the integrals of the filtered and unfiltered power spectral densities gives the power loss through the filter:

$$L_{fpy} = \frac{\int_{-2\pi/T_p}^{2\pi/T_p} |H_{RF}(j\omega)|^2 S_{py}(\omega)d\omega}{\int_{-2\pi/T_p}^{2\pi/T_p} S_{py}(\omega)d\omega} \tag{15}$$

Recall that T_p in these formulas is the P-code chipping period. Thus, these integrals are performed over the main lobe of the P(Y) signal, i.e., over the range -10.23 MHz to +10.23 MHz.

Another power loss factor is that of the C/A code. It is important because the spoofing detection calculations need to know P(Y) code power or amplitude, and they infer it from C/A code amplitude using calculations like those in Eq. (2). The C/A code loss factor must account for two effects. One is the loss in the RF front-end filter, and the other is the loss associated with the accumulation calculations in Eqs. (4a) and (4b). The latter loss arises from the use of the unfiltered C/A code $C[t]$ in the accumulation recipes. The total power loss of the C/A code at the output of the $[I_k,Q_k]$ accumulation process is:

$$L_{fca} = \left[\max_{\tau} \int_{0}^{t_{max}} h_{RF}(t) s_{ca}(t-\tau) dt \right]^2 \tag{16}$$

where $s_{ca}(t)$ is the symmetric autocorrelation function of the unfiltered C/A code. The result of the integration in Eq. (16) is the cross-correlation between the filtered and unfiltered versions of the C/A code. Its maximum value is less than 1, but it approaches 1 as the filter bandwidth increases towards infinity [19].

III. Codeless Spoofing Detection Technique

This section develops an implementation of the codeless spoofing detection algorithm of Refs. 12, 13, and 14. A significant amount of this material is taken from Ref. 13, but the notation has been changed in a number of places in order to conform with the models in Section II of the present paper. In addition to notation changes, the present developments include significant new implementation details. This section also presents an analysis of spoofing detection power as a function of accumulation interval and received carrier-to-noise ratio.

A. Computation of the Raw Codeless Spoofing Detection Statistic

The raw codeless spoofing detection statistic is the sum of products of quadrature samples from Receivers A and B. In other words, it is the sum of products of Eq. (6a) samples and Eq. (6b) samples. It constitutes an optimal test statistic in a reasonable limiting case, as will be discussed at the end of Subsection III.B.

Before forming products, it is necessary to map sample times in the two receivers to identical values as measured relative to their respective tracked C/A codes. This inter-receiver time mapping relies on the DLL estimates of the C/A code start/stop times, τ_{a1}, τ_{a2}, ..., τ_{ak}, τ_{ak+1}, ... and τ_{b1}, τ_{b2}, ..., τ_{bk}, τ_{bk+1}, ...

Suppose, in addition, that there is a known differential relative timing offset between the filtered P(Y) code and the DLL estimate of the filtered C/A code. This offset is denoted by δt_{ab}, and it represents a difference between the two receivers. It is a measure of the amount by which the filtered P(Y) code in Receiver B is delayed relative to that receiver's DLL-generated C/A code replica when compared to the filtered P(Y) code in Receiver A. Nominally, one would expect this differential timing offset to be zero or nearly so. A non-zero value is allowed in the present analysis in order to make it more general and to facilitate an experimental study of the magnitude of this delay.

Suppose that the correlation calculation seeks the correct quadrature sample from Receiver B to correlate with sample y_{qai} from Receiver A, which was sampled at Receiver A clock time t_{ai}. Suppose that the delayed sample time $(t_{ai}+\delta t_{ab})$ lies in the Receiver A DLL's estimate of the reception interval of the k^{th} C/A PRN code period. That is, suppose that $\tau_{ak} \leq (t_{ai}+\delta t_{ab}) < \tau_{ak+1}$. Then the first step in the correlation process is to compute the corresponding time according to Receiver B's clock. Using linear interpolation between DLL code start/stop times, it is:

$$\tilde{t}_{bi} = \tau_{bk} + \left(\frac{\tau_{bk+1} - \tau_{bk}}{\tau_{ak+1} - \tau_{ak}} \right) (t_{ai} + \delta t_{ab} - \tau_{ak}) \tag{17}$$

This Receiver B time estimate can be used to interpolate between Receiver B quadrature samples from Eq. (6b) in order to synthesize the "sample" of the Receiver-B quadrature signal that corresponds to the Receiver-A sample y_{qai}. Suppose that the interpolated time $\tilde{t}_{bi}$ from Eq. (17) lies between Receiver-B RF sample times t_{bj} and t_{bj+1}. Then the synthesized quadrature sample of Receiver B is the linearly interpolated value:

$$\tilde{y}_{qbi} = y_{qbj} + \left(\frac{y_{qbj+1} - y_{qbj}}{t_{bj+1} - t_{bj}} \right) (\tilde{t}_{bi} - t_{bj}) \tag{18}$$

The Receiver-A quadrature samples from Eq. (6a) and the synthesized Receiver-B quadrature samples from Eq. (18) are multiplied together and summed in order to form the un-normalized codeless spoofing detection statistic:

$$\gamma_{ul} = \sum_{i=i_l}^{i_l+M-1} y_{qai} \tilde{y}_{qbi} \tag{19}$$

The index i_l in this formula is the initial sample of the correlation accumulation interval, and M is the total number of samples used in each accumulation. This l^{th} un-normalized spoofing detection statistic spans a data interval of length $T_{corr} = M\Delta t$ sec, where $\Delta t = t_{ai+1} - t_{ai}$ is the RF front-end sample period. The mid-point of this interval is

$$t_{cl} = t_{ai_l} + \frac{(M-1)\Delta t}{2} \tag{20}$$

according to the Receiver-A clock.

B. Hypothesis Test for Spoofing based on a Normalized Codeless Detection Statistic

The spoofing detection statistic in Eq. (19) has significantly different properties depending on whether or not the C/A code signal tracked by Receiver B is a spoofed signal. If the signal is not spoofed, then the synthesized $\tilde{y}_{qbi}$ quadrature sample is assumed to be modeled by Eq. (6b). If the signal is spoofed, however, then the P(Y) code is presumed to be absent from the quadrature channel of Receiver B. In this case, Eq. (6b) is modified by setting the P(Y)-code amplitude to $A_{pb} = 0$. In truth, the P(Y) signal is still present, but with a large code phase offset relative to the spoofed C/A code. Given the narrow P(Y) correlation peak and low correlation side lobes, the net effect is well approximated by setting $A_{pb} = 0$.

Under the hypothesis of spoofing, hypothesis H_1, the mean and variance of the spoofing detection statistic γ_{ul} are

$$\begin{aligned}\bar{\gamma}_{u|H1} &= E\{\gamma_{ul} \mid H_1\}\\ &= \sum_{i=i_l}^{i_l+M-1} [\tfrac{1}{2} A_{pa} P_{Yf}(t_{ai}) + E\{n_{qai}\}] E\{\tilde{n}_{qbi}\}\\ &= 0 \qquad\qquad (21a)\end{aligned}$$

$$\begin{aligned}\sigma^2_{\gamma u|H1} &= E\{\gamma^2_{ul} \mid H_1\}\\ &= E\left\{\left[\sum_{i=i_l}^{i_l+M-1} [\tfrac{1}{2} A_{pa} P_{Yf}(t_{ai}) + n_{qai}]\tilde{n}_{qbi}\right]^2\right\}\\ &= \sum_{i=i_l}^{i_l+M-1} [\tfrac{1}{4} A^2_{pa} P^2_{Yf}(t_{ai}) + \tfrac{1}{2}\sigma^2_{RFa}]\tfrac{1}{2}\sigma^2_{RFb}\\ &= \frac{M}{4}\sigma^2_{RFb}[\sigma^2_{RFa} + \tfrac{1}{2} A^2_{pa}\bar{P}^2_{Yf}]\\ &= \frac{M}{4}\sigma^2_{RFa}\sigma^2_{RFb}[1 + 2\Delta t(C/N_0)_{pya}] \qquad (21b)\end{aligned}$$

where $\tilde{n}_{qbi}$ is the noise in the synthesized quadrature sample $\tilde{y}_{qbi}$, which is assumed to obey the same statistics as n_{qbi} in Eqs. (7b) and (7c). The quantity $\bar{P}^2_{Yf}$ is the mean value of P^2_{Yf}, i.e., it is the power of the distorted P(Y) code at the output of the RF front-end filter. The quantity $(C/N_0)_{pya}$ = $A^2_{pa}\bar{P}^2_{Yf}/(4\sigma^2_{RFa}\Delta t)$ is the filtered P(Y)-code carrier-to-noise ratio in Receiver A. The derivations in Eqs. (21a) and (21b) depend on the assumptions that $E\{\tilde{n}_{qbi}\tilde{n}_{qbj}\} = 0$ for all (i,j) such that $i \neq j$ and that $E\{n_{qai}\tilde{n}_{qbj}\} = 0$ for all (i,j).

Under the hypothesis of no spoofing, hypothesis H_0, the mean and variance of γ_{ul} are

$$\begin{aligned}\bar{\gamma}_{u|H0} &= E\{\gamma_{ul} \mid H_0\}\\ &= \sum_{i=i_l}^{i_l+M-1} [\tfrac{1}{2} A_{pa} P_{Yf}(t_{ai}) + E\{n_{qai}\}]\times\\ &\qquad [\tfrac{1}{2} A_{pb} P_{Yf}(\tilde{t}_{bi}) + E\{\tilde{n}_{qbi}\}]\\ &= \frac{M}{4} A_{pa} A_{pb} \bar{P}^2_{Yf}\\ &= M\sigma_{RFa}\sigma_{RFb}\Delta t\sqrt{(C/N_0)_{pya}(C/N_0)_{pyb}} \qquad (22a)\end{aligned}$$

$$\begin{aligned}\sigma^2_{\gamma u|H0} &= E\{\gamma^2_{ul} \mid H_0\} - \bar{\gamma}^2_{u|H0}\\ &= E\Big\{\Big[\sum_{i=i_l}^{i_l+M-1} [\tfrac{1}{2} A_{pa} P_{Yf}(t_{ai}) + n_{qai}]\times\\ &\qquad [\tfrac{1}{2} A_{pb} P_{Yf}(\tilde{t}_{bi}) + \tilde{n}_{qbi}]\Big]^2\Big\} - \bar{\gamma}^2_{u|H0}\\ &= \left[\tfrac{1}{4} A_{pa} A_{pb} \sum_{i=i_l}^{i_l+M-1} P_{Yf}(t_{ai}) P_{Yf}(\tilde{t}_{bi})\right]^2\\ &\quad + \tfrac{1}{8} A^2_{pa}\sigma^2_{RFb} \sum_{i=i_l}^{i_l+M-1} P^2_{Yf}(t_{ai})\\ &\quad + \tfrac{1}{8} A^2_{pb}\sigma^2_{RFa} \sum_{i=i_l}^{i_l+M-1} P^2_{Yf}(\tilde{t}_{bi})\\ &\quad + \frac{M}{4}\sigma^2_{RFa}\sigma^2_{RFb} - \bar{\gamma}^2_{u|H0}\\ &= (\frac{M^2}{16} A^2_{pa} A^2_{pb} [\bar{P}^2_{Yf}]^2 - \bar{\gamma}^2_{u|H0})\\ &\quad + \frac{M}{8}[A^2_{pa}\sigma^2_{RFb} + A^2_{pb}\sigma^2_{RFa}]\bar{P}^2_{Yf}\\ &\quad + \frac{M}{4}\sigma^2_{RFa}\sigma^2_{RFb}\\ &= \frac{M}{4}\sigma^2_{RFa}\sigma^2_{RFb}\{1 + 2\Delta t[(C/N_0)_{pya}\\ &\qquad + (C/N_0)_{pyb}]\} \qquad (22b)\end{aligned}$$

where $(C/N_0)_{pyb}$ = $A^2_{pb}\bar{P}^2_{Yf}/(4\sigma^2_{RFb}\Delta t)$ is the P(Y)-code carrier-to-noise ratio in Receiver B.

The derivations in Eqs. (22a) and (22b) assume that the mean value of the product $P_{Yf}(t_{ai})P_{Yf}(\tilde{t}_{bi})$ also equals $\bar{P}^2_{Yf}$. This is reasonable when the RF front-end filters are similar because the Receiver A time t_{ai} and the Receiver B time $\tilde{t}_{bi}$ are the same times relative to their respective P(Y) codes by virtue of the construction of $\tilde{t}_{bi}$ in Eq. (17). Of course, a stricter use of notation would have created slightly different function names for $P_{Yf}(t)$ in the two receivers in order to allow them to take on the same value at the different input time arguments t_{ai} and $\tilde{t}_{bi}$.

The carrier-to-noise ratios $(C/N_0)_{pya}$ and $(C/N_0)_{pyb}$ in the final forms of Eqs. (21b)-(22b) are used in place of terms involving $A^2_{pa}\bar{P}^2_{Yf}$ and $A^2_{pb}\bar{P}^2_{Yf}$. This convention is adopted because it is convenient to deduce the carrier-to-noise ratios. The determination of $(C/N_0)_{pya}$ and $(C/N_0)_{pyb}$ begins with a determination of the corresponding C/A-code carrier-to-noise ratios. Given a time history of prompt accumulations I_k and Q_k for the C/A code, the calculation starts by determining the mean amplitude of the accumulation vector $[I_k; Q_k]$ and the noise variance in each of this vector's components:

$$A_{IQ} = (\bar{z}^2 - \sigma_z^2)^{1/4} \tag{23a}$$

$$\sigma_{IQ}^2 = 0.5(\bar{z} - \sqrt{\bar{z}^2 - \sigma_z^2}) \tag{23b}$$

where $\bar{z}$ and σ_z^2 are, respectively, the mean and variance of the accumulation power:

$$\bar{z} = E\{I_k^2 + Q_k^2\} = \frac{1}{K}\sum_{k=1}^{K}(I_k^2 + Q_k^2) \tag{24a}$$

$$\sigma_z^2 = E\{[I_k^2 + Q_k^2]^2\} - \bar{z}^2 = \frac{1}{K}\sum_{k=1}^{K}(I_k^2 + Q_k^2)^2 - \bar{z}^2 \tag{24b}$$

As a side benefit, the accumulation variance in Eq. (23b) can be used to estimate the effective variance of the noise in the raw RF samples:

$$\sigma_{RF}^2 = \frac{2}{\overline{N}_{accum}}\sigma_{IQ}^2 \tag{25}$$

where $\overline{N}_{accum} = (N_1+N_2+...+N_K)/K$ is the average number of samples in an accumulation. The value of this variance for each receiver is needed in Eqs. (21b) to (22b).

The C/A-code carrier-to-noise ratio is computed from the accumulation amplitude and variance in Eqs. (23a) and (23b). Given the accumulation interval $T_{accum} = \Delta t\,\overline{N}_{accum}$, the carrier-to-noise ratio is:

$$(C/N_0)_c = \frac{A_{IQ}^2}{2\sigma_{IQ}^2 T_{accum}} \tag{26}$$

Given the C/A-code carrier-to-noise ratio, the P(Y) code carrier-to-noise ratio can be computed. This calculation considers the effects of filter loss and distortion, as per Eqs. (15) and (16), and the transmitted power decrement of the P(Y) code in comparison to the C/A code, as per Eq. (2). The resulting formula is

$$(C/N_0)_{py} = L_{fpy}L_p\left[\frac{10^{-0.04/10}(C/N_0)_c}{L_{fca}}\right] \tag{27}$$

The power of 10 in this equation adjusts for the fact that the L_{fca} loss calculation in Eq. (16) presumes an infinite bandwidth of the transmitted C/A code instead of the actual 20.46 MHz bandwidth. The term in square brackets on the right-hand side of this equation is what the received C/A-code carrier-to-noise ratio would have been had there been no loss in the filter or in the prompt accumulation calculations.

The formulas in Eqs. (23a)-(27) apply to Receivers A and B. The usual "*a*" and "*b*" subscripts can be added to each of the quantities in order to denote the receiver to which it applies.

Typically the variance results in Eqs. (23b), (24b), and (25) are computed only once when the receiver is operating on a quiescent signal with very little actual amplitude fluctuation. These quantities tend to remain constant over time due to the actions of the RF front-end's automatic gain control.

The signal power quantities in Eqs. (23a) and (24a) and the associated carrier-to-noise ratios in Eqs. (26) and (27) are typically re-computed continually. One might re-compute them for each spoofing detection accumulation interval. This approach enables the spoofing detection test to adapt to the time variations in signal power that typically occur.

Before developing the spoofing test, it is helpful to normalize the test statistic. A suitable normalization is to divide γ_{ul} by its standard deviation under the spoofed hypothesis H_1, $\sigma_{\gamma ul|H1}$. This produces the normalized spoofing test statistic:

$$\gamma_l = \frac{\gamma_{ul}}{\sigma_{\gamma ul|H1}} = \frac{\sum_{i=i_l}^{i_l+M-1} y_{qai}\tilde{y}_{qbi}}{\sigma_{RFa}\sigma_{RFb}\sqrt{\frac{M}{4}[1+2\Delta t(C/N_0)_{pya}]}} \tag{28}$$

The results in Eqs. (21a)-(22b) can be used to compute the means and standard deviations of this statistic under the respective hypotheses of spoofing on Receiver B, H_1, and no spoofing, H_0. These quantities are:

$$\bar{\gamma}_{H1} = 0 \tag{29a}$$

$$\sigma_{\gamma|H1} = 1 \tag{29b}$$

$$\bar{\gamma}_{H0} = 2\Delta t\sqrt{\frac{M(C/N_0)_{pya}(C/N_0)_{pyb}}{1+2\Delta t(C/N_0)_{pya}}} \tag{29c}$$

$$\sigma_{\gamma|H0} = \sqrt{\frac{1+2\Delta t[(C/N_0)_{pya}+(C/N_0)_{pyb}]}{1+2\Delta t(C/N_0)_{pya}}} \tag{29d}$$

The means and variances in Eqs. (29a)-(29d) can be used to design and analyze a sensible spoofing detection test. The necessary derivations require knowledge of the spoofed and un-spoofed probability density functions $p(\gamma_l|H_1)$ and $p(\gamma_l|H_0)$. The exact formulas for these probability density functions are complicated because the γ_l statistic involves products of the Gaussian noise terms n_{qai} and $\tilde{n}_{qbi}$. Fortunately, the randomness in γ_l is the result of many such product terms. Therefore, the central limit theorem can be invoked in order to model these two probability density functions as Gaussian distributions.

Given the Gaussian assumption and given the allowable false-alarm probability α_{FA}, the spoofing detection threshold γ_{th} can be computed by solving the following equation:

$$\alpha_{FA} = \int_{-\infty}^{\gamma_{th}} p(\gamma_l | H_0) d\gamma_l$$

$$= \frac{1}{\sqrt{2\pi}\sigma_{\gamma_l|H0}} \int_{-\infty}^{\gamma_{th}} exp\{-\frac{(\gamma_l - \bar{\gamma}_{H0})^2}{2\sigma_{\gamma_l|H0}^2}\} d\gamma_l \tag{30}$$

This threshold is used to determine whether the signal in Receiver B is being spoofed according to the following rule: If $\gamma_l \geq \gamma_{th}$, then accept the H_0 hypothesis that there is no spoofing, but if $\gamma_l < \gamma_{th}$, accept the H_1 hypothesis that there is spoofing. This threshold and spoofing test lead to the following probability of a successful detection:

$$\mathcal{P}_{detect} = \int_{-\infty}^{\gamma_{th}} p(\gamma_l | H_1) d\gamma_l$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\gamma_{th}} exp\{-0.5\gamma_l^2\} d\gamma_l = 1 - \mathcal{P}_{misdet} \tag{31}$$

Note that the H_0 un-spoofed hypothesis is somewhat unusual: It has a non-zero mean that is calculated by factoring down the measured C/A carrier-to-noise ratio in order to estimate the P(Y) carrier-to-noise ratio. It is important to use the proper calculation of the C/A carrier-to-noise ratio in Eqs. (23a)-(26) and the proper attenuation to get the P(Y) carrier-to-noise ratio in Eq. (27). Errors in these calculations will cause errors in the un-spoofed expected value $\bar{\gamma}_{H0}$ and in the spoofing detection threshold γ_{th}. These errors will cause the detection test to have a different false-alarm probability and a different probability of detection than are given in Eqs. (30) and (31).

The analysis of this section assumes that the noise in the quadrature baseband-mixed signal is purely white noise. This assumption is violated to some extent in any real receiver. For the receivers considered in the present study, their departures from the white-noise assumption do not appear to be large enough to have a significant impact on the spoofing detection results. If the non-whiteness of the noise were an issue, then it would be straight-forward to adapt the foregoing analysis appropriately. This adaptation is omitted for the sake of brevity.

The detection statistic γ_l would be the optimal Neyman-Pearson detection statistic [20] if the noise in Receiver A were negligible and if the prediction of the P(Y) carrier-to-noise ratios for the two receivers were exact. In that case, the Receiver-A quadrature signal would yield a perfect scaled replica of the receiver's distorted version of the encrypted P(Y) code. One could use this replica and the P(Y) amplitudes on Receivers A and B in order to derive the joint probability density functions for y_{qbi} for $i = i_l, \ldots, i_l+M-1$ under the two hypotheses. One could demonstrate a monotonic, one-to-one correspondence between the ratio of these two probability density functions and the γ_l test statistic. This correspondence would prove the optimality of the γ_l statistic. Sub-optimality of the test statistic is tolerated because of Receiver A's imperfect knowledge of its distorted P(Y) signal.

C. Potential for Cross-Talk between Channels

There is a potential for the P(Y) code or even the C/A code of another GPS signal to affect the spoofing detection statistic γ_l in Eq. (28). This can happen if the Doppler shifts and code delays of the other GPS signal line up in a certain way with those of the signal for which spoofing detection is being performed. The necessary Doppler alignment to cause interference is that of a zero-valued or nearly zero-valued Doppler double difference between the two receivers and the two signals. That is, if the carrier Doppler shift difference between the two GPS signals is the same at both the reference receiver and the defended receiver, then there is a potential interference. This difference must be smaller than the correlation accumulation frequency $1/T_{corr}$. Otherwise, the averaging action of the accumulation in Eq. (28) will attenuate the interference.

An additional requirement for interference between two signals is that their double-differenced PRN code phase be zero or nearly zero. That is, the C/A code period start/stop time difference between the two signals for the reference receiver must equal this same difference for the defended receiver. If this code-phase double difference is less than the correlation time of the filtered P(Y) code, then unintended cross-correlations of the P(Y) code of the other signal can appear in the γ_l spoofing detection statistic of Eq. (28). Similarly, if this code-phase double difference is less than a C/A code PRN chip length, then un-intended cross-correlations of the other signal's C/A code can appear in γ_l. The C/A code of the second signal could affect the P(Y) cross-correlation of the signal in question because the second C/A code could lie nearly in phase quadrature with the C/A code of the original signal.

This type of interference was noted in the study of codeless cross-correlation spoofing detection found in Ref. 13. In that study, the two receivers were both located in Ithaca, NY. Given this close proximity, the carrier Doppler shift double differences and the code phase double differences were likely to be small, and interference was likely to occur.

Under normal conditions, it is unlikely that two signals will interfere due to small double differences in Doppler shift and code phase. Large double differences will normally be caused by the necessary receiver separation between the secure reference receiver and the defended receiver. If both double differences are small, however, then this fact will be noticeable from the C/A code tracking, and the spoofing detection calculations for the signals in question must be ignored or modified. Otherwise, the computed γ_l can be much larger than expected, much smaller than expected, or even negative [13]. These possibilities arise because additional non-zero correlations of the second signal can add constructively or destructively to alter the mean value of γ_l.

It is possible to reduce or even eliminate this type of interference at the reference station. The necessary infrastructure would be a high-gain antenna system with independently steerable beams, such as could be provided by a phased array. Given sufficient gain, the interference effects of other signals on γ_l would be negligible even with zero-valued double differences of Doppler shift and code phase.

D. Analysis of Spoofing Detection Correlation Intervals

The probability of codeless spoofing detection, $\mathcal{P}_{detect}$ in Eq. (31), has been calculated as a function of the correlation interval, T_{corr}. This functional dependence is plotted in Fig. 2 for four values of the P(Y) code carrier-to-noise ratio, $(C/N_0)_{py}$. This analysis assumes that the carrier-to-noise ratio is identical in the two receivers. The four $(C/N_0)_{py}$ values of Fig. 2 span the range of observed values in the narrow-band RF front-end used in the present study. This figure indicates that reliable spoofing detection can be achieved with correlation intervals as short as T_{corr} = 0.1 sec when $(C/N_0)_{py}$ = 44 dB-Hz and as long as T_{corr} = 1.6 sec when $(C/N_0)_{py}$ = 38 dB-Hz.

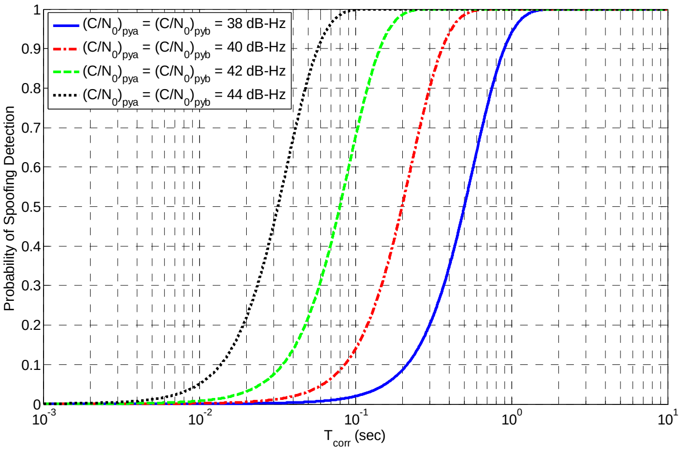

Fig. 2. Spoofing detection power as a function of correlation interval for four representative narrow-band carrier-to-noise ratios (false alarm probability = 0.01%; i.e., α_{FA} = 0.0001)

IV. Experimental Spoofing Detection Results

A. Cases Considered

This paper's spoofing detection algorithm has been implemented and tested on actual data. The algorithm runs in MATLAB software receiver code that operates on recorded RF data in an off-line mode. The RF data have been collected simultaneously from reference Receiver A operating in Ithaca, NY and from defended Receiver B operating in Austin, TX. Both receivers were connected to roof-mounted patch antennas.

The RF front-ends of the 2 receivers have 3 dB bandwidths of 2.4 MHz (Ithaca) and 2.6 MHz (Austin). The former front-end attenuates the P(Y) signal power by 5.6 dB, and the latter by 5.4 dB.

In a first test, the Austin receiver was not subjected to a spoofing attack. The first test was conducted in Sept. 2010. In a second type of test, the Austin receiver was attacked using an advanced version of the spoofer that is described in Refs. 5 and 6. Various versions of the second test were conducted in Sept. 2010 and in July 2011. Results for the second type of test will be reported only for the July 2011 data because that data employed the most sophisticated version of the spoofer.

The spoofing attack was carried out by combining the signal from the spoofer with the signal from the Austin, TX roof-mounted patch antenna. This combining operation was carried out electronically before input to the RF front-end of the defended receiver. This approach avoided violation of FCC regulations because the spoofing signal was never broadcast. The spoofer also had access to the signal from a roof-mounted antenna, as required by the spoofer design of Refs. 5 and 6. It used this data to lay the spoofed signal exactly on top of the true signal during the initial attack. This attack profile allowed the victim receiver to continue tracking C/A code without interruption and seemingly without problems during the attack.

A special spoofing protocol has been used for the July 2011 spoofed case. The initial 60 seconds of data have no spoofing. The spoofer turns on at about 60 seconds, but it keeps its spoofed C/A code exactly on top of the true C/A code for about the first 60 seconds of spoofing. During this initial period, there is zero carrier Doppler shift of the spoofed signal relative to the true signal. The spoofing detection algorithm will still see the true P(Y) code on the quadrature channel in this phase unless the spoofed C/A code has exactly a 90 deg phase offset from the true C/A code. In this situation, however, the true P(Y) code will not have the correct amplitude relationship to the spoofed C/A code because the latter will have a higher amplitude than the true C/A code in order to take control of the receiver's tracking loops. At about 120 seconds into the spoofing run, i.e., about 60 seconds after the onset of the attack, the spoofer starts to move the spoofed C/A code phase away from the true code. This process is necessary if the spoofer wants to deceive the receiver about its position or time. The receiver's C/A-code tracking loops are dragged away from the true C/A code by the spoofed signal during this latter phase of the attack. This causes the P(Y) code in the quadrature channel of the victim receiver to have a very large timing offset relative to the tracked, spoofed C/A code, and the spoofing detection test statistic should drop to a mean of zero at this point of the attack.

Only a subset of the visible GPS satellites had their C/A PRN codes spoofed in the attack. There were 9 signals present in the data, but only 6 of them were spoofed.

B. Performance of Codeless Spoofing Detection

Results for the codeless spoofing detection test are shown in Figs. 3 and 4. Figure 3 corresponds to an un-spoofed case. It plots the detection statistic γ (solid blue curve), the statistic's predicted mean value $\bar{\gamma}_{H0}$ (dotted red curve), and the 0.01% false-alarm spoofing detection threshold γ_{th} (dashed green curve). The γ statistic has been computed using the cross-correlation interval T_{corr} = 1.2 sec. These curves apply to PRN 17, a typical tracked signal. The mean and threshold values have been computed based on the assumption that the P(Y) code is transmitted with a power level that is $10log_{10}(L_p)$ = -3.04 dB down from that of the C/A code. This is the value that causes $\bar{\gamma}_{H0}$ to equal the mean of γ -- note the correspondence between the level of the dotted red curve and the mean value of the solid blue curve. This case demonstrates the efficacy of the spoofing detection test: It clearly recognizes that this signal is not being spoofed. It also demonstrates the reasonableness of the statistical signal modeling that went into deriving the mean value $\bar{\gamma}_{H0}$ and the detection threshold γ_{th}.

Figure 4 demonstrates the codeless detection method's performance during a spoofing attack. Again, this figure plots time histories of the detection statistic γ, its predicted mean value $\bar{\gamma}_{H0}$, and the corresponding 0.01% false-alarm spoofing detection threshold γ_{th}, all calculated using 1.2 sec cross-correlation intervals. These quantities are plotted for two signals: PRN 13, which undergoes a spoofing attack starting at t = 60 sec, and PRN 23, which remains un-spoofed for the duration of the test. Unlike Fig. 3, the $\bar{\gamma}_{H0}(t)$ and $\gamma_{th}(t)$ time histories fluctuate because their levels are computed based on time-varying averages of the two receivers' C/A-code carrier-to-noise ratios. Each average is taken over the corresponding spoofing detection cross-correlation interval. The C/A to P(Y) transmitted power loss factors that have been used to produce these $\bar{\gamma}_{H0}(t)$ and $\gamma_{th}(t)$ plots are $10log_{10}(L_p)$ = -3.93 dB for PRN 13 and $10log_{10}(L_p)$ = -3.80 dB for PRN 23. These values have been chosen to make the $\bar{\gamma}_{H0}(t)$ curves lie close to the $\gamma(t)$ curves during the un-spoofed first 60 seconds of this case.

Figure 4 shows clear responses at the time of the initial attack and further response changes as the attack progresses to carry the tracking loops away from the true signal. The spoofing detector correctly identifies the fact that PRN 13 is spoofed starting at t = 60 sec and that PRN 23 is never spoofed. PRN 13's solid blue spoofing detection statistic drops below its dashed green detection threshold and remains below that value except for a short interval from t = 164 to 169 sec. During this latter interval, the detection fails briefly because the detection power falls to low levels. This happens because the spoofed and true C/A codes briefly interfere with each other to produce a short, sharp power fade on that signal. PRN 23, on the other hand, never generates a spoofing (false) alarm. Its solid turquoise detection statistic never drops below its corresponding dashed brown detection threshold.

It is interesting to note the behavior of spoofed PRN 13's detection statistic during the two phases of the attack. During the interval from t = 60 sec to t = 150 sec, the spoofed signal exactly overlays the true signal. The detection statistic drops a small amount, but not to a mean value of 0. The residual non-zero mean value is the result of the P(Y) code still being present, though not with the same amplitude as before the attack. One of the reasons for the amplitude reduction is the larger overall power entering the RF front-end at the onset of the attack. The spoofing signals must have higher power than the true signals in order to capture the receiver's tracking loops. This extra power affects the RF front-end's Automatic Gain Control (AGC), causing it to lower the gain. This lowered gain translates into a lowered received power of the true P(Y) code in Receiver B. This lower power reduces the value of the detection statistic. A second possible reason for the drop in the statistic during the middle interval is that the

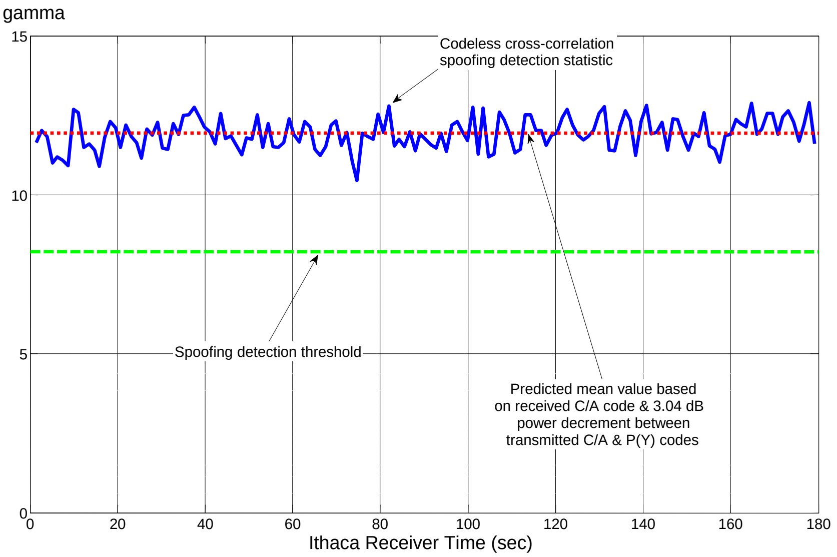

Fig. 3. Codeless spoofing detection statistic time history for PRN 17, un-spoofed case (T_{corr} = 1.2 sec, α_{FA} = 0.0001).

spoofed C/A code phase probably does not match the true C/A code phase. Therefore, the quadrature baseband mixing will not exactly capture the P(Y) code, thus reducing the detection statistic's amplitude. In an extreme situation, the detection statistic could take on a negative mean value during this phase. Starting at about t = 150 sec, the spoofer drags the receiver away from the true C/A code. It also drags the quadrature channel away from the true P(Y) code, and the spoofing detection statistic drops to a mean value of zero, as expected.

One might think that the spoofing detection test would not detect the attack until the last phase, when the spoofer drags the receiver away from the true C/A code phase. In fact, the detection is successful at the very outset of the attack. This happens because the spoofing detection threshold rises suddenly: Note the sudden jump of the green dashed curve at t = 60 sec. This rise is caused by the increased C/A code power of the combined spoofed plus true signal during this phase of the attack. This rise is sufficient to cause the spoofing alarm to be issued. Note, however, that there could be situations for well executed attacks where the spoofing attack would not be detected until the last phase, the phase of C/A code drag-off. Such a situation is acceptable because a spoofing attack with the spoofed C/A code exactly aligned to the true code represents a benign event.

The detection statistic for un-spoofed PRN 23, the solid turquoise curve, undergoes a sudden drop at the onset of the attack at t = 60 sec. This occurs because the receiver lowers its AGC gain in response to the extra power of the spoofing signals. The effect on an un-spoofed signal is to lower its C/A and P(Y) power, and this lowering of power is what causes the spoofing detection statistic for PRN 23 to decrease suddenly. One might think that this sudden decrease would give rise to a false spoofing alarm. This does not happen because the spoofing detection threshold for PRN 23, the dashed brown curve in Fig. 4, drops at the same time. It drops because it is keyed to the PRN 23 C/A signal power, which also drops in response to the AGC adjustment. Thus, the connection between the C/A-code signal power and the design of the spoofing detection threshold is important to the proper operation of this test.

The results in Fig. 4 might tempt one to suggest a simpler method of detecting the spoofing attack: Look for sudden changes of the AGC and of the C/A code power. If the AGC gain suddenly drops while the C/A power suddenly rises for some of the channels, then declare a spoofing attack. Additionally, small transient carrier phase glitches in the PLL tracking performance are evident on some of the spoofed channels at the onset of the spoofing attack. One might be tempted to look for such glitches and use them to detect a spoofing attack. Unfortunately, these detection methods can be defeated by slowly ramping up the power of the spoofed signals at the beginning of the attack. A slow attack was not used here only because the authors wanted to minimize the amount of data that needed to be tracked using offline MATLAB software receiver code. Such code runs very slowly, and its use on long data sets can be time-consuming.

In addition, the proposed simple detection scheme would work only if applied at or very near the initial moment of the spoofing attack. If the attack were not detected at its onset, then the simple detection methods would fail. This paper's cross-correlation-based detection methods function well during all phases of an attack.

The results in Fig. 4 and related results for other data sets represent the first successful spoofing detections using a single-antenna system at the defended receiver when attacked by the sophisticated spoofer of Refs. 5 and 6. The only other successful detection used a multi-antenna system [15]. This also represents the first successful detection of an actual spoofing attack using the cross-correlation method of Refs. 12, 13, and 14. This demonstration is important because it proves that the vestigial P(Y) code in a narrow-band receiver can form the basis of a powerful spoofing detection test.

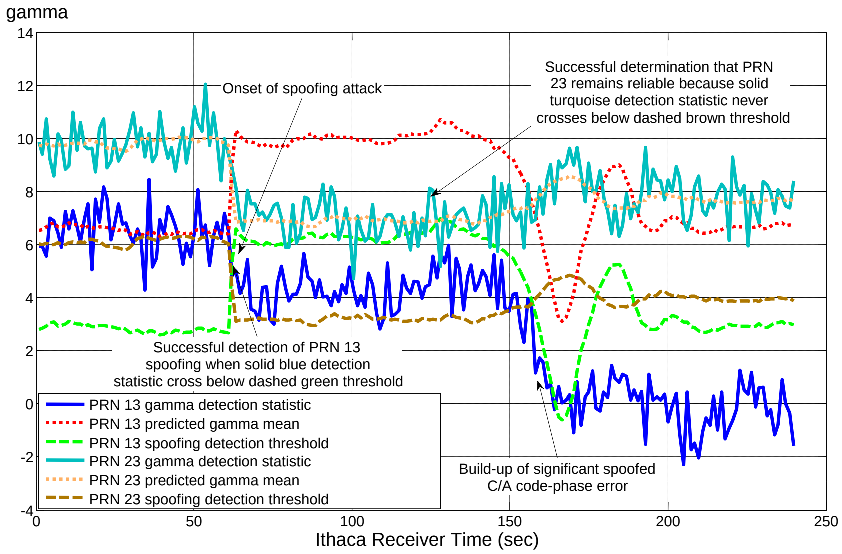

Fig. 4. Codeless spoofing detection statistic time histories for spoofed PRN 13 and un-spoofed PRN 23 (T_{corr} = 1.2 sec, α_{FA} = 0.0001).

The detection powers in all 3 cases associated with Figs. 3 and 4 remain above 0.995, except for PRN 13 during the short interval from t = 160 to 174 sec. As already mentioned, this short anomaly is caused by a drop in the C/A code amplitude due to transient interference between the true and spoofed signals. During steady-state spoofing, no such interference would occur due to the temporal separation between the two codes. The nominally high probabilities of detection indicate that the T_{corr} = 1.2 sec cross-correlation intervals are more than sufficient for a powerful test. They probably could be shortened significantly.

Two additional spoofed signals have been processed for the case associated with Fig. 4, those of PRN 03 and PRN 16. They both required P(Y) transmitted power decrements of $10log_{10}(L_p)$ = -3.37 dB in order to achieve good agreement between $\gamma(t)$ and $\bar{\gamma}_{H0}(t)$ during the initial un-spoofed phase. Spoofing detection worked well for these two signals, similar to the results for PRN 13 in Fig. 4.

C. Investigation of the Effects of Relative Time Offsets between the C/A and P(Y) Codes

A study has been made of the effect on codeless spoofing detection of varying the differential relative time parameter δt_{ab}. Recall from Subsection III.A that this is a differential between Receivers A and B of the timing of the received, filtered P(Y) code relative to the tracked C/A code. Variations of this offset, as propagated through Eqs. (17) and (18), have been assessed in order to determine how they affect the mean cross-correlation amplitude. The correct value of δt_{ab} should give the peak amplitude.

All studies to date show that the peak cross-correlation amplitude occurs at $\delta t_{ab} = 0$ for the receivers and tracking loops that have been considered. The precision of this finding is significantly better than 0.025 C/A code chips (24 nsec). Given that the two receivers' RF front-ends and tracking software were identical to within manufacturing tolerance, this result is not surprising.

If there were significant differences between the receiver RF front-ends, the DLL discriminators, or the DLL tracking loops, then this result might change. In any application of codeless spoofing detection to a new receiver design, this issue should be investigated. If necessary, the optimal value of δt_{ab} should be determined, recorded, and applied as a calibration parameter during regular codeless cross-correlation calculations.

V. Vulnerability to Alternate Methods of Spoofing Attack

This paper's spoofing detection test has been developed by using the methods of statistical hypothesis testing. The test statistic distinguishes between two precisely defined hypotheses. The null hypothesis is that the P(Y) code signal is present in quadrature with the C/A code in the defended receiver and that it has a well defined amplitude ratio relative to the C/A code. This is the un-spoofed hypothesis. The spoofed hypothesis presumes that there is no signal on the quadrature channel.

If the spoofer suspects that this paper's cross-correlation algorithms are being used, then it may elect to do something different than leaving no signal on the quadrature channel. The spoofer may put pseudo P(Y) code on the quadrature channel. This possibility has been considered, and the only effect that is anticipated on the codeless spoofing detection is an increase in the random variability of the spoofing detection statistic about the spoofed hypothesis mean value of 0. This increased variability can be compensated by an increase of the cross-correlation detection interval.

Another possibility for attack is a Security Code Estimation and Replay (SCER) attack [17]. This type of attack actively seeks to estimate the W chips on-line, and it uses its imprecise W-chip estimates in an attempt to spoof the true P(Y) code. This type of attack will dilute the spoofing detection power of a cross-correlation method in direct proportion to the percentage of its correct W-chip estimates. Of course, a large dilution can only be achieved by a high-gain antenna system. If the number of correctly estimated W chips in the spoofer were not too large and if the cross-correlation spoofing detection algorithm had enough power, then this type of attack would be detected. An effective SCER spoofer would have to estimate most of the W chips correctly, which would be expensive in terms of the needed antenna gain.

Alternatively, an SCER attack might try to compensate for mis-estimation of a significant fraction of the W chips by turning up the power of the spoofed P(Y) code. This strategy might thwart the codeless cross-correlation detection test of Section III. Alternate detection statistics, as developed in the original conference version of the present paper [21], could detect this attack mode by performing semi-codeless spoofing detection that involves estimation of the w_j encryption chips.

An SCER spoofer might need to induce a delay of the spoofed C/A code relative to the true C/A code in order to gain time to form its W-chip estimates. The necessary delaying action might be noticeable in the defended receiver at the onset of the attack.

There are other possible attack types. The spoofer might try to locate a second spoofer near the secure receiver. If both spoofers used a common false P(Y) code, then they would defeat this technique. A defense against such an attack would be to distribute an array of secure receivers over a large area and to connect them in a network that aggregated their P(Y) code quadrature samples. If there were enough physically secure receivers, then it would be unlikely that enough of them could be spoofed in a way that would defeat the detection system. Reference stations could employ phased-array antennas with independently steerable beams in order to ensure their security. They could use beam steering to attenuate the signal of any spoofer that was not directly on their line-of-sight vector to a given GPS satellite.

A meaconing attack could also defeat this method. This technique receives and replays the entire GNSS spectrum with some unavoidable delay [17]. This type of attack can even defeat a secure military receiver if the replayed bandwidth is wide enough to contain the P(Y) or M codes. A sophisticated meaconing attack might use differential delays for different signals, which it could implement by using a phased array with independently steerable beams for signal reception prior to replay. This type of attack, however, would be very expensive. A simple meaconing attack with only one delay for all signals would cause the spoofed receiver to determine a location equal to the spoofer's location, which could prove dangerous for the spoofer. Also, a victim receiver with a very stable oscillator might detect the attack because of the necessary time delay.

Other types of spoofing attacks might be mounted against this paper's cross-correlation detection methods. Perhaps a problematic attack would be to raise the noise floor on the quadrature channel instead of putting estimated or false P(Y) code there. The analysis of all such attack scenarios and the performance of this paper's detectors under threat of such attacks is beyond this paper's scope. Several preliminary analyses of this subject suggest that this paper's spoofing detection technique would perform well under many attack scenarios if the power of detection were sufficiently close to 1 for the simple attack scenario discussed in this paper.

VI. Summary and Conclusions

A spoofing detection method has been developed for publicly-known/civilian GNSS signals. It relies on the presence of an encrypted/military signal on the same transmitted frequency. It also relies on knowledge of the timing and carrier-phase relationship of the encrypted signal to the publicly-known signal. The publicly-known signal is tracked in a secure reference receiver and in a defended receiver that might be the victim of a spoofing attack. The publicly-known signal tracking data are used to isolate the part of the received signal that is encrypted. The encrypted parts of the signals from the two receivers are cross-correlated after being brought together via a communications link. This use of cross-correlation obviates the need for a priori knowledge of the PRN code of the encrypted signal. If a high cross-correlation statistic is obtained, then no spoofing has been detected because this large value indicates the presence of the encrypted signal in both receivers. If the cross-correlation statistic is too low, then a spoofing alert is issued. The low cross-correlation is likely due to the absence of the encrypted part of the received signal in the defended receiver. The only explanation for this absence is that the tracked publicly-known signal is a false spoofing signal.

A codeless cross-correlation spoofing detection test has been developed, analyzed, and tested. The analyses enable design of the spoofing detection threshold based on hypothesis testing theory, and they enable prediction of the detection power. The threshold depends on the chosen false alarm probability, on the received power of the publicly-known signal, and on the known power of the encrypted signal relative to the publicly-known signal.

The new technique has been applied to detect actual GPS spoofing attacks using recorded RF data and off-line signal processing. The technique has successfully detected spoofing of the GPS L1 C/A code by cross-correlating the military P(Y) code over accumulation intervals of 1.2 sec. It is likely that a reduction of this interval could be tolerated while maintaining a high detection power.

A surprising aspect of these results is that they have been obtained using low-gain patch antennas and narrow-band receivers. Each receiver's RF front-end had a 2.5 MHz wide filter and a 5.714 MHz sampling rate. These front-ends attenuate the P(Y) code by 5.5 dB and drastically distort its chips. Nevertheless, sufficient P(Y) power remains for successful spoofing detection based on short cross-correlation intervals.

References

[1] Anon., "Vulnerability Assessment of the Transportation Infrastructure Relying on the Global Positioning System," Tech. Report, John A. Volpe National Transportation Systems Center, 2001.

[2] Warner, J.S., and Johnston, R.G., “A Simple Demonstration that the Global Positioning System (GPS) is Vulnerable to Spoofing,” *Journal of Security Administration*, 2003.

[3] Scott, L., "Anti-Spoofing and Authenticated Signal Architectures for Civil Navigation Systems," *Proc. ION GPS/GNSS 2003*, Sept. 9-11, 2003, Portland, OR, pp. 1543-1552.

[4] Scott, L., "Location Assurance," *GPS World*, Vol. 18, No. 7, July 2007, pp. 14-18.

[5] Humphreys, T.E., Ledvina, B.M., Psiaki, M.L., O'Hanlon, B.W., and Kintner, P.M., Jr., "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," *Proc. ION GNSS 2008*, Sept. 16-19, 2008, Savannah, GA, pp. 2314-2325.

[6] Humphreys, T.E., Kintner, P.M., Jr., Psiaki, M.L., Ledvina, B.M., and O'Hanlon, B.W., "Assessing the Spoofing Threat," *GPS World*, Vol. 20, No. 1, Jan. 2009, pp. 28-38.

[7] Pozzobon, O., "Keeping the Spoofs Out, Signal Authentication Services for Future GNSS," *Inside GNSS*, Vol. 6, No. 3, May/June 2011, pp. 48-55.

[8] Brown, R.G., "Receiver Autonomous Integrity Monitoring," in *Global Positioning System: Theory and Applications, Vol. II*, B.W. Parkinson and J.J. Spilker, Jr., eds., American Institute of Aeronautics and Astronautics, (Washington, 1996), pp. 143-165.

[9] Dovis, F., Chen, X., Cavaleri, A., Ali, K., and Pini, M., "Detection of Spoofing Threats by Means of Signal Parameters Estimation," *Proc. ION GNSS 2011*, Sept. 20-23, 2011, Portland, OR, pp. 416-421.

[10] Pini, M., Fantino, M., Cavaleri, A., Ugazio, S., and Lo Presti, L., "Signal Quality Monitoring Applied to Spoofing

Detection," *Proc. ION GNSS 2011*, Sept. 20-23, 2011, Portland, OR, pp. 1888-1896.

[11] Wesson, K.D., Shepard, D.P., Bhatti, J.A., and Humphreys, T.E., "An Evaluation of the Vestigial Signal Defense for Civil GPS Anti-Spoofing," *Proc. ION GNSS 2011*, Sept. 20-23, 2011, Portland, OR, pp. 2646-2656.

[12] Lo, S., De Lorenzo, D., Enge, P., Akos, D., and Bradley, P., "Signal Authentication, A Secure Civil GNSS for Today," *Inside GNSS*, Vol. 4, No. 5, Sept./Oct. 2009, pp. 30-39.

[13] O'Hanlon, B.W., Psiaki, M.L., Humphreys, T.E., and Bhatti, J.A., "Real-Time Spoofing Detection in a Narrow-Band Civil GPS Receiver," *Proc. ION GNSS 2010*, Sept. 21-24, 2010, Portland, OR, pp. 2211-2220.

[14] Levin, P., De Lorenzo, D.S., Enge, P.K., and Lo, S.C., "Authenticating a Signal Based on an Unknown Component Thereof," U.S. Patent No. 7,969,354 B2, June 2011.

[15] Montgomery, P.Y., Humphreys, T.E., and Ledvina, B.M., "A Multi-Antenna Defense: Receiver-Autonomous GPS Spoofing Detection," *Inside GNSS*, Vol. 4, No. 2, March/April 2009, pp. 40–46.

[16] Wesson, K., Rothlisberger, M., and Humphreys, T., "Practical Cryptographic Civil GPS Signal Authentication," submitted to *Navigation*, in review, 2011 (Available at http://radionavlab.ae.utexas.edu/publications/practical-cryptographic-civil-gps-signal-authentication).

[17] Humphreys, T.E, "Detection Strategy for Cryptographic GNSS Anti-Spoofing," submitted to *IEEE Transactions on Aerospace and Electronic Systems*, in review, 2011 (Available at http://radionavlab.ae.utexas.edu/publications/detection-strategies-for-cryptographic-civil-gnss-anti-spoofing).

[18] Anon., "Global Positioning System Wing (GPSW) Systems Engineering and Integration Interface Specification," IS-GPS-200E, Science Applications International Corporation, El Segundo, CA, June 2010.

[19] Psiaki, M.L., and O'Hanlon, B.W., "System Identification of a GNSS Receiver's RF Filter Impulse Response Function," *Proc. ION GNSS 2011*, Sept. 20-23, 2011, Portland, OR, pp. 3690-3708.

[20] Poor, H.V., *An Introduction to Signal Detection and Estimation*, Springer-Verlag, (New York, 1988), pp. 7-195.

[21] Psiaki, M.L., O'Hanlon, B.W., Bhatti, J.A., Shepard, D.P., and Humphreys, T.E., "Civilian GPS Spoofing Detection based on Dual-Receiver Correlation of Military Signals," *Proc. ION GNSS 2011*, Sept. 20-23, 2011, Portland, OR, pp. 2619-2645.