

Information Processing Using Circulant Matrices

By

VAMSI SASHANK KOTAGIRI

Bachelor of Technology in Computer Science

and Engineering

GITAM University

Visakhapatnam, AP, India

2012

Submitted to the Faculty of the Graduate College of the
Oklahoma State University in partial fulfillment of the
requirements for the Degree of

MASTER OF SCIENCE

May, 2014

Information processing using circulant matrices

Thesis approved:

Dr. Subhash Kak

Thesis Adviser

Dr. Johnson Thomas

Dr. David Cline

Name: VAMSI SASHANK KOTAGIRI

Date of Degree: MAY 2014

Title of Study: INFORMATION PROCESSING USING CIRCULANT MATRIX

Major Field: COMPUTER SCIENCE

ABSTRACT:

Circulant matrices may be used to process certain kinds of signals in computer science applications. Specifically, they can be used as signal transforms. In this thesis several new applications of circulant matrices are described. New results have been obtained in number theoretic Hilbert transform (NHT), which is a generalization of discrete Hilbert transform (DHT). The NHT matrix generates ideal orthogonal sequences named as random residue sequences, since the NHT matrix with its transpose computes all correlation in the block. Random residue sequences can be used as carriers for wireless communications. We also investigate applications of circulant matrices to store and reproduce patterns as neural memories.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
1.1 Matrix transforms in computer science applications.....	1
1.2 Discrete Fourier transform.....	1
1.3 The Discrete Hilbert Transform.....	2
1.4 Circulant matrices.....	4
1.5 DHT in Data Hiding.....	6
1.6 Problem Statement.....	8
II. REVIEW OF RELATED LITERATURE.....	9
2.1 NHT-circulant Matrices.....	9
2.2 Number Theoretic Hilbert Transform.....	9
2.3 Random sequences.....	10
2.4 D sequences.....	11
2.5 PN-sequences.....	12
2.6 Neural Networks.....	12
2.7 The Back Propagation Algorithm.....	16
2.8 Hopfield Network.....	17
III. NUMBER THEORETIC HILBERT TRANSFORM AND RANDOM RESIDUE SEQUENCES.....	19
3.1 The 10-point and 12-point NHT.....	19
3.2 Development of 14-point and 16-point NHT.....	21
3.3 Random residue sequences.....	28
IV. ARCHITECTURE OF WIRELESS COMMUNICATION SYSTEM USING RANDOM RESIDUE SEQUENCES.....	37
4.1 Spread spectrum based communication system.....	37
4.2 Communication system architecture using rr sequences.....	39
4.3 Implementation of wireless system architecture using 16 bit sequence...40	
V. MEMORY CAPACITY OF NEURAL NETWORKS USING CIRCULANT MATRICES.....	42
5.1 Motivation.....	42
5.2 Artificial neurons used.....	43
5.3 Structure of Memories generated by circulant matrices.....	46

VI.CONCLUSION	51
FUTURE WORK.....	52
REFERENCES.....	53

LIST OF TABLES

Table	Page
1. The 14-point NHT for modulus 29.....	23
2. The 16-point NHT for modulus 13.....	25
3. Examples for 16-bit long NHT sequences.....	31

LIST OF FIGURES

Figure	Page
1 Information hiding system.....	7
2 Neural network.....	13
3 Artificial neuron.....	14
4 Feed-forward network.....	15
5 Feed-back network.....	16
6 Hopfield network.....	18
7 Input graph for mod 7283.....	32
8 Autocorrelation function for the given input.....	32
9 Input graph for mod 2185.....	33
10 Autocorrelation function for the given input.....	33
11 Cross correlation function between example 1 and 2 mod $n=7283$	34
12 Cross correlation function between example 1 and 2 mod $n=21851$	34
13 Cross correlation function between example 1 and 3 mod $n=7283$	35
14 Cross correlation function between example 1 and 3 mod $n=3121$	35
15 Spread spectrum communication system.....	37
16 Wireless system architecture using random residue sequences.....	40
17 Zero auto correlation produced when user produces his sequence.....	41
18 Artificial neuron used in Hopfield network.....	44
19 Memories occupied by each order circulant matrix.....	49
20 Memories occupied by even order circulant matrix.....	49
21 Memories occupied by odd order circulant matrix.....	50

CHAPTER I

INTRODUCTION

1.1 Matrix Transforms in Computer Science Applications

A matrix is a rectangular array of number of rows and columns and a matrix consisting of m rows and n columns is known as $m \times n$ matrix. Often matrices are used to store data or to solve problems using certain matrix calculations. Matrix transforms are performed on matrices through matrix multiplication of a point matrix by a transform matrix. Let T be a transform matrix, P be a point matrix and N be a new transformed point matrix then $N = T P$. We can perform operations including rotation, scaling, shearing, reflection and orthogonal projection by using 2D matrix transforms. With the help of matrix theory methods and linear algebra matrix transforms can be developed with applications in areas of computer science like computer graphics, robotics, image processing, signal processing, cryptography, animation and so on [1]-[6].

1.2 Discrete Fourier Transform

Fourier transform is the operation that helps to decompose a signal into its constituent frequency components. The Fourier transform for a continuous time signal $x(t)$ may be defined as

$$X(w) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt, \omega \in (-\infty, \infty)$$

The Fourier transform comes in three varieties: the plain-old Fourier transform, the Fourier series and the discrete Fourier transform. Discrete Fourier transform is a specific kind of Fourier transform which converts or transforms finite function of equally spaced samples to a function of finite sampled coefficients. The input samples are complex numbers and the output coefficients are complex as well. From the above definition it means that discrete Fourier transform requires input function that is discrete and non-zero values must have a limited duration. From the Fourier transform we can easily obtain discrete Fourier transform with finite summation of limits over a signal x , which may be defined as

$$X(w_k) \triangleq \sum_{n=0}^{N-1} x(t_n) e^{-j\omega_k t_n} \quad k = 0, 1, 2, \dots, N-1$$

where \triangleq means “is defined as” or “equals by definition”. Discrete Fourier transforms are used for data compression, spectral analysis and data convolution.

1.3 The Discrete Hilbert Transform

The Hilbert transform has many applications in signal processing, imaging, modulation and demodulation of instantaneous frequency and in cryptography. The discrete Hilbert transform (DHT) has several forms [7]-[11].

The Discrete Fourier Transform (DFT) has a number theoretic version that has many applications, we would like to have similar number theoretic version of the Discrete Hilbert Transform. The basic Discrete Hilbert Transform (DHT) of discrete data $f(n)$ where $n = (-\infty, \dots, -1, 0, 1, \dots, \infty)$ is given by [7]

$$\text{DHT } \{f(n)\} = g(k) = \begin{cases} \frac{2}{\Pi} \sum_{n \text{ odd}} \frac{f(n)}{k-n}; & k \text{ even} \\ \frac{2}{\Pi} \sum_{n \text{ even}} \frac{f(n)}{k-n}; & k \text{ odd} \end{cases}$$

The inverse Discrete Hilbert Transform (DHT) is given as:

$$f(n) = \begin{cases} -\frac{2}{\Pi} \sum_{k \text{ odd}} \frac{g(k)}{n-k}; & n \text{ even} \\ -\frac{2}{\Pi} \sum_{k \text{ even}} \frac{g(k)}{n-k}; & n \text{ odd} \end{cases}$$

The matrix form of the DHT requires data the data of finite length. Since the DHT transform is defined for an infinite number of points, limitations of the DHT transform signal to a finite set would set up an approximation in the signal that is recovered.

The DHT is given below for data $n=0, 1, 2, \dots$

$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \\ g(4) \\ g(5) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \frac{2}{\pi} \begin{bmatrix} 0 & \frac{1}{-1} & 0 & \frac{1}{-3} & 0 & \frac{1}{-5} & 0 & \frac{1}{-7} & \cdot \\ \frac{1}{1} & 0 & \frac{1}{-1} & 0 & \frac{1}{-3} & 0 & \cdot & \cdot & \cdot \\ 0 & \frac{1}{1} & 0 & \frac{1}{-1} & 0 & \frac{1}{-3} & \cdot & \cdot & \cdot \\ \frac{1}{3} & 0 & \frac{1}{1} & 0 & \frac{1}{-1} & 0 & \cdot & \cdot & \cdot \\ 0 & \frac{1}{3} & 0 & \frac{1}{1} & 0 & \frac{1}{-1} & \cdot & \cdot & \cdot \\ \frac{1}{5} & 0 & \frac{1}{3} & 0 & \frac{1}{1} & 0 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{7} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

1.4 Circulant Matrices

In this section we deal with circulant matrices and their properties [13]. An $N \times N$ circulant matrix can be formed by starting with a vector having n components. This vector becomes the first row of the matrix and the subsequent rows shift the elements of the previous rows to the right.

$$C = \begin{bmatrix} a & b & c & \cdot & \cdot & k \\ k & a & b & c & \cdot & \cdot \\ \cdot & k & a & b & c & \cdot \\ \cdot & \cdot & k & a & b & c \\ c & \cdot & \cdot & k & a & b \\ b & c & \cdot & \cdot & k & a \end{bmatrix}$$

The eigenvectors of a circulant matrix are given by

$$u_j = (1, w_j, w_j^2, \dots, w_j^{N-1})^T, \text{ where } j = 0, 1 \dots N-1 \text{ and}$$

$$w_j = \exp\left(\frac{2\pi i j}{N}\right) \text{ are the } n\text{th root of unity and } i = \sqrt{-1}.$$

The determinants of circulant matrices are given by

$$\begin{aligned} \det \text{circ} \{(v_0, v_1, \dots, v_{n-1})\} &= (-1)^{n-1} \det \text{circ} \{(v_1, v_2, \dots, v_{n-1}, v_0)\} \\ &= (-1)^{n-1} \det \text{circ} \{(v_{n-1}, v_{n-2}, \dots, v_1, v_0)\} \end{aligned}$$

and iterations of these yield that $\det V = (-1)^{k(n-1)} \det T^k \cdot V$ for each integer $0 \leq k < n$.

Circulant matrices form a commutative algebra, since for any two given circulant matrices A and B, the sum A+B is circulant, the product AB is circulant and AB=BA.

The number of elements in the multiplicative group of circulant matrices of size $N \times N$ modulo elements p is [13],[14]

$$P^N - 1 - \text{part}(p, N) - (p-1)$$

where $\text{part}(p, N)$ is number of partitions of p in up to N parts.

1.5DHT in Data Hiding

The data hiding or digital watermarking technique is used for the purpose of authentication, annotation and copyright protection. Not only the imperceptibility but also

the robustness against common signal is concerned as the performance of data hiding. Data hiding in audio signals exploits imperfection of Human auditory system (HAS) known as audio masking. As HAS has a wider dynamic and differential range compared to the other human senses. Hiding data in audio signals present a variety of challenges.

The specific application of information hiding is one to which DHT lends itself naturally since phase shift in speech makes no difference as far as perception is concerned. In one method [15], the secret information which could be an image or some other linear sequence is encoded in binary form. The steganographic signal is hidden in the phase differences that will be produced based on whether the direct speech convert signal or its DHT has been transmitted.

Since DHT does not affect the spectrum, for it only shifts the phase and the human perception system is insensitive to it, the fact that the steganographic signal carries additional secret information will not be obvious.

The above model presents a schematic of the information hiding system. The secret sequence is determined by considering the DHT processed information for its shifts in phase according to a clock. The speech waveform will otherwise not be effected and therefore form a perception basis it will be unaltered.

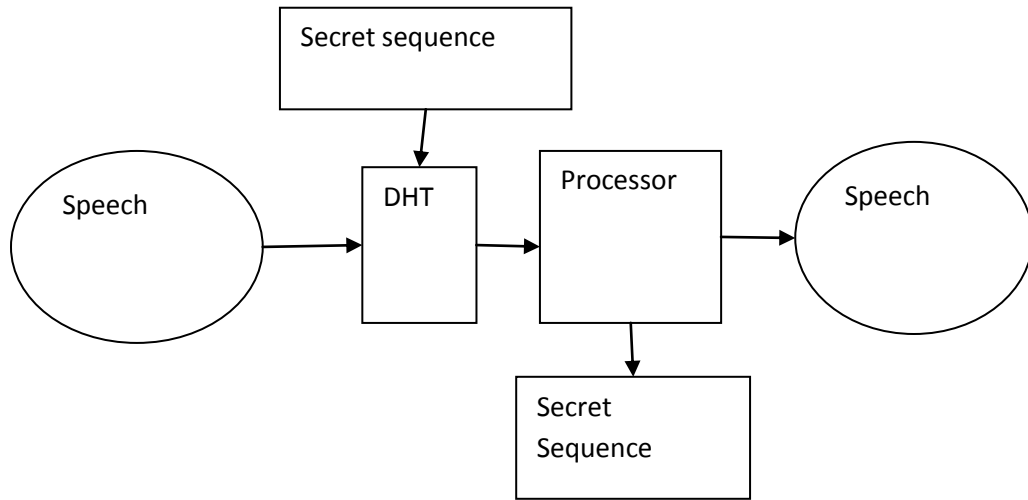


Figure 1: Information Hiding System

1.6 Problem Statement

Circulant matrix transforms are used to process certain kinds of signals in computer science applications. Further properties of the recently proposed number theoretic Hilbert transform [12] have been obtained [16], [17]. A new class of orthogonal sequences called as random residue sequences can be developed from the number theoretic Hilbert transform (NHT) matrix. Random residue sequences can be used as carriers in wireless communications in place of shift register sequences [18], [19]. Circulant matrices [14], [20] have the capability to store and reproduce patterns as neural memories [21], [22] and they could be models of neural networks at birth and other structured networks [23]-[43].

Although Hebbian learning is the most popular model used in neural networks, we are interested in determining if the regular connectivity in the networks that may be assumed to be circulant can also store memories. This is likely to have applications in understanding the memories that newly-born organism seem to possess. We show that such memories are possible and we provide some results.

The remainder of this thesis document is organized as follows. In Section II we briefly review the related work. In Section III we go through the random residue sequences and number theoretic Hilbert transform. The architecture of wireless communication system using random residue sequences is described in Section IV. In Section V we have a look at capacity of circulant matrices to store neural memories. The conclusions of the thesis are presented in Section VI.

CHAPTER II

REVIEW OF RELATED LITERATURE

2.1 NHT-circulant Matrices

A NHT circulant matrix can be formed by purging the diagonal zeroes in circulant matrix and adding alternative zeroes to all non-zero elements present. The NHT-circulant matrix has some additional constraints like the sum of the squares of the entries is 1 and the other requirement is that $NN^T=1$.

$$N = \begin{bmatrix} 0 & a & 0 & b & 0 & c & 0 & . & . & k \\ k & 0 & a & 0 & b & 0 & c & 0 & . & . \\ . & k & 0 & a & 0 & b & 0 & c & 0 & . \\ . & . & k & 0 & a & 0 & b & 0 & c & 0 \\ 0 & . & . & k & 0 & a & 0 & b & 0 & c \\ c & 0 & . & . & k & 0 & a & 0 & b & 0 \\ 0 & c & 0 & . & . & k & 0 & a & 0 & b \\ b & 0 & c & 0 & . & . & k & 0 & a & 0 \\ 0 & b & 0 & c & 0 & . & . & k & 0 & a \\ a & 0 & b & 0 & c & 0 & . & . & k & 0 \end{bmatrix} \text{mod } m$$

where N is the NHT transformation obtained and m is appropriate value of modulus.

2.2 Number Theoretic Hilbert Transform

With the help of NHT circulant matrix, a new class of transformation called the Number theoretic Hilbert Transform (NHT) can be obtained. NHT is a generalization of the standard discrete Hilbert transform (DHT). The notation for NHT transform is as follows:

F is the data block vector and G is the NHT transformed data block, N is the NHT transform matrix and the computations are with respect to the modulus m, the inverse of the NHT matrix is $N^T \text{ mod } m$. We can represent in mathematical form as

$$G = NF \text{ mod } m \text{ and}$$

$$F = N^T G \text{ mod } m \quad (1)$$

From the autocorrelation of a NHT, we will be able to derive a sequence which is zero for all non-zero shifts, which illustrates that these non-zero shifts are self-orthogonal sequences which can also be called as Random Residue Sequences.

2.3 Random Sequences

Random numbers are used in different fields in cryptography for generating encryption keys, in simulating and modeling complex phenomena. Random numbers may be classified as pseudo-random and true-random [44]-[47]. True-random numbers are unpredictable and cannot be generated by physical processes. Randomness into computers is introduced in the form of pseudo-random numbers. A truly random sequence of binary symbols would be one for which a knowledge of past history of sequence would be of no assistance in predicting the next symbol. Such a sequence is sometimes referred to as digital noise.

A random sequence is the one that appears to be perfectly random for K output symbols but then repeats i.e. it is periodic with a cycle time of K symbols. If K can be

made large enough then any interval up to K symbols will appear to be perfectly random. Random sequences can be constructed using shift registers. For a N stage shift register, $K = 2^N - 1$ for a maximal length shift register sequence. All settings cannot produce a maximal-length random sequence, but only when they have a cycle period less than K . Digital random sequences are used in error correcting and error detection codes. As random sequences have peaked autocorrelation properties they are widely used in radar ranging, GPS systems and spectrum communication systems such as digital cell phones. D sequences and PN sequences are two commonly used types.

2.4 D Sequences

Prime reciprocal or D sequences are obtained in expansion of fractions or irrational numbers and thus are “decimal” sequences to arbitrary bases [44]-[45]. Decimal sequences are obtained when a number is represented in a decimal form in a base r and they may terminate, repeat or be periodic. For a certain class of decimal sequences of $1/q$, q is prime, the digits are spaced half a period apart add up to $r - 1$, where r is the base in which the sequence is expressed as

$$a_i = (2^i \bmod p) \bmod 2$$

The D-sequences are periodic and their randomness properties are checked only in one period. D-sequences may be generated by using feedback shift registers. Decimal

sequences are also known to have good cross correlation properties and they can be used in applications for encryption and error correction coding.

2.5 PN-Sequences

A Pseudo-random Noise (PN) sequence is a sequence of binary numbers, which appears to be random, but in fact it is perfectly deterministic [18]. The sequence appears to be random in sense that binary values and groups of binary value occur in the sequence in the same proportion. For a sequence to be a pseudo noise sequence it should follow the following basic rules.

- The relative frequency of 0's and 1's are each $1/2$.
- The run lengths of 0's and 1's are, $1/2$ for all run lengths of length 1, $1/4$ for all run lengths of length 2 and so on.
- If a PN sequence is shifted by any non-zero number of elements, then the resulting sequence will have an equal number of agreements and disagreements with respect to the original sequence.

These properties are known as balance property, run property and correlation property respectively. PN sequences are known as maximal length sequences are generated by using linear feedback shift register. PN sequences are widely used in digital communications, instrumentation etc. The good autocorrelation property of PN sequences makes them suitable for frame synchronization in digital communications.

2.6 Neural Networks

An artificial neural network is an information processing paradigm that is inspired by the way nervous systems, such as brain, process information [21]. The key element of this paradigm is the novel structure of information processing system. It is composed of a large number of highly interconnected processing elements called neurons working in parallel to solve a specific problem. A neural network like people learn by example i.e. neural network is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true even in the case of artificial neural networks as well.

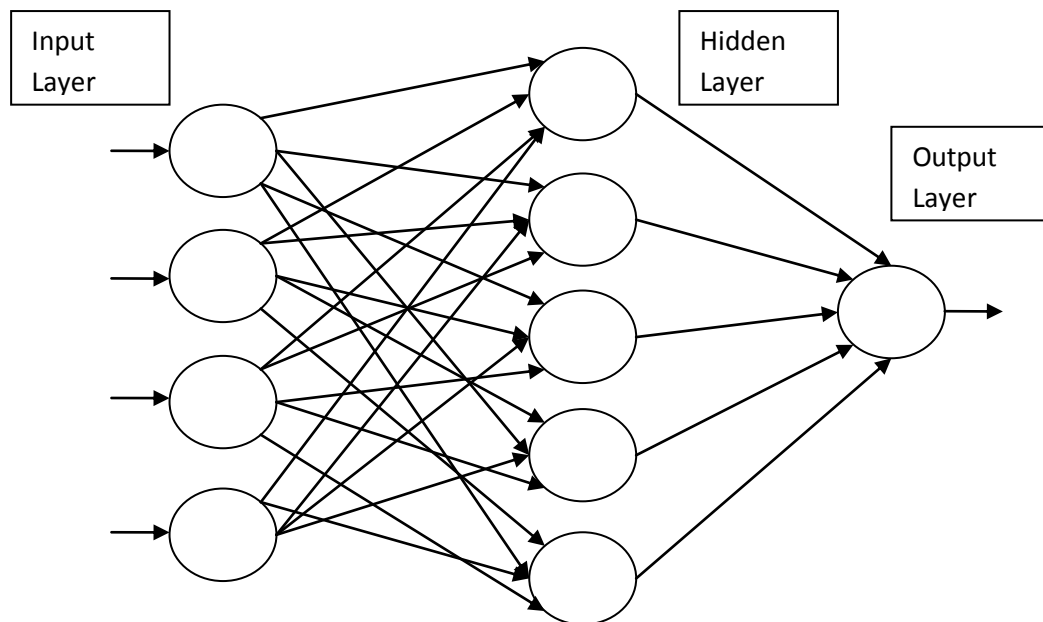


Figure 2: Neural network

A trained neural network can be thought of as an expert in the category of information it has been given to analyze. A neural network can create its own organization or representation of information it receives during learning time. Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage this tells us that neural networks are fault tolerant.

A simple neuron. A typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin structure known as an axon, which splits into thousands of branches. At end of each branch, a structure called a synapse converts the activity from the axon into electrical effects, which inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron changes the other.

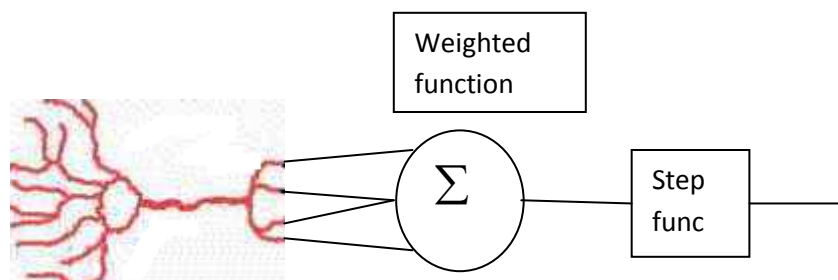


Figure 3: An artificial neuron

Architecture of neural networks

Based on the topology of neural networks, they can be classified into two types: feedforward and feedback. A feed-forward neural network allows signals to travel one way only, from input to output. There is no feedback i.e. the output of any layer does not affect the current layer. Feed-forward neural networks tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition.

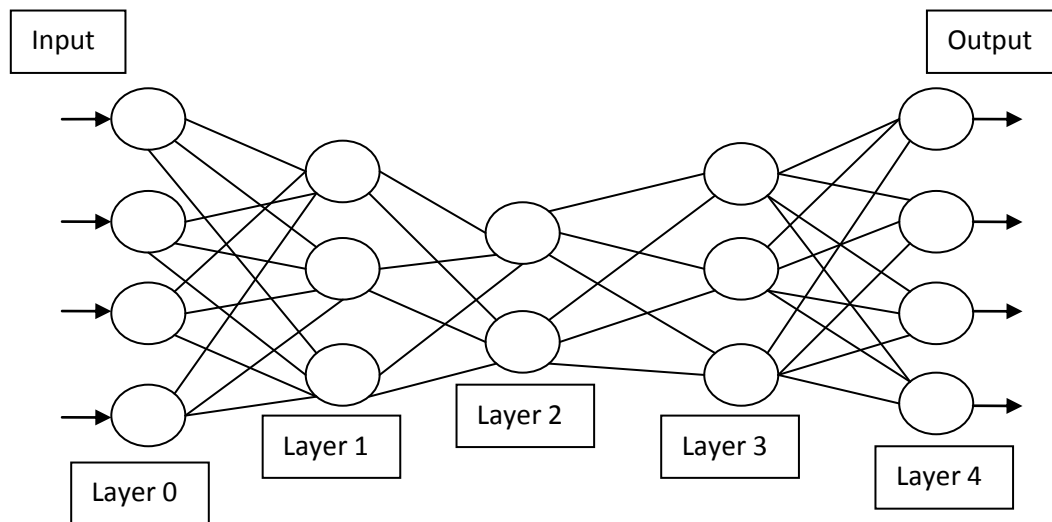


Figure 4: Feed-forward network

Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback architectures are also referred to as interactive or recurrent, although the later term is often used to denote feedback connections in single-layer organizations.

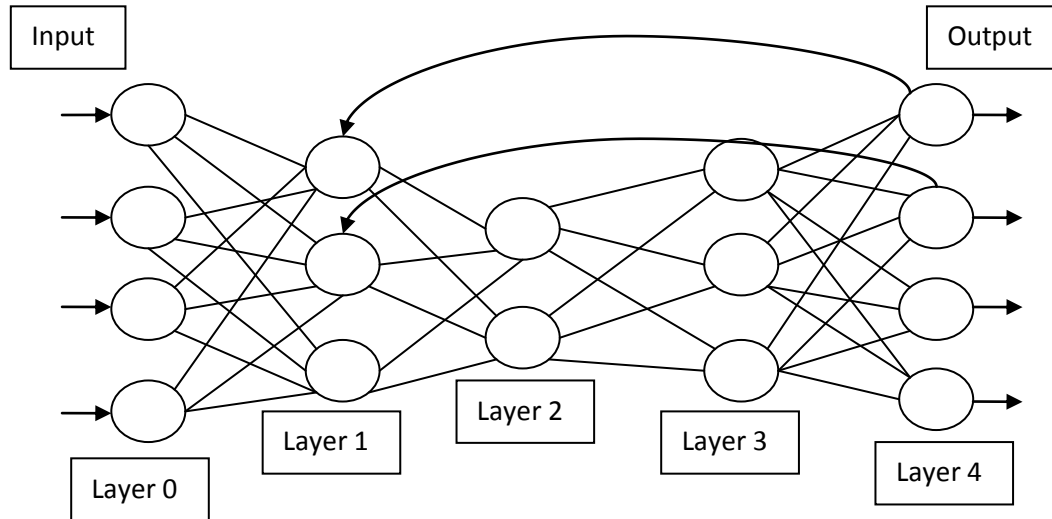


Figure 5: Feed-back network

2.7 The Back Propagation Algorithm

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (EW). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is most widely used for determining EW.

The back propagation algorithm is easiest to understand if all the units in the network are linear. The algorithm computes each EW by first computing the EA, the rate at which the error changes as the activity level of a unit is changed. For output units, the EA is simply the difference between the actual and desired output. To compute EA

for a hidden unit in the layer before the output layer, we first need to identify all the weights between the hidden unit and the output units to which it is connected. Then we multiply those weights by the EAs of output units and add the products. This sum equals to the EA for the chosen hidden unit.

After calculating all the EAs in the hidden layer just before the output layer, we compute EAs for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is why it gets the name back propagation. Once the EA has been computed for a unit, it is straight forward to compute EW for each incoming connection of the unit. The EW is the product of the EA and activity through incoming connection. The back propagation algorithm with non-linear units includes an extra step. Before back-propagating, the EA must be converted, the rate at which the error changes as the total input received by a unit is changed.

2.8 Hopfield network

The Hopfield network is created by supplying input data vectors, or pattern vectors, corresponding to the different classes. These patterns are called class patterns. In an n-dimensional data space the class patterns should have n binary components $\{1,-1\}$ i.e. each class pattern corresponds to a corner of a cube in an n-dimensional space. A Hopfield network can be used as an associative memory. If we want to “imprint” m different stable states in the network if we have to find adequate weights for the

connections. Hebbian learning can be implemented into Hopfield network by loading the m selected n -dimensional states x_1, x_2, \dots, x_m on the network and by updating the network's weights which are initially set to zero after each presentation according to the rule

$$w_{ij} \leftarrow w_{ij} + x_i^k x_j^k, \quad i, j = 1, \dots, n \text{ and } i \neq j.$$

x_i^k and x_j^k denote i th and j th component of the vector x_k .

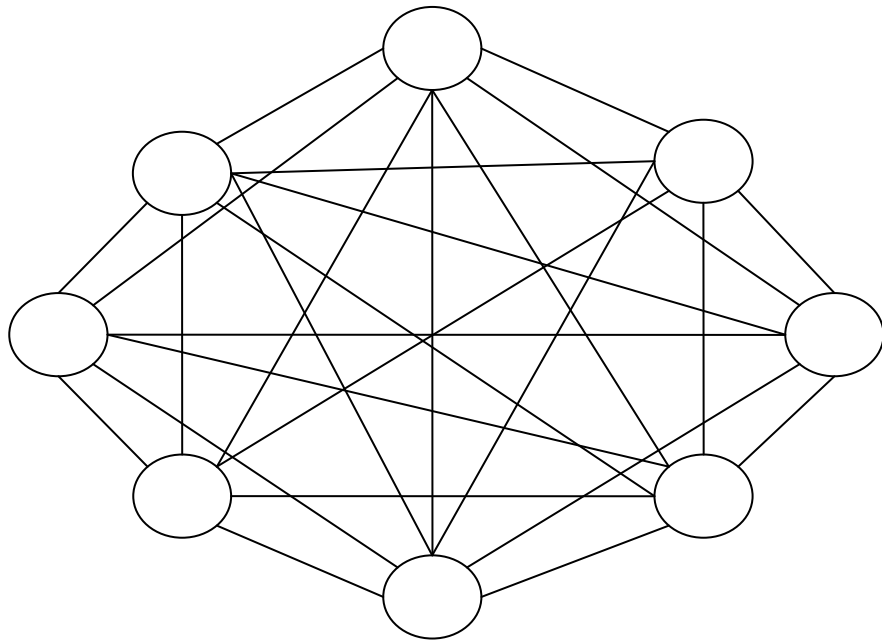


Figure 6: Hopfield network

Quantum processing may also play a role in neural networks [48]-[51]. Classical processing is described in [52]-[71].

CHAPTER III

NUMBER THEORETIC HILBERT TRANSFORM AND RANDOM RESIDUE SEQUENCES

In this chapter, we present theory of NHT transforms and random residue sequences derived from NHT transform. In this chapter we have developed NHT matrices up to 16-point NHT. The NHT transformation can be used as a primitive to create cryptographically useful scrambling transformations. In [13], NHT matrices up to 8-point were presented.

3.1 The 10-point and 12-point NHT

Here we are going to deal with 10-point NHT and 12-point NHT in detail with examples which provide greater flexibility in their use.

10-point NHT

The first row of the 10-point NHT matrix will be given by integers a, b, c, d, e that alternate with 0s. If we multiply the 10-point NHT with its transpose we observe that

$$a^2 + b^2 + c^2 + d^2 + e^2$$

is the diagonal term and the non-diagonal terms are

$$(b+e)a + (c+e)d + bc \text{ and } (a+e)c + (b+a)d + eb.$$

There are many solutions which satisfy the equation if we randomly choose the values of $a=2$ $b=1$ $c=2$ $d=5$ $e=3$ and in order to get a valid NHT matrix we need to assume a suitable modulus such that the non- diagonal elements of NN^T will become zero and only the diagonal elements remain.

$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \\ g(4) \\ g(5) \\ g(6) \\ g(7) \\ g(8) \\ g(9) \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 & 1 & 0 & 2 & 0 & 5 & 0 & 3 \\ 3 & 0 & 2 & 0 & 1 & 0 & 2 & 0 & 5 & 0 \\ 0 & 3 & 0 & 2 & 0 & 1 & 0 & 2 & 0 & 5 \\ 5 & 0 & 3 & 0 & 2 & 0 & 1 & 0 & 2 & 0 \\ 0 & 5 & 0 & 3 & 0 & 2 & 0 & 1 & 0 & 2 \\ 2 & 0 & 5 & 0 & 3 & 0 & 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 5 & 0 & 3 & 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 & 5 & 0 & 3 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 & 0 & 5 & 0 & 3 & 0 & 2 \\ 2 & 0 & 1 & 0 & 2 & 0 & 5 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \\ f(9) \end{bmatrix} \pmod{7}$$

It is easy to check $NN^T = I \pmod{7}$

12-point NHT

The first row of the 12-point NHT matrix will be given by integers 0, a, 0, b, 0, c, 0, d, 0, e, 0, f. If we multiply the 12-point NHT with its transpose we observe that the squares of the non-zero integer values of the first row should equal 1 modulo the chosen m .In other words

$$a^2+b^2+c^2+d^2+e^2+f^2$$

The other non-diagonal element terms in the product NN^T

$$\begin{aligned} &(a + e) f + (e + c) d + (c + a) b, \\ &2(ad + be + fc) \text{ and} \\ &(e + c) a + ec+ (b + d) f +bd. \end{aligned}$$

There are many solutions which satisfy the equation if we randomly choose the values of

a=14 b=28 c=18 d=27 e=23 f=7 and thus we can write the 12-point NHT transformation as $G=HT \pmod{29}$.

$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \\ g(4) \\ g(5) \\ g(6) \\ g(7) \\ g(8) \\ g(9) \\ g(10) \\ g(11) \end{bmatrix} = \begin{bmatrix} 0 & 14 & 0 & 28 & 0 & 18 & 0 & 27 & 0 & 23 & 0 & 7 \\ 7 & 0 & 14 & 0 & 28 & 0 & 18 & 0 & 27 & 0 & 23 & 0 \\ 0 & 7 & 0 & 14 & 0 & 28 & 0 & 18 & 0 & 27 & 0 & 23 \\ 23 & 0 & 7 & 0 & 14 & 0 & 28 & 0 & 18 & 0 & 27 & 0 \\ 0 & 23 & 0 & 7 & 0 & 14 & 0 & 28 & 0 & 18 & 0 & 27 \\ 27 & 0 & 23 & 0 & 7 & 0 & 14 & 0 & 28 & 0 & 18 & 0 \\ 0 & 27 & 0 & 23 & 0 & 7 & 0 & 14 & 0 & 28 & 0 & 18 \\ 18 & 0 & 27 & 0 & 23 & 0 & 7 & 0 & 14 & 0 & 28 & 0 \\ 0 & 18 & 0 & 27 & 0 & 23 & 0 & 7 & 0 & 14 & 0 & 28 \\ 28 & 0 & 18 & 0 & 27 & 0 & 23 & 0 & 7 & 0 & 14 & 0 \\ 0 & 28 & 0 & 18 & 0 & 27 & 0 & 23 & 0 & 7 & 0 & 14 \\ 14 & 0 & 28 & 0 & 18 & 0 & 27 & 0 & 23 & 0 & 7 & 0 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \\ f(9) \\ f(10) \\ f(11) \end{bmatrix} \pmod{29}$$

It is easy to check $NN^T = I \pmod{29}$.

3.2 Development of 14-point and 16-point NHT

Let us have a look in detail about 14-point NHT and 16-point NHT in detail with examples. The first row of the 14-point NHT matrix will be given by integers a, b, c, d, e, f, g that alternate with 0s. The problem is to find the circulant matrix with these values in the first row that satisfies the conditions given by (1). When we multiply the 14-point NHT with its transpose we observe that the squares of the non-zero integer values of the first row should equal 1 modulo the chosen m. In other words

$$a^2 + b^2 + c^2 + d^2 + e^2 + f^2 + g^2 = 1 \pmod{m} \quad (2)$$

The other non-diagonal element terms in the product NN^T

$$\begin{aligned}
& ab + bc + cd + de + ef + fg + ga \\
& ac + bd + ce + df + eg + fa + gb \\
& ad + be + cf + dg + ef + fg + ga
\end{aligned}$$

(3)

should be all zero with respect to the same modulus.

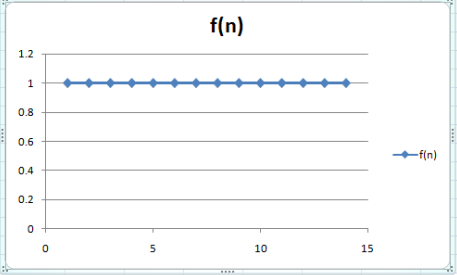
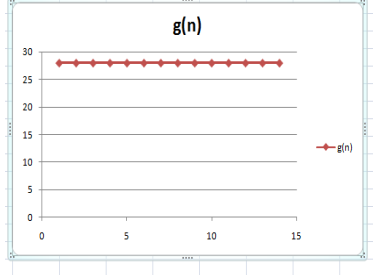
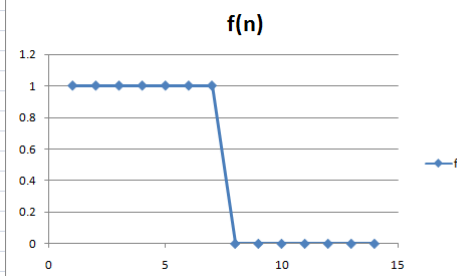
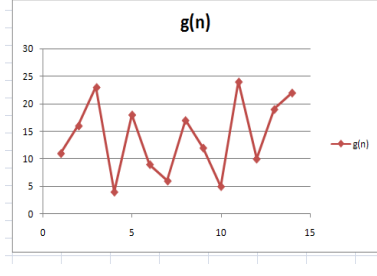
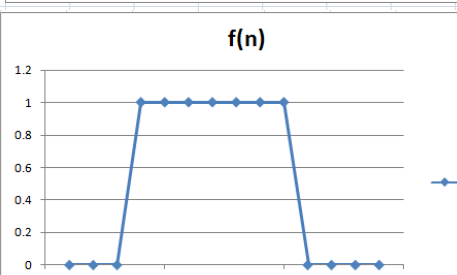
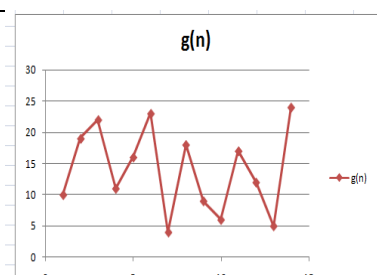
In order to get a valid NHT matrix we need to assume a suitable modulus in such that all the non-diagonal elements of the matrix product NN^T will become zero and only the diagonal elements of the product matrix remain. There are many solutions which satisfy the equation if we randomly choose the values of $a=3$ $b=15$ $c=22$ $d=11$ $e=20$ $f=10$ and $g=5$ and thus we can write the 14-point NHT transformation as $G=HT \pmod{29}$ which is shown as below:

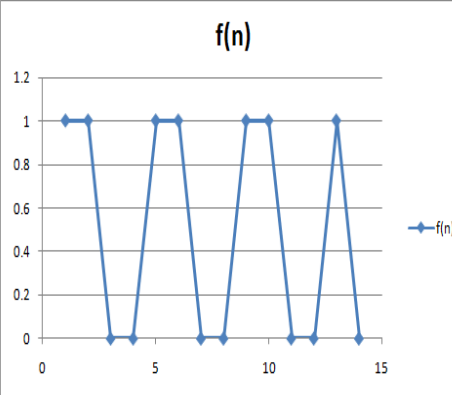
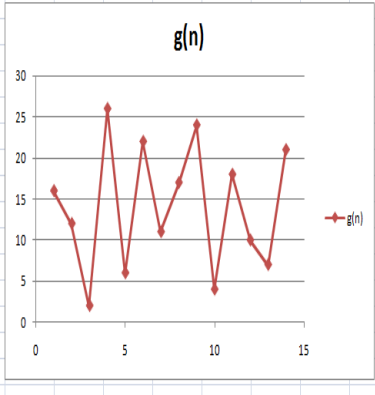
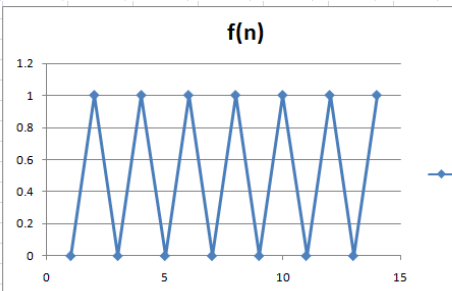
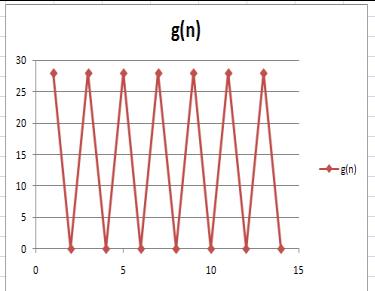
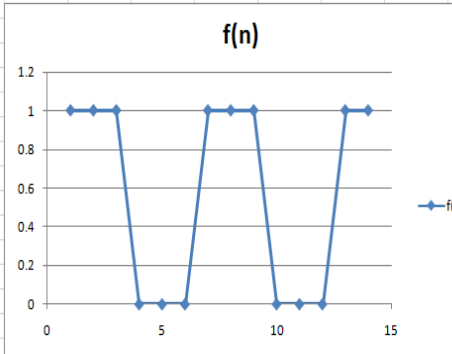
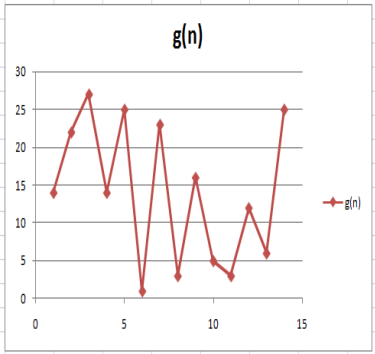
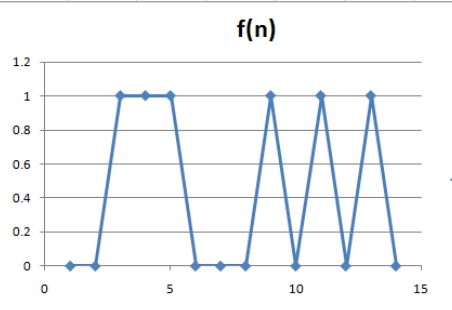
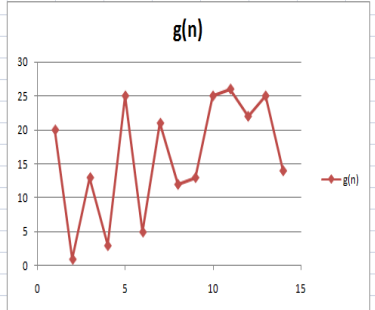
$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \\ g(4) \\ g(5) \\ g(6) \\ g(7) \\ g(8) \\ g(9) \\ g(10) \\ g(11) \\ g(12) \\ g(13) \end{bmatrix} \begin{bmatrix} 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 \\ 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 \\ 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 \\ 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 \\ 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 \\ 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 \\ 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 & 11 \\ 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 & 0 \\ 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 & 22 \\ 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 & 0 \\ 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 & 15 \\ 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 & 0 \\ 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 & 3 \\ 3 & 0 & 15 & 0 & 22 & 0 & 11 & 0 & 20 & 0 & 10 & 0 & 5 & 0 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \\ f(9) \\ f(10) \\ f(11) \\ f(12) \\ f(13) \end{bmatrix}$$

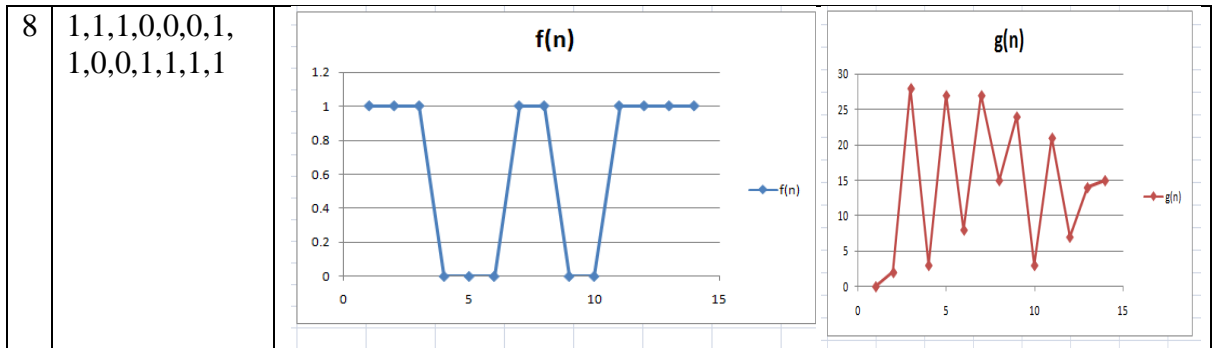
It is easy to check $NN^T = I \pmod{29}$.

We now present the data and transform block pairs for different choices of the data values.

Table 1. The 14-point NHT for modulus 29 ($a=3$ $b=15$ $c=22$ $d=11$ $e=20$ $f=10$ $g=5$)

	$f(n)$		$g(n)$
1	1,1,1,1,1,1,1, 1,1,1,1,1,1,1		
2	1,1,1,1,1,1,1, 0,0,0,0,0,0,0		
3	0,0,0,1,1,1,1, 1,1,1,0,0,0,0		

4	1,1,0,0,1,1,0, 0,1,1,0,0,1,0	 <p>$f(n)$</p>	 <p>$g(n)$</p>
5	0,1,0,1,0,1,0, 1,0,1,0,1,0,1	 <p>$f(n)$</p>	 <p>$g(n)$</p>
6	1,1,1,0,0,0,1, 1,1,0,0,0,1,0	 <p>$f(n)$</p>	 <p>$g(n)$</p>
7	0,0,1,1,1,0,0, 0,1,0,1,0,1,1	 <p>$f(n)$</p>	 <p>$g(n)$</p>



16-point NHT:

Given the first row of the 16-point NHT matrix is $0, a, 0, b, 0, c, 0, d, 0, e, 0, f, 0, g, 0, h$ and is given by the following matrix shown below. By multiplying the 16-point matrix with its transpose we observe that

$$a^2 + b^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2$$

is the diagonal element term and the non-diagonal element terms are

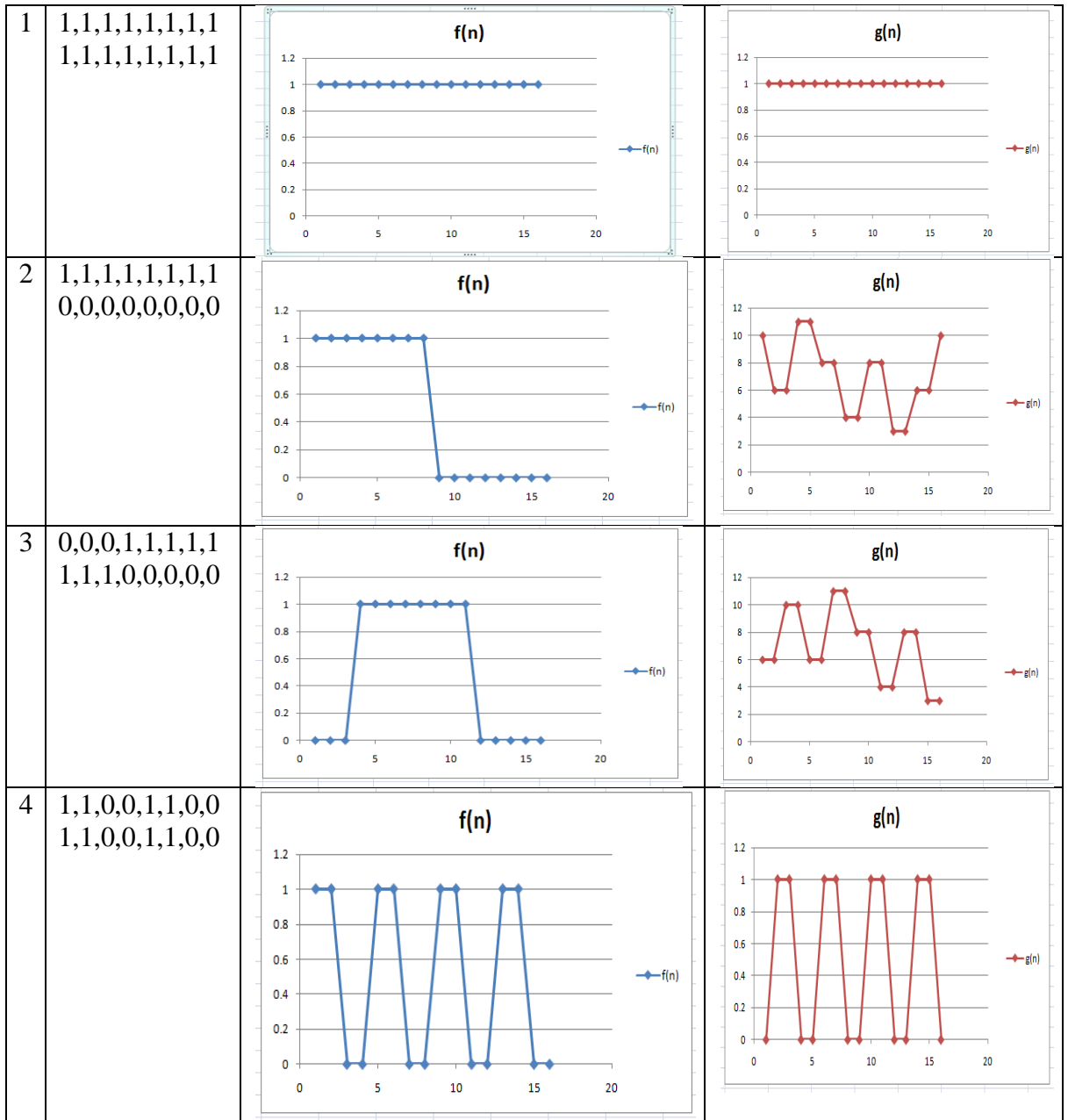
$$(a + g) d + (e + c) h + (c + a) f + (e + g) b,$$

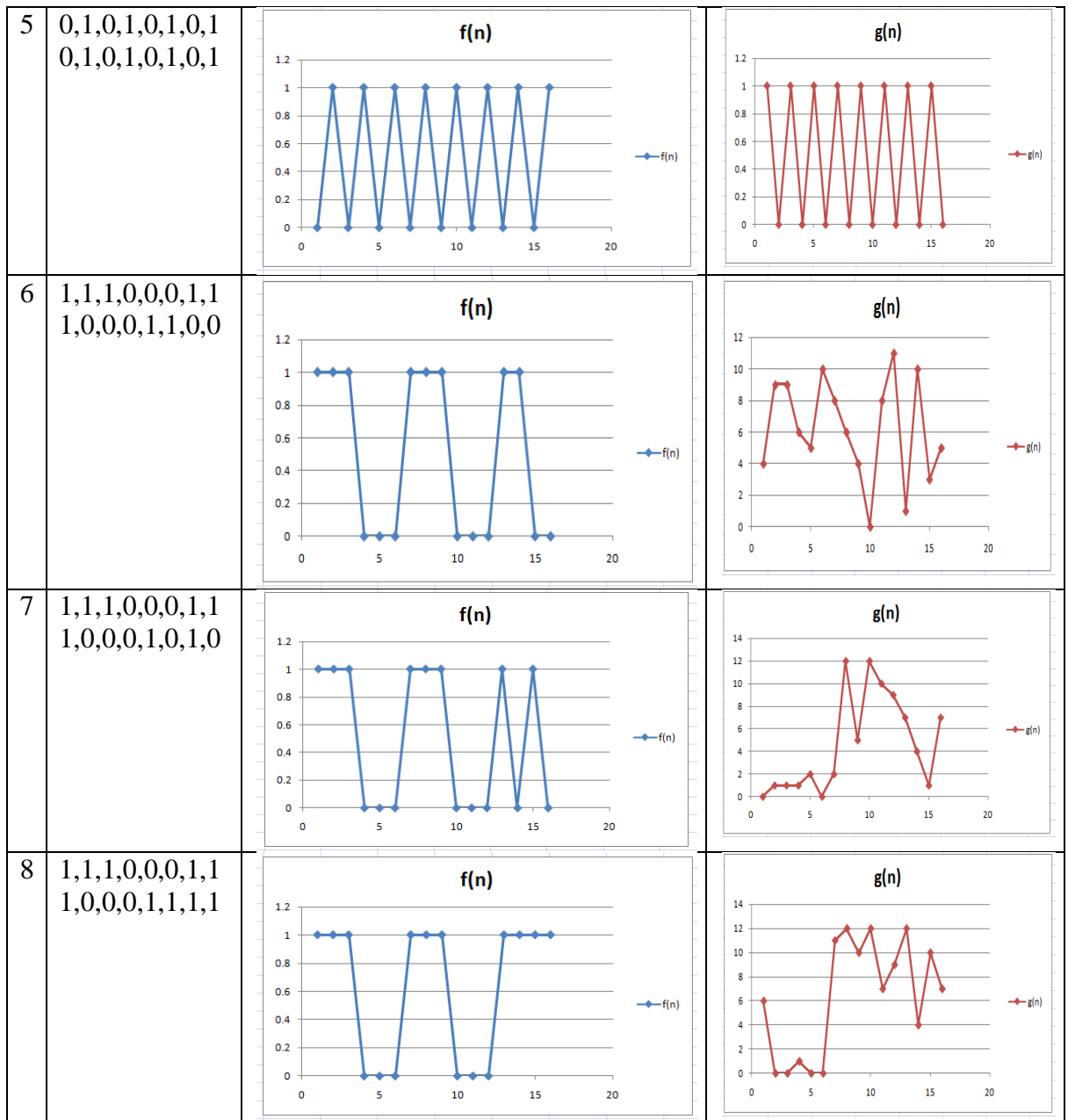
$$(a + e)(g + c) + (h + d)(b + f) \text{ and}$$

$$(a + g) h + (e + c) d + (c + a) b + (e + g) f$$

Table 2.16-point NHT for 13 ($a=7$ $b=11$ $c=12$ $d=6$ $e=3$ $f=8$ $g=4$ $h=2$)

	$f(n)$		$g(n)$
--	--------	--	--------





To get correct NHT matrix we need to select an appropriate matrix in such a way that all of the non diagonal elements in the NN^T product matrix will become zero and only the diagonal elements will remain. Though there are infinite number of solutions we choose

the values as $a=7, b=11, c=12, d=6, e=3, f=8, g=4, h=2$ randomly and we can write the

NHT transformation as $G=HT \pmod{13}$ which is as follows with $NN^T = I \pmod{13}$.

$$\begin{array}{l}
 \left[\begin{array}{l} g(0) \\ g(1) \\ g(2) \\ g(3) \\ g(4) \\ g(5) \\ g(6) \\ g(7) \\ g(8) \\ g(9) \\ g(10) \\ g(11) \\ g(12) \\ g(13) \\ g(14) \\ g(15) \end{array} \right] \left[\begin{array}{cccccccccccccccc}
 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 \\
 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 \\
 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 \\
 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 \\
 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 \\
 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 \\
 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 \\
 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 \\
 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 & 6 \\
 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 & 0 \\
 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 & 12 \\
 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 & 0 \\
 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 & 11 \\
 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 & 0 \\
 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0 & 7 \\
 7 & 0 & 11 & 0 & 12 & 0 & 6 & 0 & 3 & 0 & 8 & 0 & 4 & 0 & 2 & 0
 \end{array} \right] \left[\begin{array}{l} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \\ f(9) \\ f(10) \\ f(11) \\ f(12) \\ f(13) \\ f(14) \\ f(15) \end{array} \right]
 \end{array}$$

3.3RANDOM RESIDUE SEQUENCES:

The search for random sequences with ideal randomness properties is an important area of computer science. Shift-register sequences provide near-ideal autocorrelation function but the period constrained to be 2^n-1 for different values of n . Random sequence family is that of “decimal sequence” By obtaining residue sequences modulo prime that have ideal autocorrelation function that is it is zero for all non-zero values of the argument. In other words rather than binary sequences we wish to deal with sequences where individual items are integers modulo a prime. We do so by using the structure of the NHT matrix.

The idea of using NHT matrix is to generate random residue sequences comes from the fact that the product of the NHT matrix with its transpose computes all correlations on the block. Therefore the circulant part of the NHT matrix should be able to generate ideal random sequences. We represent the first row of matrix by using integers $a,b,c,d,e,f,g,h,i,j,k,l, m,n,o,p$ and alternate with 0s.

$$a^2 + b^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2 + i^2 + j^2 + k^2 + l^2 + m^2 + n^2 + o^2 + p^2$$

The other non-diagonal element terms in the product NN^T are:

$$2(ai + bj + ck + dl + em + fn + go + hp)(3)$$

$$a(h+j)+b(i+k)+c(j+l)+d(k+m)+e(l+n)+f(m+o)+g(n+p)+pi+oh(4)$$

$$a(g+k)+b(h+l)+c(i+m)+d(j+n)+e(k+o)+f(l+p)+gm+hn+io+jp(5)$$

$$a(f+l)+b(g+m)+c(h+n)+d(i+o)+e(j+p)+k(f+p)+gl+hm+in+jo(6)$$

$$a(e+m)+b(f+n)+c(g+o)+d(h+p)+i(e+m)+j(f+n)+k(g+o)+l(h+p) (7)$$

$$a(d+n)+b(e+o)+c(f+p)+g(d+j)+h(e+k)+i(f+l)+m(j+p)+kn+lo(8)$$

$$a(c+o)+b(d+p)+e(c+g)+f(d+h)+i(g+k)+j(h+l)+m(k+o)+n(l+p)(9)$$

$$ab + bc + cd + de + ef + fg + gh + hi + ij + jk + kl + lm + mn + no + op + pa.(10)$$

The autocorrelation function captures the correlation of data with itself .For a data

sequence $a(n)$ of N points the autocorrelation function $C(k)$ is represented by

$$C_a(k) = \frac{1}{N} \sum_{j=1}^N a(j)a(j+k)$$

For a noise sequence the autocorrelation function $C_a(k) = E(a(i)a(i+k))$ is two valued with value of 1 for $k=0$ and a value approaching zero for $k \neq 0$ for a zero-mean random variable. Assuming periodicity such a sequence will have $C(k)$ as 1 for $k=0$ and approximately μ^2 for non-zero k . μ is the mean of the variable.

We now present an algorithm for generating the NHT generator sequence. The basic idea that has worked very well is to pick a number that is prime and then pick numbers that are powers of 2.

Algorithm for Generating the Sequence

1. Enter the number of rows and columns of the circulant matrix generally the number of rows and columns will be equal to desired NHT i.e (16 by 16 for 16-point NHT).
2. Then choose the elements of circulant NHT matrix in such a way that one element need to be a prime number and the remaining elements need to be 2 and multiples of 2.
3. Then find the transpose of the circulant NHT matrix.
4. Multiply the circulant NHT matrix with its transpose matrix and we will get a product NHT matrix.

5. Find the gcd of all the non-diagonal elements in the obtained product NHT matrix.
6. The gcd of non-diagonal elements will be the modulus of the circulant NHT matrix, in most of the cases it will be a prime modulus.
7. If the desired format is $I \bmod n$ then we need to normalize the elements of the NHT matrix with the remainder obtained by taking modulus of the diagonal elements.
8. We will get the elements $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \dots \mathbf{a}_n$ and modulus which will be a prime number.

Pseudo code:

1. Input rows[m] and columns[n] of matrix .i.e matrix[m][n]
2. for i<-0 to m do // Input circulant matrix
3. for j<-0 to n do
4. Matrix[i][j] <- values.
5. for i<-0 to m do
6. for j<-0 to n do //Finding transpose of matrix
7. transpose[j][i]=matrix[i][j]
8. for i<-0 to n do
9. for j<-0 to m do
10. transpose[i][j]
11. for i<-0 to m do
12. for j<-0 to n do
13. for k<-0 to m do //Multiply circulant with its transpose
14. sum <-0
15. sum = sum + matrix[i][k]*transpose[k][j];
16. mul[i][j]=sum;
17. for i<-0 to 1 do //Initialize non diagonal elements of matrix
18. for j<-1 to n do
19. mul[i][j];
20. gcd=greatestcommondvisor(gcd,mul[i][j]);
21. greatestcommondvisor(a , b) //gcd of non diagonal elements.
22. while (a % b != 0) do

- 23. x=b;
- 24. y=a%b;
- 25. a=x;
- 26. b=y;
- 27. return b;

Experimental Results for Autocorrelation Function:

We have done random experiments by using above algorithm for different values of input sequences. As seen from Figures 1-4 their amplitudes have a variety of relationships.

Table 3: Example 16-bit long NHT sequences

Example	a	b	c	d	e	f	g	h	i
1	911	1821	3642	1	2	4	8	16	32
2	12747	3642	7284	14568	7285	14570	7289	14578	7305
3	3	2	4	8	16	32	64	128	256
4	2	2	4	8	16	32	64	128	256
5	11	2	4	8	16	32	17	34	21
6	13	2	4	8	16	32	64	128	256

j	k	l	m	n	o	p	mod q
64	128	256	512	1024	2048	4096	7283
14610	7369	14738	7625	15250	8649	17298	21851
512	1024	2048	975	1950	779	1558	3121
181	31	62	124	248	165	330	331
42	37	27	7	14	28	9	47
512	1024	61	122	244	488	976	1987

Random sequences from the first four examples of Table 3, when plotted on a graph

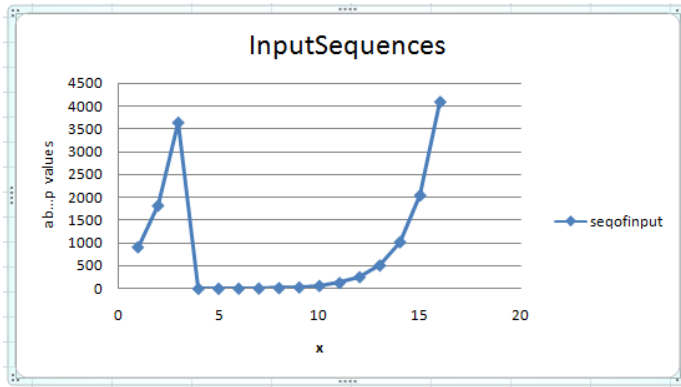


Figure 7: Input graph for the above values

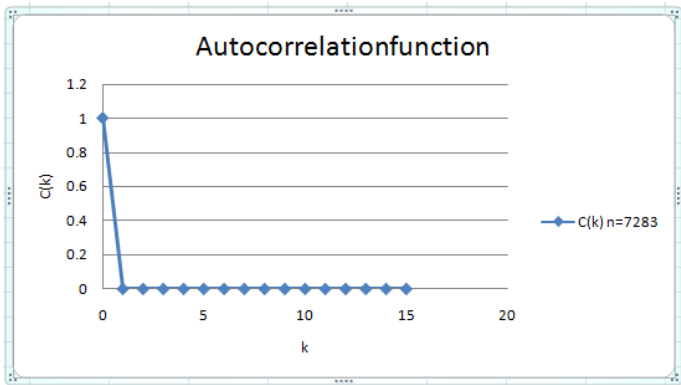


Figure 8: Autocorrelation function for the above input

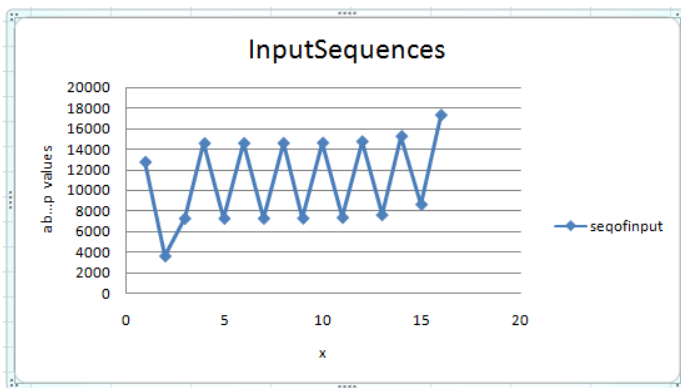


Figure 9: Input graph for the example 2 values mod 2185

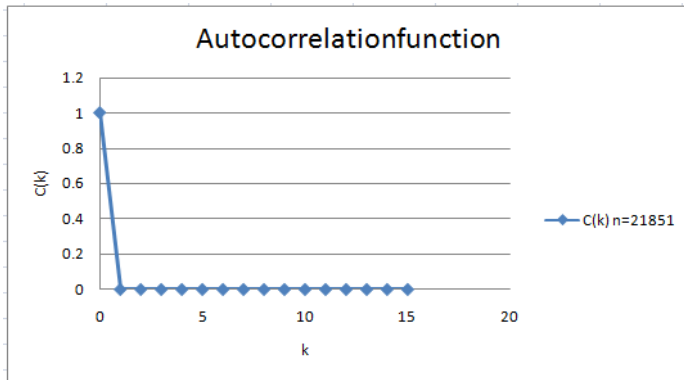


Figure 10: Autocorrelation function

Cross-Correlation Function:

The cross-correlation function captures the correlation of the data with other sequences.

For a data sequence $a(n)$ of N points cross correlation function $C(k)$ is represented by

$$C_c(k) = \frac{1}{N} \sum_{j=1}^N a(j)b(j+k)$$

where the value $b(j+k)$ corresponds to the next sequence result.

For a noise sequence, the cross correlation function $C_c(k) = E(a(i)b(i+k))$. If the two sequences are independent the cross correlation will be the product of their individual means.

Experimental results for Cross-correlation function:

We continue with the first four examples of Table 3 and we compute their mutual cross-correlation function.

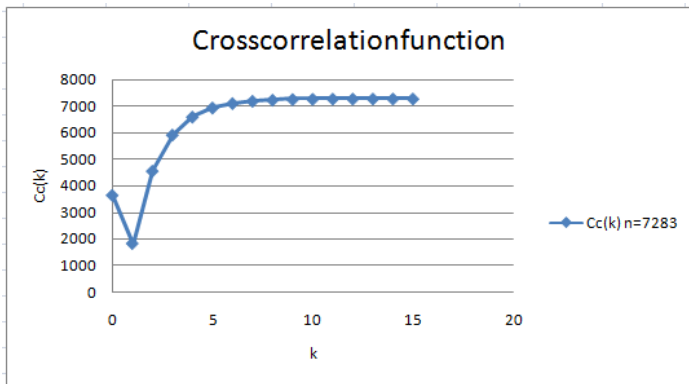


Figure 11: Cross correlation function between example 1 and 2 when mod n=7283

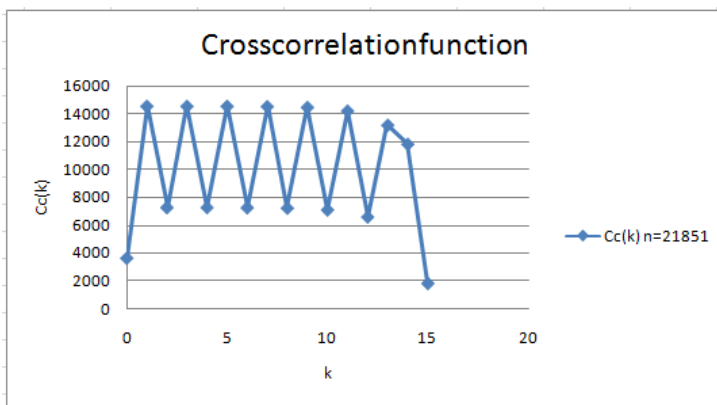


Figure 12: Cross correlation function between example 1 and 2 when mod n=21851

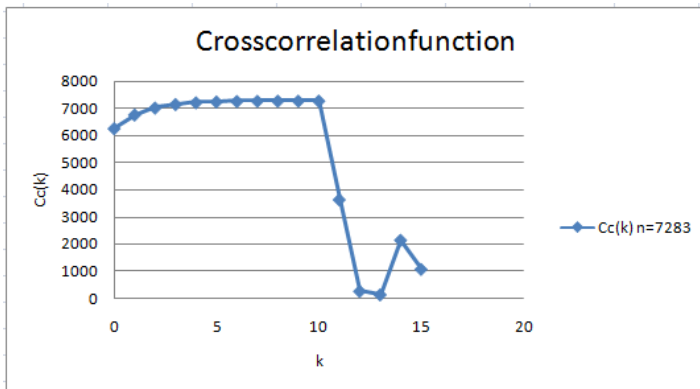


Figure 13: Cross correlation function between example 1 and 3 when mod n=7283

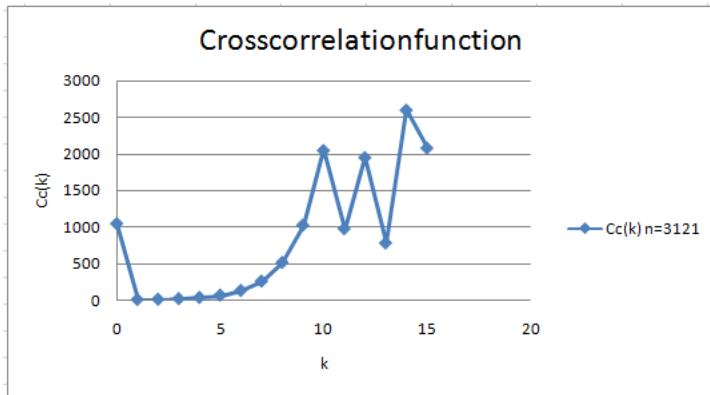


Figure 14: Cross correlation function between example 1 and 3 when mod n=3121

This chapter presents random residue sequences defined modulo of a prime that have perfect autocorrelation properties and variable cross correlation properties. The fact of zero autocorrelation function for all non-zero lags makes them suitable for use in applications where orthogonal sequences are needed.

The fact that the NHT related circulant matrix allows us to generate perfect random residue sequences opens up the larger question of the property of such random residue sequences .In particular it raises the question whether there is a general algorithm to generate the elements $a_1 a_2 a_3 a_4 \dots a_n$.Such an algorithm was discussed in this chapter along with pseudo code. Orthogonal sequences could have application as keys in environments which are extremely noisy since these strings satisfy certain properties of minimum mutual distance amongst themselves.

CHAPTER IV

ARCHITECTURE OF WIRELESS COMMUNICATION SYSTEM USING RANDOM RESIDUE SEQUENCES

In this chapter, we are going to describe the architecture of wireless communication system using the random residue sequences.

4.1 Spread Spectrum based communication system

The main idea behind spread spectrum was to use more bandwidth than the original message by maintaining same signal power [19]. As the spread spectrum signal does not have a clear distinguishable peak in the time domain this makes the signal difficult to distinguish from its noise. Spread spectrum is used to provide secure communication by spreading the given signal over a large frequency band, because of this reason spread spectrum signals can transmit with low spectral power density.

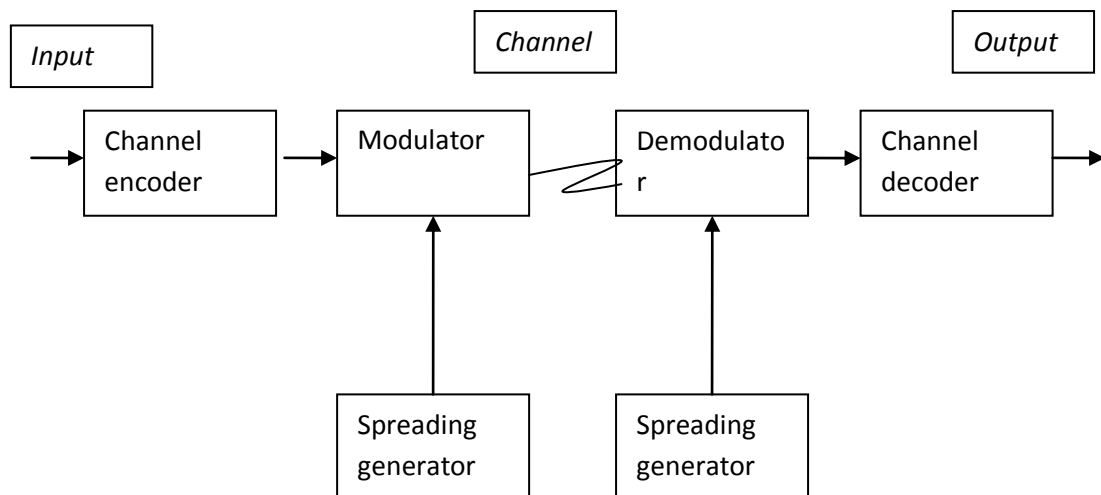


Figure 15: Spread spectrum communication system

A large number of users want to share a common channel to transmit information to a receiver in a multiple access system scenario. As the communication system has a fixed amount of resources, spectrum and channels, the system has to manage resources appropriately as multiple units are trying to access the system at the same time. To solve this problem we can employ three common technologies, frequency-division multiple access (FDMA), time-division multiple access (TDMA), code-division multiple access (CDMA). As both TDMA and FDMA has drawbacks and inefficient for multiple access system limits, we need to find an alternative i.e. by allowing more than one user to share a channel by use of direct sequence spread spectrum signals (DS-SS).

Each user is assigned a unique code sequence that allows the user to spread the information signal across the assigned frequency band. Signals from the various users are separated at the receiver by cross-correlation of the received signal with each of user code sequences. The cross-correlation and the cross-talk inherent in demodulated signals received are minimized as a result of designing the code sequences. The above multiple access method is CDMA, in order to classify a system as SS modulation technique, the transmission bandwidth must be much larger than the information bandwidth and the resulting RF bandwidth must be determined by a function other than the information being sent.

The intentional and unintentional interference and jamming signals are rejected because

they do not contain the spread spectrum key. Only the desired signal which has the key will be seen at the receiver while demodulating the received signal.

4.2 Communication system architecture using RR sequences

A wireless network system consists of several components that support communications by converting information signals into suitable form for transmission through the air medium. Wireless networks include computer devices, base stations and a wireless infrastructure. User is the one who initiates and terminates the information signal in wireless networks and can directly utilize the wireless network.

In our proposed architecture we will be having N users, where each user wants to transmit his own message or information signal securely without being intercepted by other users' signals and only the authentic user will be able to receive his information signal. The first user will be having his message M_1 to transmit, he will mix message M_1 with a random residue sequence $V_1(t)$ before transmission. Same procedure will be followed by the remaining users who wish to transmit their message signal, but each user will be getting a random left or right shift of the original random residue sequence, which each user will be mixing to his message before transmission.

At the receiver end, if a user wants to get back his original message or information signal the he has to mix his own random residue sequence to the message signal received. As it was discussed in the previous chapter that the autocorrelation of the random residue

sequence for all non-diagonal shifts is zero, so only the authentic user who is having a correct RR sequence will be able to retrieve original message signal. Similarly the remaining users can retrieve their original message by using their valid RR sequences. Thus only the authentic users can retrieve their message signal and all intruders cannot get access to the original message which also provides security to the information signal.

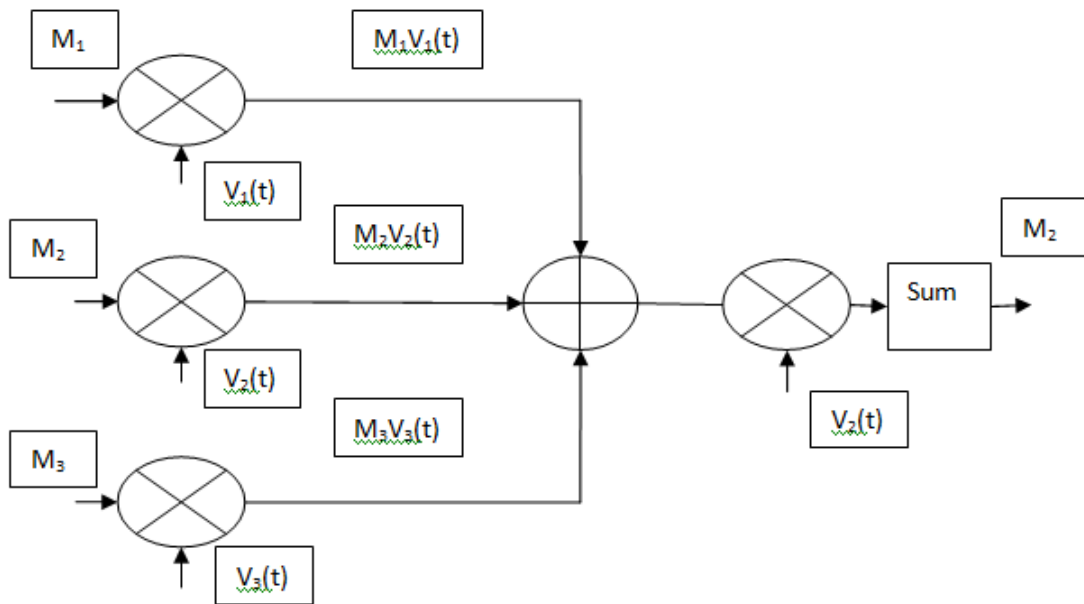


Figure 16: Wireless system communication architecture using random residue sequences

4.3 Implementation of wireless system architecture using a 16 bit RR sequence

Let the 16 bit sequence be $a=11$, $b=2$, $c=4$, $d=8$, $e=16$, $f=32$, $g=17$, $h=34$, $i=21$, $j=42$, $k=37$, $l=27$, $m=7$, $n=14$, $o=28$, $p=9$. In a multiple access system, more than one user wants to communicate through the same common channel. As the users share the same common channel to transmit their message, each user is assigned unique rr sequence i.e.

the user M_1 will be assigned the original sequence 11, 2, 4, 8, 16, 32, 17, 34, 21, 42, 37, 27, 7, 14, 28, 9 the user M_2 sequence might be left shift or right shift of the original sequence 14, 28, 9, 11, 2, 4, 8, 16, 32, 17, 34, 21, 42, 37, 27, 7 and the user M_3 sequence is 4, 8, 16, 32, 17, 34, 21, 42, 37, 27, 7, 14, 28, 9, 11, 2 which is 2 bits left shift of the original sequence. The combined signal along with the user's unique RR sequence will be transmitted and the signal will be separated at the receiver by autocorrelation with each user unique RR sequence.

User M_1 RR sequence: 11, 2, 4, 8, 16, 32, 17, 34, 21, 42, 37, 27, 7, 14, 28, 9

User M_2 RR sequence: 14, 28, 9, 11, 2, 4, 8, 16, 32, 17, 34, 21, 42, 37, 27,

User M_3 RR sequence: 4, 8, 16, 32, 17, 34, 21, 42, 37, 27, 7, 14, 28, 9, 11, 2

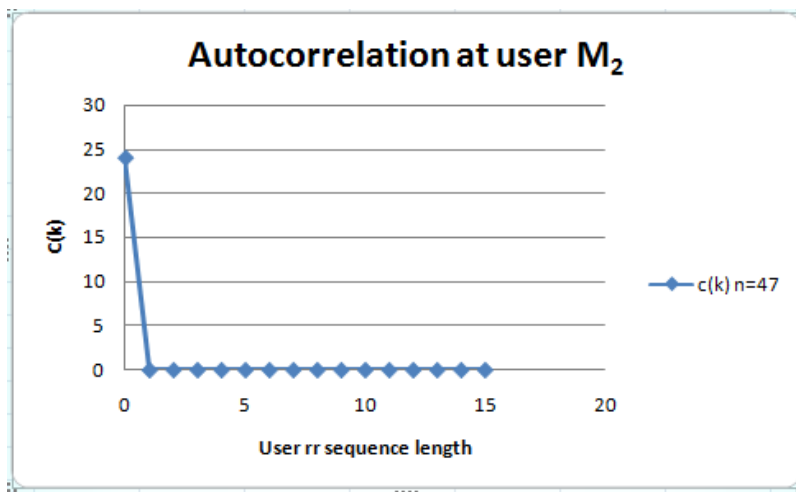


Figure 17: Zero autocorrelation produced when user M_2 produces his sequence at receiver
 As each user will be having his own RR sequence, so the cross talk inherent in the demodulated signals received will be minimized

CHAPTER V

MEMORY CAPACITY OF NEURAL NETWORKS USING CIRCULANT MATRICES

In this chapter, we present the memory capacity of generalized feedback neural networks using a circulant matrix.

5.1 Motivation

Children are capable of learning soon after birth [23]-[25] which indicates that the neural networks of the brain have prior learnt capacity that is a consequence of the regular structures in the brain's organization. The simplest regularity that can be conceived is that of a circulant structure. Therefore, we consider the memory storage behavior and capacity of feedback networks that have circulant structure in the weight matrix.

We wish to determine if circulant matrix based processing might have applications also to larger questions underlying the storage memories. These include non-classical processing [26] and abstract concept formation. Our approach is basically Hebbian. We have shown that these networks can store substantial number of patterns and their shifted versions. The memory capacity of even sized networks is higher than their nearly situated odd sized networks.

5.2 Artificial neurons used

Attempts have been made to reproduce some of the capacities of the human brain using neural networks. Several models have been implemented to exhibit features of human brain, of those models available Hopfield Network model is the simplest and widely used model by the Neural Networks. The potential of Neural Networks relies massively on finite number of artificial neurons connected by edges with variable weights. There will be pattern in which each neuron is connected to other neurons referred to as topology of neural network, based on topology Neural Networks can be classified as Feed-back Neural Networks and Feed-forward Neural Networks. A Hopfield Network is a fully connected feed-back Neural Network.

Hopfield Networks are constructed from artificial neurons which have N inputs. With each input i there will be a weight w_i and also have an output. The state of output is maintained until the neuron is updated. The following operations help us in updating these neurons:

- The value of each input x_i is determined and the sum of all weighted inputs $\sum w_i x_i$ is calculated.
- The output state of the neuron is set to +1 if the weighted input sum is larger or equal to 0. It is set to -1 if the weighted input sum is smaller than 0.
- A neuron can retain its output state until it is updated again.

- When written as a formula:

$$o = \begin{cases} 1: \sum w_i x_i \geq 0 \\ -1: \sum w_i x_i < 0 \end{cases}$$

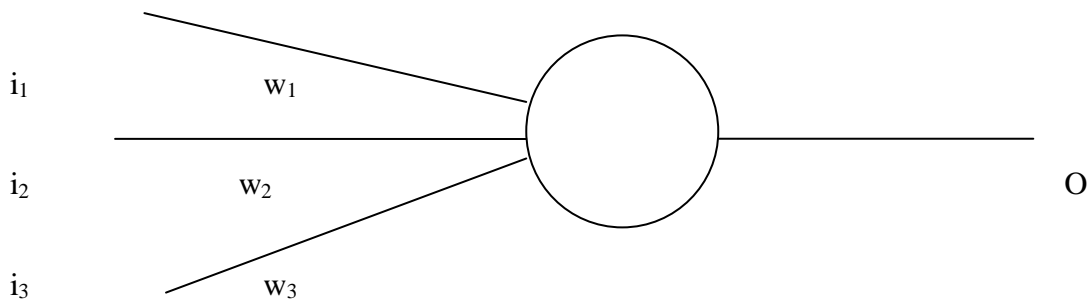


Figure 18: Artificial neuron used in Hopfield network

The weight matrix W is a $N \times N$ symmetric matrix whose components w_{ij} and w_{ji} are same, in other words as the matrix is a symmetric matrix so the values of $w_{ij} = w_{ji}$.

$$\begin{bmatrix} 0 & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & 0 & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & 0 & w_{34} & w_{35} \\ w_{41} & w_{42} & w_{43} & 0 & w_{45} \\ w_{51} & w_{52} & w_{53} & w_{54} & 0 \end{bmatrix}$$

Using the Hopfield Network as a reference we are going to use an $N \times N$ circulant matrix and try to store a new set of neural memories which are relevant to the circulant matrix. The structures of memories stored by different orders of circulant matrices are as follows:

Memories stored by 4×4 circulant matrix:

Let us consider a 4×4 circulant matrix whose elements are 0, a, b, c for experimental purpose. When a random value is assigned to each of the variable then the matrix will be

$$\begin{bmatrix} 0 & 5 & -6 & 3 \\ 3 & 0 & 5 & -6 \\ -6 & 3 & 0 & 5 \\ 5 & -6 & 3 & 0 \end{bmatrix}$$

As the matrix is of order 4×4 the total number of available memories is $2^4 = 16$. The above circulant matrix can hold 6 memories.

$$[+ \ + \ - \ -]$$

$$[- \ + \ + \ -]$$

$$[- \ - \ + \ +]$$

$$[+ \ - \ - \ +]$$

$$[- \ - \ - \ -]$$

$$[+ \ + \ + \ +]$$

Memories stored by 5x5 circulant matrix:

Consider a 5x5 circulant matrix whose elements are 0, a, b, c, d. where a=-2 b=3 c=3 and d=-2.

$$\begin{bmatrix} 0 & -2 & 3 & 3 & -2 \\ -2 & 0 & -2 & 3 & 3 \\ 3 & -2 & 0 & -2 & 3 \\ 3 & 3 & -2 & 0 & -2 \\ -2 & 3 & 3 & -2 & 0 \end{bmatrix}$$

As the matrix is of order 5x5 the total number of available memories is $2^5 = 32$. The above matrix can hold 7 memories. In general a 5x5 circulant matrix can hold two classes of memories. One of them is ++++- & its shifts, the other is +- -++ & its shifts.

$$[+ + + + +][+ - - + +]$$

$$[+ + - + -][+ + - - +]$$

$$[- + + - +][+ + + - -][+ - + + -][- + + + -]$$

$$[- + - + +][- - + + +][+ - + - +]$$

$$[- - - - -]$$

5.3 GENERAL STRUCTURE OF MEMORIES GENERATED

For the 4x4 circulant case, the memories are basically ++ - - & its circular shifts.

1. For the 5x5 case, most of the memories are shifts of ++ - + - & ++++

2. For the 6×6 case, the situation is more interesting. You have two classes; ++ - +- & + - - + - and their complements.
3. For the 7×7 case, we have the all +, the all - sequences and shifts of +++ - +- -.
4. For the 8×8 case, we have three classes +++-++++- , +++-+-+-- & +-+-+--+ and their complements.
5. For the 9×9 case, we have all the sequences as shifts of +++++-++++-.
6. For the 10×10 case, we have three classes +++++-++++- , +-----+---- & +-+-+--+ their shifts and complements.
7. For the 11×11 case, we have all the sequences as shifts of +++++-++++-.
8. For the 12×12 case, we have four classes +++++-++++- , +-+-+--+ , +-----+-----, +-+-+--+ their shifts and complements.
9. For the 13×13 case, we have all the sequences as shifts of +++++-++++-.
11. For the 14×14 case, we have four classes +++++-++++- , +-+-+--+ , +-----+-----, +-----+----- their shifts and complements.
12. For the 15×15 case, we have all the sequences as shifts of +++++-++++-.
13. For the 16×16 case, we have seven classes +++++-++++- , +-+-+--+ , +- , +-----+----- , +-----+----- , +-----+----- , +-----+----- , +-----+----- their shifts and complements.
14. For the 17×17 case, we have all sequences as shifts of +++++-++++-.
15. For the 18×18 case, we have six classes +++++-++++- , +-+-+--+

+-+ +- , +-+-----+-+-----, ++++-----++++-----, ++++-----++++-----, ++++-----
 ++++----- their shifts and complements.

16. For the 19×19 case, we have all sequences as shifts of ++++++-----+-----.

17. For the 20×20 case, we have eight classes ++++++-----+-----, +-+-----

+-----, +++++-----++++-----, +-+-----+-----+-----+-----, +-----+-----+-----+-----,
 +-----+-----+-----, +-----+-----+-----+-----, +-----+-----+----- their shifts
 and complements.

18. For the 21×21 case, we have all sequences as shifts of ++++++-----

+-----.

19. For the 22×22 case, we have six classes ++++++-----+-----, +-+-----

-+-----+-----, +-+-----+-----, +-----+-----+-----, +-----+-----+-----
 -----, +-----+-----+----- their shifts and complements.

20. For the 23×23 case, we have all sequences as shifts of ++++++-----

+-----.

21. For the 24×24 case, we have ten classes ++++++-----+-----, +-+-----

-----+-----, +-----+-----+-----, +-----+-----+-----, +-----+-----
 +-----+-----, +-----+-----+-----+-----, +-----+-----+-----+-----, +-+-----+-----
 +-+-----+-----+-----, +-----+-----+-----+----- their shifts and complements.

22. For the 25×25 case, we have all sequences as shifts of ++++++-----

-+-----+-----

The above results when plotted on a graph with first instance as memories held by the $N \times N$ circulant matrices of order up to 25×25 .

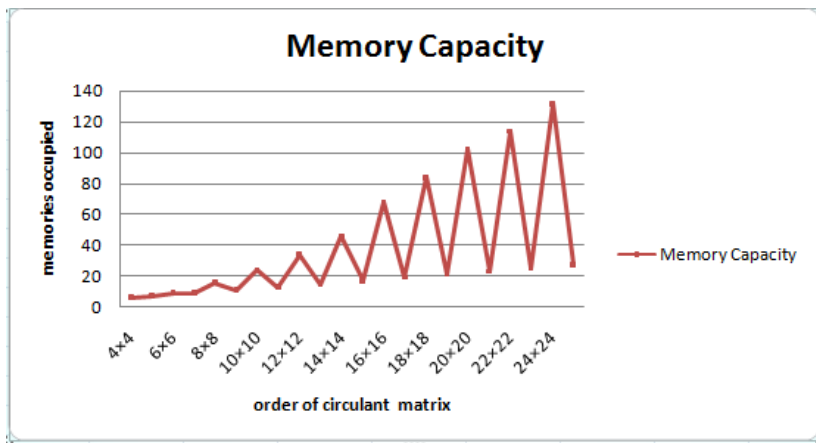


Figure 19: Memories occupied by each circulant matrix

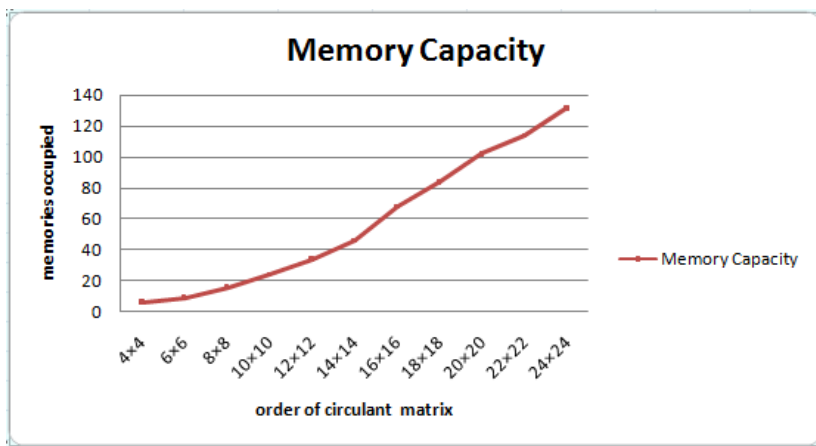


Figure 20: Memories occupied by even order circulant matrices up to 24×24

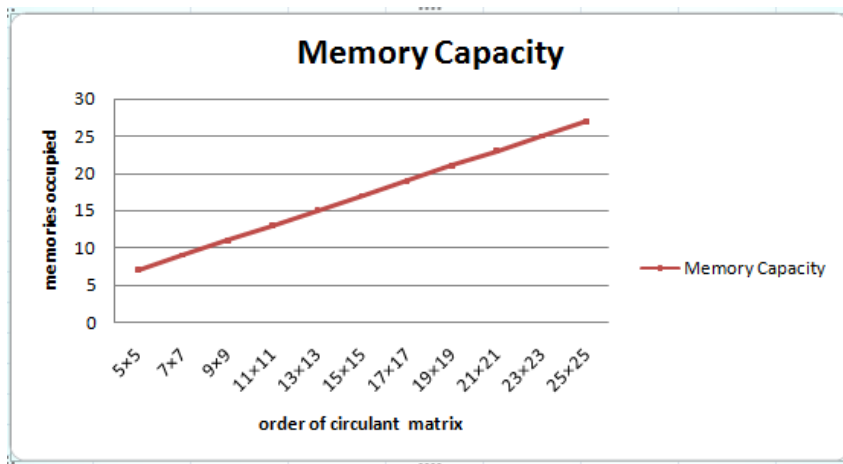


Figure 21: Memories occupied by odd order circulant matrices up to 25×25

The above graphs explain the memories held by each of the matrix out of the total outcomes and it shows that how efficiently a circulant matrix can hold memories.

We presented different classes of memories stored by circulant matrices of different order. Other interesting thing is that the memories stored by a particular circulant matrix are also in circulant nature when shifted by one memory bit at a time. If quantum processing were to be occurring in neural structures there would be further need to examine probability constraints on such processes [72]-[76].

CHAPTER VI

CONCLUSION

In this thesis, we have obtained new results in number theoretic Hilbert transform (NHT) theory by proposing an algorithm to generate valid NHT matrices. NHT matrices of various lengths up to 16 have been found. This allowed us to find ideal orthogonal sequences that can be used as carriers for wireless communications and sequences of lengths up to 24 have been computed. We proposed an architecture that uses random residue sequences as unique code sequence assigned to each user in multiple access system. We also investigated applications of circulant matrices to store and reproduce certain patterns as neural memories.

Our proposed NHT algorithm does not work for all values and, therefore, it must be further generalized. Likewise, the problem of generation of memories and their properties needs to be further investigated.

REFERENCES

- [1] P. Schreier, L. Scharf, Statistical signal processing of complex-valued data: the theory of improper and noncircular signals. Cambridge University Press (2010).
- [2] S. Kak and N.S. Jayant, Speech encryption using waveform scrambling. Bell System Technical Journal, vol. 56, pp. 781-808, May-June 1977.
- [3] N.S. Jayant and S. Kak, Uniform permutation privacy system, US Patent No. 4,100,374, July 11(1978).
- [4] S.K. Padala and K.M.M Prabhu, Systolic arrays for the discrete Hilbert transform. Circuits, Devices and Systems, IEE Proceedings, vol.144, 259-264 (1997).
- [5] J.M. Pollard, Implementation of Number Theoretic Transforms, Electron Letters 12,378-379 (1976).
- [6] R.C. Agarwal and C.S. Burrus, Number Theoretic Transforms to implement Fast Digital Convolution, Proc. IEEE 63 (1975).
- [7] S. Kak, The discrete Hilbert transform. Proc. IEEE 58, 585-586 (1970)
- [8] S. Kak, Hilbert transformation for discrete data. International Journal of Electronics, vol. 34, pp. 1385-1390, 1977.
- [9] S. Kak, Multilayered array computing. Information Sciences 45, 347-365 (1988)
- [10] S. Kak, The discrete finite Hilbert transform. Indian Journal Pure and Applied Mathematics 8, 1385-1390 (1977)

- [11] S. Kak, A two-layered mesh array for matrix multiplication. *Parallel Computing* 6, 383-385 (1988)
- [12] S. Kak, The number theoretic Hilbert transform. *Circuits, Systems and Signal Processing*, 2014; arXiv:1308.1688
- [13] S. Kak, Properties of NHT-Circulant Matrices.
http://www.cs.okstate.edu/~subhashk/nhtcirculant_properties.pdf
- [14] P.J. Davis, *Circulant Matrices*. Wiley, New York (1970)
- [15] R. Kandregula, The basic discrete Hilbert transform with an information hiding application. 2009. arXiv:0907.4176
- [16] V.K. Kotagiri, The 10-point and 12-point number theoretic Hilbert transform. arXiv:1310.3221
- [17] V.K. Kotagiri, New results on the number theoretic Hilbert transform. arXiv:1310.6924
- [18] S. Golomb. *Shift Register Sequences*. San Francisco, Holden-Day, 1967
- [19] L.B Milstein and M.K Simon, *Spread Spectrum communications*, CRC Press, 1999.
- [20] G.H. Hardy and E.M Wright, *An Introduction to the Theory of Numbers*. Oxford University Press, 1954.
- [21] S. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 2008.
- [22] D.O. Hebb, *The Organization of Behavior*. Wiley 1949.

- [23] D. Perani et al, Neural language networks at birth. *Proc. Natl. Acad. Sci. U.S.A.*, 108:16056-16061, 2011.
- [24] R.C. Berwick, A.D. Friederici, N. Chomsky, J.J. Bolhuis, Evolution, brain, and the nature of language. *Trends in Cognitive Sciences*, 17:89-98, 2013
- [25] P. Fransson et al. The functional architecture of the infant brain as revealed by resting-statefMRI. *Cereb Cortex* 21:145-154, 2010.
- [26] R.F. Thompson, In search of memory traces. *Annu. Rev. Psychol.* 56, pp. 1-23, 2005.
- [27] S. Kak, Artificial and biological intelligence. *ACM Ubiquity* vol. 4, no. 42, 2005; arXiv:cs/0601052
- [28] D. Prados and S. Kak, Non-binary neural networks. *Lecture Notes in Computing and Control*, vol. 130, pp. 97-104, 1989.
- [29] S. Kak, Feedback neural networks: new characteristics and a generalization. *Circuits, Systems, and Signal Processing*, vol. 12, pp. 263-278, 1993.
- [30] M.C. Stinson and S. Kak, Bicameral neural computing. *Lecture Notes in Computing and Control*, vol. 130, pp. 85-96, 1989.
- [31] S. Kak, The three languages of the brain: quantum, reorganizational and associative. In: K. Pribram, J. King (Eds.), *Learning as Self-Organization*, Lawrence Erlbaum, London, 1996, pp. 185-219.

- [32] J.J. Hopfield, Neural networks and physical systems with emergent collective computational properties. Proc. Nat. Acad. Sci. (USA), vol. 79, pp. 2554-2558, 1982.
- [33] M. Lowe, On the storage capacity of Hopfield models with correlated patterns. Ann. Appl. Probab. 8: 975-1349, 1998.
- [34] R.Q. Quiroga, L. Reddy, G. Kreiman, C. Koch and I. Fried, Invariant visual representation by single neurons in the human brain. Nature 435, pp. 1102-1107, 2005.
- [35] S. Kak and J.F. Pastor, Neural networks and methods for training neural networks, US Patent 5,426,721, 1995.
- [36] L. Lin, R. Osan and J.Z. Tsien, Organizing principles of real-time memory encoding: neural clique assemblies and universal neural codes. Trends in Neuroscience, vol. 29, pp. 48-57, 2006.
- [37] S. Kak, New algorithms for training feed forward neural networks. Pattern Recognition Letters, vol. 15, pp. 295-298, 1994.
- [38] K.W. Tang and S. Kak, Fast classification networks for signal processing. Circuits, Systems, Signal Processing, vol. 21, pp. 207-224, 2002.
- [39] G. Bi and M. Poo, Synaptic modification by correlated activity: Hebb's postulate revisited. Annu. Rev. Neurosci. 24, pp. 139-166, 2001.
- [40] W.R. Softky, Simple codes versus efficient codes. Curr. Opin. Neurobiol. 5, pp. 239-247, 1995.

- [41] S.R. Schweinberger, E.C. Pickering, I. Jentzsch, M. Burton and J.M. Kaufmann, Event-related brain potential evidence for a response of inferior temporal cortex of familiar face repetitions, *Cognitive Brain Research*, vol. 14, pp. 398-409, 2002.
- [42] E.T. Rolls, The representation of information about faces in the temporal and frontal lobes, *Neuropsychologia*, vol. 45, pp. 124-143, 2007.
- [43] L. Pessoa, To what extent are emotional visual stimuli processed without attention and awareness? , *Current Opinions in Neurobiology*, vol. 15, pp. 188-196, 2006.
- [44] S Kak, A Chatterjee, On decimal sequences. *IEEE Transactions on Information Theory* IT-27: 647-652,1981.
- [45] S. Kak, Encryption and error correction coding using D sequences. *IEEE Transactions on Computers* C-34: 803-809, 1985
- [46] N. Mandhani and S. Kak, Watermarking using decimal sequences. *Cryptologia*, vol. 29, pp. 50-58, 2005.
- [47] S. Kak, Classification of random binary sequences using Walsh-Fourier analysis. *IEEE Trans. on EMC*, vol. EMC-13, pp. 74-77, August 1970.
- [48] S. Kak, Are quantum computing models realistic? *ACM Ubiquity*, vol. 7, no. 11, pp. 1-9, 2006; arXiv:quant-ph/0110040
- [49] S. Kak, Artificial and biological intelligence. *ACM Ubiquity* vol. 4, no. 42, 2005; arXiv:cs/0601052
- [50] S. Kak, Can we define levels of artificial intelligence? *Journal of Intelligent Systems*, vol. 6, pp. 133-144, 1996.

- [51] S. Kak, Active agents, intelligence, and quantum computing. *Information Sciences*, vol. 128, pp. 1-17, 2000.
- [52] U. Neisser and I. Hyman, *Memory Observed: Remembering in Natural Contexts*. Worth Publishing, 1999.
- [53] G. Orbán, J. Fiser, R.N. Aslin, and M. Lengyel, Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences USA*, vol. 105, pp. 2745-2750, 2008.
- [54] J. Fiser and R.N. Aslin, Encoding multielement scenes: statistical learning of visual feature hierarchies. *J Exp Psychol Gen*, vol. 134, 521–37, 2005.
- [55] M. Lengyel and P. Dayan, Rate-and phase-coded autoassociative memory. *Advances in Neural Information Processing Systems*, vol. 17, 769-776, 2005.
- [56] M.A. Bobes, I. Quinonez, J. Perez, I. Leon, and M. Valdes-Sosa, Brain potentials reflect access to visual and emotional memories for faces. *Biological Psychology*, vol. 75, pp. 146-153, 2007.
- [57] S. Kak, *The Nature of Physical Reality*. Peter Lang, OSU, 2011.
- [58] S. Kak, *The Architecture of Knowledge*. CSC, 2004.
- [59] C. Ranganath and G. Rainer, Neural mechanisms for detecting and remembering novel events, *Nature Reviews. Neuroscience*, vol. 4, pp. 193–202, 2003.
- [60] J.J. Hopfield, Neural networks and physical systems with emergent collective computational properties. *Proc. Nat. Acad. Sci. (USA)*, vol. 79, pp. 2554-2558, 1982.

- [61] S. Kak and M.C. Stinson, A bicameral neural network where information can be indexed. *Electronics Letters*, vol. 25, pp. 203-205, 1989.
- [62] D.L. Prados and S. Kak, Neural network capacity using the delta rule. *Electronics Letters*, vol. 25, pp. 197-199, 1989.
- [63] S. Kak, Self-indexing of neural memories. *Physics Letters A*, vol. 143, pp. 293-296, 1990.
- [64] M. Stowe and S. Kak, Neural network capacity for multilevel inputs.
arXiv:1307.8104
- [65] S. Kak, On training feedforward neural networks. *Pramana*, vol. 40, pp. 35-42, 1993.
- [66] K.-W. Tang and S. Kak, A new corner classification approach to neural network training. *Circuits, Systems, and Signal Processing*, vol. 17, pp. 459-469, 1998.
- [67] S. Kak, On generalization by neural networks. *Information Sciences*, vol. 111, pp. 293-302, 1998.
- [68] S. Kak, A class of instantaneously trained neural networks. *Information Sciences*, vol. 148, pp. 97-102, 2002.
- [69] D. Perani et al, Neural language networks at birth. *Proc. Natl. Acad. Sci. U.S.A.*, 108: 16056–16061, 2011.
- [70] R.C. Berwick, A. D. Friederici, N. Chomsky, J. J. Bolhuis, Evolution, brain, and the nature of language. *Trends in Cognitive Sciences*, 17: 89-98, 2013

- [71] F. Homae et al. Development of global cortical networks in early infancy. *J Neurosci* 30:4877–4882, 2010.
- [72] A. Gautam and S. Kak, Symbols, meaning, and origins of mind. *Biosemiotics*, vol. 6, number 3, pp. 301-309, 2013.
- [73] S. Kak, Probability constraints and the classical/quantum divide. *NeuroQuantology*, vol. 11, pp. 600-606, 2013.
- [74] S. Kak, Biological memories and agents as quantum collectives. *NeuroQuantology*, vol. 11, pp. 391-398, 2013.
- [75] S. Kak, Information and learning in neural networks. *NeuroQuantology*, vol. 9, pp. 393-401, 2011.
- [76] S. Kak, From the no-signaling theorem to veiled non-locality. *NeuroQuantology*, vol. 12, pp. 12-20, 2014.

VITA

Vamsi Sashank Kotagiri

Candidate for the Degree of

Master of Science

Thesis: Information Processing Using Circulant Matrices

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in May, 2014.

Completed the requirements for the Bachelor of Science in Computer Science and Engineering at GITAM University, Visakhapatnam, India in May, 2012.

Experience:

Graduate Research Assistant, Department of Computer Science Sept 2013– May 2014
Oklahoma State University: Stillwater, OK

Software Engineer Intern, Symbiosis Technologies May 2011– January 2012
Visakhapatnam, AP, India