Simulated Signal Acquisition of Multiple Pseudolite Signals

Charles Tytler
Purdue University

Abstract:

As GPS continues to become a more widely used tool every-day, many are looking for unique ways to improve the technology. Pseudolites (short for pseudo-satellites) are local transmitters on Earth that output GPS satellite signals in order to augment the GPS system for use in locations where satellite signals may be obstructed. Broadcasting pseudolite signals in pulses is a method to overcome the near-far problem which is their main limitation. This paper explores through simulation the use of multiple pseudolites broadcasting pulsed signals, and their effects on the signal acquisition capabilities of GPS receivers. In this simulation, using the RTCM SC-104 pseudolite pulsing scheme, a receiver is easily able to acquire satellite signals with the interference of one pseudolite; however this performance quickly degrades as more pseudolite signals are introduced. By modifying receivers to "blank" out pseudolite signal pulses while searching for satellite signals, cross-correlation capabilities increase drastically, allowing uncomplicated acquisition of satellite signals with as many as four pseudolites transmitting nearby.

Introduction:

Global Navigation Satellite Systems (GNSS) have become a key enabler to a wide variety of industries and new uses continue to emerge. The system however is limited in areas where there is not always direct line-of-sight to satellites. This is important to users operating indoors, underwater, or in mines or caves. Several augmentation systems have been designed to improve GNSS performance, one concept that this paper will be examining is pseudolites (short for pseudo-satellites). Pseudolites are transmitters that mimic the function of a GNSS satellite by broadcasting GPS-like signals, but from a local area, often from the ground. The introduction of local signals has several benefits. When placed appropriately, they can provide additional signals to users in locations that obscure signals from GPS satellites. Operators that require a high level of accuracy or reliability can also benefit from the introduction of additional signals, as pseudolites could act as a redundant fail-safe signal in addition to the extra measure for ranging.

One of the largest problems concerning pseudolites is known as the near-far problem. Pseudolites act just like a satellite signal, but the problem is satellites are transmitting from thousands of kilometers away, while pseudolites are very close to the user. This can create a problem of the pseudolite signal overpowering the satellite signals when the user is too close; but if a user is located too far from the pseudolite the opposite occurs and the signal will be too weak for the receiver. Pseudolite signals must be designed in a way to minimize the near-far problem in a way that does not interfere with the satellite signals.

One solution to the near-far problem is transmitting pulsed signals from the pseudolites, a type of Time Division Multiple Access (TDMA). In this process the pseudolite signals would be broadcast in short pulses that receivers could pick up, and transmit nothing in between pulses. By only transmitting a fraction of the time, receivers could acquire pseudolite signals from the pulses and the satellite signals in between them. This paper will simulate a receiver acquiring signals from pulsed pseudolites in addition to GPS satellites. It will examine the effects that multiple pseudolites would cause, from one pseudolite transmitting to four. The maximum number of pseudolites tested is four since that is the required number of satellites for full positioning capabilities. Through similar reasoning, three GPS satellites are present, as this would be necessary if only one pseudolite is operating.

The paper will examine the signal acquisition characteristics of a GPS receiver when the pseudolites are present. Also, the advantages of a technique termed "blanking" will be examined for this situation. To do this, the simulation will also be run with a modification to the receiver, in which it will blank out its input (set it equal to zero) in the presence of a pseudolite pulse, in the hopes of improving satellite signal acquisition.

SIMULATING GPS AND PSEUDOLITE SIGNALS

GPS satellites output signals in two forms, the Coarse Acquisition Code (C/A Code), and the encrypted Precision Code (P(Y) Code). The codes use a form of Code Division Multiple Access (CDMA) so they can all transmit at the same frequency. This paper will use the standards set by The Radio Technical Commission for Maritime Services subcommittee, RTCM SC-104, for pulsing pseudolite signals, which uses the C/A Code.

The C/A Code is made up of a Pseudo-Random Noise (PRN) code that is modulated onto a carrier wave at frequency 1.023 MHz; such that one period is equivalent to 1 millisecond. This signal is also modulated by a Data message which transmits at 50 bps.

Generating PRN codes

The PRN codes are 1023 bit long codes generated through a 10-stage Linear Feedback Shift Register with Gold Codes. The PRN codes for the satellites and pseudolites are generated as defined in GPS system reference document GPS-IS-200D. The document contains a set of pre-defined PRN codes for satellites (PRN 1 – 32) and for ground transmitters (PRN 33 – 37).

For this simulation the three satellites will use PRN 8, PRN 15, and PRN 22, and the four pseudolites will use PRN 33, PRN 34, PRN 35, and PRN 36, all selected randomly. The codes are generated with a sampling rate of 4 samples per chip.

Pseudolite Pulsing

Pulsing of a pseudolite is normally defined with two variables: the pulse duration, the length of one pulse, and the pulse duty cycle (PDC), the percentage of time it transmits.

The recommended pulsing scheme set by RTCM SC-104 is to have a pulse duration of 1/11 of a code period (1 millisecond). This is equivalent to 93 code chips per pulse, with one pulse occurring each period. Every 10^{th} period, however, 2 pulses are transmitted to create an average pulse duty cycle of 10%. The position of the pulse within the period is altered for each pulse, such that after 10 periods of C/A code, a complete pseudolite PRN code is received.

SIMULATING RECEIVED SIGNAL

The signals generated for the GPS satellites and pseudolites were modulated with a sine wave of frequency 1.023MHz, to match signals transmitted in the L1 frequency domain. A Simulink model of the modulation for the pseudolite PRN 35 code is seen below in Figure 1. An amplification was applied to each of the sine waves to account for the different levels of signal power for the pseudolites and satellites. All signals were set to have zero Doppler shift to simplify cross-correlation calculations, as well as view the received signal with interference from all signals combined in a single frequency. The modulated satellite and pseudolite signals were then added together along with a function of noise to produce one overall signal at 1.023MHz as seen in Figure 2. The ephemeris data message was not included in the simulation as its transmission rate is so slow that it does not impact signal acquisition substantially.

The simulation runs for 20 milliseconds, or 20 periods of the C/A codes. This allows two full PRN codes from the pseudolites to be transmitted, to ensure that the receiver will be able to cross-correlate the signals with the locally generated pseudolite PRN codes.

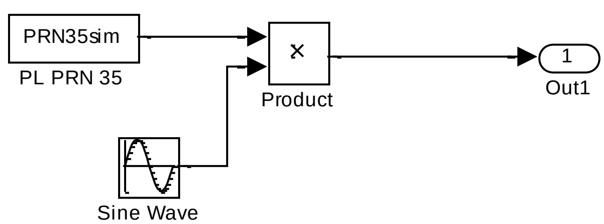


Figure 1: Modulation of PRN Code with Carrier Code Sine Wave

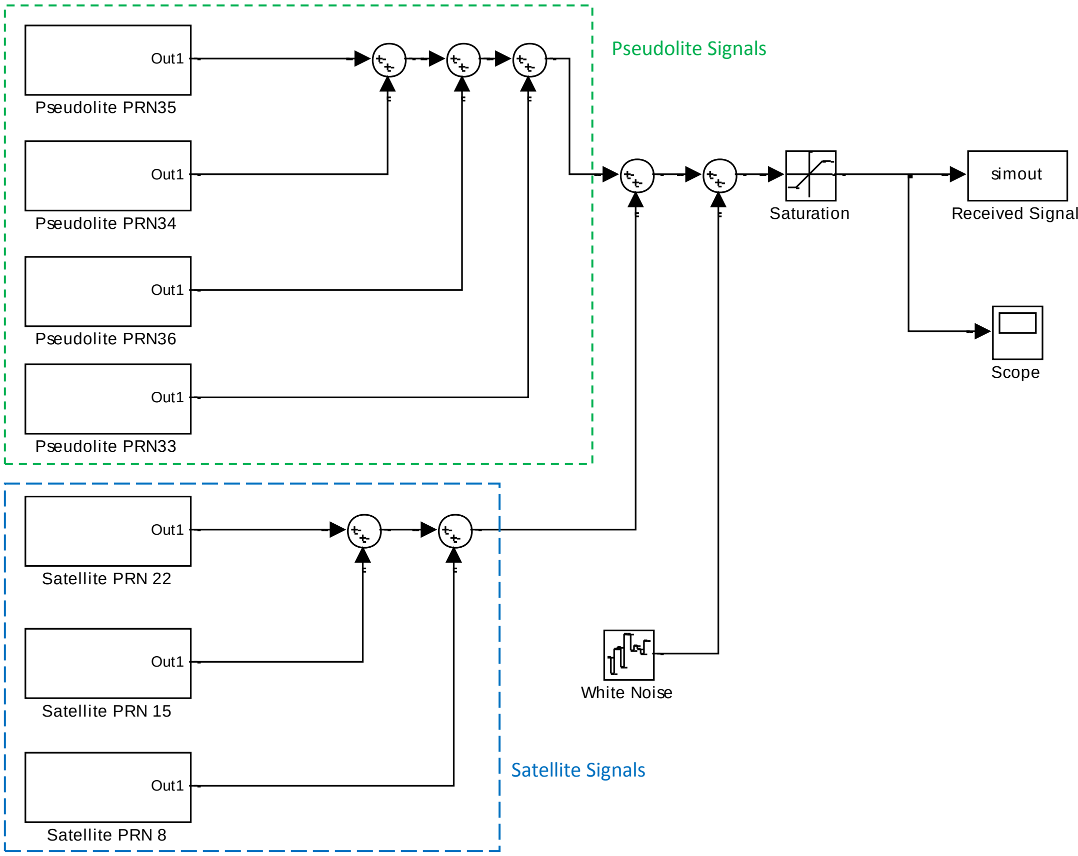


Figure 2: Simulink model to simulate GPS/Pseudolite signals, noise, and receiver

Noise

For GPS receivers, the only transmitted signals they encounter in their frequency range are almost entirely those emitted by GPS satellites, and in this case, pseudolites too. Ambient noise is the only real interference experienced in the GNSS L1 frequency band other than the signals from different satellites. The ambient noise can be classified as white noise, noise that has components in all frequencies with equal strength. The power of the noise for a given bandwidth can be calculated in relation to ambient temperature with the equation

$$P_N = kTB_N \text{ [Watts]}$$

where k is Boltzmann's constant, T is the ambient temperature (in Kelvin), and B_N is the bandwidth (in Hertz). By assuming a standard ambient temperature of 290K, and setting the bandwidth to that of the C/A code, 2 MHz, an estimate of the ambient noise can be calculated as -141dBW. The noise in simulation is modeled with a Simulink Band-Limited White Noise generator, that outputs normally distributed random numbers, with a power level of -141dBW.

Signal Power

GPS satellites are required to transmit their signals such that receivers on Earth will receive them with a power of no less than -160dBW. While receiver antennas will usually receive the C/A code with a higher signal power than this by several decibels, we will use -160dBW as the basis for our simulated signals to ensure code-correlation is possible for all situations. In comparison to our calculated noise, this provides a Signal to Noise Ratio (SNR) of -19dBW.

Since pseudolites broadcast at such a close distance from the receiver, there is a wide range of possible received power levels (as mentioned with the Near-Far Problem earlier). If the receiver is close enough, the pseudolite signal can actually saturate the receiver. In this simulation the pseudolites will be set to a power such that they saturate the receiver, as this provides an easy way to ensure overpowering all other signals. Overpowering the other signals is beneficial for pseudolite acquisition since it is only being transmitted for 10% of the time that the satellites are. When the receiver is saturated it can be seen as only receiving that one signal (one of the problems of the Near-Far Problem), which will help when cross-correlating with the local PRN code.

Saturation

Once all of the signals and noise have been added together, the signal goes through a Saturation block that models the saturation a GPS receiver would experience with such a signal. All of the pseudolite signals saturate the receiver by design, and the signal that the receiver ends up with is seen below in Figure 3. The signal plot shows a signal with 4 pseudolites and 3 satellites transmitting. The tall vertical bars are the pseudolite pulses, and the satellite signals are buried in the noise between the pulses.

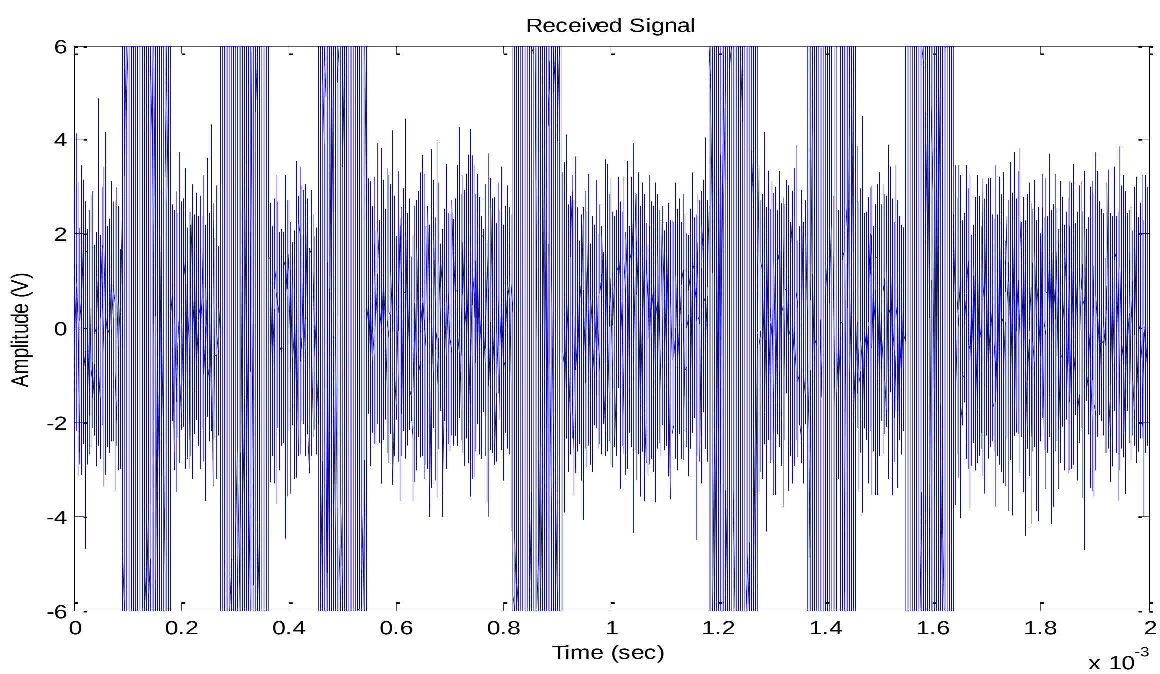


Figure 3: A 2 millisecond Sample of the Received Signal from the Simulink Model

SIGNAL ACQUISITION WITH RECEIVED SIGNAL

Quantization of Received Signal

The Simulink model samples the received signal at 12 MHz. The data is then converted with a 1.5 bit Analog to Digital Converter (3 levels of quantization).

Local Generation of PRN Codes

In order to track a satellite, the receiver must generate its own local copy of the PRN codes and search for correlations of them within its received signal. For this paper, copies of the PRN codes that were generated for the satellite and pseudolite signals are used. They are then multiplied by the carrier wave of the form

$$\sin\left(2\pi f_{C/A}\, t\right).$$

Once modulated onto the carrier wave, the local code is sampled at 12 MHz to match the output of the receiver from the Simulink model.

Cross-correlation of PRN Codes and Signal

In order for a GPS receiver to track and make use of satellite signals, it must be able to acquire them without any prior information other than their PRN codes. The local copies of the PRN codes are cross-correlated with the input signal from the receiver through a Fast Fourier Transform (FFT) method, which outputs an ambiguity function versus time delay of the local and transmitted code. Plotted below are the cross-correlations for satellite signals, PRN 22, PRN 15, and PRN 8, for the cases of 1 to 4 pseudolites present. The satellite signals were generated so that there is no time delay for easier presentation. When a satellite signal is acquired, the plot of the cross-correlation should have a peak for every code period; so peaks at zero delay as well as 1 millisecond (1×10^{-3} s) apart indicate an acquired signal.

As you can see in Figure 4, with one pseudolite the receiver has no problem acquiring all of the satellite signals. However, as more pseudolites are introduced to the signal, the cross-correlation properties degrade. With two pseudolites transmitting, PRN 22 is still visible as well as PRN 15, but PRN 8 can no longer be distinguished. When 3 or 4 pseudolites are present, none of the satellite signals can be acquired. This is not surprising as the satellite signals were already below the noise level, and when up to 40% of the incoming signal is taken by pseudolite transmissions saturating the receiver, chances of acquiring those satellite signals seem pretty slight.

The ability to track the pseudolite signals is seen in Figure 8. When all four pseudolites are transmitting, each is able to be identified very clearly.

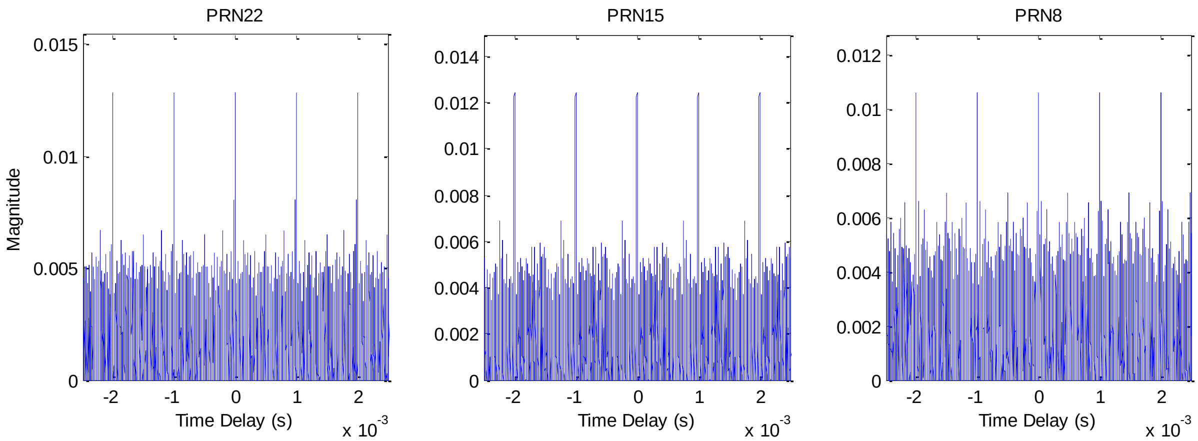


Figure 4: Satellite Code-correlation with 1 Pseudolite Transmitting

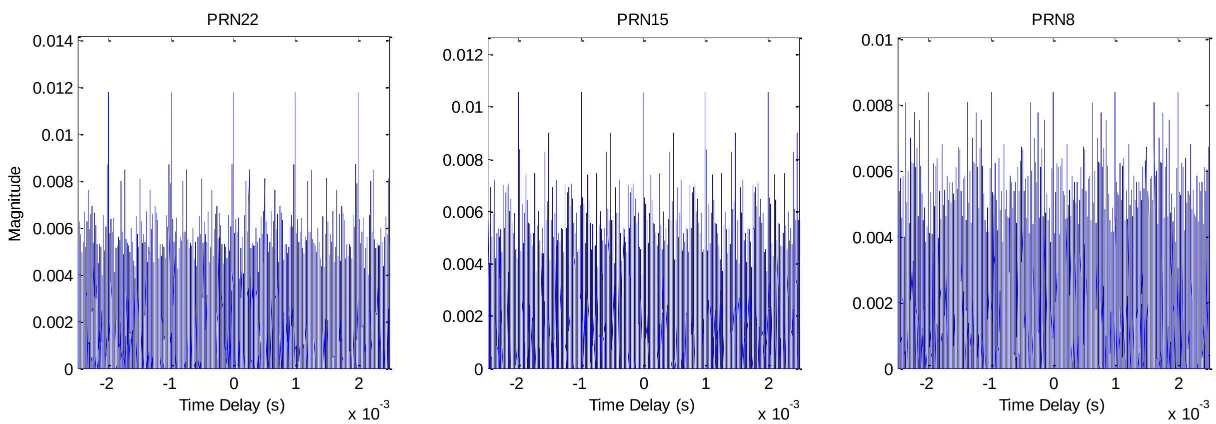


Figure 5: Satellite Code-correlation with 2 Pseudolites Transmitting

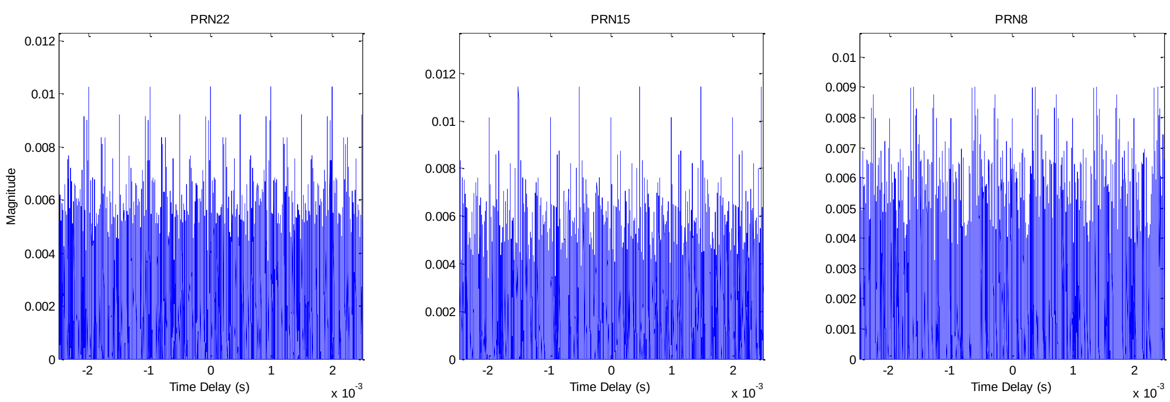


Figure 6: Satellite Code-correlation with 3 Pseudolites Transmitting

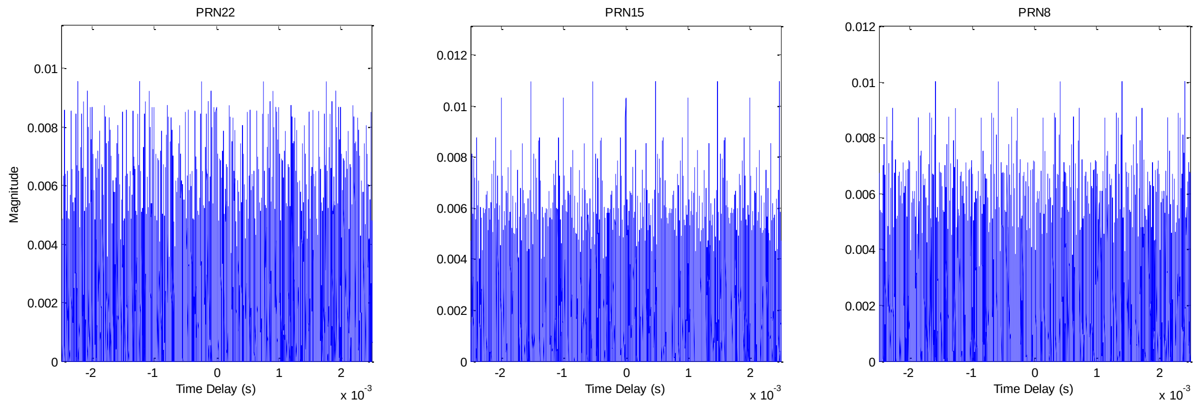


Figure 7: Satellite Code-correlation with 4 Pseudolites Transmitting

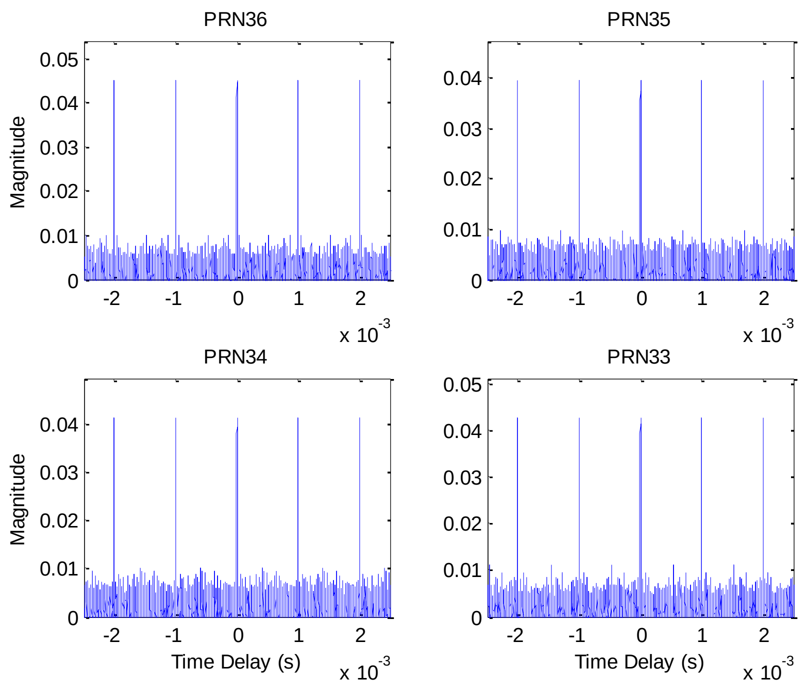


Figure 8: Pseudolite Code-correlation for 4 Pseudolites Transmitting

PSEUDOLITE BLANKING

A suggested method for overcoming the interference caused by pseudolite signals that saturate the receiver is to blank them from input into the receiver. If a receiver is modified so that it is able to blank its input when it detects saturation, it will result in a received signal as seen below in Figure 9.

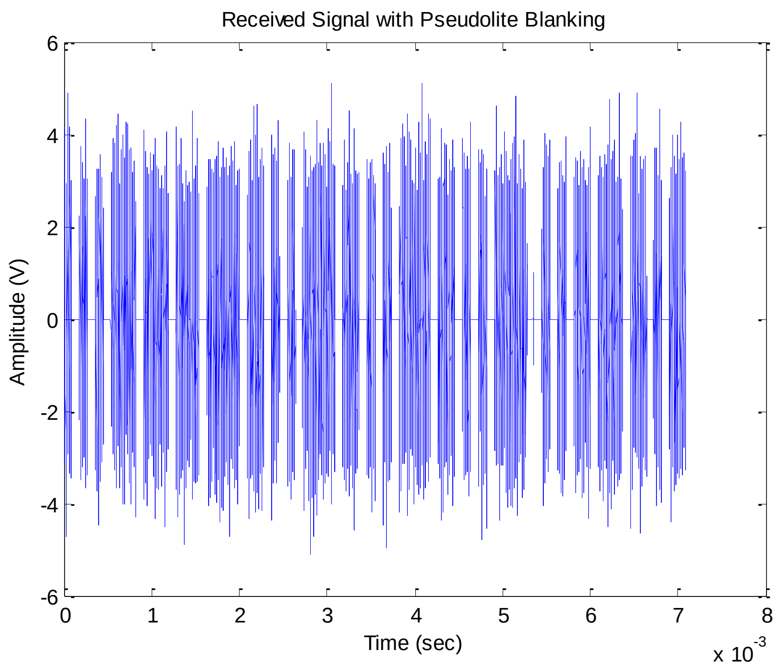


Figure 9: A 7 millisecond sample of the received signal with Pseudolite Blanking with 4 pseudolites transmitting (Compare to Figure 3)

The pseudolite blanking results in a signal that maintains its average amplitude while still in the presence of pseudolites. If this is done while attempting to acquire satellite signals, the results are impressive. Figure 10 below shows the cross-correlations with the pseudolite-blanked signal with 4 pseudolites present.

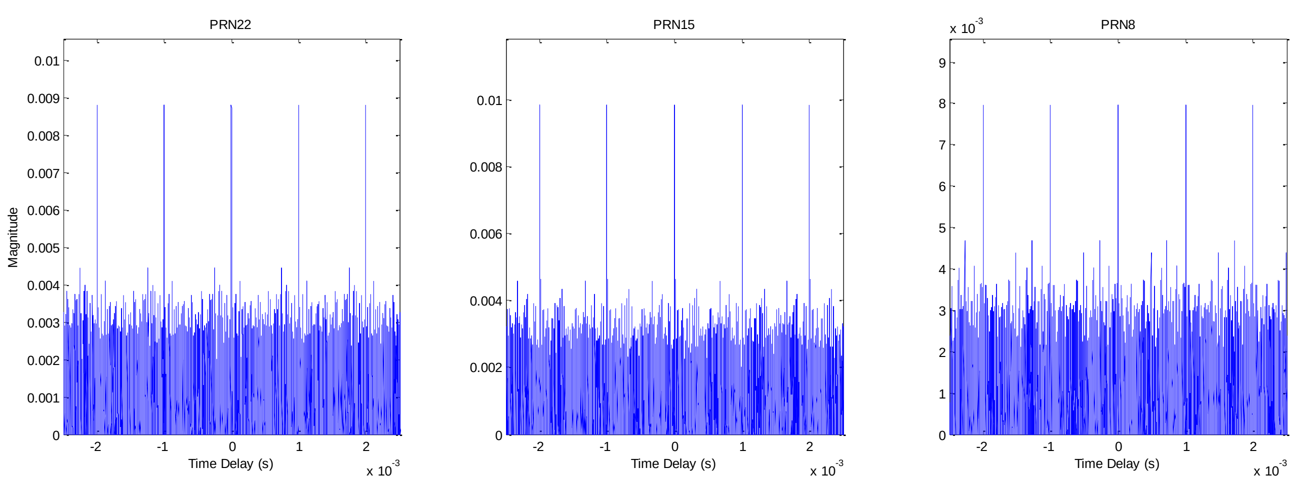


Figure 10: Satellite Code-correlation with 4 Pseudolites Transmitting and Pseudolite Blanking (Compare to Figure 7)

In the same situation as in Figure 7, unidentifiable satellite signals become very clear through pseudolite blanking.

Conclusion

While pseudolites can bring many benefits to GPS users, it must be ensured that nearby GPS receivers not intending to use them do not experience interference. It was shown that with only one pseudolite transmitting, receivers can easily track satellite signals as if the pseudolite were not present. However the only case considered is when pseudolite signals saturate the receiver. If pseudolites are placed in remote locations (as most uses would require anyways), this is not a large concern, as receivers beyond a reasonable distance would not experience signals of significant power. Further investigation should be done into pseudolite placement to allow sufficient GPS augmentation without interruption of signal to other receivers nearby.

Pseudolite blanking provides a promising technique to allow GPS receivers to operate in environments with multiple pseudolites. However, it does require hardware upgrades to receivers. Research into cost effectiveness for specific uses should be conducted to test feasibility.

REFERENCES

Cobb, H. S., “GPS Pseudolites: Theory, Design, and Applications,” Ph.D. Thesis, Stanford University, September 1997.

Abt, T. L., Soualle, F., Martin, S., “Optimal Pulsing Schemes for Galileo Pseudolite Signals,” *Journal of Global Positioning Systems*, 2007, Vol. 6, No. 2: p166-141.

Anyaegbu, E., et al, “An Integrated Pulsed Interference Mitigation for GNSS Receivers,” The Journal of Navigation, 2008, Vol. 61, No. 2: p239-255.

Braasch, M., Van Dierendonck, A. J., “GPS Receiver Architectures and Measurements,” *Proceedings of the IEEE*, January 1999, Vol. 87, No. 1.

Jovancevic, A., et al, “Piercing the Veil,” *GPS World*, March 2007, Vol. 18 Issue 3, p30-37.
< http://www.gpsworld.com/defense/navigation/piercing-veil-3155?page_id=1>

Zhang, Lei, “Simulation on C/A Codes and Analysis of GPS/Pseudolite Signals Acquisition,” *Science in China Series E: Technological Sciences*, Science in China Press, May 2009, Vol. 52, No. 5: p.1459-1462.

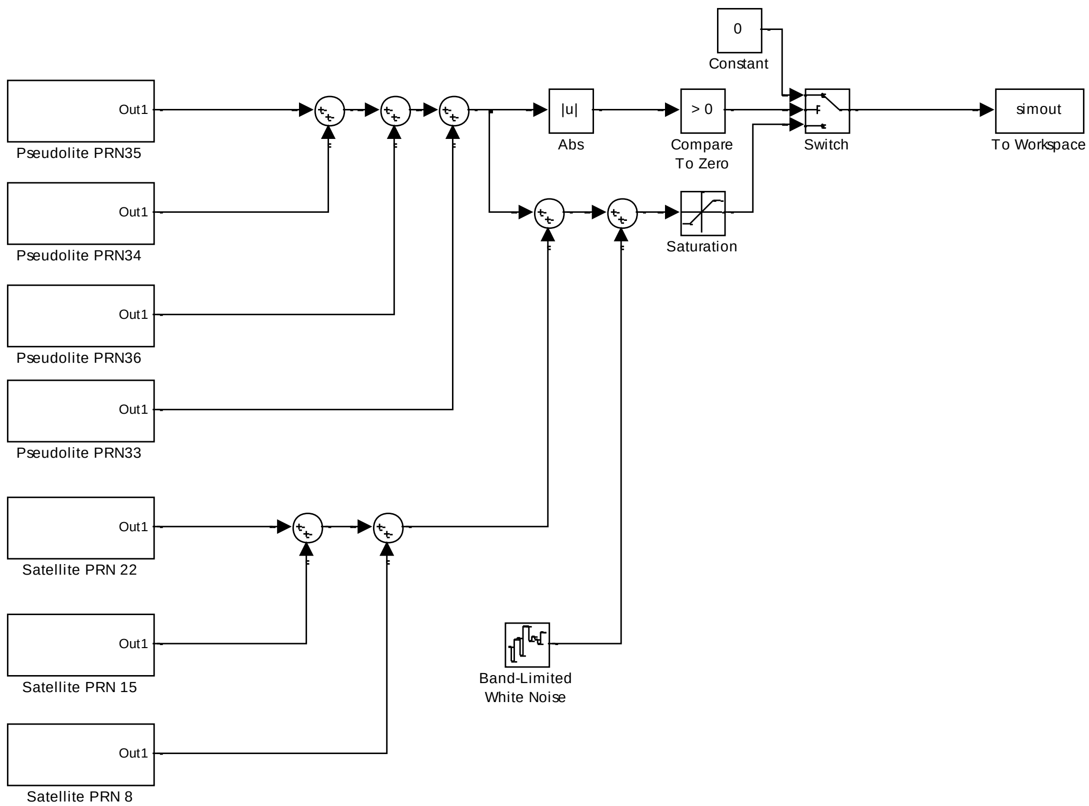
Borre, K., et al, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*. Birkhauser, Boston, MA, 2007.

Misra, P., Enge, P., *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, Lincoln, MA, 2006, 2nd Edition.

APPENDIX

Simulink Models

Simulation Model with Pseudolite Blanking Applied



MatLab Code

PRN Code Generator function:

```
function [PRN, time, Ts] = PRNcodeGen(n, numberCycles, samplesPerChip)

%[PRN Ts] = PRNcodeGen(n,numberCycles,samplesPerChip)
%
% GPS PRN C/A Code Generator
% Outputs code for PRN 'n' and its sampling rate, Ts (in seconds)
% With given number of cycles of code and number of samples per chip
%
% PRN is generated from two 10-stage ML LFSR Gold Codes (G1 & G2)
%
% Charles Tytler
% AAE 575



%CONSTANT VARIABLES
chipRate = 1.023e6;  %cps
cycle = 2^(10)-1;    %1023 chips (for 10 stage)
period = cycle/chipRate;

%Table of C/A Code Tap Selection as per IS-GPS-200
taps = [2 6;
        3 7;
        4 8;
        5 9;
        1 9;
        2 10;
        1 8;
        2 9;
        3 10;
        2 3;
        3 4;
        5 6;
        6 7;
        7 8;
        8 9;
        9 10;
        1 4;
        2 5;
        3 6;
        4 7;
        5 8;
        6 9;
        1 3;
        4 6;
        5 7;
        6 8;
        7 9;
        8 10;
        1 6;
        2 7;
```

```
        3 8;
        4 9
        5 10
        4 10
        1 7
        2 8
        4 10];

%INPUTS
%Identify taps for PRN definition
tap1 = taps(n,1);
tap2 = taps(n,2);
%Set Sample Rate according to specified number of samples per chip
Ts = 1/(samplesPerChip*chipRate); %sec
%m-file will produce a specified number of cycles of the PRN codes
Tmax = numberCycles*period;

%CODE GENERATION
codeIndex = 1;
PRN = zeros(1,round(Tmax/Ts)); %PRN code initialized
time = zeros(1,round(Tmax/Ts)); %time array initialized
%Set Gold Code Initial Conditions
G1 = ones(1,10);
G2 = ones(1,10);

%Generate individual chips for the calculated length, Tmax
for t=0:Ts:Tmax-Ts
    time(codeIndex) = t;

    %Compute delayed G2 per IS-GPS-200
    G2i = xor(G2(tap1),G2(tap2));

    %Calculate current index value of PRN codes using G1 and G2i
    PRN(codeIndex) = xor(G1(10),G2i);

    %Only modify gold codes when time reaches next chip
    if(mod(codeIndex,samplesPerChip) == 0)
        %Generate Gold Code inputs as per definitions
        G1_input = xor(G1(3),G1(10));
        G2_input = mod(G2(2)+G2(3)+G2(6)+G2(8)+G2(9)+G2(10),2);

        for i=9:-1:1        %Shift Gold Code chips down 1 slot
            G1(i+1) = G1(i);
            G2(i+1) = G2(i);
        end

        G1(1) = G1_input;   %Insert calculated Gold Code input to slot 1
        G2(1) = G2_input;
    end

    codeIndex = codeIndex + 1;  %Advance to next index value for PRN codes
end

return
```

Pseudolite Pulsing function:

```
function signal = pulseSignal(PRN, Ts, t, initialPulse)

% signal = pulseSignal(PRN, Ts, t)
%
% Returns a pulsed signal of duration 't' seconds
% for given PRN code with sampling rate Ts

%Initialize Constants
Tc = 1.0e-3; %sec  Code Period
Tp = Tc/11; %sec    Pulse Duration

signal = zeros(1,floor(t/Ts));

cycleIndex = 1;
pulsePos = initialPulse;
pulse = 0;
i=1;
PRNindex = 1;

for time = 0:Ts:t-Ts

    if(time>=(cycleIndex*.001))
        cycleIndex = cycleIndex + 1;
        PRNindex = 1;
        pulsePos = mod(pulsePos+1,10);
    end

    if(time>=(pulsePos*Tp + (cycleIndex-1)*.001))
        pulse = 1;
    end

    if(pulsePos == 9)
        if(time>=(10*Tp + (cycleIndex-1)*.001))
            pulse = 1;
        end
        if(time> (11*Tp + (cycleIndex-1)*.001))
            pulse = 0;
        end
    end

    if(time> ((pulsePos+1)*Tp + (cycleIndex-1)*.001))
        pulse = 0;
    end

    signal(i) = pulse*PRN(PRNindex);

    PRNindex = PRNindex + 1;
    i = i + 1;
end
```

```
return;
```

Correlation function:

```
function [R, lag] = circcorr(x, y, Ts);
%
% Circular correlation of two vectors, x and y, through FFT methods.
%
% Output is R(lag), where R is the cross correlation of x and y and
%  Lag is in the same dimesnions as Ts.
%
% Prof. Jim Garrison, Purdue University, AAE575 Fall 2009
%
npts = size(x,1);
X = fft(x);
Y = fft(y);
FTXY = X.*conj(Y);
R = fftshift(ifft(FTXY))/npts;
lag = [-floor(npts/2):floor((npts-1)/2)]*Ts;

return
```