# Modeling Signal Data of Unknown Source from Antenna 443A

Jeremiah Simmons

April 23, 2016

# 1 Abstract

On 21 March 2016 signals of unknown source were received by Antenna 443A. Signals from an unknown source at data points arrived every .2 milliseconds. This collection of data has been analyzed and synthesized in order for precise predictions to allow other antenna sites to compare the measurements. Applying pattern analytics using a linear least square fitting method achieved a "goodness of fit" of 6.7781e-6 and are able to predict signals received outside the known interval. At .2 ms the predicted signal strength is .7100 +/- .0050 mW, and at 14 ms, .9959 +/- 8.18e-5 mW.

# 2 Introduction

The methodology used was linear least square fitting. Linear least square fitting is the mathematical procedure for finding a best fit curve along a given set of points while minimizing the distance from the points from the curve. Essentially the process allows for a best fit or goodness of fit to be established. Using an extended variation of the linear model:

$$F(x) = ax + b \tag{1}$$

We can derive a similar model for a systems of equations:

$$\begin{gathered} y^1 = ax^1 + b; \\ y^2 = ax^2 + b; \\ y^n = ax^n + b \end{gathered}$$

or

$$Y = AX \tag{2}$$

where

$$Y = [y^1; y^2; y^n], A = [1x^-1; 1x^-2; 1x^-n], X = [a; b] \tag{3}$$

Using the model of equation (3) and the data gathered we assume the best fit equation has a model such as:

$$Y(x) = A0 + sigma(n = 1 : N)(Anx^(-n)) \tag{4}$$

The summation of this equation allows a precision up to a certain N where A0 through An are constants. Converting our summation into a general matrix equation we obtain the following:

$$Y = CA \tag{5}$$

Where Y is our dependent matrix, C is the coefficients, and A is the matrix of Xns.We will be using the notation from the following equation throughout the duration of the report. Next we manipulate equation (5) to calculate our summation equations coefficients.

$$Y * A = A * A * C \tag{6}$$

$$(Y * A) * (A * A) = C \tag{7}$$

In order to account for the uncertainty of the data points a matrix W needs to be constructed. Using the n by one column matrix of uncertainties it needs to be converted to a zeros matrix with the uncertainties creating a diagonal. Inserting this matrix, W, into equation (7) it becomes:

$$(Y * W * A) * (A * W * A) = C \tag{8}$$

Using the newly found C matrix it is possible to sue equation (5) and calculate a matrix of theoretical Y values. We will name this yFit(xFit). In order to find goodness of fit the following equation is used:

$$Chi^2 = sigma|n = 1 : N|((yFit(xFit)yInitial)^2)/N \tag{9}$$

, where N is the number of data points in the matrices individually.

# 3 Results

Using the Matlab script the appropriate N, or iteration count, should be three. Below is a table of iteration count (how specific the equation becomes) with the respective chi squared value and predicted signal strength at times .2 ms and 14 ms

Table 1:

| N | Chi Squared | .2 ms | 14 ms |
|---|---|---|---|
| 0 | 3.3073e-4 | .9842 +/- .0050 mW | .9842 +/- 8.18e-5 mW |
| 1 | 7.0138e-6 | .7802 +/- .0050 mW | .9954 +/- 8.18e-5 mW |
| 2 | 6.8878e-6 | .7963 +/- .0050 mW | .9957 +/- 8.18e-5 mW |
| 3 | 6.7781e-6 | .7100 +/- .0050 mW | .9959 +/- 8.18e-5 mW |
| 4 | 6.3990e-6 | 2.0759 +/- .0050 mW | .9966 +/- 8.18e-5 mW |
| 5 | 6.3279e-6 | -4.0393 +/- .0050 mW | .9970 +/- 8.18e-5 mW |
| 6 | 6.3262e-6 | -16.5021 +/- .0050 mW | .9969 +/- 8.18e-5 mW |
| 7 | 6.3242e-6 | 173.0105 +/- .0050 mW | .9968 +/- 8.18e-5 mW |
| 8 | 6.3221e-6 | -2.7490e+3 +/- .0050 mW | .9967 +/- 8.18e-5 mW |
| 9 | 6.8465e-6 | 6.2864e+5 +/- .0050 mW | .9952 +/- 8.18e-5 mW |
| 10 | .0062 | 2.2023e+6 +/- .0050 mW | .9952 +/- 8.18e-5 mW |
| 11 | 2.4783e-4 | 3.8203e+6 +/- .0050 mW | .9953 +/- 8.18e-5 mW |

N = 0 is not a reasonable argument for goodness of fit due to a higher value of chi squared than 1 through 3 and beyond N = 4, the equation breaks down. Signal strength for .2 ms begins to explode at N = 4. This can be seen by graph 1:

*Graph 1: See the back of the document. Page 7.

Graph 2 shows when the equation runs to the iteration N = 3 compared to original data.

*Graph 2: See the back of the document. Page 8.

Using the following equation we can find the uncertainty of our coefficients when N = 3:

$$cUncertainty = (A * W * A) \tag{10}$$

Table 2: Coefficients and their uncertainties for N = 3.

| | |
|---|---|
| A0 | .9995 +/- .0032 |
| A1 | -.0504 +/- .0200 |
| A2 | .0086 +/- .0236 |
| A3 | -.0017 +/- .0067 |

## 4 Conclusion

With the assistance of Matlab we built a program to synthesize a collection of data from an unknown origin. By achieving this we can now compare data received by other antenna sites. The following equation is shown to be the most accurate system:

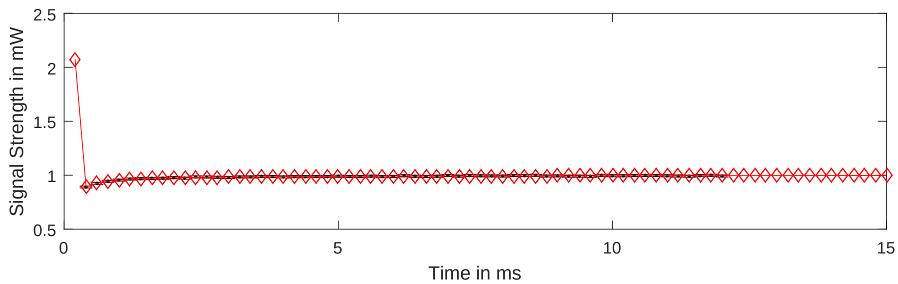
$$y = A0 + A1 * x^(-1) + A2 * x^(-2) + A3 * x^(-3) \tag{11}$$

Through this form we achieved a "goodness of fit" of 6.7781e-6 and are able to predict signals received outside the known interval. At .2 ms the predicted signal strength is .7100 +/- .0050 mW, and at 14 ms, .9959 +/- 8.18e-5 mW.

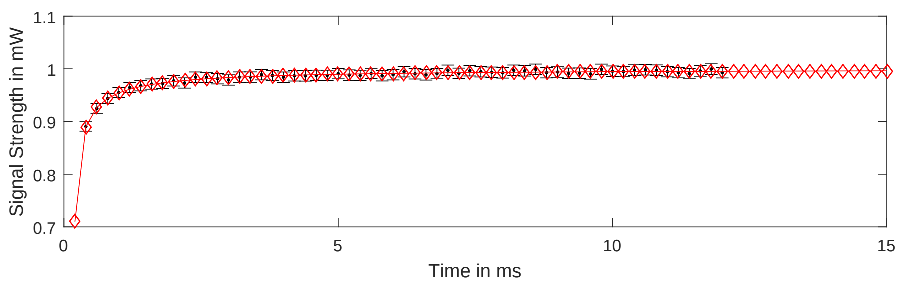
## 5 Appendix

```
load datafile
X = VarName1; %time stored in ms
Y = VarName2; %signal strength stored in mW
%uncertainty of y at time x
uncertainty = diag(VarName3.^(-2));

%to establish to what iteration the for loop runs to
prompt = 'What is your largest desired n? ';
n = input(prompt);

%A matrix for when N = 0
A = ones(59,1);
%To calculate the A matrix to find all coefficient
for count = 1:n
    A(:,count+1) = X.^(-count);
end
%c = coefficients
c = ((A' * uncertainty * A)^(-1)) * (A' * uncertainty * Y);
c;
%Uncertainties of coefficients
cUncert = ((A' * uncertainty * A)^(-1)).^(1/2);

%Theoretical y's calculated
```

```
yFit = A * c;
chi = 0;

%To calculate the chi for "goodness" of fit
for count = 1:59
    chi = chi + ((yFit(count,1) - Y(count,1))^2 / 59);
end
chi;

yTest = 0;

%Testing prediction for .2 using linear least fit
for count = 1:n+1
    yTest =  yTest + c(count,1) * .2^(1 - count);
end
yTest;

%BELOW HERE IS FOR MAKING COMPARISON GRAPHS!

x = 0:.2:15;
x = x';
A = ones(76,1);
for count = 1:n
    A(:,count+1) = x.^(-count);
end
fx = A * c;
hold off
errorbar(X, Y, VarName3, 'k.')
hold on
plot(x, fx, 'rd-')
```



Signal Strength in mW
2.5
2
1.5
1
0.5
0
5
10
15
Time in ms


Signal Strength in mW
1.1
1
0.9
0.8
0.7
0
5
10
15
Time in ms