Near Field Communication Simulation & Identification

Collection Editor:

Kiran Pathakota

Near Field Communication Simulation & Identification

Collection Editor:

Kiran Pathakota

Authors:

Ayana Andalcio
Paul Melvin
Kiran Pathakota

Online:

< http://cnx.org/content/col11398/1.1/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: December 19, 2011
PDF generated: December 19, 2011
For copyright and attribution information for the modules contained in this collection, see p. 21.

Table of Contents

Chapter 1

Introduction[1]

Near field communication (NFC) is a new technology that allows for wireless, contactless communication between devices. It works by utilizing magnetic coupling between devices. There can either be one active device and one passive device or two active devices.

NFC technology is currently being put into cell phones. There is excellent commercial potential for the technology. For example, it can allow for people to simply tap their phones to pay for their purchases instead of pulling out a card. Also, one could tap their phone to an NFC enabled poster to obtain further information about the advertisement or to purchase the advertised item. Another example of possible NFC use is the ability to sit down at a restaurant, tap your phone on the table, pull up the menu, and order from your phone.

One of the primary concerns with all of these examples is that it is crucial that the device that is in contact with the phone is able to accurately identify the phone. Even if the active device correctly identifies the phone 99% of the time, this would be unacceptable because that 1% of error represents instances in which some random unlucky individual is charged for something that someone else bought.

Our goal was to accurately model an NFC system and then develop an algorithm that accurately identifies a passive device.

[1] This content is available online at <http://cnx.org/content/m42010/1.1/>.

Chapter 2

Principle Components[1]

2.1 About the components

An NFC system consists of a passive network (the card/circuit with information) and an active circuit. We must also take into account the environment surrounding the device. A working knowledge of these three componenents is essential to understanding the inner mechanics of near field communication.

2.1.1 The Active Circuit

The active circuit has a transmitting antenna that broadcasts RF waves and a number of electrodes that read values corresponding to how the environment reacts to the RF waves. All the processing power is centered in this active circuit and that is where the decoding will take place. This would be considered the reader in the traditional sense. In the following mathematical discussion, the electrode voltages of the active circuit will be values in a vector **v1.**

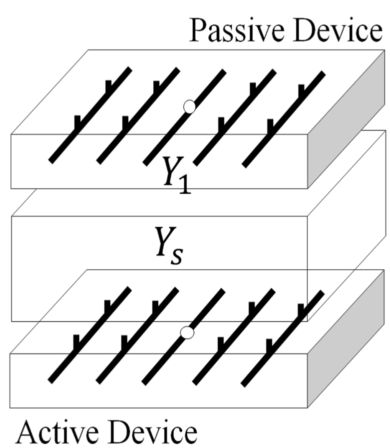

Figure 2.1

[1]This content is available online at <http://cnx.org/content/m42015/1.1/>.

2.1.2 The Passive Network

The passive network consists of a series of electrodes which are attached to impedances with values of infinity or zero (open/closed switches). This network might be located inside a phone or other device and is associated with the client side identification. The configuration of the passive network uniquely identifies the user, and thus, our problem becomes one of reverse engineering in which we try to determine the impedances. In our analysis, the passive network voltages are stored in the vector **v1** and the impedances are represented by the matrix **Y1.** Since we are assuming no mutual impedances, this is a diagonal matrix. Rather than using infinity, we have represented the open switches with the very high impedance value of 10,000 ohms.

2.1.3 The Environment

The "evil channel" is our environment. It converts our beautiful binary impedances into ugly complex voltages. It is our job to understand its effects and make sure that our output is decipherable. The channel matrix we used was given to us and was measured experimentally. It can be modeled by a large square matrix which is of size N x N where N is the total number of voltages (in this case 41). The matrix is labeled **Ys** and is divided into 9 separate matrices, each representing the mutual interaction of different elements in our matrix.

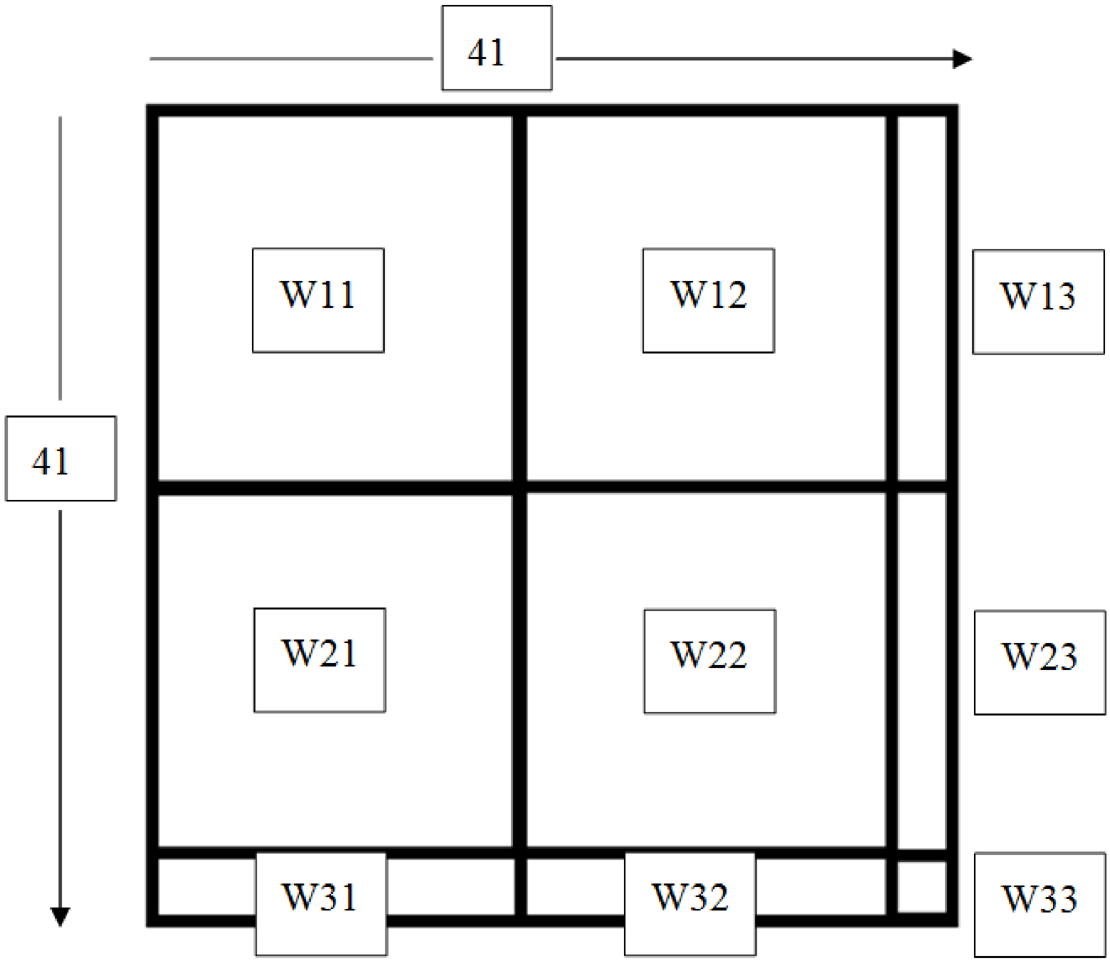

Figure 2.2

For example, the elements in matrix W12 would reperesent how the electrodes with voltages **v1** interact with the electrodes with voltages **v2.** Consequently, our answer lies in simplifying this matrix and trying to nullify it's effect on the input (the **Y1** matrix).

2.2 Problem Analysis

As previously stated, the relationship between input impedences and output voltages is non-linear by nature. Thus, our task will be to find some underlying patterns between input and output so that we may adequately decode the voltage values. When presented with a problem as open-ended as this, a number of techniques can be employed. We must make a preliminary analysis of the data through the use of well-constructed plots and principle component analysis.

Chapter 3

Simulation[1]

3.1 Code to produce v1

This code takes the input of the Y1 matrix and the initial impedance to produce a vector of v1 values, using equations discovered in previous research. This code also produces v2 as well as yin which is not needed for our purposes.

```
    %function to produce values for v1 given a Y1 and S0
function v1 = nfc(Y1,S0)
n  = 41;
Z0 = 50;
A = S0;
S = zeros(n,n);

for k = 1:n,
    for l = 1:n,
        S(k,l)=A(k,2*l-1)*cos(A(k,2*l)*pi/180)+i*A(k,2*l-1)*sin(A(k,2*l)*pi/180);
    end
end

Zparam = Z0*inv(eye(n)-S)*(eye(n)+S); % Zparameters of n-port system
Yparam = inv(Zparam);
W = Yparam; % Ys matrix

m = 20; % Number of receivers
r = n-1; % Maximum n-1

W11 = W(1:m,1:m);
W12 = W(1:m,(m+1):r);
W13 = W(1:m,r+1);
W21 = W((m+1):r,1:m);
W22 = W((m+1):r,(m+1):r);
W23 = W((m+1):r,r+1);
W31 = W(r+1,1:m);
W32 = W(r+1,(m+1):r);
W33 = W(r+1,r+1);
```

[1] This content is available online at <http://cnx.org/content/m42017/1.1/>.

```
Yc = inv(W22 - W21*inv(W11)*W12 + Y1); % Complex Y matrix
Y = real(Yc); % Matrix M in paper
YY = imag(Yc); % Matrix N in paper
P = W22-W21*inv(W11)*W12; %*****W11+Y1 changed to W11

P1 = W31*inv(W11)*W12-W32; %*****W11+Y1 changed to W11

M = real(P);
M = 0.5*(M+M');

v2 = P1*(Y+YY*i); % Voltages as input to Y1
v1 = -v2*W21*inv(W11)-W31*inv(W11); % Voltages not connected to Y1 %*****W11+Y1 changed to W11

yin = v1*W13+v2*W23+W33;
return
```

Using this code, we could feed the function a variety of Y1 values to try and determine a pattern between the Y1 and the v1. This way, we were able to modify Y1 in whichever fashion we liked to see if there were visible patterns that could be seen within the v1 values. The code that calls this function then uses the value of v1 and Y1 to plot graphs and look at specific values.

Chapter 4

Results[1]

4.1 Direct Relationship

Our first step was to generate a Y1 with random ones and zeros and see if there is any direct relationship to v1. The following demonstrates the code used to produce random binary values (impedances for Y1), while Fig 1 shows the output of the comparison.

```
    for c = 1:m,
    if rand > 0.5,
        y1(c) = one;
    else
        y1(c) = 0;
    end
end

Y1 = diag(y1);
v1 = nfc(Y1, S0);

subplot(2,1,1)
stem(1:20, y1, 'r')
title('Magnitude of Y1')

subplot(2,1,2)
stem(1:20, abs(v1), 'b')
title('Magnitude of v1')
```

[1] This content is available online at <http://cnx.org/content/m42016/1.1/>.

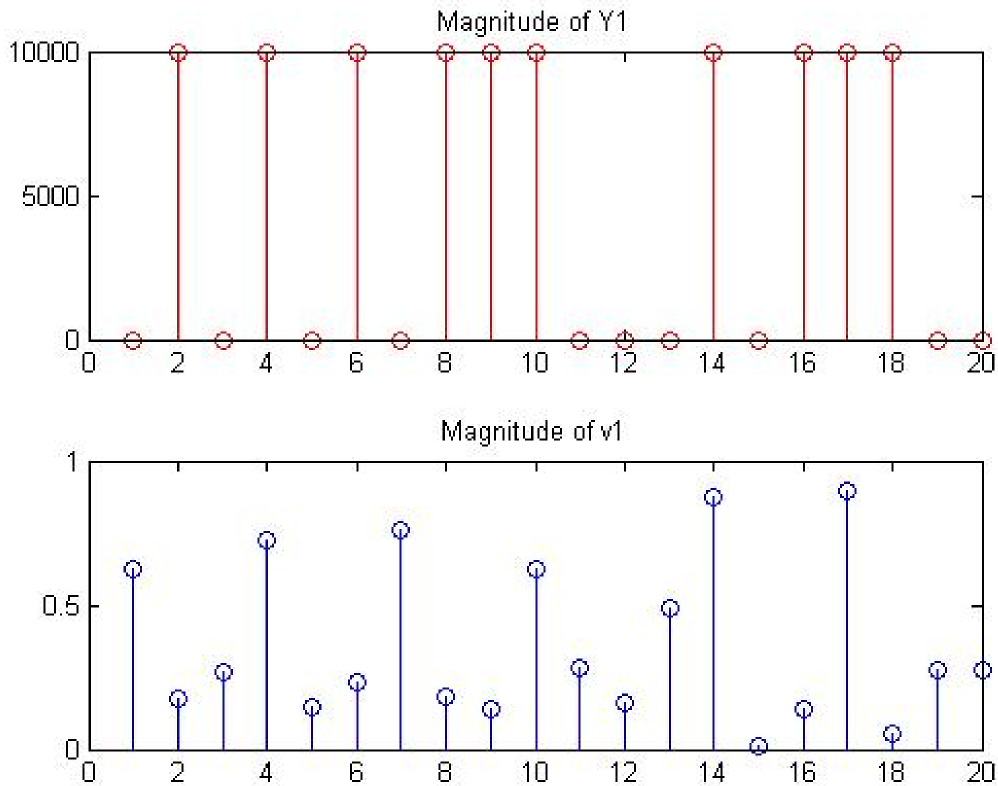

Figure 4.1: The diagonal of Y1 and the corresponding v1 values.

From these graphs, there was no obvious correlation between the "0's" and "1's" in Y1 and the values of v1. From here, a linear relationship was tested.

4.2 Testing for Linearity

To determine if there was a linearity in computing the values of v1, we decided to compute the values of v1 for a Y1 matrix that had a diagonal of 11000000000000000000. As such, we first computed a set of v1 where there was a single 1 in the first port of Y1, and then a second set of v1 where there was a single 1 in the second port of Y1, and then added those two sets of values together. We were hoping to see if the result would produce a set of values of v1 that is identical to or similar to that which would be produced by a matrix with a diagonal of two 1's followed by 0's. The results of this test are shown in Fig 2 and Fig 3.

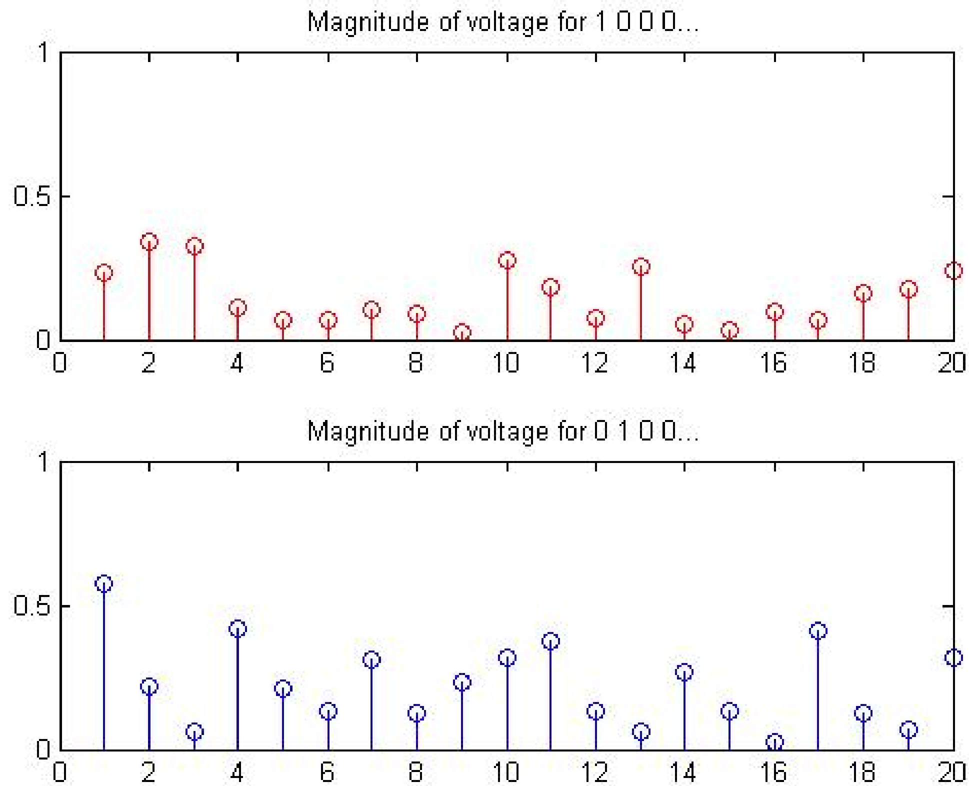

Figure 4.2: The separate v1 values.

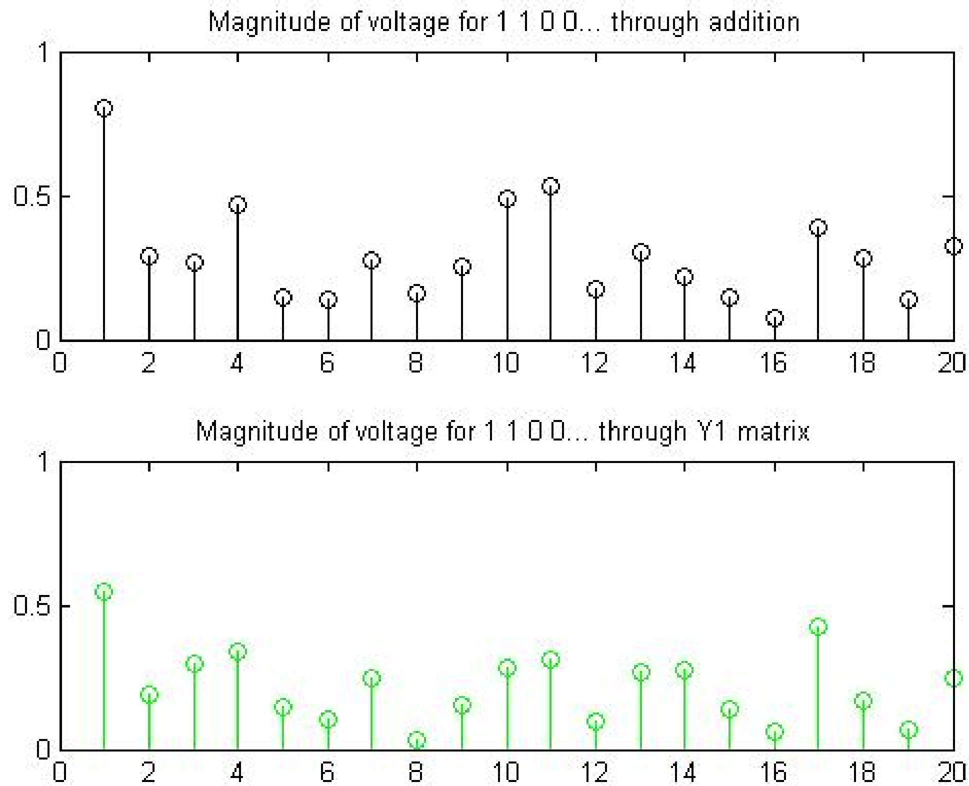

Figure 4.3: Comparison for linearity.

Looking at Fig 3. shows that the methods produce similar looking graphs, but each port has a slightly different magnitude. From these results we were able to conclude that the relationship between Y1 and v1 was non-linear.

4.3 Fourier Transform

Our next approach was to look at the Fourier transform of several different patterns, to see if a Fourier transform of the values of v1 could provide any insight into a relationship. The following set of graphs shows the values of the diagonal of the Y1 matrix, and the corresponding v1 values computed from it.

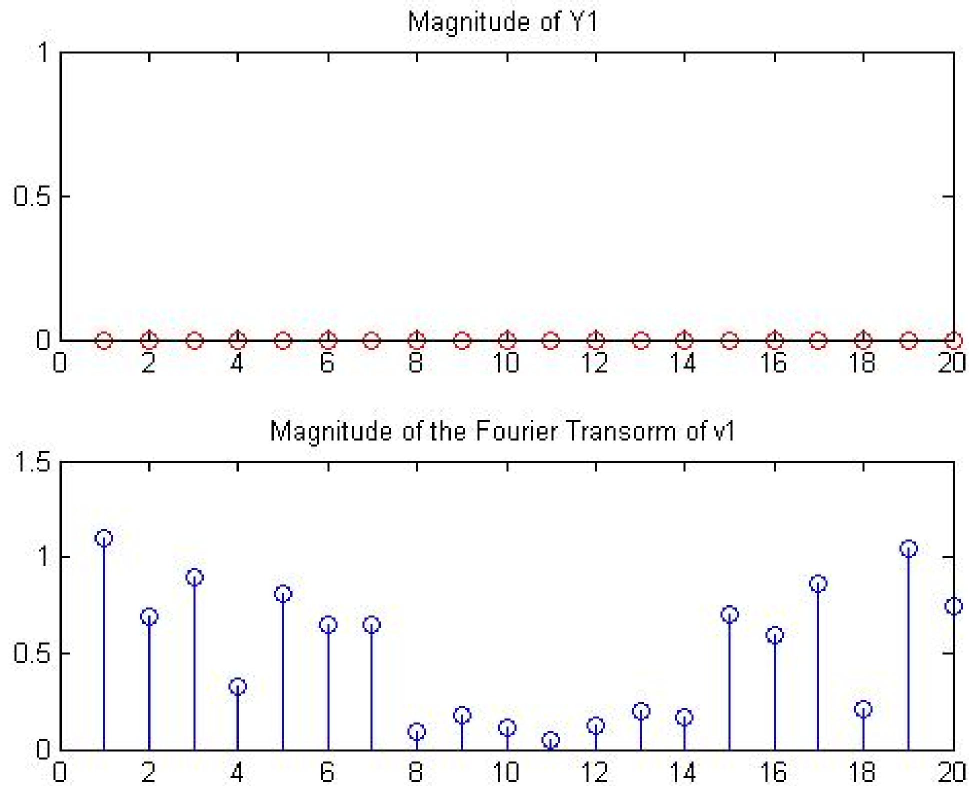

Figure 4.4: Pattern 1 and the Fourier transform of the corresponding v1 values.

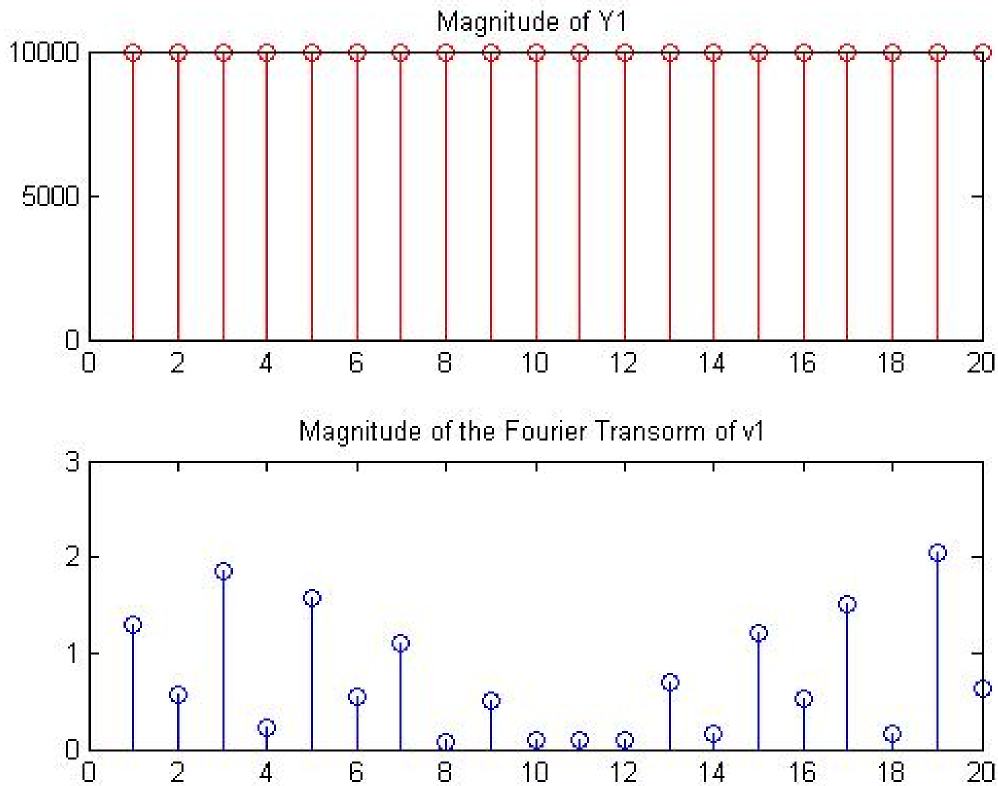

Figure 4.5: Pattern 2 and the Fourier transform of the corresponding v1 values.

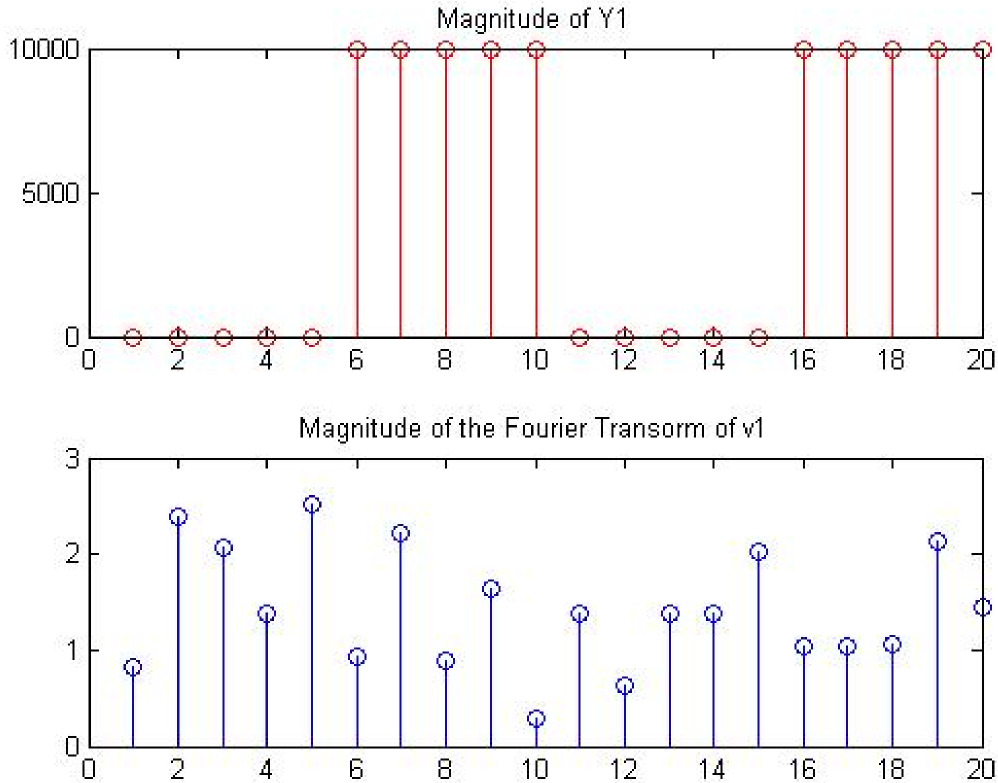

Figure 4.6: Pattern 3 and the Fourier transform of the corresponding v1 values.

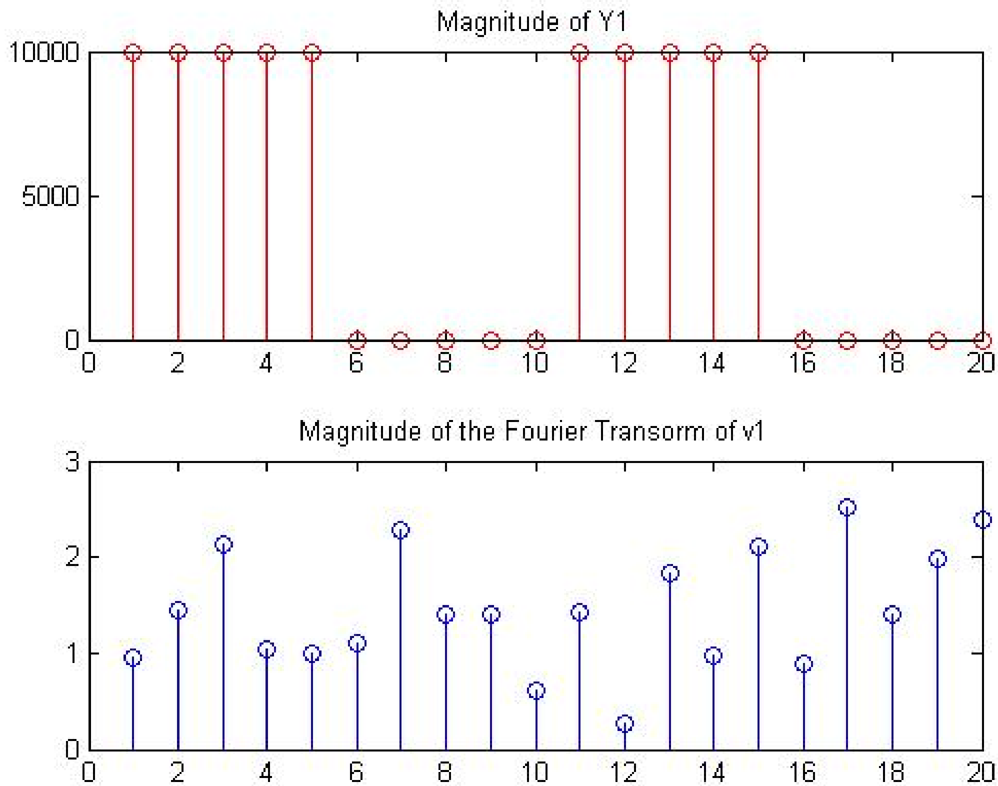

Figure 4.7: Pattern 4 and the Fourier transform of the corresponding v1 values.

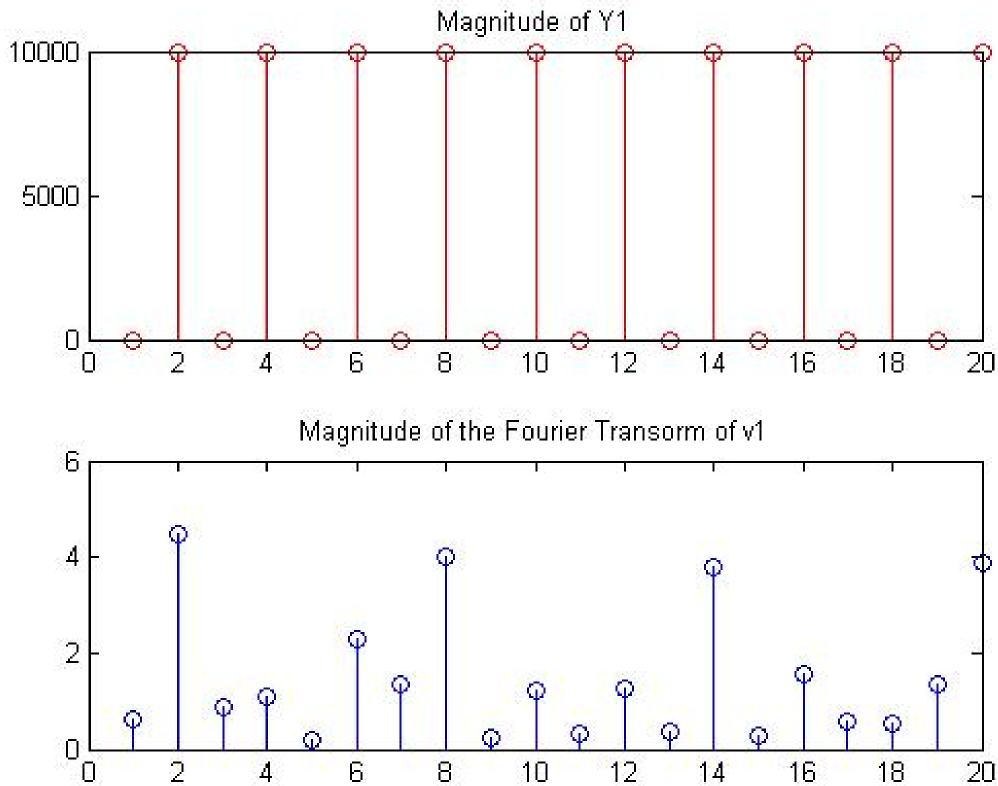

Figure 4.8: Pattern 5 and the Fourier transform of the corresponding v1 values.

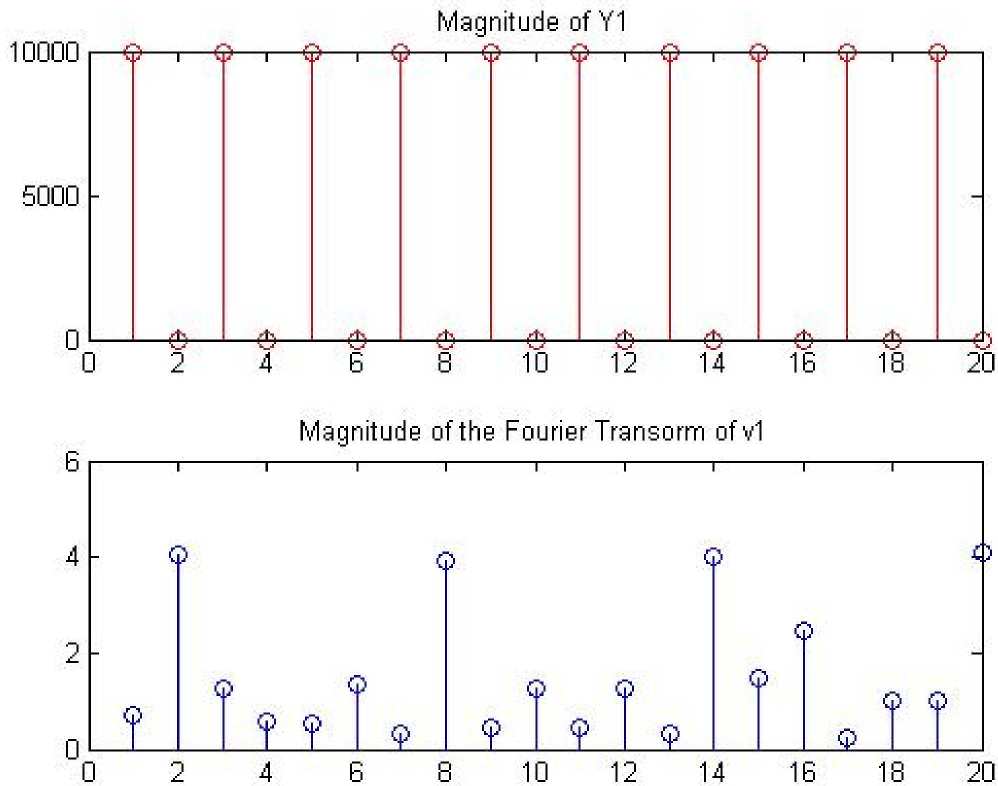

Figure 4.9: Pattern 6 and the Fourier transform of the corresponding v1 values.

From these graphs, once again, there is no evidence that there is any relationship between any of the patterns and the values of v1, nor any relationship between Y1 and the Fourier transform of the values in v1.

Chapter 5

Conclusion[1]

We were able to successfully model an NFC system. It gave us accurate outputs for whatever input we chose to give it. However, due to the nonlinearity of the system, we were unable to find an algorithm that sufficiently accurately could identify a passive device by looking at the voltages produced at the output.

We did determine that there was a way to accurately identify a device given the limitation of our system. We chose not to implement this method because it does not scale well and it would require that any active device receive the same signal from one passive device regardless of the environment that the devices are in. This method would require the active device to have all possible received signals store in their memories so that for every unique device, it would recognize the unique signal it produces.

Instead of implementing this method, we tried to identify patterns in the output signals. The patterns that we were looking for, however, simply weren't there. With more time, there is a possibility that we might find a useful pattern if we were to make our input impedances continuous instead of the 0's and infinities that we were experimenting with.

While our lack of a concrete algorithm is disappointing, the robustness of our model is promising. It is possible that with further analysis, we might find the type of pattern that we are looking for to help us create a working algorithm.

[1] This content is available online at <http://cnx.org/content/m42014/1.1/>.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Near Field Communication Simulation & Identification
This is a report of our exploration of the mathematics of Near Field Communication. In this report, we set up the problem, explore various solutions and then come to an interesting conclusion. This project was for our Elec 301 class at Rice University.