Design of Anti-GPS for Reasons of Security

Hazem M. El-Bakry

Faculty of Computer Science & Information Systems,
Mansoura University, EGYPT
E-mail: helbakry20@yahoo.com

Nikos Mastorakis

Technical University of Sofia, BULGARIA

Abstract – In this paper, the design considerations of Global Positioning System (GPS) is presented. Simulation issues related to implementing a GPS jamming signal in a laboratory test environment are introduced. These issues pertain to jamming accuracy requirements. Furthermore, important design parameters that may affect jamming system performance are discussed. In addition, an example of the Navigation Laboratory jamming system is given. It addresses fabrication issues, data requirements, error handling, local and remote operations, and how to attain high accuracy and repeatability during the generation and measurement of jamming. Moreover, a new approach for fast detection of GPS signal is presented. The entire data are collected together in a long vector and then tested as a one input pattern. Proposed fast time delay neural networks (FTDNNs) use cross correlation in the frequency domain between the tested data and the input weights of neural networks. It is proved mathematically and practically that the number of computation steps required for the presented time delay neural networks is less than that needed by conventional time delay neural networks (CTDNNs). Simulation results using MATLAB confirm the theoretical computations.

Keywords— Global Positioning System, Active/Passive tracking devices, Jamming System, Fast Neural Networks, Cross Correlation, Frequency Domain.

I. Introduction

In recent years, GPS has become a major application in military and civilian devices. GPS is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense. GPS was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS. Figure 1 shows GPS applications. This device helps people in many aspects, but some of them misuse GPS and use this device to kill, spy and thief others [7,49,54]. In this paper, the advantages and disadvantages of using GPS are discussed. How to prevent bad people from misuse of GPS, and how to generate a jamming signal that will block GPS receiver is described.

GPS has a variety of applications on land, at sea and in the air. Basically, GPS is usable everywhere except where it's impossible to receive the signal such as inside most buildings, in caves and other subterranean locations, and underwater. The most common airborne applications are for navigation by general aviation and commercial aircraft. At sea, GPS is also typically used for navigation by recreational boaters, commercial fishermen, and professional mariners. Land-based applications are more diverse. The scientific community uses GPS for its precision timing capability and position information. Surveyors use GPS for an increasing portion of their work. GPS offers cost savings by drastically reducing setup time at the survey site and providing incredible accuracy. Basic survey units, costing thousands of dollars, can offer accuracies down to one meter. More expensive systems are available that can provide accuracies to within a centimeter [6,54]. Recreational uses of GPS are almost as varied as the number of recreational sports available. GPS is popular among hikers, hunters, snowmobiles, mountain bikers, and cross-country skiers, just to name a few. Anyone who needs to keep track of where he or she is, to find his or her way to a specified location, or know what direction and how fast he or she is going can utilize the benefits of the global positioning system [13]. GPS is now common place in automobiles as well. Some basic systems are in place and provide emergency roadside assistance at the push of a button (by transmitting your current position to a dispatch center). More sophisticated systems that show your position on a street map are also available. Currently these systems allow a driver to keep track of where he or she is and suggest the best route to follow to reach a designated location.

"Kill, spy and thief" is an easy word that people could say today, by using cheap and small devices and tools to achieve what they want. GPS was originally intended for military applications, to confuse the enemy on where their exact location is or where the enemies GPS guided missiles. Also GPS allows just about anyone to track anyone else and know where and when, this technology is auto tracker. Finally GPS help thieves, what's happening lately is that the fascists and criminals are finding it much cheaper and less man-hour intensive to simply place a GPS receiver with digital recorder in victim's vehicles, then the record of where the victims traveled can be retrieved either by removal of the device, or by remote retrieval. This is possible with the device a criminal planted in the Colorado couple's vehicle. So GPS Tracking is an important field in GPS application and the types of auto tracking system will be discussed.

II. GPS Tracking Technology

A GPS tracking unit is a device that uses the Global Positioning System to determine the precise location of a vehicle, person, or other asset to which it is attached and to record the position of the asset at regular intervals. The recorded location data can be stored within the tracking unit, or it may be transmitted to a central location data base, or internet-connected computer, using a cellular (GPRS), radio, or satellite modem embedded in the unit. This allows the asset's location to be displayed against a map backdrop either in real-time or when analyzing the track later, using customized software. Such systems are not new; amateur radio operators have been operating their free GPS based nationwide real time Automatic Position Reporting System since 1982.

A) TYPES OF GPS TRACKING DEVICES

The types of GPS tracking devices technology available to the public:

1- logger tracking device (Passive)
Passive tracking is used to tell where a vehicle has been over a certain period of time, Figure 2 shows passive tracking life cycle. Passive tracking devices is so difficult to detect because they use different technologies to report the GPS locations, and most of them don't report constantly-they may only send the location every 20 minutes.
Those devices can be as small as a matchbook and can be hidden just about anywhere. They are attached to a vehicle or individual and after a specific amount of time they must be retrieved. After retrieval the device is usually attached to a personal computer and the information is downloaded into a database that is provided by the device manufacture. Probably the best defense against passive/logger tracking devices comes in the form of a GPS blocker. These small units normally plug into your automobiles cigarette lighter port and provide protection for about 30 feet in any direction. Small handheld units are also available that run on batteries that also provide short range blocking. These units can range in price from $200 to $850.Best bet would be to if one of RF detectors/scanners is used under "Counter Surveillance" category. These will detect the transmissions of the GPS unit, but only if the unit is transmitting at the time you are scanning. So, if the unit just sent the location, and then you scan it, you will not detect anything. The unit must be transmitting at the same time you are scanning the vehicle.

2-Real time tracking devices (active)
Active tracking is intended to show where a vehicle is now, Figure 3 show real time tracking life cycle.
Real type of tracking device is relatively easy to detect with a combination cell phone/GPS blocking device or RF detectors/scanners. These units range in cost from $300 to more than $1000. Of course either a cell phone or a GPS blocking device may accomplish the same affect.
So the best solution to block GPS receiver devices is generating jamming signal.

B) JAMMING AND BLOCKING GPS SIGNAL

Jamming and blocking is process of generating noise signal that concatenating and jamming with GPS signal and generate new signal that receiver can't understand and translate this new signal and this receivers loss the signal [3,5,49].

C) GPS SPOOFING

GPS Spoofing is process to feed the receiver false information so that it computes an erroneous time or location. This technique is so difficult because it is lay to GPS receivers [50].
How GPS Spoofing work?
1. Feed the receiver false information so that it computes an erroneous time or location.
2. The device thinks it is in Place A when it’s really in Place B.
3. The simulator produces fake satellite radio signals that are stronger than the real signals coming from outer space. Most current GPS receivers are totally fooled, happily accepting these stronger signals while ignoring the weaker, authentic signals.
4. Current GPS receivers are relatively stupid. They eagerly accept fake GPS satellite signals that are thousands of times stronger than any real satellite could possibly produce [50-53].

D) GPS JAMMING TECHNIQUES AND CHARACTERISTICS

Most jamming techniques fall into three major types usually based on bandwidth. Continuous wave, or CW jamming, is usually defined as occupying less than 100 kHz of bandwidth. CW jamming will be defined as one frequency only. Narrowband (NB) jamming will be defined as any unwanted signal occupying more than one MHz of bandwidth but less than or equal to the entire ±1.023 MHz bandwidth of C/A code. NB is usually centered about L1 or L2 but not necessarily so. Wideband (WB) jamming will be defined as jamming signals occupying the entire ±10.23 MHz bandwidth about L1 or L2 [3].

Characteristics common to all types of jamming are as follows:

1. Pulsed NB, WB, and CW—Each of the previously mentioned jamming types can be pulsed at a maximum pulse repetition frequency (PRF) of 20 kHz. The minimum pulse repetition frequency (PRF) is 10 Hz. Duty cycle (DC) can range from 10 to 90%.Large values for both PRF and DC will make the jamming look continuous. very small values for both will not affect the GPS receiver noticeably [3,5].

2. Jamming levels—Variable from 20 to 80 dB J/S with an accuracy of ±0.5 dB in 0.5 dB increments of precision. Low values of J/S will have little effect on receiver performance. very high values provide little useful information because the GPS receiver has long since lost lock. The minimum value of 20 dB J/S was chosen because C/A acquisition at 24 dB J/S is a common military requirement. The maximum value was chosen because no GPS receivers can track at 80 dB J/S against a WB jammer without employing beam steering, nulling, or some other multi- element antenna technique [3].

3. Frequency offset—Allowable frequency offset of ±9 MHz in 1 kHz increments for CW and NB jamming types. Only wideband noise may not be offset in frequency. We chose to limit the frequency offset to ±9.0 MHz to avoid generating jamming outside the band, and to ensure that all the jamming energy enters the GPS receiver under test [3].

III. GPS Jamming Signal

Anti-GPS is a device that prevents the GPS loggers, trackers and GPS/GSM devices to get positions from the Satellites. In order to generate those radio jamming signals, the structure of GPS signal must be known.

A) GPS SIGNAL

To design a simulation software-defined signal GPS receive it is necessary to know the characteristics of the signal and data transmitted from the GPS satellites and received by the GPS receiver antenna.

The GPS signals are transmitted on two radio frequencies in the Ultra High Frequency (UHF) band. The UHF band covers the frequency band from 500MHz to 3GHz. These frequencies are referred to as L1 and L2 and are derived from a common frequency, L0 = 10.23MHz

L1 = 154*f0 = 575.42MHz,

L2 = 120*f0 = 1227.60MHz.

It’s Navigation data have a bit rate of 50 bps, The navigation data contain information regarding satellite orbits. Signals are modulated onto the carrier signal using the binary phase shift keying (BPSK), The GPS signal use two pseudorandom number (PRN) code, the first one is the Coarse / Acquisition (C/A) code and the other one is the Precision code (P(Y)) code, C/A code is only modulated onto the L1 carrier while the P(Y) code is modulated onto both the L1 and the L2 carrier. Figure 4 show GPS signal structure and how to generate L1 signal and L2 signal.

The C/A code is a 1,023 bit long pseudorandom number (PRN) which, when transmitted at 1.023 megabits per second (Mbit/s), repeats every millisecond. Pseudorandom numbers only match up, or strongly correlate, when they are exactly aligned. Each satellite transmits a unique PRN code, which does not correlate well with any other satellite's PRN code. In other words, the PRN codes are highly orthogonal to one another. The P-code is also a PRN, however each satellite's P-code PRN code is 6.1871 × 1012 bits long (6,187,100,000,000 bits) and only repeats once a week (it is transmitted at 10.23 Mbit/s). Figure 5 show GPS signal structure blocks diagrams.

It follows that the signal transmitted from satellite k can be described as:

$$s^k(t) = \sqrt{2P_C}\left(C^k(t) \oplus D^k(t)\right)\cos(2\pi f_{L1}t) + \sqrt{2P_{PL1}}\left(P^k(t) \oplus D^k(t)\right)\sin(2\pi f_{L1}t) + \sqrt{2P_{PL2}}\left(P^k(t) \oplus D^k(t)\right)\sin(2\pi f_{L2}t),$$

where PC, Ppl1,and Ppl2 are the powers of signals with C/A or P code, Ck is the C/A code sequence assigned to satellite number k, Pk is the P(Y) code sequence assigned to satellite number k, Dk is the navigation data sequence, and fL1 and fL2 are the carrier frequencies of L1 and L2, respectively.

B) DEFINITIONS

Radio jamming signals have the following characteristics:

1. The jamming signals have same frequency.
2. The jamming signals have the same type of modulation.
3. The jamming signals have enough power to override any signal at receiver.

Most jamming techniques fall into three major types usually based on bandwidth. Continuous wave, or CW jamming, is usually defined as occupying less than 100 kHz of bandwidth. In this paper, CW jamming will be defined as one frequency only. Narrowband (NB) jamming will be defined as any unwanted signal occupying more than one MHz of bandwidth but less than or equal to the entire ±1.023 MHz bandwidth of C/A code. NB is usually centered about L1 or L2 but not necessarily so. Wideband (WB) jamming will be defined as jamming signals occupying the entire ±10.23 MHz bandwidth about L1 or L2. All discussions of jamming signal ratio (J/S) will be related to dBm. J/S ratios are with respect to L1 or L2 P(Y) code only, where L1 = -133 dBm, L2 = -136 dBm. Three dB are added for C/A-code J/S comparisons.

IV. Jamming Design Issues

A) ACCURACY

When measuring the GPS jamming signals, it is also important to note that the complete 20.46 MHz of signal bandwidth for L1 or L2 P(Y) code should be measured to ensure that all the jamming energy is accounted for when performing the J/S calculations. GPS simulator signals must also have correct signal power levels. Adjusting the power output to correct for the testing system's losses or gains usually attains this. Incorrect GPS simulator output levels will cause J/S to be artificially high or low, nullifying test results, even if the GPS jamming levels are correct.

It is needed to be able to determine J/S as accurately as possible. Any error in determining this parameter can have a large impact on system performance. For example, suppose it is required to determine how a +1.0 dB error in measuring the amplitude of jamming would affect the overall accuracy of our J/S calculation. Assume there is no amplitude error in S, the GPS signal level, and that it is a constant value. Given

$$1 \text{ dB} = 10 \text{ Log } JM / JT$$

Where

JM = Measured jamming power

JT = True jamming power

Then

JT= 1.26 JM or a positive 26% error in measurement

For a -1.0 dB error in measuring jamming amplitude, we have

$$-1 \text{ dB} = 10 \text{ Log } JM / JT$$

$$\text{or} \quad JT = 0.79 \text{ JM}$$

This corresponds to a negative 21% error in our measurement.

B) FREQUENCY

There are other important jamming parameters besides amplitude tolerances. Frequency is also an important issue. CW jammer or NB jammer center frequency location relative to the GPS signal is important, especially if testing a GPS receiver that can notch-out CW and NB jammers in the frequency domain. Too much drift from the commanded center frequency of the signal generators could nullify test results. We chose to limit the frequency offset to ±9.0 MHz to avoid generating jamming outside the band, and to ensure that all the jamming energy enters the GPS receiver under test.

C) PULSE

For pulse jamming (turning CW, NB, and WB jamming on and off at some rate), care should be taken to limit the two pulse description parameters, pulse repetition frequency (PRF) and duty cycle (DC). Large values for both PRF and DC will make the jamming look continuous, and very small values for both will not affect the GPS receiver noticeably. Realistic values found through experimentation are:

Minimum PRF: 1 Hz, Maximum PRF: 20 kHz

Minimum DC: 10%, Maximum DC: 90%

These values should only be considered as a starting point and are tailor able for specific requirements.

D) MODULATION

The overriding goal of any GPS jamming modulation/mixing scheme is to completely fill a given bandwidth of frequency with energy that will cause the GPS receiver to lose lock or never attain lock. There are many types of modulation options available. Some standard modulation types are amplitude modulation (AM), frequency modulation (FM), and biphase shift keying (BPSK). Mixing of noise with a carrier frequency to produce WB jamming is another common practice. Other options are as follows: sweeping the center frequency, summing two different kinds of modulation together (AM and FM, for instance), RF summation of multiple signal generator jamming signals, and others. The possibilities are almost endless.

E) J/S RANGE

Another design parameter not already addressed is the absolute limits placed upon the J/S values of the GPS jamming system. Low values of J/S will have little effect on receiver performance, and very high values provide little useful information because the GPS receiver has lon since lost lock. Values chosen for the jamming system located inside the Navigation Laboratory were 7 to 80 db J/S in 0.50 dB increments of precision. The minimum value of 7 dB J/S was chosen because C/A acquisition a 24 dB J/S is a common military requirement. The maximum value was chosen because no GPS receivers can track at 80 dB J/S against a WB jammer without employing beam steering, nulling, or some other multi-element antenna technique.

F) JAMMING SYSTEM SPECIFICATIONS

The system offers the following GPS jamming types:

• CW—Successive oscillations that are identical under steady-state conditions.

• NB—Generated from a pseudorandom Gaussian distributed noise sequence. A 2 MHz bandwidth contained within a 20.46 MHz band usually centered about the L1 or L2 frequency.
• WB—Generated from a pseudorandom Gaussian distributed noise sequence. A 20.46 MHz bandwidth centered on the L1 or L2 frequency.

The different jamming signals used were:
* Non-Coherent Continuous Wave (NCW) Frequency: 1450~1600 MHz.
* Coherent CW (CCW) Frequency: 1450~1600 MHz.
* Amplitude Modulation (AM): Carrier frequency: 1450~1600 MHz, Modulation waveform: Sine, Modulation frequency: 1 kHz, Modulation depth: 50.0 percent.
* Frequency Modulation (FM): Carrier frequency: 1450~1600 MHz ,Modulation waveform: Sine, Modulation frequency: 1 kHz, Frequency deviation: [+ or -]50 kHz.
* Band-limited White Noise (WB): Center frequency: 1450~1600 MHz Bandwidth: 20 MHz.

Common characteristics to all types of jamming are:
• Pulsed NB, WB, and CW. Each of the previously mentioned jamming types can be pulsed at a maximum pulse repetition frequency of 20 kHz. The minimum PRF is 10 Hz. Duty cycle can range from 10 to 90%.
• Jamming levels—Variable from 7 to 80 dB J/S with an accuracy of ±0.5 dB in 0.5 dB increments of precision.
• Frequency offset—Allowable frequency offset of ±9 MHz in 1 kHz increments for CW and NB jamming types. Only wideband noise may not be offset in frequency.

G) SIMULATION OF JAMMING SIGNAL

There are many simulation tools to simulate jamming signal but the best tool is using matlab simulink, matlab V7.7.0 (R2008b) and simulink are used. Figure 6 show simulink library browser. First of all I simulate Band-Limited White Noise, The Band-Limited White Noise block generates normally distributed random numbers that are suitable for use in continuous or hybrid systems. The primary difference between this block and the Random Number block is that the Band-Limited White Noise block produces output at a specific sample rate, which is related to the correlation time of the noise.

Theoretically, continuous white noise has a correlation time of 0, a flat power spectral density (PSD), and a covariance of infinity. In practice, physical systems are never disturbed by white noise, although white noise is a useful theoretical approximation when the noise disturbance has a correlation time that is very small relative to the natural bandwidth of the system.

In Simulink software, you can simulate the effect of white noise by using a random sequence with a correlation time much smaller than the shortest time constant of the system. The Band-Limited White Noise block produces such a sequence. The correlation time of the noise is the sample rate of the block. For accurate simulations, use a correlation time much smaller than the fastest dynamics of the system. You can get good results by specifying:

$$t_c \approx \frac{1}{100}\frac{2\pi}{f_{max}}$$

Where fmax is the bandwidth of the system in rad/sec.

To produce the correct intensity of this noise, the covariance of the noise is scaled to reflect the implicit conversion from a continuous PSD to a discrete noise covariance. The appropriate scale factor is 1/tc, where tc is the correlation time of the noise. This scaling ensures that the response of a continuous system to the approximate white noise has the same covariance as the system would have to true white noise. Because of this scaling, the covariance of the signal from the Band-Limited White Noise block is not the same as the Noise power (intensity) dialog box parameter. This parameter is actually the height of the PSD of the white noise. While the covariance of true white noise is infinite, the approximation used in this block has the property that the covariance of the block output is the Noise Power divided by tc.

Figure 7 show Band-Limited White Noise block diagram, a model using Band-Limited White Noise block and Scope block is made. Band-Limited White Noise block have three parameters noise power and sample time and seed. Figure 8 show scope diagram result.

Second frequency modulation passband is simulated. The FM Modulator passband block modulates using frequency modulation. The output is a passband representation of the modulated signal. The output signal's frequency varies with the input signal's amplitude. Both the input and output signals are real sample-based scalar signals.

If the input is u(t) as a function of time t, then the output is:

$$\cos\left(2\pi f_c t + 2\pi K_c \int_0^t u(\tau)d\tau + \theta\right)$$

where:

fc is the Carrier frequency parameter.

θ is the Initial phase parameter.
Kc is the Modulation constant parameter.
Typically, an appropriate Carrier frequency value is much higher than the highest frequency of the input signal. By the Nyquist sampling theorem, the reciprocal of the model's sample time (defined by the model's signal source) must exceed twice the Carrier frequency parameter.
To use those blocks go to simulink library browser. Figure 9 show FM modulation Block diagram, we make a model using sine wave block and FM modulation passband block and Scope block to see result. FM modulation passband block have parameters carrier frequency and initial phase and frequency deviation. Figure 10 show scope diagram result.
Third The Multiband-Limited White Noise is simulated. This block models multiband noise using four different band limited white noise blocks, each one with its own power and time scale. A first order lowpass filter with a selectable cut off frequency is applied to the last three noises (have look under the mask).The output sampling time has to be multiple of the first sample time. Figure 12 shows Multiband-Limited White Noise Block Diagram and Figure 11 show Multiband-Limited White Noise subsystem Block Diagram, we make a model using Band-Limited White Noise block and Scope block to see result. Band-Limited White Noise block have three parameters noise power and sample time and seed. Then Figure 13 show scope diagram result.

V. ANTI-GPS in a Laboratory Environment

In this section, implementation of anti-GPS in lab is explained. You must understand what are techniques and their Characteristics to help you understand these test equipments. Each piece of test equipment is described next, including how that equipment functions in the jamming system

1. Personal Computer:
Computer contains a GPIB interface card and Cables, an IRIG B timing card, a SCSI card, an A/D & D/A card, a video card, and an Ethernet card. 100MB of random access memory (RAM) is also included to prevent any virtual memory swapping to disk during operations. Of course, it would be better if we have more advanced devices.

2. Arbitrary Waveform Generator:
A piece of electronic test equipment used to generate electrical waveforms, generate Gaussian distributed pseudorandom noise sequences consisting of 10,000 data points clocked at a 2 MHz rate. The arbitrary waveform generator can clock through the modulating sequences at a maximum rate of 250 MHz. The values can lie anywhere between ±1.0 VDC. We need AWG2021 this is the best one for Anti-GPS application with 2 channels (2ch) and the frequency from 2-250 MHZ, FM modulation.

3. RF Signal Generator:
Output a pure sine wave only that can be offset ±9 MHz from L1 or L2.

4. Spectrum Analyzer:
Measure all generated jamming power (in dBm) across the 20.46 MHz frequency spectrum surrounding L1 or L2.

5. RF Switch Driver:
Routes signals during BIT and is not used for normal operations. Controlled by the JamCtrl, it routes signals during BIT and is not used for normal operations.

6. GPS Timing Receiver:
Provides UTC to PC. PC uses UTC to timestamp saved data.

7. GPS Device:
Any Device receives and sends GPS signal L1&L2.

As mentioned above, Power received at L1 from each GPS satellite is -133 dBm and Power received at L1 from each GPS satellite is -136. So in order to generate jamming signal we must generate jamming signal at range 1450-1600 and frequency offset is +- 9 MHZ and PRF between 1HZ and 20 KHZ and DC is between 10% and 90% and J/S is between 7 dB and 80 dB. Finally, the best jamming modulation signals are Frequency Modulation and Band-limited White Noise. The problem is how to create enough power to override any signal at receiver and what is the range of jamming. Results are shown in Tables 1 and 2. Band-limited White Noise Jamming requires 13-15 dB higher JSR values for loss of lock than FM jamming. Specifications of GPS Jammer is shown in table 3.

VI. Fast Detection of GPS Signals by using High Speed Neural Networks

Finding GPS signal, in the incoming data, is a searching problem. First neural networks are trained to classify the GPS signal from other signals and this is done in time domain. In information detection phase, each position in the incoming matrix is tested for presence or absence of the GPS signal. At each position in the input one dimensional matrix, each sub-matrix is multiplied by a window of weights, which has the same size as the sub-matrix. The outputs of neurons in the hidden layer are multiplied by the weights of the output layer. When the final output is high, this means that the sub-matrix under test contains the GPS signal and vice versa. Thus, we may conclude that this searching problem is a cross correlation between the incoming data and the weights of neurons in the hidden layer.

The convolution theorem in mathematical analysis says that a convolution of f with h is identical to the result of the following steps: let F and H be the results of the Fourier Transformation of f and h in the frequency domain. Multiply F and H* in the frequency domain point by point and then transform this product into the spatial domain via the inverse Fourier Transform. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain, speed up in an order of magnitude can be achieved during the detection process [14-47].

Assume that the size of the GPS signal is 1xn. In detection phase of GPS signal, a sub matrix I of size 1xn (sliding window) is extracted from the tested matrix, which has a size of 1xN. Such sub matrix, which may be an intrusion code, is fed to the neural network. Let W_i be the matrix of weights between the input sub-matrix and the hidden layer. This vector has a size of 1xn and can be represented as 1xn matrix. The output of hidden neurons h(i) can be calculated as follows:

$$h_i = g\left(\sum_{k=1}^{n} W_i(k)I(k) + b_i\right) \qquad (1)$$

where g is the activation function and b(i) is the bias of each hidden neuron (i). Equation 1 represents the output of each hidden neuron for a particular sub-matrix I. It can be obtained to the whole input matrix Z as follows:

$$h_i(u) = g\left(\sum_{k=-n/2}^{n/2} W_i(k)\ Z(u+k) + b_i\right) \qquad (2)$$

Eq.2 represents a cross correlation operation. Given any two functions f and d, their cross correlation can be obtained by:

$$d(x) \otimes f(x) = \left(\sum_{n=-\infty}^{\infty} f(x+n)d(n)\right) \qquad (3)$$

Therefore, Eq. 2 may be written as follows [14-47]:

$$h_i = g\left(W_i \otimes Z + b_i\right) \qquad (4)$$

where h_i is the output of the hidden neuron (i) and h_i (u) is the activity of the hidden unit (i) when the sliding window is located at position (u) and (u) $\in$ [N-n+1].

Now, the above cross correlation can be expressed in terms of one dimensional Fast Fourier Transform as follows [14]:

$$W_i \otimes Z = F^{-1}\left(F(Z) \bullet F^*\left(W_i\right)\right) \qquad (5)$$

Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to conventional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u) = g\left(\sum_{i=1}^{q} W_o(i)\, h_i(u) + b_o\right) \qquad (6)$$

where q is the number of neurons in the hidden layer. O(u) is the output of the neural network when the sliding window located at the position (u) in the input matrix Z. W_o is the weight matrix between hidden and output layer.

The complexity of cross correlation in the frequency domain can be analyzed as follows:

1- For a tested matrix of 1xN elements, the 1D-FFT requires a number equal to $N\log_2 N$ of complex computation steps [15]. Also, the same number of complex computation steps is required for computing the 1D-FFT of the weight matrix at each neuron in the hidden layer.

2- At each neuron in the hidden layer, the inverse 1D-FFT is computed. Therefore, q backward and (1+q) forward transforms have to be computed. Therefore, for a given matrix under test, the total number of operations required to compute the 1D-FFT is $(2q+1)N\log_2 N$.

3- The number of computation steps required by FTDNNs is complex and must be converted into a real version. It is known that, the one dimensional Fast Fourier Transform requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions [48]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. Therefore, the total number of computation steps required to obtain the 1D-FFT of a 1xN matrix is:

$$\rho=6((N/2)\log_2 N) + 2(N\log_2 N) \qquad (7)$$

which may be simplified to:

$$\rho=5N\log_2 N \qquad (8)$$

4- Both the input and the weight matrices should be dot multiplied in the frequency domain. Thus, a number of complex computation steps equal to qN should be considered. This means 6qN real operations will be added to the number of computation steps required by FTDNNs.

5- In order to perform cross correlation in the frequency domain, the weight matrix must be extended to have the same size as the input matrix. So, a number of zeros = (N-n) must be added to the weight matrix. This requires a total real number of computation steps = q(N-n) for all neurons. Moreover, after computing the FFT for the weight matrix, the conjugate of this matrix must be obtained. As a result, a real number of computation steps = qN should be added in order to obtain the conjugate of the weight matrix for all neurons. Also, a number of real computation steps equal to N is required to create butterflies complex numbers ($e^{-jk(2\Pi n/N)}$), where 0<K<L. These (N/2) complex numbers are multiplied by the elements of the input matrix or by previous complex numbers during the computation of FFT. To create a complex number requires two real floating point operations. Thus, the total number of computation steps required for FTDNNs becomes:

$$\sigma=(2q+1)(5N\log_2 N)+6qN+q(N-n)+qN+N \qquad (9)$$

which can be reformulated as:

$$\sigma=(2q+1)(5N\log_2 N)+q(8N-n)+N \qquad (10)$$

6- Using sliding window of size 1xn for the same matrix of 1xN pixels, q(2n-1)(N-n+1) computation steps are required when using CTDNNs for detection GPS unit or processing (n) input data. The theoretical speed up factor η can be evaluated as follows:

$$\eta=\frac{q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \qquad (11)$$

Time delay neural networks accept serial input data with fixed size (n). Therefore, the number of input neurons equals to (n). Instead of treating (n) inputs, the proposed new approach is to collect all the incoming data together in a long vector (for example 100xn). Then the input data is tested by time delay neural networks as a single pattern with length L (L=100xn). Such a test is performed in the frequency domain as described before. The combined information in the incoming data may have real or complex values in a form of one or two dimensional array. Complex-valued neural networks have many applications in fields dealing with complex numbers such as telecommunications, speech recognition and image processing with the Fourier Transform []. Complex-valued neural networks mean that the inputs, weights, thresholds and the activation function have complex values. In this section, formulas for the speed up ratio with different types of inputs (real /complex) will be presented. Also, the speed up ratio in case of a one and two dimensional incoming input matrix will be concluded. The operation of FTDNNs depends on computing the Fast Fourier Transform for both the input and weight matrices and obtaining the resulting two matrices. After performing dot multiplication for the resulting two matrices in the frequency domain, the Inverse Fast Fourier Transform is determined for the final matrix. Here, there is an excellent advantage with FTDNNs that should be mentioned. The Fast Fourier Transform is already dealing with complex numbers, so there is no change in the number of computation steps required for FTDNNs. Therefore, the speed up ratio in case of complex-valued time delay neural networks can be evaluated as follows:

1) In case of real inputs

A) For a one dimensional input matrix

Multiplication of (n) complex-valued weights by (n) real inputs requires (2n) real operations. This produces (n) real numbers and (n) imaginary numbers. The addition of these numbers requires (2n-2) real operations. The multiplication and addition operations are repeated (N-n+1) for all possible sub matrices in the incoming input matrix. In addition, all of these procedures are repeated at each neuron in the hidden layer. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(2n-1)(N-n+1) \qquad (12)$$

The speed up ratio in this case can be computed as follows:

$$\eta=\frac{2q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \qquad (13)$$

The theoretical speed up ratio for searching short successive (n) code in a long input vector (L) using complex-valued time delay neural networks is shown in Tables 4, 5, and 6. Also, the practical speed up ratio for manipulating matrices of different sizes (L) and different sized weight matrices (n) using a 2.7 GHz processor and MATLAB is shown in Table 7.

B) For a two dimensional input matrix

Multiplication of (n^2) complex-valued weights by (n^2) real inputs requires ($2n^2$) real operations. This produces (n^2) real numbers and (n^2) imaginary numbers. The addition of these numbers requires ($2n^2$-2) real operations. The multiplication and addition operations are repeated $(N-n+1)^2$ for all possible sub matrices in the incoming input matrix. In addition, all of these procedures are repeated at each neuron in the hidden layer. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(2n^2-1)(N-n+1)^2 \qquad (14)$$

The speed up ratio in this case can be computed as follows:

$$\eta=\frac{2q(2n^2-1)(N-n+1)^2}{(2q+1)(5N^2\log_2 N^2)+q(8N^2-n^2)+N} \qquad (15)$$

The theoretical speed up ratio for detecting (nxn) real valued submatrix in a large real valued matrix (NxN) using complex-valued time delay neural networks is shown in Tables 8, 9, 10. Also, the practical speed up ratio for manipulating matrices of different sizes (NxN) and different sized code matrices (n) using a 2.7 GHz processor and MATLAB is shown in Table 11.

2) In case of complex inputs

A) For a one dimensional input matrix

Multiplication of (n) complex-valued weights by (n) complex inputs requires (6n) real operations. This produces (n) real numbers and (n) imaginary numbers. The addition of these numbers requires (2n-2) real operations. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(4n-1)(N-n+1) \tag{16}$$

The speed up ratio in this case can be computed as follows:

$$\eta = \frac{2q(4n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \tag{17}$$

The theoretical speed up ratio for searching short complex successive (n) code in a long complex-valued input vector (L) using complex-valued time delay neural networks is shown in Tables 12, 13, and 14. Also, the practical speed up ratio for manipulating matrices of different sizes (L) and different sized weight matrices (n) using a 2.7 GHz processor and MATLAB is shown in Table 15.

B) For a two dimensional input matrix

Multiplication of (n^2) complex-valued weights by (n^2) real inputs requires ($6n^2$) real operations. This produces (n^2) real numbers and (n^2) imaginary numbers. The addition of these numbers requires ($2n^2$-2) real operations. Therefore, the number of computation steps required by conventional neural networks can be calculated as:

$$\theta=2q(4n^2-1)(N-n+1)^2 \tag{18}$$

The speed up ratio in this case can be computed as follows:

$$\eta = \frac{2q(4n^2-1)(N-n+1)^2}{(2q+1)(5N^2\log_2 N^2)+q(8N^2-n^2)+N} \tag{19}$$

The theoretical speed up ratio for detecting (nxn) complex-valued submatrix in a large complex-valued matrix (NxN) using complex-valued neural networks is shown in Tables 16, 17, and 18. Also, the practical speed up ratio for manipulating matrices of different sizes (NxN) and different sized code matrices (n) using a 2.7 GHz processor and MATLAB is shown in Table 19.

An interesting point is that the memory capacity is reduced when using FTDNN. This is because the number of variables is reduced compared with CTDNN.

It should be noted that most GPS jammers are illegal to build or use in many countries or localities due to the potential for misuse. For instance, a GPS jammer can confuse aircraft and other vehicle instruments, possibly causing mishaps. Some GPS jammers state that they are only for civilian GPS jamming only; however some military equipment, must first sign onto the civilian GPS radio frequency in order to gain access to the military GPS frequency.

VII. Conclusion

The major issues and problems associated with generating GPS jamming signal has been presented. Furthermore, Simulation of jamming signal using Matlab has been introduced. Moreover, a brief description about the design of GPS jamming signal suitable for laboratory use has been given. Furthermore, a fast neural algorithm for detecting GSP Signals has been presented. Theoretical computations have shown that FTDNNs require fewer computation steps than conventional ones. This has been achieved by applying cross correlation in the frequency domain between the input data and the input weights of time delay neural networks. Simulation results have confirmed this proof by using MATLAB. The proposed high speed neural networks can be applied to fast detect any signal successfully.

References

[1] K. Borre, D. M. Akos , N. Bertelsen , P. Rinder , S. H. Jensen, “A Software-Defined GPS and Galileo Receiver”. Birkh¨ auser Boston, ISBN: 978-0-8176-4390-4, NOV 11, 2006.

[2] B. Iyidir, Y. Ozkazanc, “Jamming of GPS receivers”, Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE 12th, ISBN. 0-7803-8318-4, April 28-30, 2004.

[3] Lt. Trond, B. Olsen, B. Forssell, ” Jamming GPS”, GPS World, Jan. 1, 2006.

[4] GPS World Conference & Expo, Rosemont, Illinois, June 11-13, 2007.

[5] G. D. Rash, “GPS Jamming in A Laboratory Environment,” Naval Air Warfare Center Weapons Division (NAWCWPNS)/China Lake, Jun 17, 2004.

[6] The Asian GPS Conference, India International Centre, New Delhi, India, 24-30 October 2002.

[7] A. El-Rabbany, "Introduction to GPS: The Global Positioning System," Second Edition,. Norwood, Aug 31, 2006.

[8] 2006 IEEE AESS Tutorial on GPS, Von Braun Center Huntsville, AL, Conference #: 12469, 16 Nov 2006,

[9] S. W. Hinch, "Outdoor Navigation with GPS," Annadel Press, April 20, 2007.

[10] K. Michael, A. McNamee, MG. Michael,” The Emerging Ethics of Humancentric GPS Tracking and Monitoring” IEEE Computer Society Washington, DC, USA, ISBN.0-7695-2595-4,PP.34,2006 .

[11] J. H. Doty, “Revolution in GPS”, GPS World, Sep 1 2004.

[12] J. A. Lee,V. M. Thomas,” GPS and radio tracking of end-of-life products [recycling and waste disposal applications]”, IEEE Computer Society Washington, DC, USA, ISBN.0-7803-8250-1,PP. 309-312, 2004.

[13] Rong-Jyue, Fang, Ken-ISu, Hsin-Chang Lu, Cheng-Chung Wang,Chin-Chih Lin ," Application of global positioning system (GPS) in earth sciences teaching", World Scientific and

Engineering Academy and Society (WSEAS), vol.6, pp. 267-271, January 2007.

[14] H. M. El-Bakry, " Fast Virus Detection by using High Speed Time Delay Neural Networks ," Accepted for publication in journal of computer virology.

[15] H. M. El-Bakry, " A New Neural Design for Faster Pattern Detection Using Cross Correlation and Matrix Decomposition," Accepted for publication in Neural World journal.

[16] H. M. El-Bakry, " New Fast Principal Component Analysis For Real-Time Face Detection," Accepted for publication in MG&V Journal.

[17] H. M. El-Bakry, " Fast Record Detection in Large Databases Using New High Speed Time Delay Neural Networks," Accepted for publication in International Joint Conference on Neural Networks, June 14-19, 2009, Atlanta, USA.

[18] H. M. El-Bakry, "New Faster Normalized Neural Networks for Sub-Matrix Detection using Cross Correlation in the Frequency Domain and Matrix Decomposition," Applied Soft Computing journal, vol. 8, issue 2, March 2008, pp. 1131-1149.

[19] H. M. El-Bakry, "A Novel Modified Hopfield Neural Network for Perfect Calculation of Magnetic Resonance Spectroscopy," International Journal on Biomedical Science, vol. 3, no. 1, 2008, pp. 63-75.

[20] H. M. El-Bakry, and N. Mastorakis, "A New Technique for Detecting Dental Diseases by using High Speed Neuro-Computers," WSEAS Trans. on Signal Processing, vol. 8, issue 2, October 2008, pp. 1131-1149.

[21] H. M. El-Bakry, and N. Mastorakis, "A Real-Time Intrusion Detection Algorithm for Network Security," WSEAS Transactions on Communications, Issue 12, vol. 7, December 2008, pp. 1222-1234.

[22] H. M. El-Bakry, and N. Mastorakis, "A New Fast Forecasting Technique using High Speed Neural Networks ," WSEAS Transactions on Signal Processing, Issue 10, vol. 4, October 2008, pp. 573-595.

[23] H. M. El-Bakry, and N. Mastorakis, "A Perfect QoS Routing Algorithm for Finding the Best Path for Dynamic Networks ," WSEAS Transactions on Communications, Issue 12, vol. 7, December 2008, pp. 1123-1136.

[24] H. M. El-Bakry, and N. Mastorakis, "A Novel Fast Kolmogorov's Spline Complex Network for Pattern Detection," WSEAS Transactions on Systems, Issue 11, vol. 7, November 2008, pp. 1310-1328.

[25] H. M. El-Bakry, and N. Mastorakis, " A Modified Hopfield Neural Network for Perfect Calculation of Magnetic Resonance Spectroscopy," WSEAS Transactions on Information Science and Applications, issue 12, vol. 5, December 2008, pp. 1654-1666.

[26] H. M. El-Bakry, "New Fast Principal Component Analysis for Face Detection," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol.11, no.2, 2007, pp. 195-201.

[27] H. M. El-Bakry and M. Hamada, " New Fast Decision Tree Classifier for Identifying Protein Coding Regions," Lecture Notes in Computer Science, Springer, ISICA 2008, LNCS 5370, 2008, pp. 489-500.

[28] H. M. El-Bakry and M. Hamada, "A New Implementation for High Speed Neural Networks in Frequency Space," Lecture Notes in Artificial Intelligence, Springer, KES 2008, Part I, LNAI 5177, pp. 33-40.

[29] H. M. El-Bakry, and N. Mastorakis "New Fast Normalized Neural Networks for Pattern Detection," Image and Vision Computing Journal, vol. 25, issue 11, 2007, pp. 1767-1784.

[30] H. M. El-Bakry and N. Mastorakis, "Fast Code Detection Using High Speed Time Delay Neural Networks," Lecture Notes in Computer Science, Springer, vol. 4493, Part III, May 2007, pp. 764-773.

[31] H. M. El-Bakry, "Fast Painting with Different Colors Using Cross Correlation in the Frequency Domain," International Journal of Computer Science, vol.1, no.2, 2006, pp. 145-156.

[32] H. M. El-Bakry, "New High Speed Normalized Neural Networks for Fast Pattern Discovery on Web Pages," International Journal of Computer Science and Network Security, vol.6, No. 2A, February 2006, pp.142-152.

[33] H. M. El-Bakry, and Q. Zhao, "Fast Normalized Neural Processors For Pattern Detection Based on Cross Correlation Implemented in the Frequency Domain," Journal of Research and Practice in Information Technology, Vol. 38, No.2, May 2006, pp. 151-170.

[34] H. M. El-Bakry, and Q. Zhao, "Fast Time Delay Neural Networks," International Journal of Neural Systems, vol. 15, no.6, December 2005, pp.445-455.

[35] H. M. El-Bakry, and Q. Zhao, "Speeding-up Normalized Neural Networks For Face/Object Detection," Machine Graphics & Vision Journal (MG&V), vol. 14, No.1, 2005, pp. 29-59.

[36] H. M. El-Bakry, "A New High Speed Neural Model For Character Recognition Using Cross Correlation and Matrix Decomposition," International Journal of Signal Processing, vol.2, no.3, 2005, pp. 183-202.

[37] H. M. El-Bakry, and Q. Zhao, "A Fast Neural Algorithm for Serial Code Detection in a Stream of Sequential Data," International Journal of Information Technology, vol.2, no.1, pp. 71-90, 2005.

[38] H. M. El-Bakry, and Q. Zhao, "Fast Complex Valued Time Delay Neural Networks," International Journal of Computational Intelligence, vol.2, no.1, pp. 16-26, 2005.

[39] H. M. El-Bakry, and Q. Zhao, "Fast Pattern Detection Using Neural Networks Realized in Frequency Domain," Enformatika Transactions on Engineering, Computing, and Technology, February 25-27, 2005, pp. 89-92.

[40] H. M. El-Bakry, and Q. Zhao, "Sub-Image Detection Using Fast Neural Processors and Image Decomposition," Enformatika Transactions on Engineering, Computing, and Technology, February 25-27, 2005, pp. 85-88.

[41] H. M. El-Bakry, and Q. Zhao, "Face Detection Using Fast Neural Processors and Image Decomposition," International Journal of

Computational Intelligence, vol.1, no.4, 2004, pp. 313-316.
[42] H. M. El-Bakry, and Q. Zhao, "A Modified Cross Correlation in the Frequency Domain for Fast Pattern Detection Using Neural Networks," International Journal on Signal Processing, vol.1, no.3, 2004, pp. 188-194.
[43] H. M. El-Bakry, and Q. Zhao, "Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform," International Journal on Signal Processing, vol.1, no.3, 2004, pp. 182-187.
[44] H. M. El-Bakry, and H. Stoyan**,** “FNNs for Code Detection in Sequential Data Using Neural Networks for Communication Applications, ” Proc. of the First International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2004, 21-25 July, 2004. Orlando, Florida, USA, Vol. IV, pp. 150-153.
[45] H. M. El-Bakry, “Face detection using fast neural networks and image decomposition,” Neurocomputing Journal, vol. 48, 2002, pp. 1039-1046.
[46] H. M. El-Bakry, “Human Iris Detection Using Fast Cooperative Modular Neural Nets and Image Decomposition,” Machine Graphics & Vision Journal (MG&V), vol. 11, no. 4, 2002, pp. 498-512.
[47] H. M. El-Bakry, “Automatic Human Face Recognition Using Modular Neural Networks,” Machine Graphics & Vision Journal (MG&V), vol. 10, no. 1, 2001, pp. 47-73.
[48] J. W. Cooley, and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. 19, 1965, pp. 297–301.
[49] J. Ruley," GPS Jamming", Avweb, February 12, 2003, http://www.avweb.com/news/avionics /182754-1.html
[50] J. S. Warner, R. G. Johnston," GPS Spoofing Countermeasures", National Laboratory, vol. 10, December 2003, pp. 22-30.
[51] J. Kesel," How To Detect GPS Tracking Units", GPS Auto Trackers,2006, http://ezinearticles.com/?How-To-Detect-GPS-Tracking-Units&id=894204
[52] J. Kesel," GPS Auto Tracker", ACM Press New York, NY, USA, VOL 12 , no. 30 , January 2007, pp. 31 – 39.
[53] M. R. Mosavi," Think GPS Cargo Tracking", IOS Press Amsterdam, VOL 17, no. 21, April 2006, pp. 159-171.
[54] K. Chiang, Y. Huang "An intelligent navigator for seamless INS/GPS integrated land vehicle navigation applications" Elsevier Science Publishers B. V., vol. 8, no.1, January 2008, pp. 722-733.

Figure 1.GPS Applications.

Figure 2. Passive Tracking Life Cycle.

Figure 3.Real time tracking life cycle.

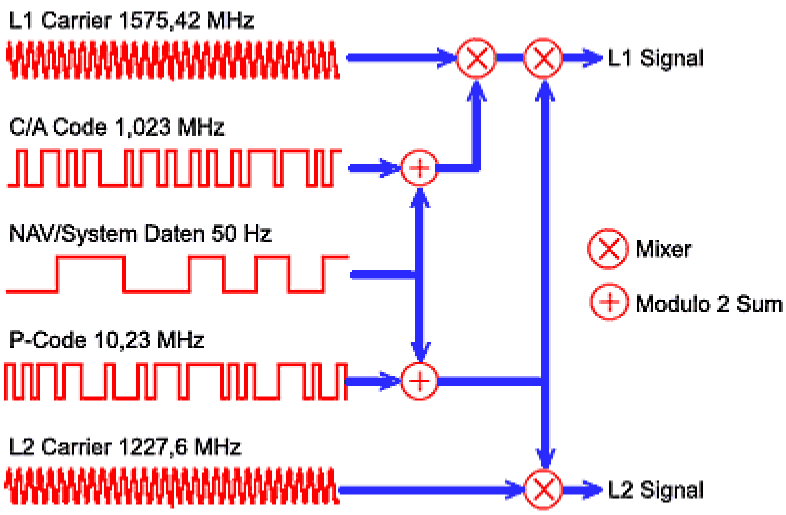

Figure 4.GPS Signal Structure.

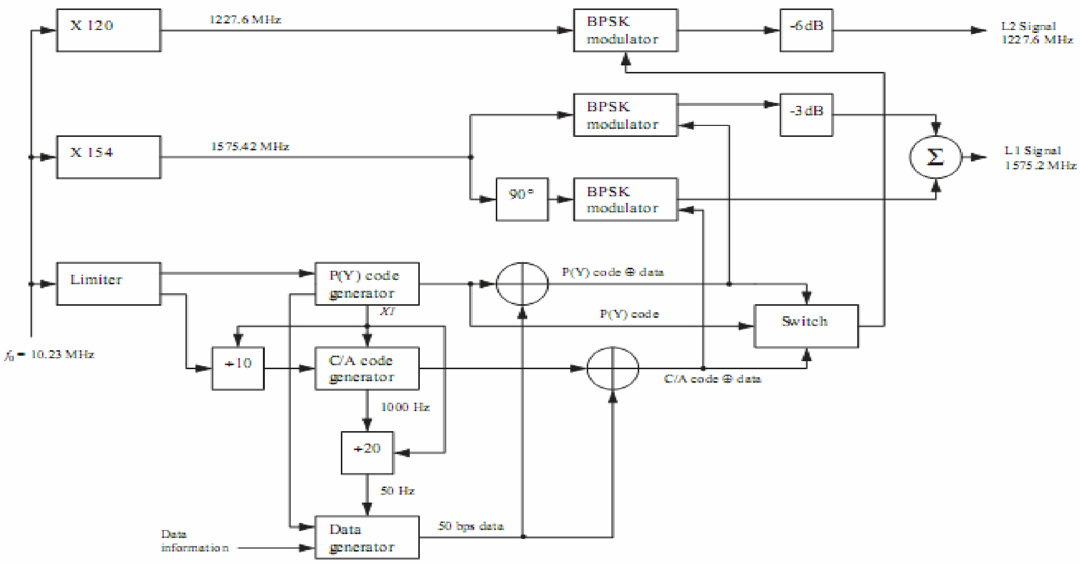

Figure 5.GPS Signal Structure block diagram.

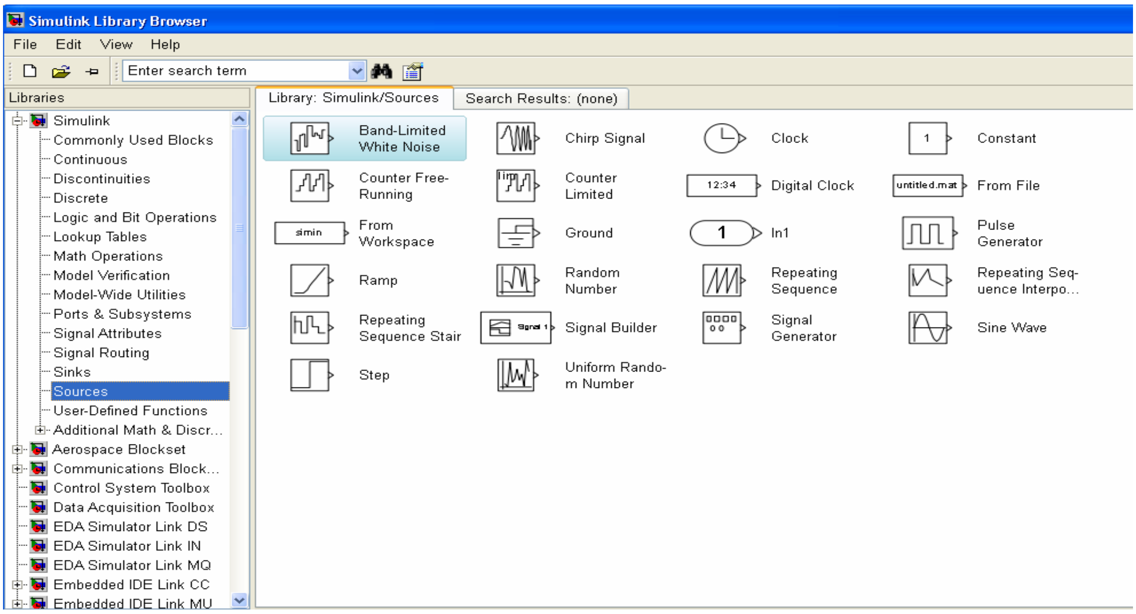

Figure 6 .Band-Limited white noise in simulink

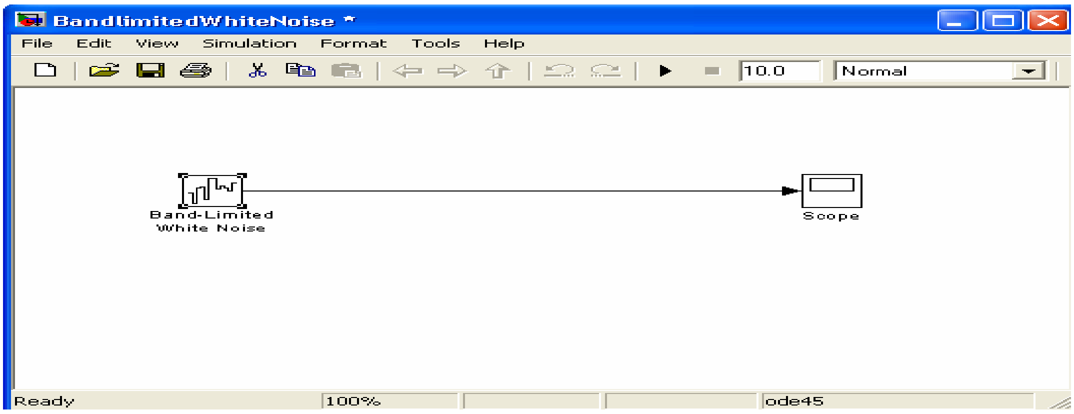

Figure 7. Band-Limited white noise Block Diagram.

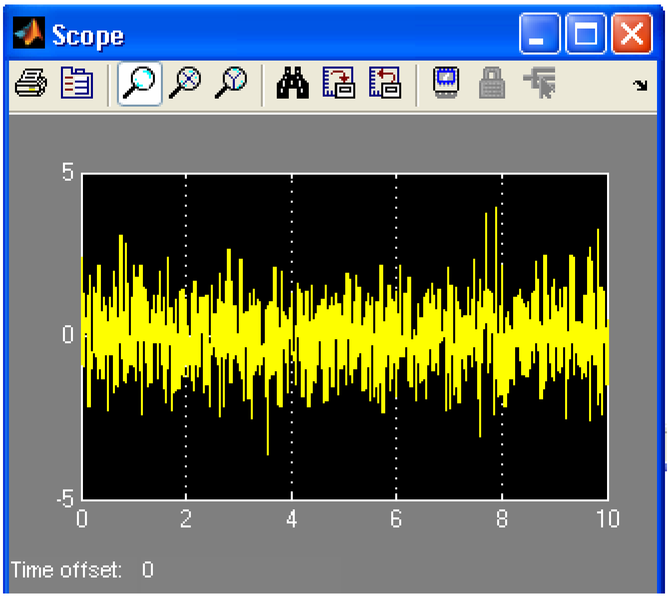

Figure 8.Band-Limited white noise Scope.

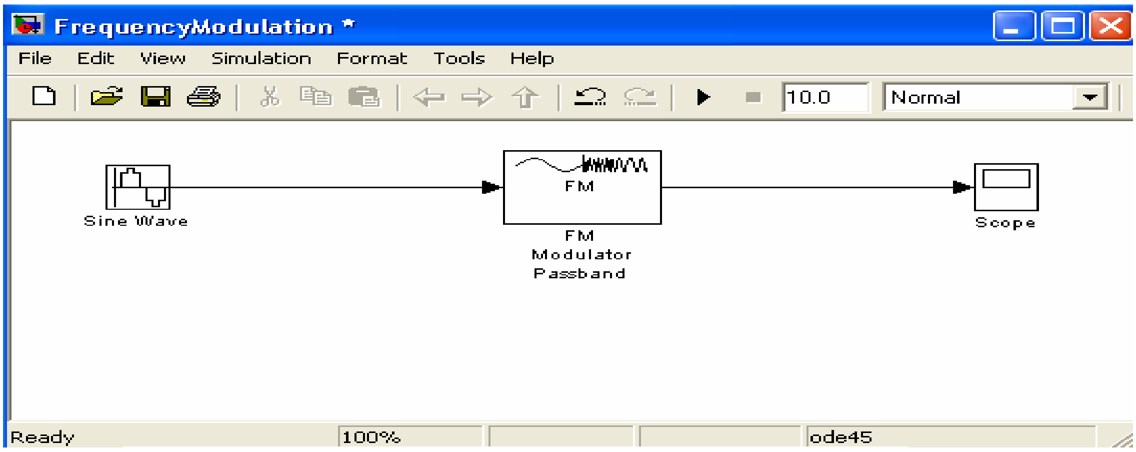

Figure 9.FM Modulation Block Diagram.

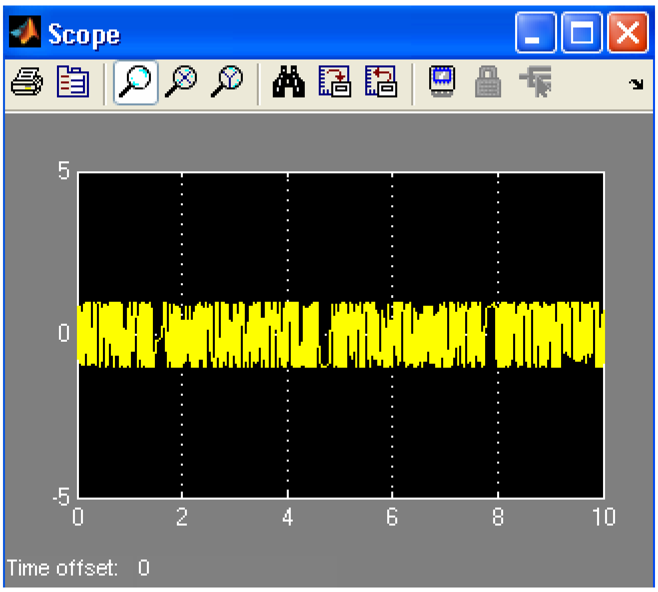

Figure 10.FM Modulation Scope.

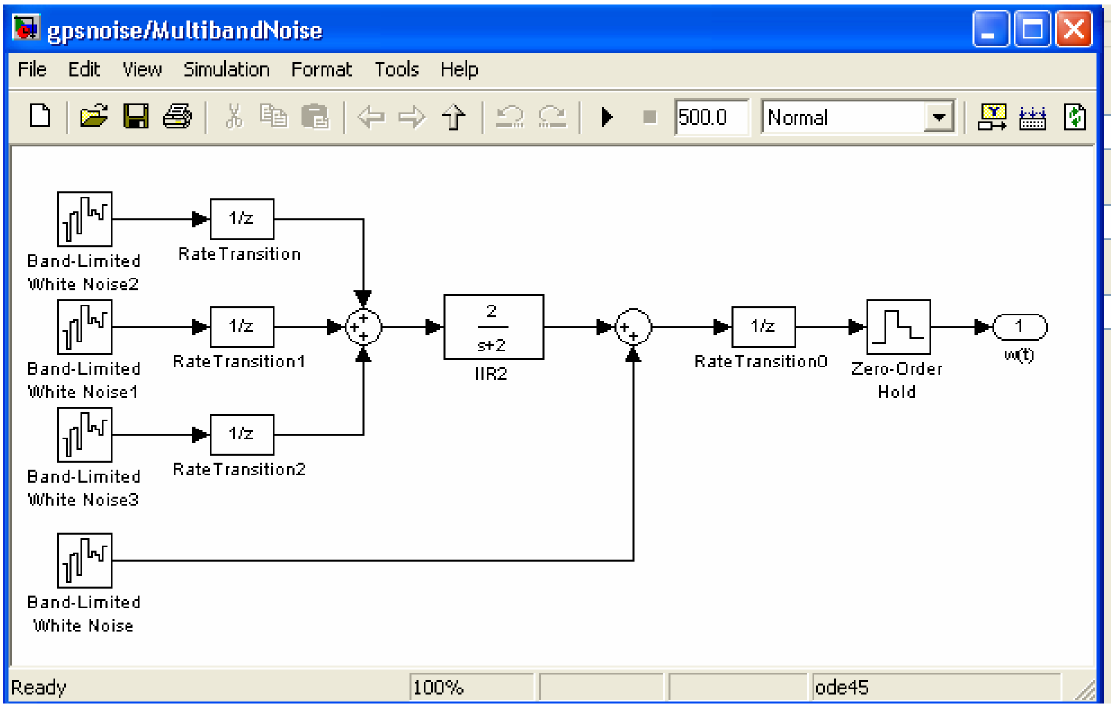

Figure 11. Multiband-Limited White Noise subsystem Block Diagram.

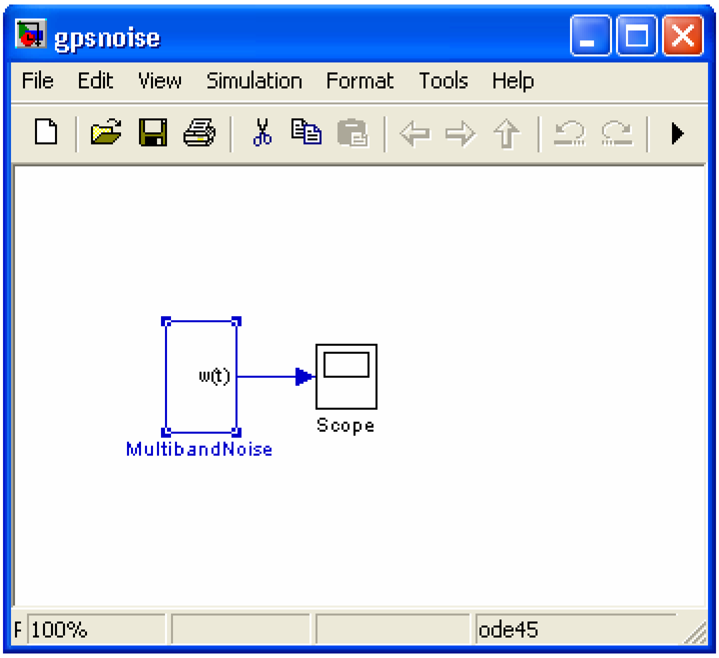

Figure 12. Multiband-Limited White Noise Block Diagram.

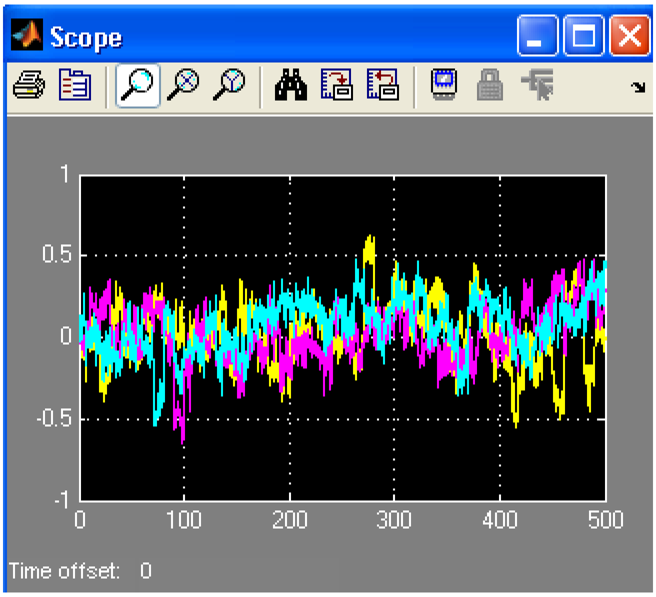

Figure 13. Multiband-Limited White Noise.

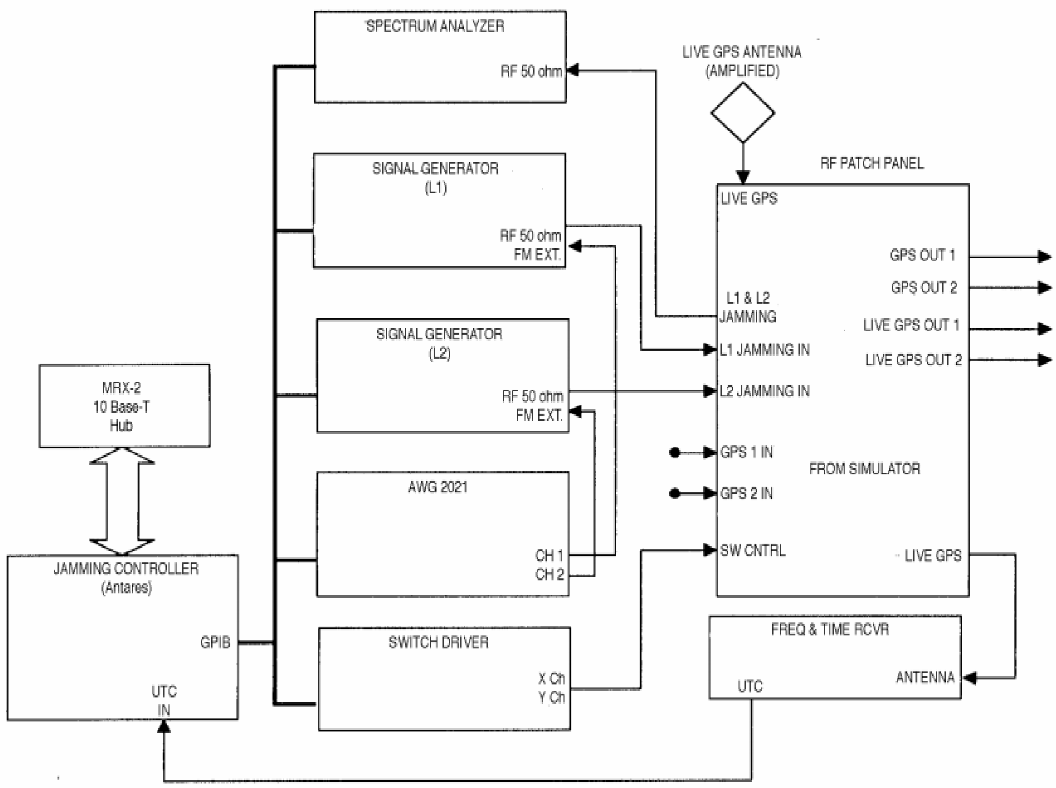

Figure 14. Anti-GPS in A Laboratory Environment.

Table 1: Using FM Jamming GPS (OEM) receiver.

Parameter	JSR (dB)
Beginning to lose track	35.5
Loss of 3D navigation	36
Complete loss of navigation	36
Regaining 2D navigation	17.5
Regaining 3D navigation	16

Table 2: Using Band-limited White Noise Jamming GPS (OEM) receiver.

Parameter	JSR (dB)
Beginning to lose track	53
Loss of 3D navigation	53
Complete loss of navigation	53.5
Regaining 2D navigation	48
Regaining 3D navigation	47.5

Table 3: Specifications of GPS Jammer.

Frequency	1450~1600MHz
Jamming Range	Average 5 meters radius
Output Power	35.5 - 36 dB using FM modulation 53 - 53.5 dB using Band-limited White Noise

Table 4: The theoretical speed up ratio for time delay neural networks (1D-real values input matrix, n=400).

Length of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
10000	4.6027e+008	4.2926e+007	10.7226
40000	1.8985e+009	1.9614e+008	9.6793
90000	4.2955e+009	4.7344e+008	9.0729
160000	7.6513e+009	8.8219e+008	8.6731
250000	1.1966e+010	1.4275e+009	8.3823
360000	1.7239e+010	2.1134e+009	8.1571
490000	2.3471e+010	2.9430e+009	7.9752
640000	3.0662e+010	3.9192e+009	7.8237

Table 5: The theoretical speed up ratio for time delay neural networks (1D-real values input matrix, n=625).

Length of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
10000	7.0263e+008	4.2919e+007	16.3713
40000	2.9508e+009	1.9613e+008	15.0452
90000	6.6978e+009	4.7343e+008	14.1474
160000	1.1944e+010	8.8218e+008	13.5388
250000	1.8688e+010	1.4275e+009	13.0915
360000	2.6932e+010	2.1134e+009	12.7433
490000	3.6674e+010	2.9430e+009	12.4612
640000	4.7915e+010	3.9192e+009	12.2257

Table 6: The theoretical speed up ratio for time delay neural networks (1D-real values input matrix, n=900).

Length of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
10000	9.823 e+008	4.2911e+007	22.8933
40000	4.2206e+009	1.9612e+008	21.5200
90000	9.6176e+009	4.7343e+008	20.3149
160000	1.7173e+010	8.8217e+008	19.4671
250000	2.6888e+010	1.4275e+009	18.8356
360000	3.8761e+010	2.1134e+009	18.3409
490000	5.2794e+010	2.9430e+009	17.9385
640000	6.8985e+010	3.9192e+009	17.6018

Table 7: Practical speed up ratio for time delay neural networks (1D-real values input matrix).

Length of input matrix	Speed up ratio (n=400)	Speed up ratio (n=625)	Speed up ratio (n=900)
10000	17.88	25.94	35.21
40000	17.19	25.11	34.43
90000	16.65	24.56	33.59
160000	16.14	24.14	33.05
250000	15.89	23.76	32.60
360000	15.58	23.23	32.27
490000	15.28	22.87	31.99
640000	14.08	22.54	31.78

Table 8: The theoretical speed up ratio for time delay neural networks (2D-real values input matrix, n=20).

Size of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	3.1453e+008	4.2916e+007	7.3291
200x200	1.5706e+009	1.9610e+008	8.0091
300x300	3.7854e+009	4.7335e+008	7.9970
400x400	6.9590e+009	8.8203e+008	7.8898
500x500	1.1091e+010	1.4273e+009	7.7711
600x600	1.6183e+010	2.1130e+009	7.6585
700x700	2.2233e+010	2.9426e+009	7.5556
800x800	2.9242e+010	3.9186e+009	7.4623

Table 9: The theoretical speed up ratio for time delay neural networks (2D-real values input matrix, n=25).

Size of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	4.3285e+008	4.2909e+007	10.0877
200x200	2.3213e+009	1.9609e+008	11.8380
300x300	5.7086e+009	4.7334e+008	12.0602
400x400	1.0595e+010	8.8202e+008	12.0119
500x500	1.6980e+010	1.4273e+009	11.8966
600x600	2.4863e+010	2.1130e+009	11.7667
700x700	3.4246e+010	2.9425e+009	11.6381
800x800	4.5127e+010	3.9185e+009	11.5163

Table 10: The theoretical speed up ratio for time delay neural networks (2D-real values input matrix, n=30).

Size of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	5.4413e+008	4.2901e+007	12.6834
200x200	3.1563e+009	1.9608e+008	16.0966
300x300	7.9272e+009	4.7334e+008	16.7476
400x400	1.4857e+010	8.8201e+008	16.8444
500x500	2.3946e+010	1.4273e+009	16.7773
600x600	3.5193e+010	2.1130e+009	16.6552
700x700	4.8599e+010	2.9425e+009	16.5160
800x800	6.4164e+010	3.9185e+009	16.3745

Table 11: Practical speed up ratio for time delay neural networks (2D-real values input matrix).

Size of input matrix	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	17.19	22.32	31.74
200x200	17.61	22.89	32.55
300x300	16.54	23.66	33.71
400x400	15.98	22.95	34.53
500x500	15.62	22.49	33.32
600x600	15.16	22.07	32.58
700x700	14.87	21.83	32.16
800x800	14.64	21.61	31.77

Table 12: The theoretical speed up ratio for time delay neural networks (1D-complex values input matrix, n=400).

Length of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	9.2111e+008	4.2926e+007	21.4586
200x200	3.7993e+009	1.9614e+008	19.3706
300x300	8.5963e+009	4.7344e+008	18.1571
400x400	1.5312e+010	8.8219e+008	17.3570
500x500	2.3947e+010	1.4275e+009	16.7750
600x600	3.4500e+010	2.1134e+009	16.3245
700x700	4.6972e+010	2.9430e+009	15.9604
800x800	3.9192e+009	6.1363e+010	15.6571

Table 13: The theoretical speed up ratio for time delay neural networks (1D-complex values input matrix, n=625).

Length of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	1.4058e+009	4.2919e+007	32.7558
200x200	5.9040e+009	1.9613e+008	30.1025
300x300	1.3401e+010	4.7343e+008	28.3061
400x400	2.3897e+010	8.8218e+008	27.0883
500x500	3.7391e+010	1.4275e+009	26.1934
600x600	5.3885e+010	2.1134e+009	25.4969
700x700	7.3377e+010	2.9430e+009	24.9324
800x800	9.5868e+010	3.9192e+009	24.4612

Table 14: The theoretical speed up ratio for time delay neural networks (1D-complex values input matrix, n=900).

Length of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	1.9653e+009	4.2911e+007	45.7993
200x200	8.4435e+009	1.9612e+008	43.0519
300x300	1.9240e+010	4.7343e+008	40.6410
400x400	3.4356e+010	8.8217e+008	38.9450
500x500	5.3791e+010	1.4275e+009	37.6817
600x600	7.7544e+010	2.1134e+009	36.6920
700x700	1.0562e+011	2.9430e+009	35.8870
800x800	1.3801e+011	3.9192e+009	35.2134

Table 15: Practical speed up ratio for time delay neural networks (1D-complex values input matrix).

Length of input matrix	Speed up ratio (n=400)	Speed up ratio (n=625)	Speed up ratio (n=900)
10000	37.90	53.58	70.71
40000	36.82	52.89	69.43
90000	36.34	52.47	68.69
160000	35.94	51.88	68.05
250000	35.69	51.36	67.56
360000	35.28	51.02	67.15
490000	34.97	50.78	66.86
640000	34.67	50.56	66.58

Table 16: The theoretical speed up ratio for time delay neural networks (2D-complex values input matrix, n=20).

Size of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	6.2946e+008	4.2916e+007	14.6674
200x200	3.1431e+009	1.9610e+008	16.0281
300x300	7.5755e+009	4.7335e+008	16.0040
400x400	1.3927e+010	8.8203e+008	15.7894
500x500	2.2197e+010	1.4273e+009	15.5519
600x600	3.2386e+010	2.1130e+009	15.3266
700x700	4.4493e+010	2.9426e+009	15.1206
800x800	5.8520e+010	3.9186e+009	14.9340

Table 17: The theoretical speed up ratio for time delay neural networks (2D-complex values input matrix, n=25).

Size of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	8.6605e+008	4.2909e+007	20.1836
200x200	4.6445e+009	1.9609e+008	23.6856
300x300	1.1422e+010	4.7334e+008	24.1301
400x400	2.1198e+010	8.8202e+008	24.0333
500x500	3.3973e+010	1.4273e+009	23.8028
600x600	4.9746e+010	2.1130e+009	23.5427
700x700	6.8519e+010	2.9425e+009	23.2856
800x800	9.0290e+010	3.9185e+009	23.0418

Table 15: The theoretical speed up ratio for time delay neural networks (2D-complex values input matrix, n=30).

Size of input matrix	Number of computation steps required for classical complex-valued neural networks	Number of computation steps required for fast complex-valued neural networks	Speed up ratio
100x100	1.0886e+009	4.2901e+007	25.3738
200x200	6.3143e+009	1.9608e+008	32.2021
300x300	1.5859e+010	4.7334e+008	33.5045
400x400	2.9722e+010	8.8201e+008	33.6981
500x500	4.7904e+010	1.4273e+009	33.5640
600x600	7.0405e+010	2.1130e+009	33.3197
700x700	9.7225e+010	2.9425e+009	33.0412
800x800	1.2836e+011	3.9185e+009	32.7581

Table 18: Practical speed up ratio for time delay neural networks (2D-complex values input matrix).

Size of input matrix	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	38.33	46.99	62.88
200x200	39.17	47.79	63.77
300x300	38.44	48.86	64.83
400x400	37.92	47.23	65.99
500x500	37.32	46.89	64.89
600x600	36.96	46.48	64.01
700x700	36.67	46.08	63.31
800x800	36.38	45.78	62.64