NAVAL POSTGRADUATE SCHOOL
Monterey, California

UNITED STATES NAVY
NAVAL POSTGRADUATE SCHOOL

THESIS

INVESTIGATION OF AN EQUIVALENT CIRCUIT
FOR THE INDUCTIVE STRIP IN
FINLINE WITH DIELECTRIC

by

Michael R. Linzey

June, 1991

Thesis Advisor: Jeffrey B. Knorr

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School

6b. OFFICE SYMBOL (If applicable): EC

7a. NAME OF MONITORING ORGANIZATION: Naval Postgraduate School

6c. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000

7b. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) INVESTIGATION OF AN EQUIVALENT CIRCUIT FOR AN INDUCTIVE STRIP IN FINLINE WITH DIELECTRIC

12. PERSONAL AUTHOR(S): LINZEY, Michael R.

13a. TYPE OF REPORT: Master's Thesis

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Year, Month, Day): 1991 June

15. PAGE COUNT: 82

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Finline; Discontinuity; Inductive; Strip

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis describes a circuit model fo rthe inductive strip in inhomogeneous finline with the following geometry: fin and strip centered in the shield, dielectric material with $\epsilon_r = 2.22$, $b/a = 4/9$, $0.5 \leq w/b \leq 1.0$, $T/a \geq 0.01$ and $0.0 \leq d/a \leq 0.1$. The model is shown to produce results that agree with data computed using the spectral domain method. The model has been generated using WR(90) waveguide operating in the TE_{10} mode.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: KNORR, Jefferey B.

22b. TELEPHONE (Include Area Code): 408-646-2815

22c. OFFICE SYMBOL: EC/ko

DD Form 1473, JUN 86 Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603 UNCLASSIFIED

Approved for public release; distribution is unlimited.

Investigation of an Equivalent Circuit for the Inductive Strip in Finline with Dielectric

by

Michael R. Linzey
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1976

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1991

Author: ______________________
Michael R. Linzey

Approved by: ______________________
Jeffrey B. Knorr, Thesis Advisor

____
David C. Jenn, Second Reader

____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis describes a circuit model for the inductive strip in inhomogeneous finline with the following geometry: fin and strip centered in the shield, dielectric material with ε_r=2.22, b/a =4/9, $0.5 \le W/b \le 1.0$, $T/a \ge 0.01$ and $0.0 \le d/a \le 0.1$. The model is shown to produce results that agree with data computed using the spectral domain method. The model has been generated using WR(90) waveguide operating in the TE_{10} mode.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

Finline is a transmission structure for electromagnetic waves that was first discussed by Meier in 1974 [Ref. 1]. Finline consists of one or more thin metal fins printed on a dielectric substrate mounted in the E-plane of a rectangular waveguide. Figure 1 depicts the particular variation under consideration here. There are several advantages to the finline structure. The fin manufacture is simplified by the presence of the dielectric substrate which allows the use of well developed etching technologies for their manufacture. Finline has less stringent tolerance requirements than

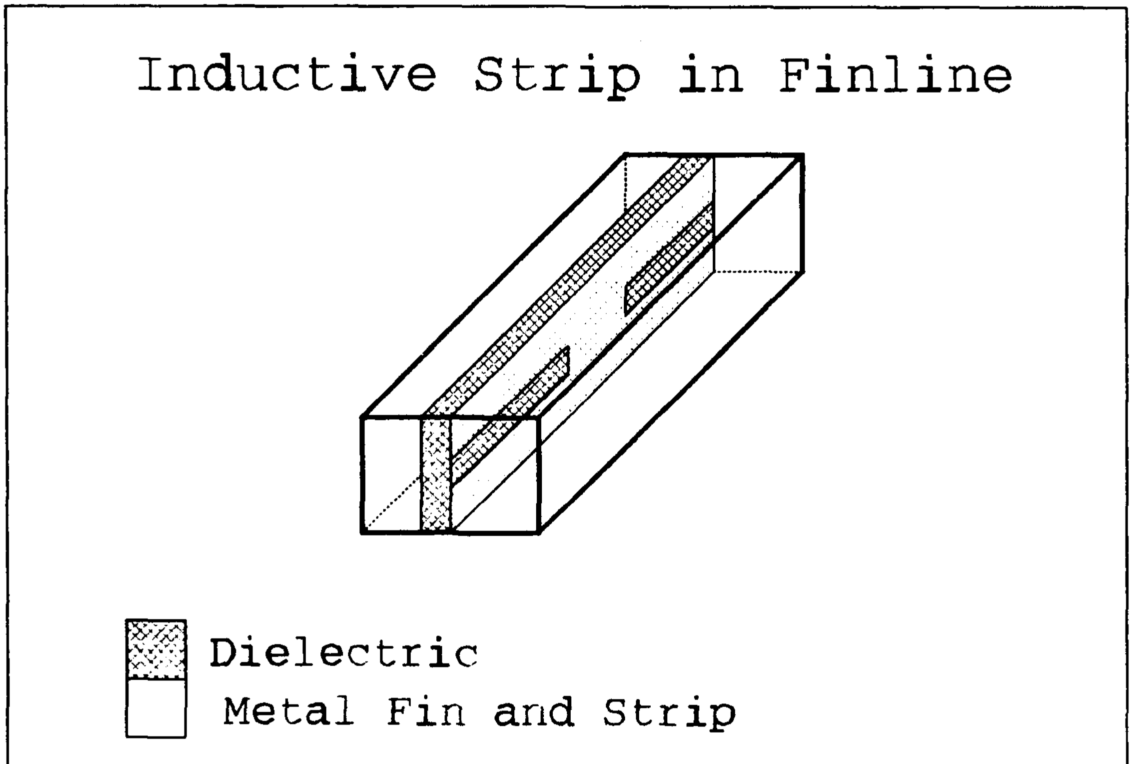

Figure 1. Arrangement of inductive strip in finline.

microstrip and the use of a rectangular waveguide shield simplifies the interfaces with other rectangular waveguides. Like conventional waveguides, finline can operate in single mode with low attenuation. [Ref. 1]

Computer assisted design (CAD) is essential to the development of complex devices in all fields of engineering. This is especially true in microwave and millimeter wave design. Numerical solutions to many electromagnetic problems result in time consuming computer implementations. To be useful in design, a workstation should be able to quickly predict the response of a known structure. The use of equivalent circuit models for common structures that respond in the same way as the general solution for a limited, but useful, range of parameters has been found to be a useful CAD technique.

B. USE OF PREVIOUS WORK

Initial work by Knorr and Shayda [Ref. 2] solved the electromagnetic fields in an arbitrary section of finline using the spectral domain method. The program IMPED implemented the solution. Knorr and Deal solved for the scattering parameters of an arbitrarily located inductive strip in finline and implemented the solution in a program called STRIP [Ref. 3],[Ref. 4]. Morua developed a circuit model for homogeneous finline, using input from both IMPED and STRIP [Ref. 5]. Grohsmeyer developed a model for inhomogeneous finline using the IMPED program. This work uses all of the previous work as a basis to begin and relies on the work of Grohsmeyer for the finline model [Ref. 6].

C. OBJECTIVE

The purpose of this study is to create a model that allows the representation of an inductive strip in inhomogeneous finline. The model must be accurate and suitable for implementation in a CAD environment.

The desired range of validity is as follows:

- $\varepsilon_r = 2.22$
- W/b - ratios from 0.5 to 1.0
- b/a - ratio of 4/9
- T/a - ratios greater than 0.01
- d/a - ratios from 0.0 to 0.1

D. CONVENTIONS

Previous work, referenced above, specified the subscript 'eq' to denote an equivalent dimension (length or dielectric constant). This model requires several equivalent dimensions to be specified. Therefore, the following subscript convention will be used.

- ' ' - actual dimension (no subscript)
- 'f' - equivalent dimension in finline
- 'd' - equivalent dimension in dielectric loaded half waveguide
- 'a' - dimension of the air filled half waveguide.

Figure 2 depicts three views of the strip in finline and labels the parameters used to describe the geometry. The dotted lines are grid lines to assist in specifying the dimensions.

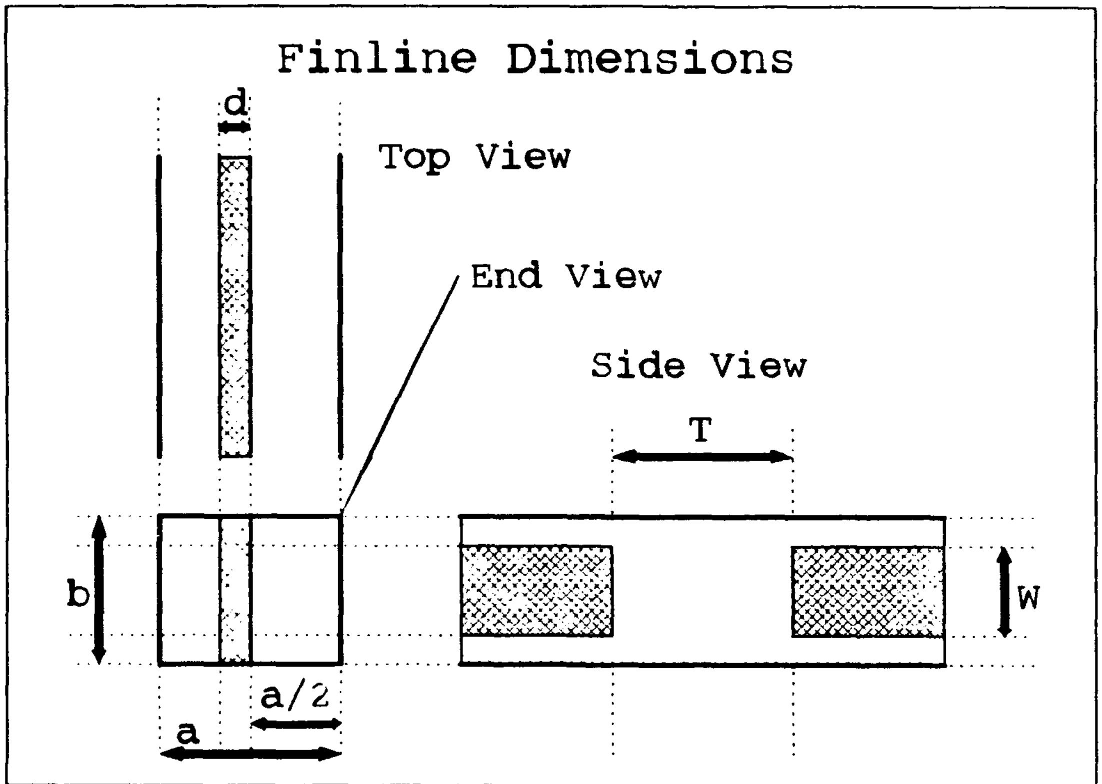

Figure 2. View of the inductive strip in finline with parameter dimensions indicated.

- a - width of the shield
- b - height of the shield
- T - length of inductive strip
- d - thickness of dielectric

II. MODEL DEVELOPMENT

A. CONCEPT

The model for an inductive strip was derived from the homogeneous inductive strip model developed by Morua [Ref. 5]. The model retains a similar structure but the calculation of element values has been modified to account for the inclusion of dielectric.

The model takes the physical structure of the finline inductive strip and divides it into two half-waveguides that are treated as separate elements, as shown in Figure 3. One

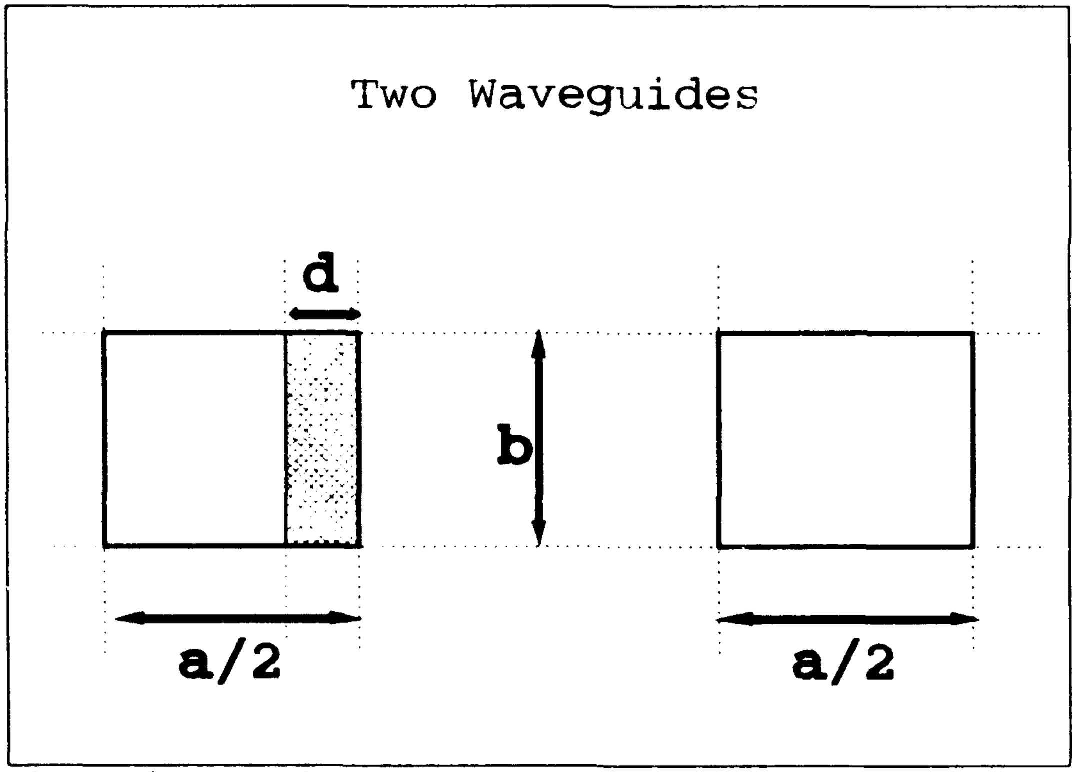

Figure 3. Looking into each of the half waveguides.

waveguide is air filled with transverse dimension a/2 and b. The second waveguide is dielectric loaded with the same dimensions as the air filled guide. At the mouth of each of these half-waveguides, there is a parallel inductor/capacitor circuit that accounts for stored electric and magnetic energy. The circuits are then connected through a perfect impedance transformer to the finline. Figure 4 shows the entire structure.

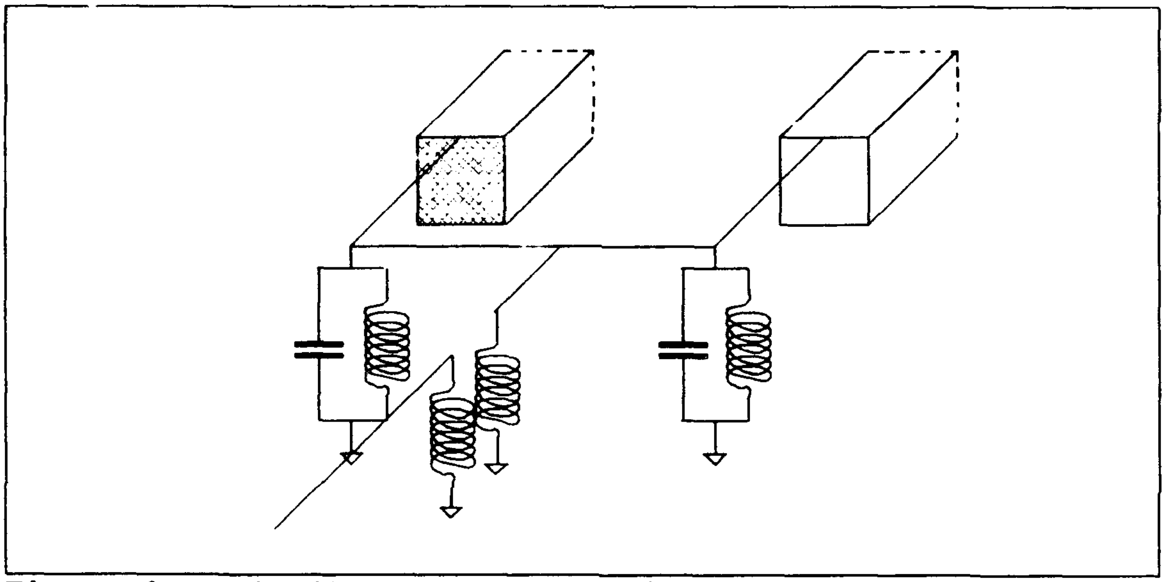

Figure 4. This figure shows the circuit at one end of the two half-waveguides. The other ends of the waveguides are connected to the mirror image of the above circuit.

B. DERIVATION OF THE MODEL

1. General Comments

A model for the loaded half-waveguide that could be represented in CAD software was found. This required that an equivalent below-cutoff homogeneous waveguide be found. A model form that replicated the scattering properties of the inductive strip as predicted by STRIP was then developed.

2. Loaded Waveguide Model

The loaded, below-cutoff waveguide was modeled by matching the propagation constant and voltage-power impedance with that of equivalent homogeneous waveguide. The first step of this process was to write the solution for the electromagnetic wave propagation for a below-cutoff dielectric loaded waveguide as shown in Figure 3. In the following discussion, it is understood that, β, will be a positive imaginary number due to the below-cutoff operation of the guide and will result in a negative real propagation constant. The full derivation of this solution is shown in Appendix A and parallels the discussion on propagating loaded waveguide in Pozar, [Ref 7;pp. 151-153]. This analysis results in the following transcendental equation

$$0=\sqrt{k_0^2-\beta^2}\,\tan(d\sqrt{\epsilon_r k_0^2-\beta^2}\,)+\sqrt{\epsilon_r k_0^2-\beta^2}\,\tan((a-d)\sqrt{k_0^2-\beta^2}\,)\,. \quad (1)$$

Since this equation must be true for all frequencies, it is possible to find a set of, β's, for various frequencies and geometries. Several MATLAB functions were written to compute β, in a half section of WR(90) waveguide for frequencies between 6 GHz and the cutoff frequency, f_c, and for 2d/a values from 0 to 0.25 . Figure 5 shows the cutoff frequency as a function of d/a for WR(90) waveguide. Figure 6 depicts β versus frequency for various dielectric thicknesses. These curves describe the propagation characteristics of a particular geometry and are the characteristics that need to be matched by the equivalent waveguide. The equation for β in a homogeneous waveguide can be written as

$$\beta=\sqrt{\epsilon_r k_0^2-\left(\frac{\pi}{a_d}\right)^2}\,. \quad (2)$$

Using another set of MATLAB routines values of a_d and ϵ_{rd} that cause a waveguide described by equation (2) to match the curves in Figure 6 can be found for each desired geometry.

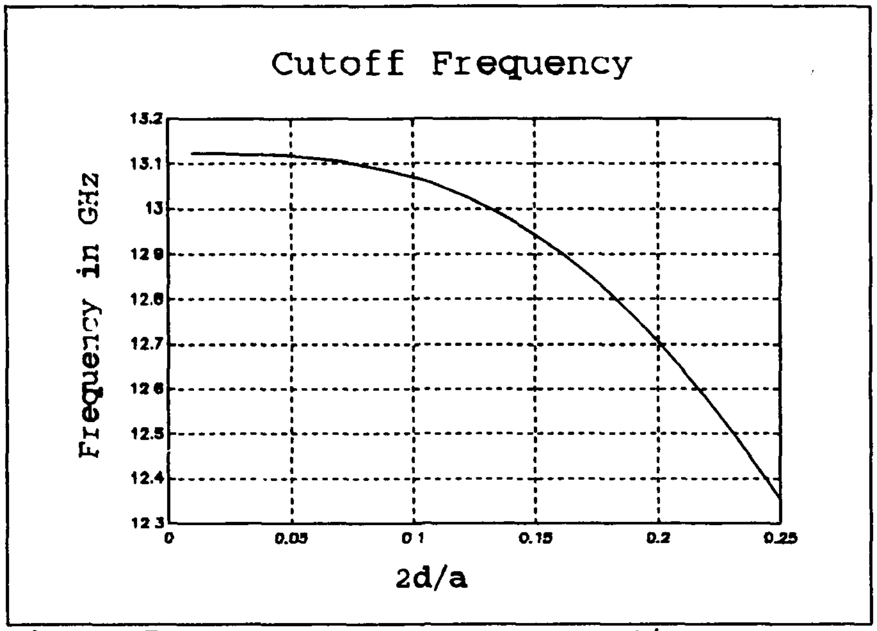

Figure 5. Cutoff frequency vs. d/a.

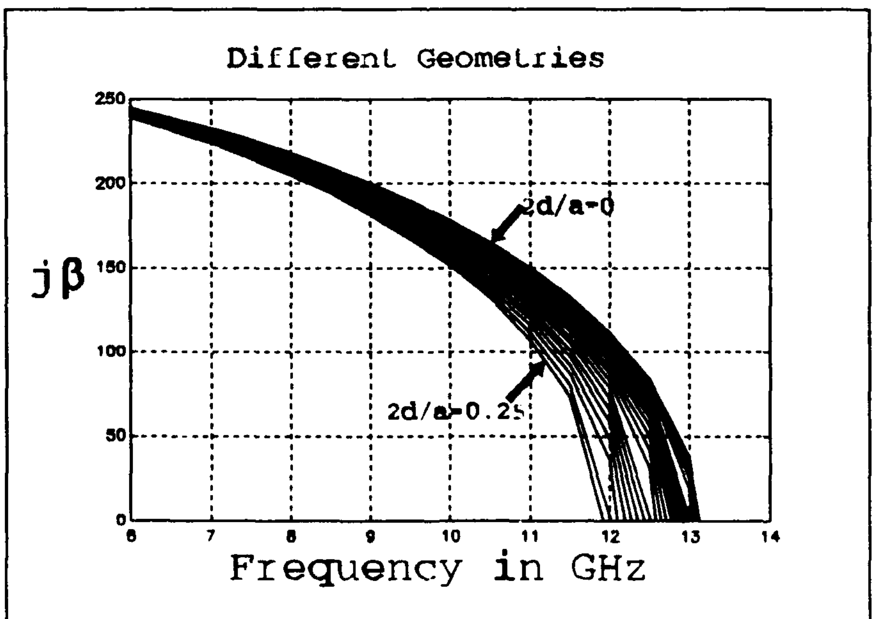

Figure 6. β vs. frequency for several values of d/a.

That data can then be reduced to the following two equations using a least mean square error curve fitting routine:

$$\epsilon_{rd}=1.0+0.0192\left(2\frac{d}{a}\right)-0.3391\left(2\frac{d}{a}\right)^2+10.6174\left(2\frac{d}{a}\right)^3+ \\ -3.6647\left(2\frac{d}{a}\right)^4+0.01356\left(2\frac{d}{a}\right)^5 \tag{3}$$

and

$$\frac{a_d}{a/2}=1.0+0.001314\left(2\frac{d}{a}\right)-0.03026\left(2\frac{d}{a}\right)^2+0.3424\left(2\frac{d}{a}\right)^3+ \\ -2.1617\left(2\frac{d}{a}\right)^4+1.7461\left(2\frac{d}{a}\right)^5 \tag{4}$$

where

$$0\leq\frac{d}{a}\leq 0.1.$$

The shapes of the curves described by the above equations are shown in Figures 7 and 8. Application of these equivalent dimensions to equation (2) results in β's that differ from those found for the loaded waveguide by less than 0.2%. MATLAB functions used in the production of the equivalent homogenous half-waveguide in this section can be found Appendix B.

The voltage-power impedance of the loaded waveguide can be calculated analytically since the electric and magnetic fields can be found to within a single constant value. The voltage-power impedance is given by

$$Z_{ov}=\frac{V^2}{2P}. \tag{5}$$

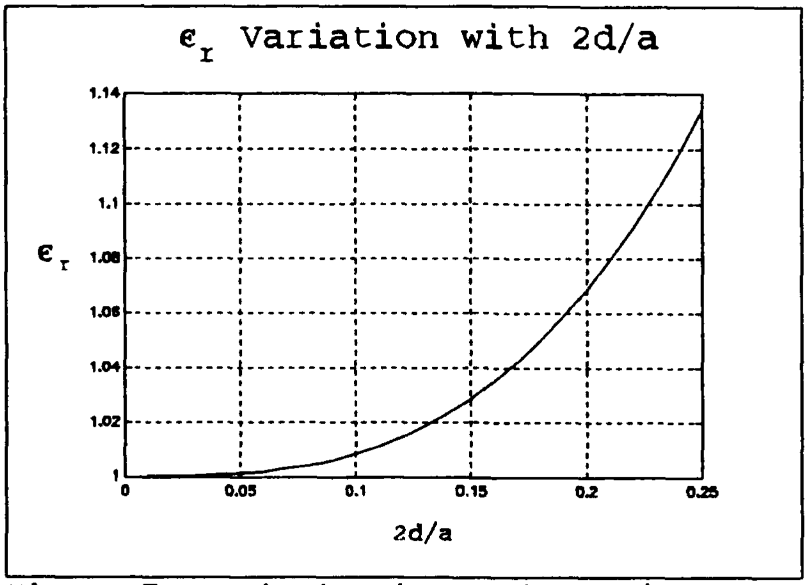

Figure 7. Variation in ϵ_r with 2d/a.

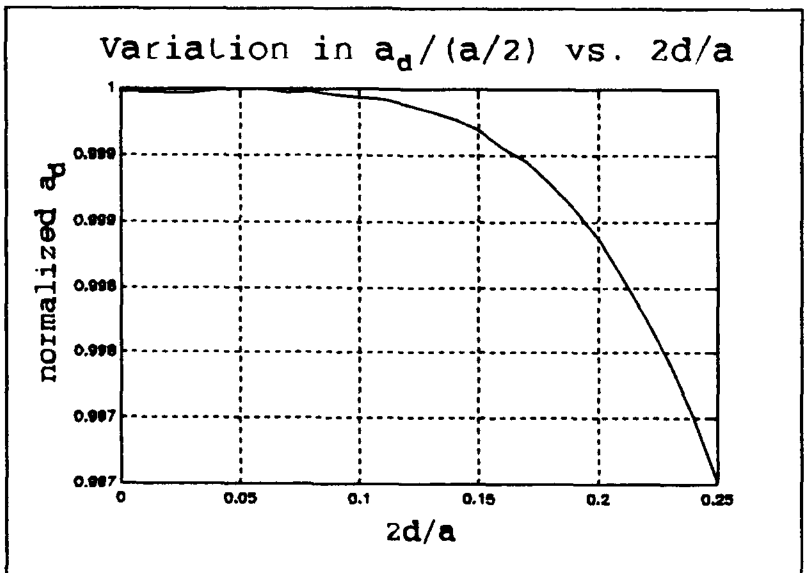

Figure 8. Variation of a with 2d/a.

In this case, since the waveguides are operating below-cutoff the power will be imaginary and will result in an imaginary impedance [Ref. 8]. The power can be found to be

$$P=b\left(-\frac{A^2\omega\beta\mu\cos^2(k_d d)\tan(k_a(a-d))}{2k_a^3}+\frac{(a-d)A^2\omega\beta\mu}{2k_a^2\cos(k_a(a-d))}\right.$$
$$\left.-\frac{A^2\omega\beta\mu\sin(k_d d)\cos(k_d d)}{2k_d^3}+\frac{A^2 d\omega\beta\mu}{2k_d^2}\right). \tag{6}$$

Equation (6) can be written more compactly as

$$P=b\ C_p \tag{7}$$

where the imaginary constant C_p represents the terms in parentheses above. The voltage in equation (5) is given by

$$V=\ -\int_0^b E_m\ dl=-E_m b \tag{8}$$

where E_m is the maximum electric field. The voltage-power impedance for the loaded half-wave guide is

$$Z_{ov}=b\frac{E_m^2}{C_p}. \tag{9}$$

The voltage-power impedance for the equivalent below-cutoff waveguide is given by [Ref. 6]

$$Z_{ov}=120\pi\frac{2b_d}{a_d\sqrt{\epsilon_{rd}}}\frac{1}{\sqrt{(1-\left(\frac{\lambda}{2a_d\sqrt{\epsilon_{rd}}}\right)^2}}. \tag{10}$$

Given that a_d and ε_{rd} are already known, b_d can be found by solving equation (10) for b_d and substituting in Z_{ov} from equation (9) which results in

$$b_d = \frac{bE_m^2}{C_p}\frac{\sqrt{4a_d^2 f^2 \varepsilon_{rd} - c^2}}{480\pi f}. \tag{11}$$

Equations (3), (4) and (11) define the loaded half-waveguide model. The variation in b_d with dielectric thickness is shown in Figure 9.

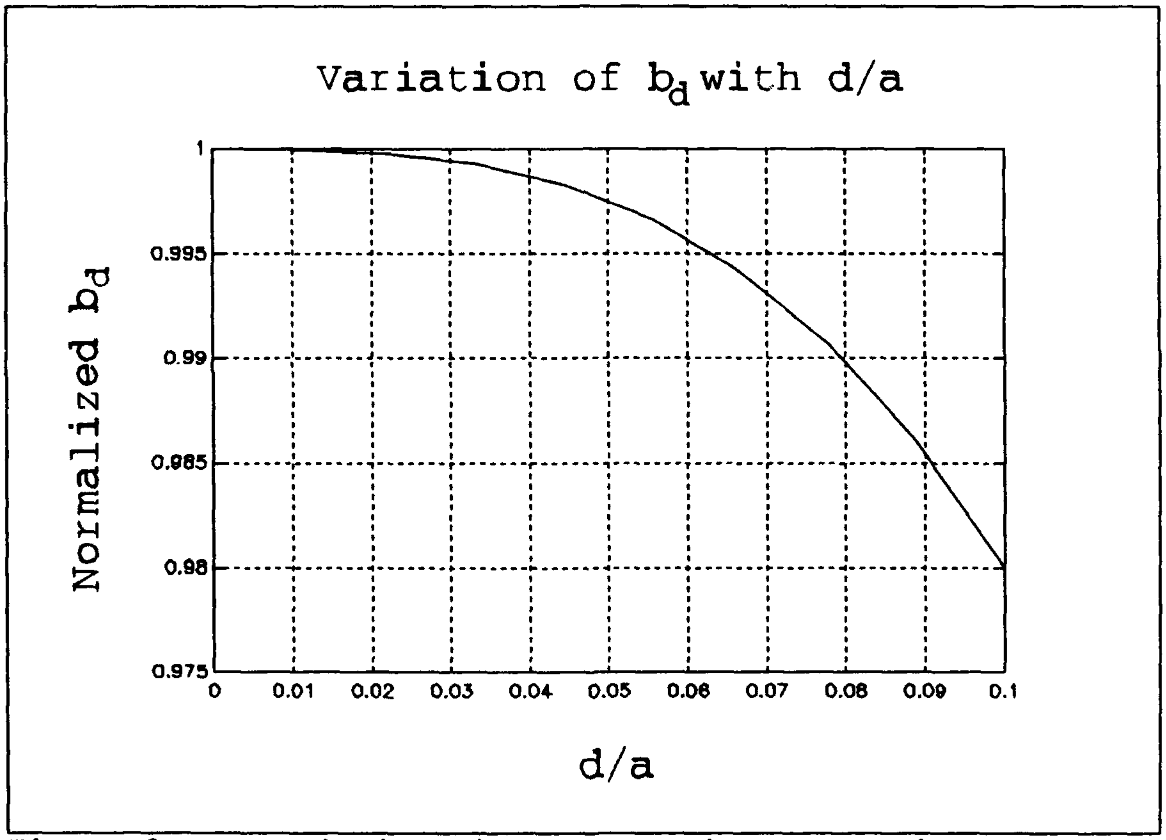

Figure 9. Variation in the equivalent height of the dielectric loaded half waveguide.

3. Integrated Model

Using the circuit model developed by Morua as a starting point, a MATLAB function that replicated the performance of the CAD workstation was written. This program was validated using arbitrary values of the inductance, capacitance, and turns ratio on the CAD workstation. The CAD program and the MATLAB functions used are located in Appendix C. A function minimization routine was then written which found values of the various elements that resulted in the model output matching the spectral domain data produced by STRIP. During the course of much trial and error, it was found that the expression used by Morua to calculate the turns ratio in the homogeneous case could not be extended to the inhomogeneous case. The model produced good results when the conductance on the strip side of the transformer was specified as a function of frequency. It was found that the conductance could be defined in the following way

$$Y_0=0.001(t_3 f_n^3+t_2 f_n^2+t_1 f_n+t_0) \qquad \textbf{(12)}$$

where

$$f_n=\frac{f-f_c}{f_c}. \qquad \textbf{(13)}$$

Figure 10 depicts the way conductance varies with frequency for a typical geometry. Once the value of the conductance needed to match the spectral domain data was found, the turns ratio of the impedance transformer was calculated using the relation

$$N=\sqrt{\frac{Y_f}{Y_0}}=\sqrt{\frac{Z_0}{Z_f}}. \qquad \textbf{(14)}$$

where N is the turns ratio, Y_f is the finline conductance, Y_0 is the conductance on the strip side of the transformer. The behavior of the turns ratio with frequency is shown in Figure 11. This model was found to be able to replicate the performance of the inductive strip for all geometries of interest with less than 1% error. Some results are shown in Figures 12 and 13 (in the figures spectral domain data is referred to as SPEC_DOM and the model is referred to as STRIP).

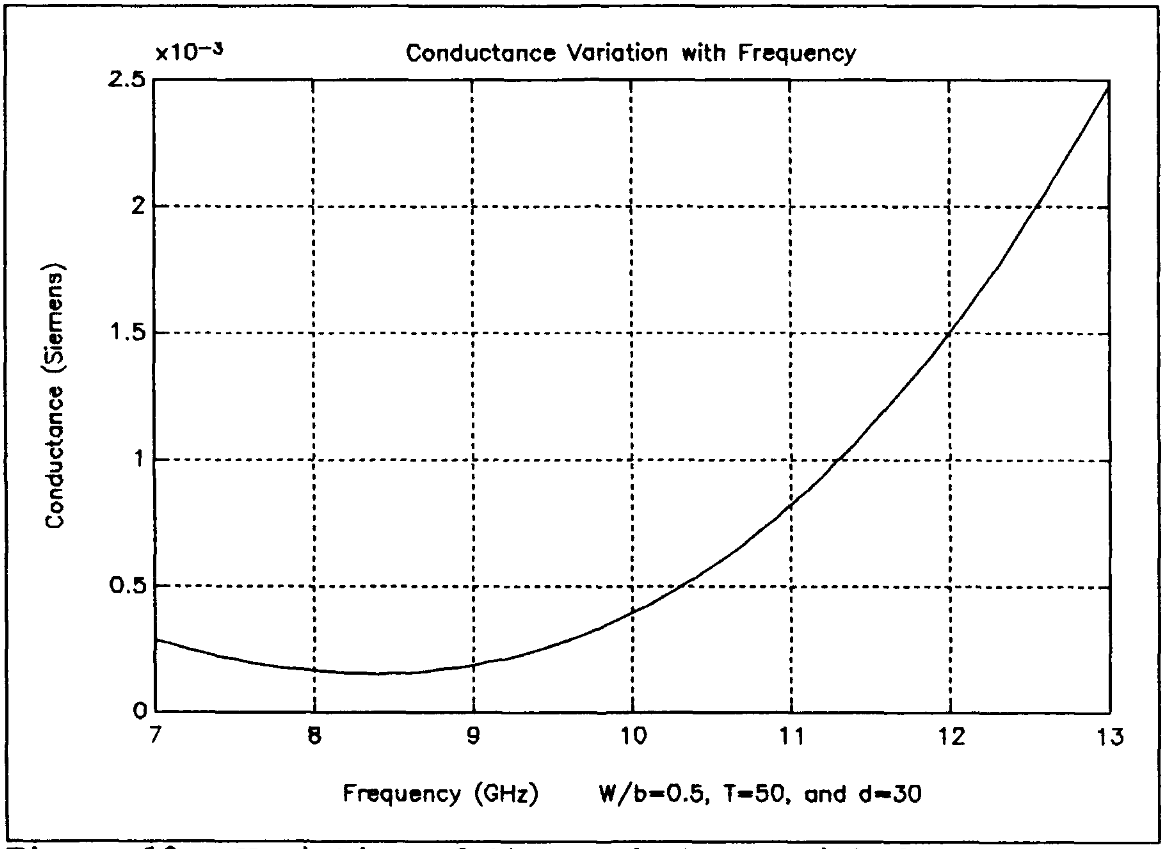

Figure 10. Variation of the conductance with frequency.

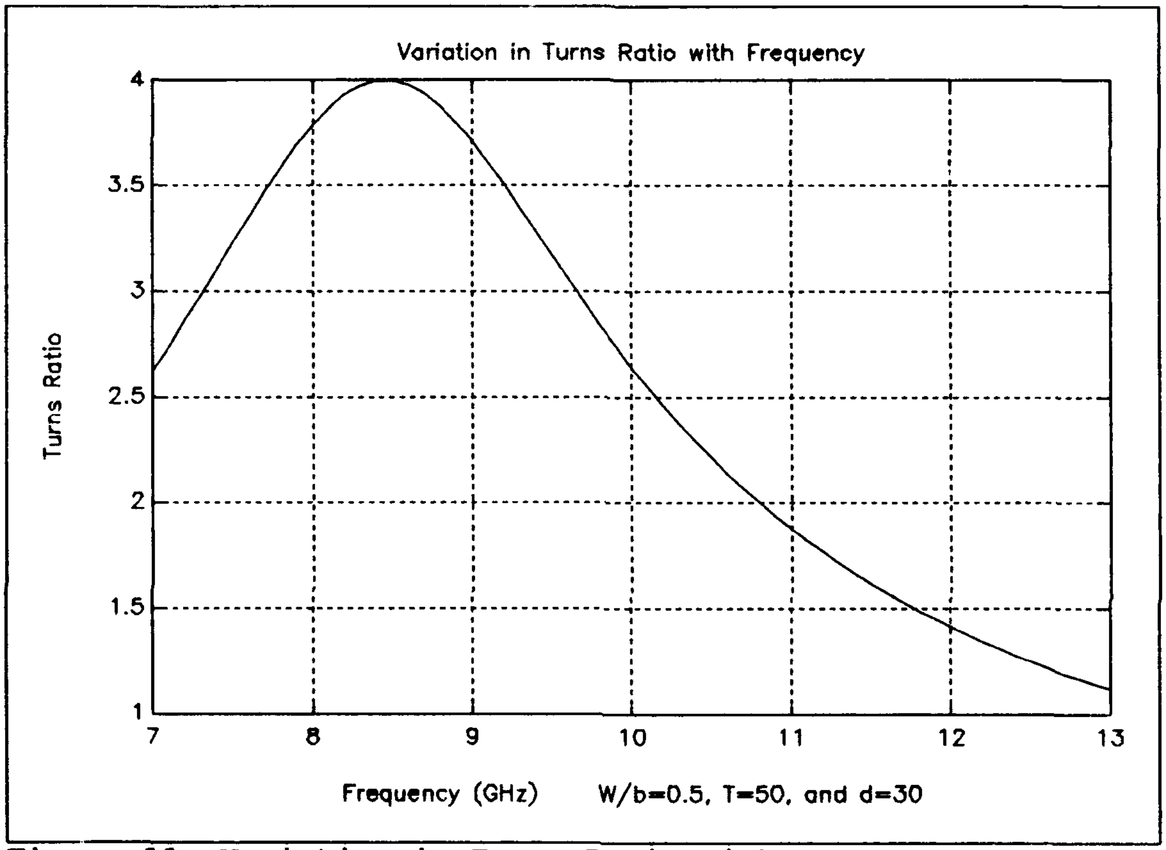

Figure 11. Variation in Turns Ratio with Frequency.

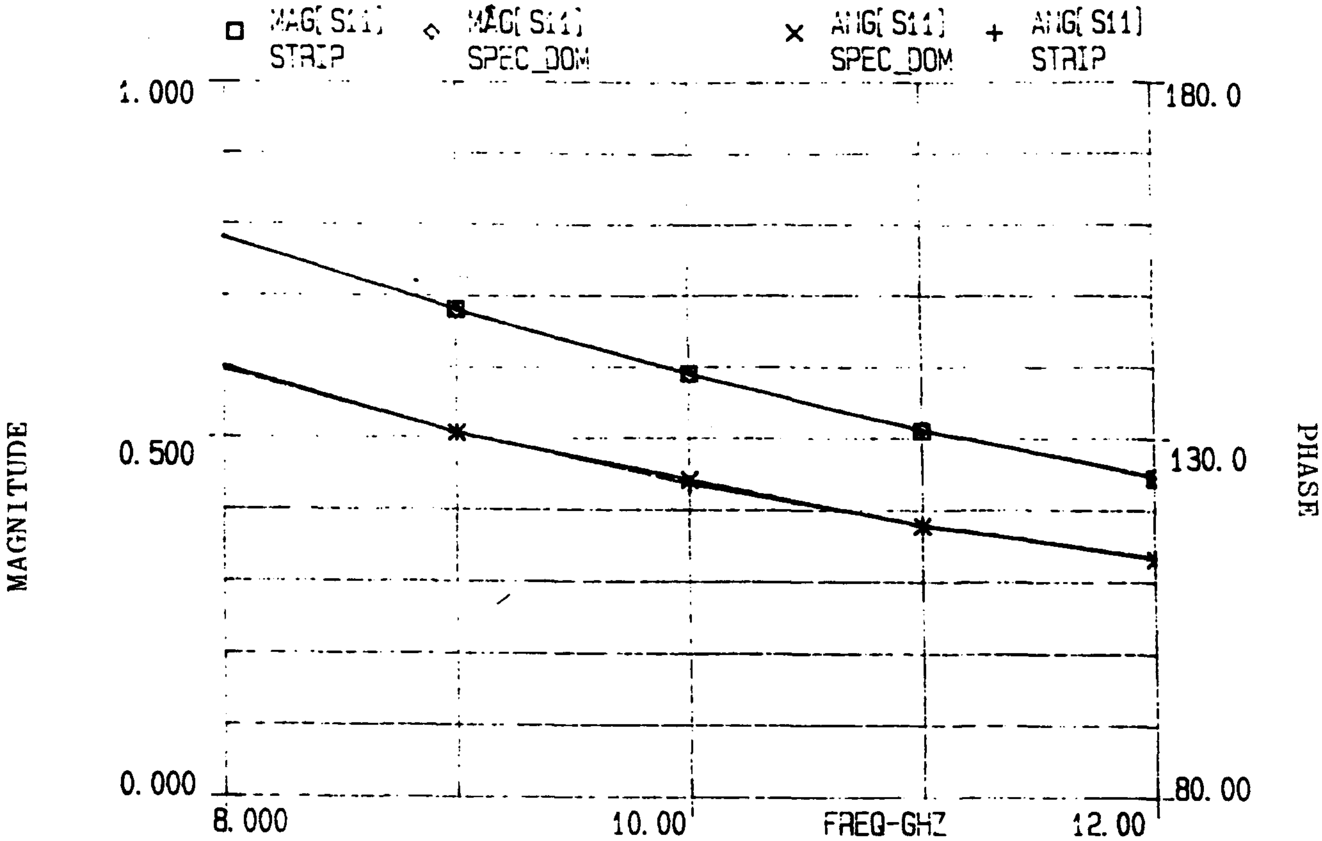

Figure 12. Response of S_{11} for Model and Spectral domain data, for W/b=0.5, T=10 mils, d=0 mils, L=14.67nH, C=0.0043pF and $Y_0=0.001(1.315+3.63f_n-3.43f_n^2+0.98f_n^3)$.

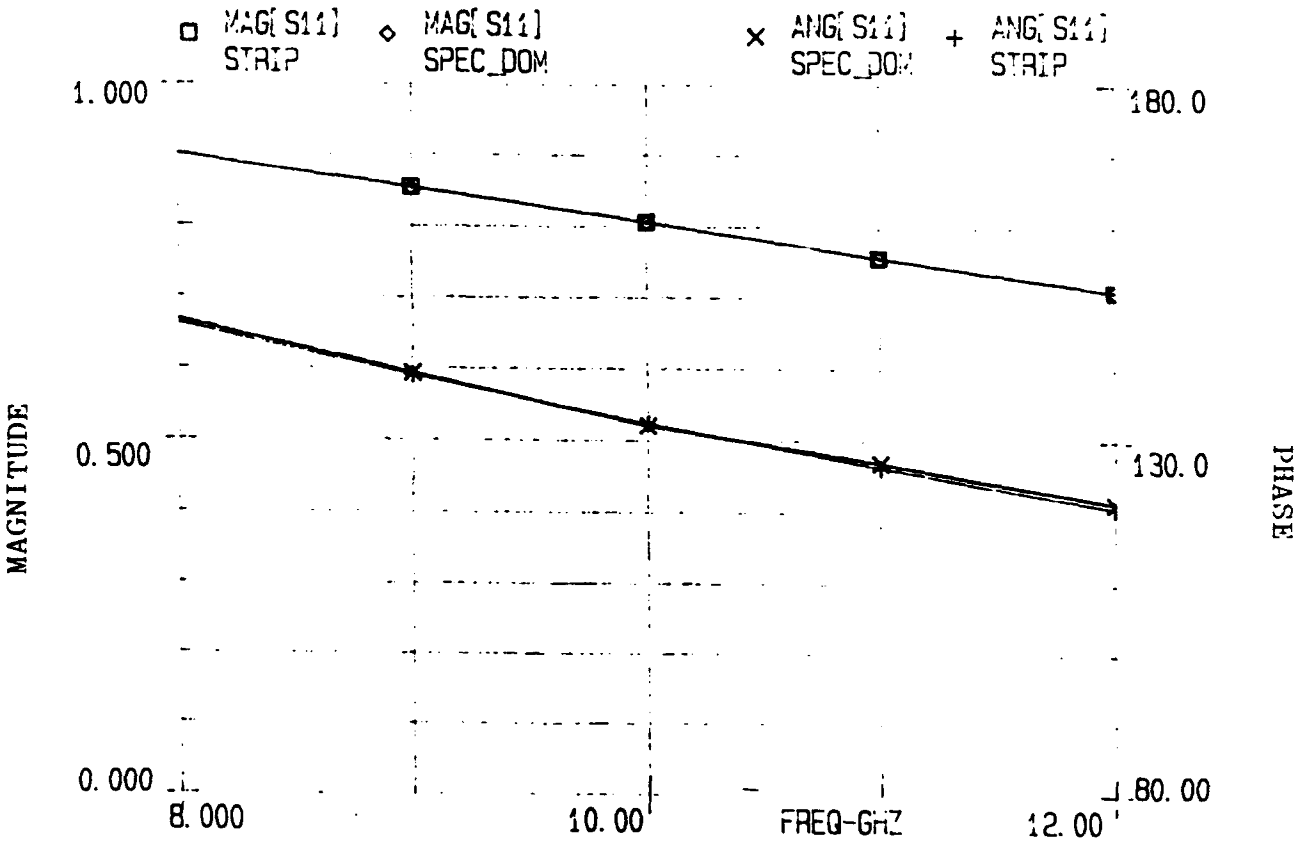

Figure 13. The response of S_{11} for Model and Spectral domain data, for W/b=0.5, T=50 mils, d=30 mils, L=12.12nH, C=.0018pF, and $Y_0=0.001(1.69+2.06f_n-1.52f_{n2}+0.3792f_n^3)$.

III. DATA COLLECTION

A. STRIP PROGRAM DESCRIPTION

The input data was generated using the STRIP program. STRIP is a Fortran program that computes the scattering parameters of the inductive strip in finline using the spectral domain method. The STRIP program and the development of the theory behind it are discussed in detail in References 3 and 4. Accuracy of 1-2% with experimental data on homogeneous finline were reported. [Ref. 3],[Ref. 4]

B. MODIFICATIONS TO THE ORIGINAL PROGRAM

During the course of the data collection both a SUN workstation and a VAX 11/785 were available to compute the scattering data. The two computers were found to produce results that differed by as much as 10%. These errors were presumed to result from differences in the way the two machines handled small numbers. This indicated a possible problem in the way the program dealt with small numbers. To correct the problem, the program was changed to re-specify some variables that were previously real as double precision. This modification elevated the problem and resulted in agreement between the solutions generated by the VAX and the SUN. Unfortunately, other intermittent problems began to appear in the output data. Further extensive trouble shooting revealed that some double precision values were being returned from subroutines into real numbers. The extra bits were written into the adjacent memory spaces, corrupting them. This problem was corrected by re-compiling the program to treat all real numbers as double precision. This final alteration corrected all the apparent problems in the operation of the program.

C. DATA GATHERING

The data on which the model was based, was calculated using parameters associated with WR(90) waveguide. WR(90) waveguide was selected for two reasons. First, it is more convenient to conduct measurements in X-band. Second, as shown by Morua in the homogeneous case, selection of b/a=4/9 may permit the model to be extended to b/a's, from 0.4 to 0.5. The following list specifies the values of the parameters used in determining the data points. All measurements in mils (1 mil = 0.001 inches):

- W/b = { 0.5, 0.7, 0.9, 1.0 }
- T = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 1000 }
- d = { 0.0, 10, 30, 50, 70, 90}

The strip length interval was increased for T>100 mils because the scattering parameters were found to vary more slowly at large values of T.

The performance of the STRIP program is determined in large part by the order of the matrix of inner product terms. Morua showed that most of the convergence for typical parameter values had occurred by matrix of order 10. Therefore, a matrix of order 10 was used as a starting point to calculate the data. Some data sets were re-computed with matrix orders of 12 to attempt to correct errors discussed below.

The STRIP program was developed based on the assumption that the finline was operated in the TE_{10} mode and the two half-waveguide sections were operated below cutoff. As the thickness of the dielectric substrate was increased the second propagating mode began to have an effect for the higher frequencies in the range of interest. In this situation, the program returns an increasing S_{11} magnitude normally 1.0. Values of the magnitude of S_{11} were assumed to be continually decreasing with increasing frequency. This assumption complies with intuition of the physical system that as the non-propagating half-waveguide approaches the conditions for propagation, more energy will be transmitted and less reflected. Therefore, data points that indicated an increasing magnitude of S_{11} with frequency were discarded. At apparently random geometries, the STRIP program returned values for the magnitude of S_{11} that did not correspond with the similar, but different, geometries or the expected behavior of the structure. Identification of this data is discussed below. The block of erroneous data would typically begin at a strip length, T, of approximately 500 mils and continue through 1000 mils for a given dielectric thickness and gap width. This type of error was found to effect 15% of the data. By not including data with T greater than 500 mils the amount of affected data was reduced to approximately 5%. For this reason, data with a strip length greater than 500 mils was discarded. An example of this behavior is shown in Table 1 (on the following page). The bad data is located at d=70 mils, T=600 mils and f=12 GHz. The numbers in the table represent the magnitude of S_{11}. The physical arrangement of the system is such that with increasing strip length, T, the reflection coefficient should continually increase. The adjacent data, at d=50, does exhibit the expected behavior. The data point described here and others that could be identified as bad in this manner were discarded.

Table 1. Computed Magnitude of S_{11} near d=70 and T=600 at 12 GHz, Illustrating Anomalous Results

f=12GHz	d=50 mils	d=70 mils
T=300 mils	0.8790	0.7877
T=400 mils	0.9643	0.8216
T=500 mils	0.9788	0.8242
T=600 mils	0.9872	0.6259

III. ELEMENT REDUCTION

A. CONCEPT

It has been shown that element values could be found which produced excellent agreement with the STRIP data for specific geometries. The development of a useful tool for the CAD environment requires that a method of generating element values as a function of the geometry be found. In the following discussions, the coefficients of the strip conductance are referred to as elements.

B. REDUCTION TECHNIQUE

Polynomials were selected as generating equations. Polynomials were selected because of their simplicity and because the element data appeared to be slowly varying. The function minimization routines resulted in solutions that were exact beyond the level needed for a successful model. Errors of less than 1% with the STRIP data were typical and the expected error for the STRIP data was 1-2%. Therefore, it was assumed that an acceptable solution could be found and that a least mean square curve fit would be an adequate method of reducing the data. Programs were written that conducted a least mean square fit over each of the three parameters of interest.

The reducing function first took an entire set of data for a given W/b and then subdivided that into sets of different d/a values. For each set of like d/a's a second order least mean square reduction was done using the T/a ratios as the input parameter. The resulting coefficients were placed in column form. This procedure was repeated for each different d/a set. The coefficients were then arranged into data sets treating each location in the column as a data point. Another least mean square fit was done using the d/a ratios as the input parameter. Another set of coefficients were generated from this reduction. These coefficients were again placed in a known column order. This procedure was repeated for each W/b set. The coefficients were again placed in a known column order and the data reduced to coefficients using a least mean square fit with W/b as the input parameter. This procedure resulted in 162 coefficients. The element values could then be regenerated by reversing this process. The programs that implemented the above algorithm were tested by picking an arbitrary set of coefficients, generating element values then reducing the element values to coefficients. The coefficient values were recovered with differences on the order of 10^{-6}. This test, of course, did not give any indication of the performance of the algorithm with unsmoothed element values. The function listings for the above routines are included in Appendix C.

Application of the procedure described above to the element values was unsatisfactory. Testing the model with elements generated from the curve fitting routine resulted in large errors. Typical error levels were found to be in the neighborhood of 80%. Apparently, small errors generated in the initial recursions were magnified by using curve fitting on the coefficient in the subsequent curve fits. After the initial attempt was found to be inadequate, several additional procedures were used to attempt to develop suitable coefficients.

The element values were examined and values that were far from the apparent trend of the data were eliminated to create a data set that could be used to generate usable coefficients. This procedure did improve the performance but the results were again found to be unsatisfactory. Several possible variations in the 'correct' trend were explored with similar unsatisfactory results.

The most promising of the coefficient sets generated above was used to generate all of the values with the exception of the value of the inductor. The inductor was selected because it was the single most significant element to the model's fit. Another program was written that found the value of the inductor needed, given that the other elements were all generated by coefficients. The elements were again reduced using the initial procedure described. A comparison between desired element values and those generated by the coefficients was made and showed that the elements other than the inductor were a perfect fit, as expected. The error in the value of the inductor was found to be in the 5-7% range, with some much larger values. This level of error in the inductor was found to result in 20-30% error when compared to the STRIP data. This procedure was applied successively to each element in the hope that the coefficient values would converge to an acceptable solution. Convergence did not occur. The output of the data was continually found to generate coefficients that produced error levels in the 20-30% range. Appendix E contains a portion of the MATLAB output, with typical output values.

Two final attempts at function minimization were made. Both attempts iterated through the entire data set to allow a best guess at the correct coefficients. The first attempt used the coefficients found using the curve fitting procedure to generate all of the element values except the value of the inductor. The 27 coefficients necessary to specify the inductor were passed to the function minimization routine. This program tested every fifth data point of the entire STRIP data set. The error at each point was defined in the same way as described in the previous chapter. The total error was defined as the summation of the error at all the test points.

The resulting coefficients produced no significant change in the performance of the model. A modification to the program was made that allowed the function minimization routine to vary all 162 of the coefficients. This modification did not result in improved performance. The failure of these last two attempts was expected for two reasons. First, the function minimization routine was designed to operate with less than five parameters. Second, the nature of the model is such that when drastically incorrect element values are applied to the model, the value of the magnitude of, S_{11}, goes to either 0 or 1 and the angle to $\pm 180^{\circ}$, as a result, no indication of increased or decreased error is given to the minimization routine.

Coefficients were not found which allowed the element values to be represented as series of equations. Possible alternative approaches are discussed in the following chapter. The reason that least mean square curve fitting was not applicable is not completely clear, but it is most likely a combination of the following factors.

- The output of the STRIP program changed after the variables were re-specified as double precision. Indicating a possible problem in the program, although the program has been confirmed experimentally for several geometries.
- The STRIP data may contain unlocated irregularities that prevent the production of smooth element values.
- Function minimization accentuates inconsistencies in the data by finding the optimum fit point by point.
- Using an element generation method that results from three iterations of curve fitting is susceptible to significant error magnification during the calculation of the elements.

V. SUMMARY

A. CONCLUSIONS

The equivalent circuit model presented has been shown to accurately reproduce the response predicted by the STRIP program for individual geometries. Within the model, there are three effects that act together to replicate the behavior of the actual inductive strip. They are the decay of the field along the length of the inductive strip, the scattering from the edge of the strip, and the coupling of the field between the finline and the half-waveguide sections. The length of the inductive strip was accounted for by the two below-cutoff waveguides, which attenuated the energy flow along their lengths. The scattering caused by the edge of the inductive strip was modeled with the fixed parallel inductor and capacitor circuit. The coupling of the field from the finline to the below-cutoff waveguides was modeled with a frequency dependent conductance defined at the beginning of the strip. This frequency dependant conductance accounted for the transformation of the finline conductance to the conductance of the below-cutoff waveguides. Viewing the model in this way, gives valuable insight into the behavior of the actual inductive strip in finline.

B. RECOMMENDATIONS

An algorithm which accurately reproduced the element values was not found. To be useful in the CAD environment, a method of generating the element values from the geometry is essential. One possibility would be to smooth the STRIP data using some curve fitting technique prior to applying the model to the data. Smoothing the data first would reduce the tendency of the function minimization routines to produce element values that are difficult to describe analytically.

Element values that can be easily and accurately reduced to equations are essential because of the three levels of generation required.

The model should be extended to other geometries. It has been tested at values of W/b as low as 0.1 and has continued to perform well for specific geometries. The model may be scalable to b/a's of 0.4 to 0.5 as was the case for homogeneous finline, but this will require further investigation.

The model for the loaded waveguide could be tested experimentally and implemented directly in CAD software.

APPENDIX A. DERIVATION OF LOADED HALF WAVEGUIDE MODEL

This appendix presents a derivation of the attenuation through a partially loaded below cutoff waveguide and the computation of the equivalent height by matching the voltage-power impedance. The variable 'a' will be used here to denote the width of the shield. The presentation here follows Pozar's development for the loaded waveguide. [Ref 7:pp. 151-153]

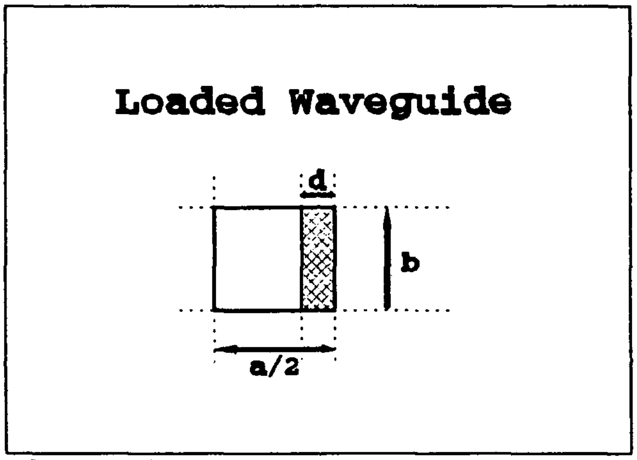

Figure 14. Loaded half-waveguide.

As can be seen in Figure 14, the geometry is uniform in the y direction and since, n=0, the TE_{10} modes have no y dependance the wave equation for H_z can be written separately as

$$\left(\frac{\partial}{\partial x}+k_d^2\right)H_z=0 \tag{1}$$

and

$$\left(\frac{\partial}{\partial x}+k_a^2\right)H_z=0 \qquad (2)$$

where k_a and k_d are the wave numbers in the two regions (a in air and d in the dielectric). These two equations are related by the fact that both waves must propagate down the waveguide together. Therefore, the propagation constant, γ, must be the same on each side. The model assumes no loss, therefore, the propagation constant reduces to $j\beta$. Since, the model also assumes below cutoff operation β must be a positive imaginary number, which will result in a negative real propagation constant and evanescent electromagnetic waves. Therefore, β can be written as

$$\beta=\sqrt{\epsilon_r k_0^2-k_d^2}=\sqrt{k_0^2-k_a^2}. \qquad (3)$$

Given the wave equations above and a knowledge of the boundary conditions, the general solution can be written as

$$\begin{aligned} H_{zd}&=A\cos(k_d x)+B\sin(k_d x) && \text{for } x<d \\ H_{za}&=C\cos(k_a(a-x))+D\sin(k_a(a-x)) && \text{for } d<x<a. \end{aligned} \qquad (4)$$

The field components of a TE field can be found from H_z

$$E_y=\frac{j\omega\mu}{k_c^2}\frac{dH_z}{dx} \qquad H_x=\frac{-j\beta}{k_c^2}\frac{dH_z}{dx}. \qquad (5)$$

The electric field in the y direction can be found to be

$$E_{yd}=\frac{j\omega\mu}{k_d}\left(B\cos(k_d\, x)-A\sin(k_d\, x)\right) \qquad (6)$$

and

$$E_{ya}=\frac{j\omega\mu}{k_a}(D\cos(k_a(x-d))-C\sin(k_a(d-x))\,. \tag{7}$$

Boundary conditions on the tangential electric field imply that B=D=0. The field components reduce to

$$H_{zd}=A\cos(k_d\,x) \qquad H_{za}=C\cos(k_a(a-x)) \tag{8}$$

$$E_{yd}=-\frac{j\omega\mu A}{k_d}\sin(k_d\,x) \qquad E_{ya}=-\frac{j\omega\mu C}{k_a}\sin(k_a(a-x)) \tag{9}$$

$$H_{xd}=\frac{-j\beta A}{k_d}\sin(k_d x) \qquad H_{xa}=\frac{-j\beta C}{k_a}\sin(k_a(d-x))\,. \tag{10}$$

Since, the electric field tangential to the dielectric boundary must be continuous, the following condition applies

$$E_{zd}(d)=E_{za}(d)\,. \tag{11}$$

Similarly, the tangential magnetic field must be continuous across the boundary

$$H_{dz}(d)=H_{az}(d)\,. \tag{12}$$

Equations (12) and (13) can be solved and result in the following transcendental equation

$$0=k_a\tan(k_d d)+k_d\tan(k_a(a-d))\,. \tag{13}$$

Note also that

$$C=A\frac{\cos(k_d d)}{\cos(k_a(a-d))}\,. \tag{14}$$

Solving equation (3) for k_a and k_d then substituting into equation (14) produces

$$0=\sqrt{k_0^2-\beta^2}\ \tan(d\sqrt{\epsilon_r k_0^2-\beta^2}\ )+\sqrt{\epsilon_r k_0^2-\beta^2}\ \tan((a-d)\sqrt{k_0^2-\beta^2}\ )\ . \quad (15)$$

Solutions for the propagation constant can then be found numerically. For a homogeneous waveguide operating in the TE_{10} mode β can be found to be

$$\beta=\sqrt{\left(\frac{2\pi f}{c}\right)^2\epsilon_{rd}-\left(\frac{\pi}{a_d}\right)^2}\ . \quad (16)$$

Equation (15) will produce a set of β's for different frequencies and geometries. By selecting a_d and ε_{rd} in equation (18) correctly, a homogeneous waveguide can be found that approximates β's frequency variation for a specific geometry.

The voltage-power impedance, Z_{ov}, is defined as

$$Z_{ov}=\frac{V^2}{2P}\ . \quad (17)$$

Since, this is a below-cutoff waveguide, no real power will propagate. The imaginary voltage-power impedance can still be defined by using the complex power [Ref. 8]. The complex power can be written as

$$\boldsymbol{P}=\int_0^b\left(\int_0^d E_{ya}H_{xa}^{*}dx+\int_d^a E_{yd}H_{xd}^{*}dx\right)dy\ . \quad (18)$$

There is no y variation, so the integration in y can be done immediately

$$\boldsymbol{P}=b\left(\int_0^d E_{ya}H_{xa}^{*}dx+\int_d^a E_{yd}H_{xd}^{*}dx\right)\ . \quad (19)$$

Substituting in the known quantities

$$\mathbf{P}=b\int_0^d \frac{j\ A\omega\mu\sin(k_d x)}{k_d}\,\frac{jA\beta\sin(k_d x)}{k_d}\,dx+$$

$$b\int_d^a \frac{-jA\omega\mu\cos(k_d d)\sin(ka(x-a))}{k_a\cos(ka(a-d))}\,\frac{jA\beta\cos(k_d d)\sin(k_a(x-a))}{k_a\cos(k_a(a-d))}\,dx. \tag{20}$$

Completing the integration gives

$$P=b\left(-\frac{A^2\omega\beta\mu\cos^2(k_d d)\tan(k_a(a-d))}{2k_a^3}+\frac{(a-d)A^2\omega\beta\mu}{2k_a^2\cos(k_a(a-d))}\right.$$
$$\left.-\frac{A^2\omega\beta\mu\sin(k_d d)\cos(k_d d)}{2k_d^3}+\frac{A^2 d\omega\beta\mu}{2k_d^2}\right). \tag{21}$$

Equation (21) can be more compactly written as

$$\mathbf{P}=bC_p \tag{22}$$

where C_p is the value of the terms in parenthesis in equation (21). Taking into account that the power is imaginary the average power can be written as

$$P=\frac{1}{2}\mathrm{IM}(\mathbf{P})\,. \tag{23}$$

The voltage is given by

$$V = -\int_0^b E_m \, dl = -bE_m \tag{24}$$

where E_m is the max field given found by finding the zero of the derivative of the y component of the electric field with respect to x

$$0 = \frac{d}{dx}\left(E_{ya} = \frac{-jA\omega\mu\cos(k_d d)\sin(k_a(x-a))}{k_a\cos(k_a(a-d))}\right). \tag{25}$$

This results in

$$0 = \cos(k_a(x-a)) \quad \rightarrow \quad x = (n+\frac{1}{2})\frac{\pi}{k_a}+a \qquad n = \ldots -1,0,1.. \tag{26}$$

which gives

$$E_m = \frac{-jA\omega\mu\cos(k_d d)}{k_a\cos(k_a(a-d))}. \tag{27}$$

The voltage-power impedance, Z_{ov}, can then be written as

$$Z_{ov} = \frac{V^2}{2P_{ave}} = bE_m^2\left(\frac{1}{C_p}\right). \tag{28}$$

The voltage-power impedance for the equivalent below cutoff waveguide can then be found as [Ref. 8]:

$$Z_{ov} = 120\pi\frac{2b_d}{a_d\sqrt{\epsilon_{rd}}}\frac{1}{\sqrt{(1-\frac{\lambda}{2a_d\sqrt{\epsilon_{rd}}}}}. \tag{29}$$

Solving equation (29) for b_d and substituting in the voltage-power impedance found in equation (28), defines the equivalent height required to produce the same voltage-power impedances for bot · ·eguides

$$b_d = \frac{bE_m^2}{C_p} \frac{\sqrt{4a_d^2 f^2 \epsilon_{rd} - c^2}}{480\pi f} . \tag{30}$$

APPENDIX B. MATLAB FUNCTIONS FOR HALF WAVEGUIDE MODEL

This appendix presents an explanation and listing of the MATLAB functions used to determine the model for the half waveguide. Variables defined as global_'name' are global variables which must be specified in the MATLAB shell. The purpose of these functions is to produce a set of equivalent a_d's and ε_{rd}'s that approximate the variation of β for a specific geometry.

The cutoff frequency was found using FIND_FC.M. FIND_FC.M uses the MATLAB function minimization routine to find the zeros of the transcendental equation contained in TRANS_FC.M.

The equivalent a_d's and ε_{rd}'s were found using MODEL_EQ.M. This function found the set of β's over the frequency of interest for a particular geometry by finding the zero's of the transcendental function found in Appendix A (equation 15). Equation (15) was implemented in TRANS.M. The equivalent dimensions were found by matching the β's just produced with β's calculated using test dimensions supplied by the function minimization routine. The routine selected the best choice of a_d's and ε_{rd}'s to fit the data.

The following functions relate to cutoff frequency calculations:

```
function fc=find_fc()
% Finds the cutoff frequency for the loaded half waveguide.
% The variable da refers to d/a and fc is the cutoff frequency.
% The following variables must be declared global. global_da

tol=1.0e-5;           % sets the accuracy of the solution
for da=.01:.01:.25    % iterates through da
  global_da=da;       % sets to global, pass to trans_fc
  freq=13e9;          % start of search
  freq=fzero('trans_fc',freq,tol)% finds solution to trans_fc
  fc=[fc; [ da freq/1e9]]% stores the solution
end
```

```
function y=trans_fc(x)
% finds the cutoff frequency.

beta=0;               % condition for cutoff
a=.02286/2;           % wr90 in meters
d=global_da*a;        % d dimensions
er=2.22;
ko=2*pi*x/3e8;        % wave numbers
kd=(er*ko^2+beta^2)^0.5;
ka=(ko^2+beta^2)^0.5;
                  % transcendental equation -> 0
global_yy=ka*tan(kd*d)+kd*tan(ka*(a-d));
y=global_yy;
end
```

The following functions relate to the computation of the equivalent dimensions:

```
function y=model_eq(pass__tol)

% global variables that must be declared: global_pass_tol,
% global_aeq, global_er, global_f, global_y_error global_yy
% g_cost

global_pass_tol=pass__tol; % sets tolerance throughout routine
tol=pass__tol;
freq=12;              % upper bound of matching
ii=1; x(ii,:)=[.01111 1.0701];% starting point for search
for i=.01:.01:.25     % iterates through geometries
  n=1;
  hhold=245;          % beginning point for beta
  global_d=i;
  for f=6:0.5:freq    % iterates through frequency
    global_f=f*1e9;   % pass frequency to global
                      % get beta at that freq
    hhold=fzero('trans',hhold,1e-6);
    g_cost(n,1)=hhold;% save the data point
    n=n+1;
  end
                  % find the aeq and er that match
  x(ii,:)=fmins('cost_eq',x(ii,:),tol)';
                  % the beta's held in the cost
                  % matrix below.
                  % save data and error
  y2(ii,:)=[i x(ii,:) global_yy global_y_error ];
  ii=ii+1;
  x(ii,:)=x(ii-1,:);
end
```

```
save y2
y=y2
end
-------------------------------------------------------------
function y=trans(x)
% This function finds the degree to which the transcendental
% equation goes to 0 for a given beta and dimension.

beta=x;              % the value being found
                 % constants
a=.02286/2; d=global_d*a; er=2.22;
ko=2*pi*global_f/3e8;% wave numbers
kd=(er*ko^2+beta^2)^0.5;
ka=(ko^2+beta^2)^0.5;
                 % transcendental equation
global_y=ka*tan(kd*d)+kd*tan(ka*(a-d));
y=global_y;
end
-------------------------------------------------------------

function y=cost_eq(x)

global_aeq=x(1);     % variables
global_er=x(2);

tol=global_pass_tol; % set tolerances
freq=12;             % set upper limit on matching
hhold=245;           % starting value for search
n=1;
for f=6:0.5:freq
  global_f=f*1e9;    % passes frequency to trans_eq.m
                 % finds the beta that solves for
                 % a particular aeq and er
  hhold=fzero('trans_eq',hhold,tol);
```

```
  g_cost(n,2)=hhold;% saves data in matrix
  n=n+1;
end
y1=0;
for i=1:n-1
                 % compares the beta vectors from
                 % load waveguide with homogeneous
                 % waveguide.  error is the square
                 % of the sum of the differences.
  y1=(g_cost(i,1)-g_cost(i,2))^2+y1;
end
y_error=y1;
y=y1;
end
------------------------------------------------------------
function y=trans_eq(x)
% function returns the beta for a specific homogeneous geometry
% at a specific frequency

beta=x;              % desired variable
ko=2*pi*global_f/3e8     % calculation of wave numbers
ka=(global_er*ko^2+beta^2)^0.5;
yy=tan(ka*(global_aeq)); % goes to zero to satisfy B.C.
y=yy;
end
```

APPENDIX C. EESOF MODEL AND MATLAB MODEL

This appendix describes the steps taken to develop the circuit model. The first step in the process was writing a MATLAB program that could replicate the performance of the CAD software. The CAD software available, and the target of this work, was the Touchstone software by EESOF. Following this discussion is a printout of the structure described in EESOF code, a typical EESOF data file and the MATLAB functions mentioned.

The approach taken to develop the MATLAB code was to represent each of the half waveguides as a π-equivalent circuit, as shown in Figure 15 (the dotted lines represent each half waveguide element). The nature of the structure lends itself most easily to use conductance for calculations. The value of the individual elements of the π-equivalent circuit for a waveguide can be found as

$$\begin{aligned} Y_{1a} &= \frac{Y_a}{\sinh(\gamma_a t)} \qquad & Y_{2a} &= Y_a \tanh(\gamma_a t/2) \\ Y_{1d} &= \frac{Y_d}{\sinh(\gamma_d t)} \qquad & Y_{2d} &= Y_d \tanh(\gamma_d t/2) \end{aligned} \tag{1}$$

where the subscript 'd' and 'a' refer to the loaded and unloaded half-waveguides. The initial implementation of the above algorithm was done in a function called STRUCTURE.M. This function and the same model design implemented in Touchstone was tested over a wide range of parameter values to ensure that the two programs produced the same results. In the interest of speed STRUCTURE.M was divided into two functions STRUCTURE_1.M and STRUCTURE_2.M. STRUCTURE_2.M computed values that were dependant only on geometry and

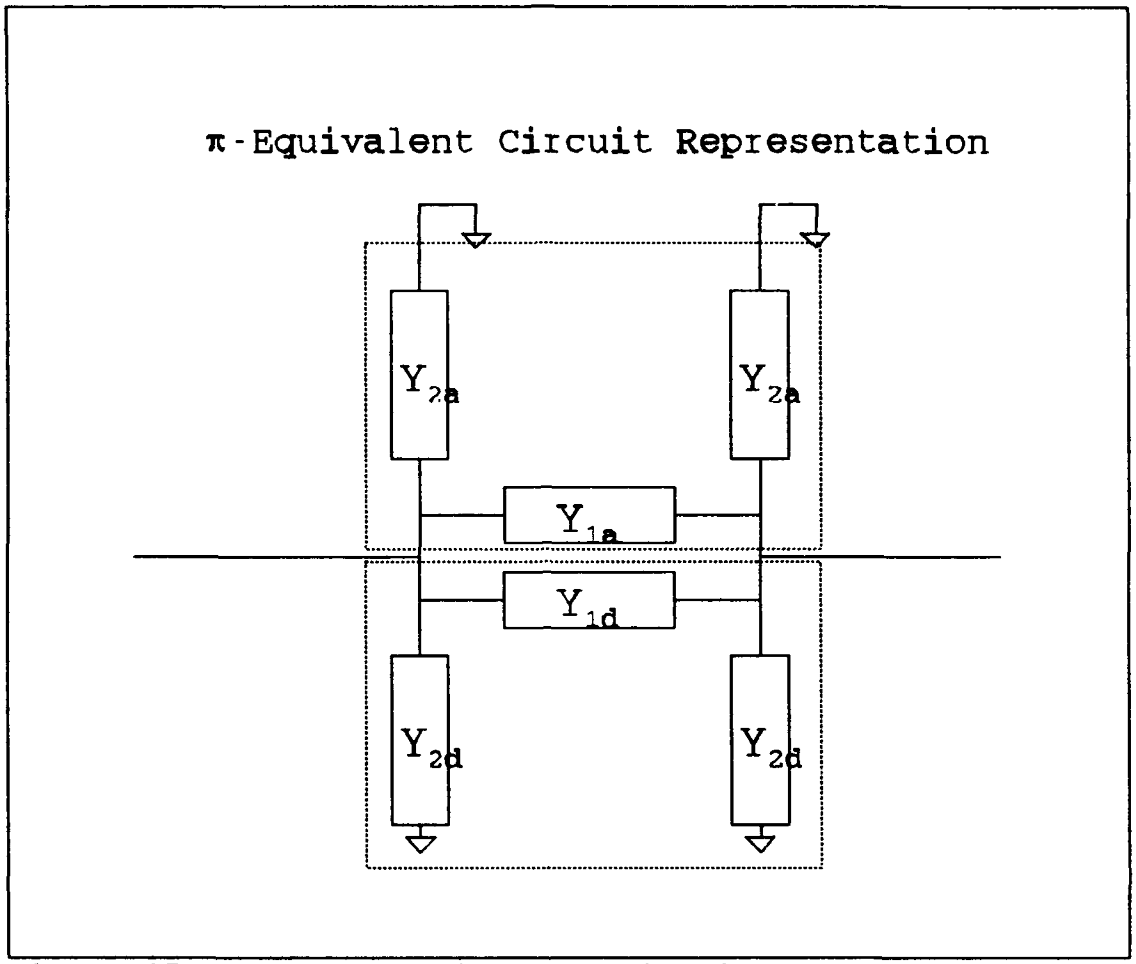

Figure 15. The π-equivalent circuit for the two half waveguides. The dotted lines separate each of the waveguides.

therefore, could be saved while the values in STRUCTURE_2.M depended on the element values and needed to be recalculated for each iteration.

The 'structure' functions were then called by a series of function minimization routines. Initially FIRST_SEARCH.M was used to get a rough idea of the progression of the element values. This program first searched over the coefficients of the, Y_0, polynomial to get a solution reasonably close to the values from STRIP. Then, another function minimization was done using all of the parameters. The purpose of this search regime was two fold: a search over six parameter is very time consuming and the results from one geometry to the next can be very discontinuous. By holding the value of the inductor and the capacitor fixed for the first search, the entire program ran faster and produced smoother results. After returning the 'best' guess solution of the parameters to the calling function, the parameter values were then stored and passed to the next iteration as the starting point of the search. The function STORE.M opened a file named STORAGE.MAT and save the results in a column form. Figure 16 contains a flow graph which summarizes the above discussion.

The performance of a function minimization is determined by the way the error is defined. The error, in this case, was

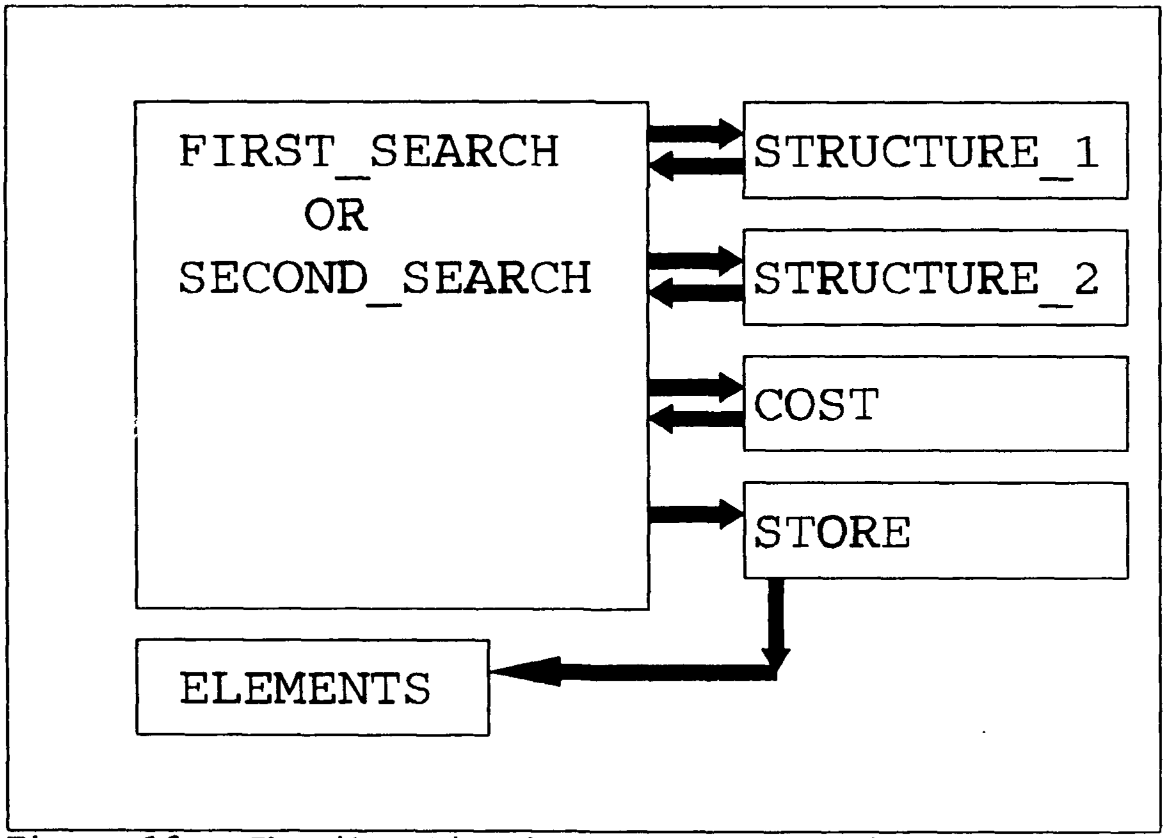

Figure 16. The iteration between the functions to produce element values can be seen above.

calculated, in the function CALC_ERROR.M using the square of the sum of the differences of the calculated values of the scattering parameters and the values generated by the STRIP program. Because both the phase and magnitude of, S_{11}, were considered equally important, the square of the difference of the two values was also taken and added to the error to force the errors to become evenly balanced. The following is an algebraic description error

$$e_m = (|S_{11s}| - |S_{11c}|)(|S_{11s}| - |S_{11c}|)^T \tag{2}$$

$$e_m = (\angle S_{11s} - \angle S_{11c})(\angle S_{11s} - \angle S_{11c})^T \tag{3}$$

where, S_{11s}, is a vector that contains values of the spectral domain data over the frequency set and, S_{11c}, is a vector that contains the values calculated by the model. The errors were combined in the following way

$$e_{total} = e_m + e_p + 10(e_m - e_p)\ . \tag{4}$$

After several runs of FIRST_SEARCH the values of elements were smooth enough to allow the calculation of equation to compute the element vales at specific geometries. The function MOD_ELEMENTS.M allowed the element values to be hand tuned to manually smooth the data prior to attempting to reduce the data. The coefficients were calculated using FIND_COEF.M and FIND_WB.M. These functions were placed into another function called MAKE_COEF.M This program iteratively called up the saved values of the elements and computed coefficients using a least mean square curve fitting routine. For a given, d/a, and, W/b, the second order polynomial for the, T/a, variation was found for each of the elements. These coefficients were then placed in column form. The set of data for the next, d/a, geometry was reduced until all the, d/a's, for a specific, W/b, geometry was computed. These coefficients were then treated as elements and converted into polynomials that depended on, d/a. These coefficients are placed in column form and the data for each, W/b, was then reduced. These coefficients were again treated as elements and polynomials were formed using the, W/b, dependance. By reversing the process in the functions GET_WB.M and GET_COEF.M, the values of the elements were regenerated. This program was tested extensively by starting with an arbitrary set of coefficients and generating elements then re-created the coefficients and verifying that the original coefficients were regenerated.

After producing a set of coefficients, the STRIP data was reapplied to the function SECOND_SEARCH.M. SECOND_SEARCH.M did the same thing as FIRST_SEARCH.M but, the initial guesses were produced by the coefficients and the function only went through one minimization routine that varied all of the element values. Figure 17 depicts the flow through the entire process.

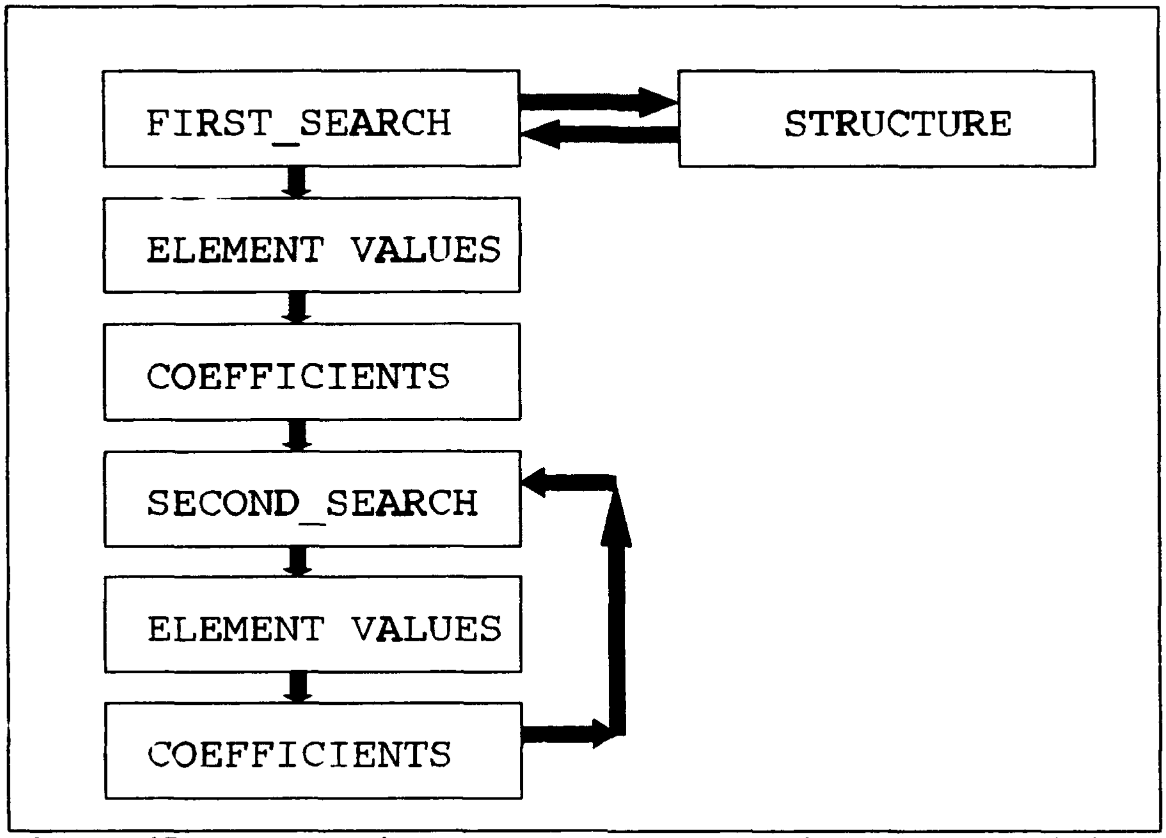

Figure 17. The entire process of developing the coefficient from the raw data is depicted above.

```
EESOF MODEL

! USER:       Michael Linzey
! DATE:       3 MAR 91
! CIRCUIT:    model

! COMMENT:    This is a model that represents the inductive strip
!             in finline.  The element values must be entered for a
!             specific geometry.

DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MIL
  TIME PS
  COND /OH
  ANG DEG

VAR
! GEOMETRY SPECIFIC
        a=900
        b=400
        W=200
        d=0
        T=10
! ELEMENTS
        l_d=12.6320
        c_d=.0064
        turn0=1.6020
        turn1=3.1142
        turn2=-2.5861
        turn3=0.1429

EQN
        PI=3.14159
        A_a=a/2

!MODEL
  balance=1
  ba=b/a
  wb=W/b
  dal=d*2/a
  da=d/a
! EQUIVELENT VALUES FOR THE LOAD HALF WAVEGUIDE
  er_dl=.999851+dal*.0149217-dal**2*.3391197
  er_d=er_dl+dal**3*10.61754-dal**4*3.664746+dal**5*.135657
  A_dl=1+dal*.001314-dal**2*.0302559+dal**3*.342420
  A_d=(A_dl-dal**4*2.161754+dal**5*1.7460658)*a/2
```

```
  c=2.998*10**8*1000/.0254
  mu=4*pi*10**(-7)/(1000/.0254)
  f=FREQ*10**9
  k0=2*pi*f/c
! beta presumed imaginary
  beta=sqrt(abs(er_d*k0**2-(pi/A_d)**2))
  kd=sqrt(2.22*k0**2+beta**2)
  ka=sqrt(k0**2+beta**2)

  CS=cos(d*kd)
  SN=sin(d*kd)
  zov_n=2*b*ka*kd**3*(2*pi*f)*mu*CS**2
  zov1=ka**3*(SN*CS-d*kd)*cos(A_d*ka-d*ka)**2
  zov2=kd**3*CS**2*sin(A_d*ka-d*ka)*cos(A_d*ka-d*ka)
  zov_d=(beta*(zov1+zov2-ka*kd**3*(A_d-d)*CS**2))
  zov=zov_n/zov_d

  B_d=-zov*sqrt(abs(4*A_d**2*er_d*f**2-c**2))/(480*f*pi)
! CALCULATES THE FINLINE EQUIVALENT DIMENSIONS
c1=-4.9723*ba**2+4.7413*ba-0.7651
Aeq=(2-(1-(ba+.45)*(1-wb)**2)**0.5+c1*(1-wb)**26)*a

c9=-20.16*da**2+6.42*da+.6494
er1=c9*(1-wb**(1-exp(-10*da)))
er=er1+wb+2.604*da+(1-da)**6*(1-wb)

c2=(-115.79*da**2+27.87*da-.4933)*ba+87.52*da**2-22.49*da-.1932
c3=0.29+0.0773*exp(1-40*da)
c4=(20.1154*da**2-3.729*da-0.0611)*ba+(-26.1788*da**2+5.537*da+1.0376)
c5=-13.5217*da**2+2.4017*da+0.0411
Beq_ave1=c2*(1-wb**(2*ba*c3))+c4+c5*(1-abs(ba-wb)**2)**4
Beq_ave=(Beq_ave1-0.025*(1-abs(.925-wb)**2)**16)*b
c6=(-76.251*(da)**2+17.23*da-.1578)*ba+111.2*da**2-20.84*da-.2936
c7=(64.82*da**2-14.77*da-.3029)*ba-107.1*da**2+22.85*da-.2936
c8=(9.696*da**2-1.449*da-.1431)*ba-12.13*da**2+1.39*da+.1195
m=c6*wb**2+c7*wb+c8
fc=(29980/2.54)/(2*a)
Beq=m*(FREQ/fc-1.56)*b+Beq_ave

!IMPEDANCE TRANSFORMER
  Lambda=(29980/2.54)/FREQ
  LpovL1=1/(1-(Lambda/(2*Aeq*er**0.5))**2)**0.5
  LpovL2=1/(1-(Lambda/(2*A))**2)**0.5
  Z1=120*PI*(2*Beq/Aeq)*LpovL1/er**0.5
  f_n=(FREQ-fc)/fc
  Z_d=1000/abs(turn0+turn1*f_n+turn2*f_n**2+turn3*f_n**3)
  Z_a=Z_d/balance

  X_a=(Z1/Z_a)**0.5
  X_d=(Z1/Z_d)**0.5
```

```
CKT
        IND       1 0 L^L_d
        CAP       1 0 C^C_d
        RWG       1 2 A^A_a   B^B L^T   ER=1   RHO=1
        IND       2 0 L^L_d
        CAP       2 0 C^C_d
        DEF2P     1 2 Air

        IND       1 0 L^L_d
        CAP       1 0 C^C_d
        RWG       1 2 A^A_d B^B_d L^T   ER^er_d   RHO=1
        IND       2 0 L^L_d
        CAP       2 0 C^C_d
        DEF2P     1 2 Dielec

        XFER      1 2 0 0 N^X_a
        XFER      1 3 0 0 N^X_d
        Air       2 4
        Dielec    3 5
        XFER      6 5 0 0 N^X_d
        XFER      6 4 0 0 N^X_a
        DEF2P     1 6 STRIP

        RWGT      1   A^Aeq B^Beq ER^er RHO=1
        DEF1P     1   WEDGE
!SPECTRAL DOMAIN DATA
        S2PA      1 2 0    t50d30w5.S2P
        DEF2P     1 2      SPEC_DOM

        SPEC_DOM 1 2
        DEF2P   1 2   SPEC_M

TERM
        STRIP WEDGE WEDGE
PROC

OUT
        STRIP    MAG[S11] GR1
        STRIP    ANG[S11] GR1A

        SPEC_DOM MAG[S11] GR1
        SPEC_DOM ANG[S11] GR1A

FREQ
        SWEEP    8        12       1

GRID
        RANGE    8        12       1
        GR1      0        1       .1
        GR1A     80       180
```

The following listing illustrates the form of the spectral domain input data file for the iductive strip model. This is the listing called by the model on the previous page.

```
!Spectral Domain data file for the following geometry
!  T=50   d=30   W=200

!Freq  |S11|   <S11     |S12|  <S12     |S21|  <S21      |S22|  <S22
 8.0   0.8992 146.6191  0.1914 56.6191  0.1914 56.6191  0.8992 146.6191
 9.0   0.8528 139.1836  0.2727 49.1836  0.2727 49.1836  0.8528 139.1836
10.0   0.8056 132.0117  0.3510 42.0117  0.3510 42.0117  0.8056 132.0117
11.0   0.7568 126.7910  0.4272 36.7910  0.4272 36.7910  0.7568 126.7910
12.0   0.7093 121.3066  0.4968 31.3066  0.4968 31.3066  0.7093 121.3066
13.0   0.6610 115.7168  0.5630 25.7168  0.5630 25.7168  0.6610 115.7168
```

The following is a listing of MATLAB functions used in element calculations:

```
function [f,mag,ang]=structure(f1,a,b,d,W,t,l_d,c_d,t0,t1,t2,t3)
% This function replicates the performance of TSTONE software for the
% configuration under consideration.
                        % calculates constants
a_a=a/2; er_a=1; wb=W/b;
fc=2.998e8*1e-9*1000/.0254/(2*a)
                        % iterates through frequencies
for n=1:length(f1)
  freq=f1(n);           % get first freq
  f=freq*1e9;
                        % get equivalent loaded waveguide
  [a_d,b_d,er_d]=half_guide(a,b,d,f);

  y0_a=yov(a_a,b,er_a,f);     % calculate the susceptance of
  y0_d=yov(a_d,b_d,er_d,f);   % the two guides
  gamma_a=real(j*beta(f,a_a,er_a)); % calculate the prob. const.
  gamma_d=real(j*beta(f,a_d,er_d));

  fn=(freq-fc)/fc;             % calculate the susptance used in
  y0_f=0.001*abs(t0+t1*fn+t2*fn^2+t3*fn^3);% turns ratio

  y0_a=y0_a/y0_f;              % normalizes the impedance
  y0_d=y0_d/y0_f;
                               %  pi equivilant for waveguide
  ya=y0_a/sinh(gamma_a*t)+y0_d/sinh(gamma_d*t);
  yb=y0_a*tanh(gamma_a*t/2)+y0_d*tanh(gamma_d*t/2);
                               % tank cuircuits
  tank_a=(j*2*pi*f*c_d+1/(j*2*pi*f*l_d))/y0_f;
  tank_d=(j*2*pi*f*c_d+1/(j*2*pi*f*l_d))/y0_f;
  yb=yb+tank_a+tank_d;         % form the complete pi equivilant
  y_eq=1/(1/(1+yb)+1/ya)+yb;  % find the equivilent suseptance
  s11=(1-y_eq)/(1+y_eq);       % calculate s11

  g(n,:)=[f*1e-9 abs(s11) angle(s11)*180/pi];
end
f=g(:,1);
mag=g(:,2);
ang=g(:,3);
```

```
function structure_1(f1,a,b,d,W,t)
% This function coupled with STRUCTURE_2 produce the same results as
% STRUCTURE.  STRUCTURE_1 and STRUCTURE_2 together repoduce the
% performance of TSTONE for the device under concideration.  STRUCTURE_1
% computes the geometry specific parts of STRUCTURE to prevent
% for each frequency iteration.

                      % constants
a_a=a/2; n=0; er_a=1; wb=W/b;
fc=1e-9*(2.998e8*1000/.0254)/(2*a);
                      % iterate through the frequencies
for ifreq=1:length(f1)
  freq=f1(ifreq);     % select first freq (in GHz)
  f=freq*1e9;
  [a_d,b_d,er_d]=half_guide(a,b,d,f);      % get the equivilent loaded guide
  y0_a(ifreq)=yov(a_a,b,er_a,f);           % calculate the impedances for
  y0_d(ifreq)=yov(a_d,b_d,er_d,f);         % the two guides
  gamma_a=real(j*beta(f,a_a,er_a));        % calculate the probigation
  gamma_d=real(j*beta(f,a_d,er_d));        % constants
  gamma_check(ifreq)=gamma_d;              % used to check if guide starts
                                 % to probigate
  sinh_a(ifreq)=sinh(gamma_a*t);           % calculates need trig fucntions
  sinh_d(ifreq)=sinh(gamma_d*t);
  tanh_a(ifreq)=tanh(gamma_a*t/2);
  tanh_d(ifreq)=tanh(gamma_d*t/2);
end
f=f1;
                      % saves calculations to CONSTANTS
save constants f y0_a y0_d gamma_check fc sinh_a sinh_d tanh_a tanh_d

----------------------------------------------------------------------

function [f,mag,ang]=structure_2(l_d,c_d,t0,t1,t2,t3)
% This function completes the calcuations started in STRUCTURE_1; includes
% the effects of the elements.
load constants    % gets constants from STRUCTURE_1
l_d=abs(l_d); c_d=abs(c_d); n=1; f1=f;  % takes the abs of L and C to
                  % prevent the use of negative #'s
for freqi=1:length(f)               % iterates through frequency
 if gamma_check(n)==0,              % checks to see if the guide is
                  % probigating, skips if yes
 else
  freq=f1(freqi); f=freq*1e9; w=2*pi*f; % variables
                  % calculates transformed impedance
  f_n=(freq-fc)/fc;
  y0_f=0.001*abs(t0+t1*f_n+t2*f_n^2+t3*f_n^3);
                      % Normalizes the suseptance of
                      % guides
  y0_a(n)=y0_a(n)/y0_f;
  y0_d(n)=y0_d(n)/y0_f;
```

```
                          % pi-equivelant of guides
  ya=y0_a(n)/sinh_a(n)+y0_d(n)/sinh_d(n);
  yb=y0_a(n)*tanh_a(n)+y0_d(n)*tanh_d(n);
                          % suseptance of the tank circuits
  tank_a=(j*w*c_d+1/(j*w*l_d))/y0_f;
  tank_d=(j*w*c_d+1/(j*w*l_d))/y0_f;
                          % total circuit
  yb=yb+tank_a+tank_d;
                          % eqivilent suseptance
  y_eq=1/(1/(1+yb)+1/ya)+yb;
                          % calculate s11
  s11=(1-y_eq)/(1+y_eq);
  g(n,:)=[f*1e-9 abs(s11) angle(s11)*180/pi];
  n=n+1;
 end
end
f=g(:,1);
mag=g(:,2);
ang=g(:,3);
```

The following functions support the 'structure' functions:

```
function b=beta(f,a,er)
%calcualtes beta for a given frequency for the TE10 mode.  Assumes that
%a is measured mils.

c=2.998e8*1000/.0254;
k=2*pi*f*(er)^.5/c;
kc=pi/a;
b=(k^2-kc^2)^.5;
end
```

```
function [a_d,b_d,er_d]=half_guide(a,b,d,f)
%This function computes the effective dimentions of a the below cutoff
%waveguide section that contains dielectric.  The model assumes that
%the dielectric is er=2.22.
                       % constants
c=2.998e8*1000/.0254; mu=pi*4e-7/(1000/.0254); d=d*2/a;
                       % calc er
er_d=.999851+d*.0149217-d^2*.3391197+d^3*10.617549-d^4*3.664746+d^5*.135657
;
a_d=a/2*(1+d*.001314-d^2*.0302559+d^3*.342420-d^4*2.161754+d^5*1.7460658);
                       % calculate Zov
k0=2*pi*f/c;
beta=sqrt(er_d*k0^2-(pi/a_d)^2);
kd=sqrt(2.22*k0^2-beta^2);
ka=sqrt(k0^2-beta^2);
```

```
CS=cos(d*kd);
SN=sin(d*kd);
zov_n=2*b*ka*kd^3*(2*pi*f)*mu*CS^2;
zov1=ka^3*(SN*CS-d*kd)*cos(a_d*ka-d*ka)^2;
zov2=kd^3*CS^2*sin(a_d*ka-d*ka)*cos(a_d*ka-d*ka);
zov_d=(beta*(zov1+zov2-ka*kd^3*(a_d-d)*CS^2));
zov=zov_n/zov_d;
b_d=real(-zov*sqrt(4*a_d^2*er_d*f^2-c^2)/(480*f*pi));
```

```
function y=yov(a,b,er,f)
% calculates voltage power suseptance
fc=(1000/.0254)*2.998e8/(2*a*er^0.5);
lambda=(2.998e8*1000/.0254)/f;
z=(376.7/er^0.5)*(2*b/a)/(1-(fc/f)^2)^0.5;
y=(1/z);
```

The following functions are connected with the search routines:

```
function error=calc_error(mag,s_mag,phase,s_phase,f,x)
% This function caclulates the error between the two input magnitudes
% and phases.
s_mag=s_mag(1:length(mag));      % sets the length of the vectors to be
s_phase=s_phase(1:length(mag));  % the same, needed in the case of
                    % propagation less than 13GHz
m=(mag-s_mag);
p=(phase-s_phase);
for i=1:length(m)                % finds the relative error at each point
  m(i)=m(i)/s_mag(i);
  p(i)=p(i)/s_phase(i);
end
global_m=max(abs(m))*100;
global_p=max(abs(p))*100;
error_mag=m'*m;
error_ph=p'*p;
                    % calculates the error returned
error=error_mag+error_ph+10*(error_mag-error_ph)^2;
                    % viewing options when running in
                    % foreground
%[global_m global_p x']
%plot(f,s_mag,'r',f,s_phase/180,'g',f,mag,'*r',f,phase/180,'*g')
----------------------------------------------------------------------

function error=model(x)
% This function takes input from the matlab function minimization
% routine passes values to STRUCTURE_2.  STRUCTURE_2 calculates the
% response and return the values of the S11 scattering parameters.
% CALC_ERROR then compares the two responses and calculates an error
% which is returned to the function minimization routine, for use in the
% production of the next guess.

load s11_data_m
l_d=x(1)*1e-9;          % element values
c_d=x(2)*1e-12;
turn0=x(3);
turn1=x(4);
turn2=x(5);
turn3=x(6);
                    % finds response for given element values
[f mag phase]=structure_2(l_d,c_d,turn0,turn1,turn2,turn3);
                    % calculates error
error=calc_error(mag,s_mag,phase,s_phase,f,x);

----------------------------------------------------------------------

function error=model_2(x)
% This function takes input from the matlab function minimization
% routine passes values to STRUCTURE_2.  STRUCTURE_2 calculates the
```

```
% response and return the values of the S11 scattering parameters.
% CALC_ERROR then compares the two responses and calculates an error
% which is returned to the function minimization routine, for use in the
% production of the next guess.
load s11_data_m
load g_hold
l_d=g_hold(1)*1e-9;        % element values
c_d=g_hold(2)*1e-12;
turn0=x(1);
turn1=x(2);
turn2=x(3);
turn3=x(4);
                  % finds response for given element values
[f mag phase]=structure_2(l_d,c_d,turn0,turn1,turn2,turn3);
                  % calculates error
error=calc_error(mag,s_mag,phase,s_phase,f,x);
------------------------------------------------------------------------

function g=first_search(g_init,tol,begin)
% This function finds and stores element values that match the scattering
% parameters found in the file S11_DATA.  The data in S11_DATA is in
% column form as follows:
%          [ freq, s11_mag, S11_phase, t, d, W]
% The data is first broken in to blocks that represent each geometry.  The
% data is then saved in S11_data_m for recall by the function minimization
% routines.  STRUCTURE_1 then calculates the geometry dependant parameters.
% Initail starting points for element values are eather given by the
% argument or from the previous iteration.  Function minimiztion routines
% are then called to find values of the elements that match the S11_DATA.

% The variables global_m and global_p must be declared to be global.

                      % initial values
g=g_init; g_select=1; d_hold=0; a=900; b=400; n=0;
load s11_data
                      % block data by geometry, using the
                      % starting frequency
for i=1:length(s11_data)
  if 7 == s11_data(i,1),
    n=n+1;
    block(n)=i;       % record begining of each block
  end
end
block(n+1)=length(s11_data)+1;              % mark the end of the last block
for i=begin:length(block)-1                  % iterate through all the data
  i_end=block(i+1)-1;  % calculate the end of the block
  f=s11_data(block(i)+1:i_end,1);            % block out the data
  s_mag=s11_data(block(i)+1:i_end,2);
  s_phase=s11_data(block(i)+1:i_end,3);
  t=s11_data(block(i),4);
  d=s11_data(block(i),5);
```

```
  W=s11_data(block(i),6);
  i_end=length(s_mag);
  for k=length(s_mag)-1:-1:2                         % test to see if magnitude of
    if s_mag(k)<s_mag(k+1)                           % s11 is increasing
      i_end=k;
    end
  end
  f=f(1:i_end);           % eliminate the bad points noted
  s_mag=s_mag(1:i_end);                              % above
  s_phase=s_phase(1:i_end);
  [d t W]                 % display block being worked on
  [f s_mag s_phase]
  save s11_data_m
  structure_1(f,a,b,d,W,t);                         % calculate geometry dependant
                          % values
  g_hold=g                % save guess of element values
  save g_hold g_hold
                          % pass turns ratio into function
                          % minimization routine
  g_hold=fmins('model_2',[g(3) g(4) g(5) g(6)],0.10);
                          % re-form element vector
  g=[g(1) g(2) g_hold(1) g_hold(2) g_hold(3) g_hold(4)]
  g_hold=g
  [global_m global_p]
  save g_hold g_hold
  [g,count]=fmins('model',g,tol);            % pass all elements into function
                        % miniminzation routine
  store(a,b,W,d,t,tol,g',count/100)          % save the element values
  if t<15,
    g_save=g            % save the value of the begining
  end                   % of one d set to start the next
  if t>950
    g=g_save
  end
end

---------------------------------------------------------------------

function g=second_search(tol,begin)
% Operation of this function is the same as the FIRST_SEARCH except
% in the way the initial guess of the elements is found.  As note
% below.

n=0; a=900; b=400;
load s11_data
for i=1:length(s11_data)
  if 7 == s11_data(i,1),
    n=n+1;
    block(n)=i;
  end
end
```

```
block(n+1)=length(s11_data)+1;
for i=begin:length(block)-1
  i_end=block(i+1)-1;
  f=s11_data(block(i)+1:i_end,1);
  s_mag=s11_data(block(i)+1:i_end,2);
  s_phase=s11_data(block(i)+1:i_end,3);
  t=s11_data(block(i),4);
  d=s11_data(block(i),5);
  W=s11_data(block(i),6);
  i_end=length(s_mag);
  for k=length(s_mag)-1:-1:2
    if s_mag(k)<s_mag(k+1)
      i_end=k;
    end
  end
  f=f(1:i_end);
  s_mag=s_mag(1:i_end);
  s_phase=s_phase(1:i_end);
  [d t W]
  [f s_mag s_phase]
  save s11_data_m
  structure_1(f,a,b,d,W,t);
                    % gets the elements as a function
                    % of parameters
  [L,C,t0,t1,t2,t3]=get_elements(a,b,d,W,t);
  g=[L C t0 t1 t2 t3]
  [g,count]=fmins('model',g,tol);
  store2(a,b,W,d,t,tol,g',count/100)
  if t<15,
    g_save=g
  end
  if t>950
    g=g_save
  end
end
```

The following functions relate to coefficient reduction and element generation.

```
function coef_d=find_coef(s)
% This function find the coefficients of the polynomials using MATLAB's
% mean square function.
Nt=2; Nd=2; Nb=2;
s(10:11,:)=abs(s(10:11,:));
s=[s zeros(16,1)];
for n=1:length(s(5,:))-1
  if s(5,n)>s(5,n+1)
    Ne=n;
    for i=10:15         % finds the coef of the elements
                        % with respect to T/a
      coef_t=[coef_t; polyfit(s(5,Nb:Ne)/s(1,1),s(i,Nb:Ne),Nt)' ] ;
    end
    coef_1=[coef_1 coef_t]                    % puts the results in a column
    d=[d s(4,Nb)/s(1,1)]                      % makes d/a list to reduce the
                        % above list
    coef_t=[];
    Nb=Ne+1;
  end
end
[row,col]=size(coef_1);
for n=1:row             % finds the coef of the coef
                        % generated above wrt d/a
  coef_d=[coef_d; polyfit(d,coef_1(n,:),Nd)' ]
end
```

```
function coef_wb=find_wb(wb,coef_in)
% Finds the coefficients with respect to W/b
[r,c]=size(coef_in);
Nw=2;
for i=1:r
  coef_wb=[coef_wb; polyfit(wb,coef_in(i,:),Nw)];
end
```

```
function [L,C,t0,t1,t2,t3]=get_elements(a,b,d,W,t)
% This function used the values of the coefficients in the file COEF
% to calculate the element values.

load coef
Nd=2;                    % the order of the d/a polynomial
                                     % needed for initial decomposition
coef_d=get_wb(Nt,coef,W/b);          % accounts for the W/b variation
[r,c]=size(coef_d);
N=r/6; da=d/a;
for i=1:N           % iterates throught the coef; accounts for
                    % the d/a variation
  coef_L=[ coef_L polyval(coef_d(i,:),da) ];
  coef_C=[ coef_C polyval(coef_d(N+i,:),da) ];
  coef_t0=[ coef_t0 polyval(coef_d(2*N+i,:),da) ];
  coef_t1=[ coef_t1 polyval(coef_d(3*N+i,:),da) ];
  coef_t2=[ coef_t2 polyval(coef_d(4*N+i,:),da) ];
  coef_t3=[ coef_t3 polyval(coef_d(5*N+i,:),da) ];
end
                    % computes the value of the elements
L=polyval(coef_L,t/a);
C=polyval(coef_C,t/a);
t0=polyval(coef_t0,t/a);
t1=polyval(coef_t1,t/a);
t2=polyval(coef_t2,t/a);
t3=polyval(coef_t3,t/a);
----------------------------------------------------------------------
function coef_d=get_wb(Nb,coef_wb,wb)
% Account for the W/b variation
for i=1:length(coef_wb)
  c0(i)=polyval(coef_wb(i,:),wb);   % calculates the W/b variation
end
for i=1:Nd+1:length(c0)             % resizes the matrix for next
  c1=[ c1; c0(i:i+Nd) ];            % operation
end
coef_d=c1;
```

APPENDIX D. COEFFICIENT REDUCTION OUTPUT

Following this page are three sample outputs that illustrate the difficulties encountered during the development of the coefficients which were to represent the element values. The first listing shows the results of a function minimization where the inductance was allowed to vary and the capacitance and the four terms of the turns ratio were calculated by the coefficients. The second listing shows the percent error in the desired value of the inductance and the value produced from the least mean square generating programs. The third listing shows the percentage error in phase and magnitude with the STRIP data when the coefficients calculated above were used to generate element values in the model.

The following listing resulted from generating element values for all of the elements except the inductor using coefficients and then using a function minimization routine to generate the inductor value. The columns are arranged as follows:

a
b
W
d
T
tolerance
(search count)*0.01
% phase error
% mag error
L   inductor in nH
    C   capacitor in pF
t0 0 order coefficient for the turns ratio
t1 1st order coefficient for the turns ratio
t2 2nd order coefficient for the turns ratio
t3 3rd order coefficient for the turns ratio

Columns 1 through 7

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
0	0	0	0	0	0	0
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
2.6978	1.8785	0.9913	0.9183	0.8394	1.4123	2.0688
0.4667	0.5811	0.8052	1.0324	1.2725	1.6186	1.9703
17.6283	17.0645	16.7005	16.4661	16.3048	16.1440	16.0190
0.0047	0.0049	0.0052	0.0054	0.0057	0.0060	0.0063
1.5539	1.4595	1.3682	1.2798	1.1946	1.1123	1.0331
2.5056	2.9212	3.3246	3.7159	4.0949	4.4617	4.8164
1.5626	0.5833	-0.3683	-1.2922	-2.1885	-3.0571	-3.8981
-3.2824	-2.6831	-2.1004	-1.5344	-0.9851	-0.4524	0.0636
0	0	0	0	0	0	0

Columns 8 through 14

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
0	0	0	0	0	0	0
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.1000	0.1000	0.0800	0.0800
2.5424	1.1586	1.1907	0.6524	0.9807	3.7443	2.2694
2.3146	0.3406	0.6124	0.9522	0.8626	0.0775	0.2237
15.8997	16.1901	16.1876	16.5842	15.8695	13.2978	9.6489
0.0065	0.0068	0.0071	0.0101	0.0136	0.0174	0.0216
0.9570	0.8838	0.8137	0.2799	0.0499	0.1236	0.5011
5.1589	5.4892	5.8073	8.3180	9.6100	9.6833	8.5378
-4.7114	-5.4970	-6.2549	-12.3128	-15.6042	-16.1289	-13.8871
0.5630	1.0457	1.5117	5.2560	7.3344	7.7469	6.4936
0	0	0	0	0	0	0

Columns 15 through 21

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
2.7920	1.9233	1.2957	0.5927	0.5989	0.5894	0.7307
1.0093	1.1187	1.2459	1.3433	1.4925	1.6206	1.7332
15.4365	14.7504	14.2990	13.9707	13.7161	13.5044	13.3290
0.0047	0.0048	0.0049	0.0050	0.0051	0.0052	0.0054
1.4917	1.4271	1.3647	1.3045	1.2465	1.1907	1.1370
3.2785	3.4819	3.6807	3.8750	4.0648	4.2501	4.4309
-0.6303	-1.1668	-1.6912	-2.2036	-2.7039	-3.1922	-3.6683
-1.8272	-1.4472	-1.0769	-0.7162	-0.3651	-0.0237	0.3081
0	0	0	0	0	0	0

Columns 22 through 28

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0800	0.0800	0.0900
0.7781	0.8612	0.9399	0.9337	0.1064	0.0487	0.0699
1.7554	1.8820	1.9826	3.8107	1.9144	1.6289	0.7899
13.1603	13.0097	12.8710	11.5923	10.9954	9.1540	7.1753
0.0055	0.0057	0.0058	0.0079	0.0109	0.0148	0.0196
1.0855	1.0362	0.9890	0.6374	0.5035	0.5874	0.8890
4.6071	4.7788	4.9460	6.3695	7.3412	7.8610	7.9289
-4.1324	-4.5845	-5.0244	-8.7601	-11.2889	-12.6106	-12.7254
0.6303	0.9428	1.2456	3.7443	5.2792	5.8502	5.4575
0	0	0	0	0	0	0

Columns 29 through 35

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
1.0938	1.0476	1.1273	0.9510	0.9825	0.9362	1.0262
0.4272	0.6167	0.7064	0.7250	0.6569	0.5621	0.3492
15.4914	14.9184	14.5766	14.3525	14.1980	14.0834	14.0077
0.0049	0.0048	0.0047	0.0046	0.0045	0.0045	0.0045
1.4964	1.4717	1.4480	1.4251	1.4032	1.3821	1.3620
4.3038	4.2170	4.1360	4.0607	3.9911	3.9273	3.8692
-3.7174	-3.6376	-3.5668	-3.5051	-3.4525	-3.4091	-3.3747
0.1571	0.2234	0.2898	0.3563	0.4231	0.4899	0.5569
0	0	0	0	0	0	0

Columns 36 through 42

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0800	0.0800	0.0800	0.0800
1.0220	0.9504	1.0565	0.6479	0.3448	0.2139	0.1880
0.1549	-0.1042	-0.3019	0.5094	0.7133	0.5648	3.6981
13.9314	13.8818	13.8258	13.2444	11.7661	9.7564	7.8159
0.0045	0.0045	0.0045	0.0055	0.0081	0.0123	0.0181
1.3428	1.3245	1.3071	1.1833	1.1507	1.2092	1.3589
3.8168	3.7701	3.7292	3.6353	4.1144	5.1667	6.7920
-3.3495	-3.3334	-3.3263	-3.7568	-5.0978	-7.3493	-10.5114
0.6241	0.6914	0.7588	1.4416	2.1394	2.8521	3.5799
0	0	0	0	0	0	0

Columns 43 through 49

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
50.0000	50.0000	50.0000	50.0000	50.0000	50.0000	50.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
0.4114	0.5845	0.7866	0.8306	0.9956	1.0635	1.0886
0.6316	0.9808	1.1108	1.2071	1.1559	1.0530	0.8981
15.5422	14.9465	14.5972	14.3616	14.2099	14.1129	14.0423
0.0050	0.0049	0.0049	0.0048	0.0047	0.0047	0.0047
1.6733	1.6621	1.6512	1.6406	1.6302	1.6200	1.6101
4.6350	4.4373	4.2488	4.0694	3.8993	3.7384	3.5867
-5.0731	-4.7357	-4.4152	-4.1116	-3.8249	-3.5552	-3.3023
0.9066	0.8258	0.7492	0.6769	0.6088	0.5450	0.4855
0	0	0	0	0	0	0

Columns 50 through 56

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
50.0000	50.0000	50.0000	50.0000	50.0000	50.0000	50.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0800	0.0800	0.0800	0.0800
1.1351	1.1735	1.1892	1.2297	1.5884	0.9332	0.9515
0.7281	0.4689	0.2374	0.5446	0.7871	0.1534	2.2812
13.9922	13.9515	13.9251	13.4752	12.1279	10.1444	7.9998
0.0047	0.0047	0.0048	0.0060	0.0090	0.0136	0.0198
1.6004	1.5910	1.5819	1.5040	1.4512	1.4235	1.4208
3.4442	3.3108	3.1867	2.4512	2.6354	3.7394	5.7630
-3.0665	-2.8475	-2.6455	-1.5561	-2.1593	-4.4551	-8.4436
0.4303	0.3793	0.3325	0.0999	0.2939	0.9145	1.9617
0	0	0	0	0	0	0

Columns 57 through 63

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
0.9719	0.4383	0.5465	0.6036	0.7790	1.0208	1.0358
0.6034	0.7000	0.7568	0.7414	0.7860	0.7618	0.6843
15.3784	14.6776	14.2433	13.9576	13.7379	13.5720	13.4432
0.0052	0.0053	0.0054	0.0056	0.0057	0.0059	0.0061
2.0224	1.9983	1.9745	1.9509	1.9275	1.9043	1.8812
4.2721	4.1426	4.0190	3.9013	3.7895	3.6835	3.5834
-4.6973	-4.4611	-4.2364	-4.0230	-3.8210	-3.6304	-3.4512
0.4213	0.3600	0.3013	0.2454	0.1922	0.1417	0.0939
0	0	0	0	0	0	0

Columns 64 through 70

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0800	0.0800	0.0800	0.0800
1.1151	1.2713	1.2846	1.2579	0.7262	0.5771	0.4570
0.6889	0.6666	0.6126	0.6307	1.7171	2.5476	2.6323
13.3299	13.2318	13.1585	12.5084	11.8911	10.5411	8.9369
0.0063	0.0065	0.0067	0.0094	0.0134	0.0185	0.0248
1.8584	1.8357	1.8133	1.5995	1.4051	1.2302	1.0749
3.4892	3.4009	3.3185	2.8174	2.9043	3.5791	4.8419
-3.2833	-3.1269	-2.9819	-2.1580	-2.4734	-3.9280	-6.5218
0.0488	0.0064	-0.0332	-0.2808	-0.2573	0.0373	0.6029
0	0	0	0	0	0	0

Columns 71 through 77

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
90.0000	90.0000	90.0000	90.0000	90.0000	90.0000	90.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0800
2.2465	0.8248	0.6130	1.0333	1.4566	1.7532	1.9895
1.1545	1.2252	1.3701	1.4892	1.6555	1.9017	2.0748
15.1502	14.2180	13.5937	13.1212	12.7360	12.4109	12.1343
0.0054	0.0059	0.0064	0.0070	0.0075	0.0080	0.0086
2.5436	2.4804	2.4179	2.3561	2.2952	2.2349	2.1754
3.2151	3.3331	3.4468	3.5563	3.6615	3.7625	3.8594
-2.5900	-2.8140	-3.0304	-3.2393	-3.4408	-3.6347	-3.8212
-1.2988	-1.1741	-1.0538	-0.9381	-0.8268	-0.7201	-0.6179
0	0	0	0	0	0	0

Columns 78 through 84

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
90.0000	90.0000	90.0000	90.0000	90.0000	90.0000	90.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
0.0800	0.0800	0.0800	0.0800	0.0800	0.0800	0.0900
3.5099	3.7978	3.9986	3.0324	2.1798	0.6948	0.3984
1.4109	1.5650	1.7247	-0.6245	0.1249	2.7824	6.8887
12.5358	12.2834	12.0626	10.5341	9.7944	10.0006	9.9825
0.0091	0.0097	0.0102	0.0157	0.0213	0.0271	0.0330
2.1166	2.0586	2.0014	1.4696	1.0123	0.6294	0.3210
3.9520	4.0403	4.1245	4.7338	4.9209	4.6859	4.0288
-4.0001	-4.1715	-4.3355	-5.5627	-6.0402	-5.7679	-4.7460
-0.5203	-0.4271	-0.3385	0.2995	0.4859	0.2205	-0.4965
0	0	0	0	0	0	0

The following is a partial listing of the output of a MATLAB function that gives the percent error in the desired element value and the value calculated from the coefficients at specific geometries. The data is arranged in columns, the first five places represent, a,b,W,d,T and the last five places represent the element values. As discussed above only the inductor shows any significant deviation from the desired values.

```
element_error =

  Columns 1 through 7

  900.0000  900.0000  900.0000  900.0000  900.0000  900.0000  900.0000
  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000
  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000
         0         0         0         0         0         0         0
   10.0000   20.0000   30.0000   40.0000   50.0000   60.0000   70.0000
  -17.3010  -13.6287  -10.9892   -9.1597   -7.9007   -6.8295   -6.1583
   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.0000         0         0   -0.0000   -0.0000
   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000         0
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000   -0.0000

  Columns 8 through 14

  900.0000  900.0000  900.0000  900.0000  900.0000  900.0000  900.0000
  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000
  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000  400.0000
         0         0         0         0         0         0         0
   80.0000   90.0000  100.0000  200.0000  300.0000  400.0000  500.0000
   -5.7242   -7.8888   -8.5581  -28.6469  -65.4846 -132.0765 -280.2833
   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000
   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000
   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000
```

Columns 15 through 21

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
-5.0626	-0.0575	3.5428	6.2640	8.3605	10.0122	11.2380
0	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
0.0000	0.0000	0	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0000

Columns 22 through 28

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
12.2612	12.9748	13.4181	6.2806	-29.8648	-91.5673	-218.4440
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

Columns 29 through 35

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
10.0000	20.0000	30.0000	40.0000	50.0000	60.0000	70.0000
-4.9390	-1.5317	0.4749	1.6842	2.3735	2.7318	2.7550
0.0000	0.0000	0.0000	0	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

Columns 36 through 42

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
80.0000	90.0000	100.0000	200.0000	300.0000	400.0000	500.0000
2.7284	2.4490	2.1606	-3.8652	-10.0880	-21.4418	-48.9710
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
-0.0000	0	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000

The following is a listing of the output of a function that iterates through the STRIP data and gives the resulting maximum percent error for that geometry with the model using element values generated using coefficients.
The columns are arranged in the following way:

a
b
W
d
T
L inductor in nH
C capacitor in pF
t0 0 order coefficient for the turns ratio
t1 1st order coefficient for the turns ratio
t2 2nd order coefficient for the turns ratio
t3 3rd order coefficient for the turns ratio
% error phase
% error magnitude

Columns 1 through 7

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
0	0	0	0	0	10.0000	10.0000
10.0000	40.0000	70.0000	100.0000	400.0000	20.0000	50.0000
14.5784	14.9579	15.0325	14.8023	-4.2655	14.7420	14.8628
0.0047	0.0054	0.0063	0.0071	0.0174	0.0048	0.0051
1.5539	1.2798	1.0331	0.8137	0.1236	1.4271	1.2465
2.5056	3.7159	4.8164	5.8073	9.6833	3.4819	4.0648
1.5626	-1.2922	-3.8981	-6.2549	-16.1289	-1.1668	-2.7039
-3.2824	-1.5344	0.0636	1.5117	7.7469	-1.4472	-0.3651
30.7504	23.6768	23.2986	21.2717	16.6859	33.6509	19.6462
4.9488	6.5312	9.8957	9.9965	58.2688	10.7641	12.8559

Columns 8 through 14

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
10.0000	10.0000	10.0000	30.0000	30.0000	30.0000	30.0000
80.0000	200.0000	500.0000	30.0000	60.0000	90.0000	300.0000
14.7739	12.3204	-8.4988	14.6459	14.4681	14.2218	10.5792
0.0055	0.0079	0.0196	0.0047	0.0045	0.0045	0.0081
1.0855	0.6374	0.8890	1.4480	1.3821	1.3245	1.1507
4.6071	6.3695	7.9289	4.1360	3.9273	3.7701	4.1144
-4.1324	-8.7601	-12.7254	-3.5668	-3.4091	-3.3334	-5.0978
0.6303	3.7443	5.4575	0.2898	0.4899	0.6914	2.1394
13.4226	7.9890	1.1516	106.1962	73.7996	56.8377	12.2073
15.3371	27.7931	31.9739	44.3994	49.0150	53.7996	66.7834

Columns 15 through 21

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
50.0000	50.0000	50.0000	50.0000	50.0000	70.0000	70.0000
10.0000	40.0000	70.0000	100.0000	400.0000	20.0000	50.0000
14.6882	14.3058	13.9310	13.5639	10.3139	14.3519	13.7979
0.0050	0.0048	0.0047	0.0048	0.0136	0.0053	0.0057
1.6733	1.6406	1.6101	1.5819	1.4235	1.9983	1.9275
4.6350	4.0694	3.5867	3.1867	3.7394	4.1426	3.7895
-5.0731	-4.1116	-3.3023	-2.6455	-4.4551	-4.4611	-3.8210
0.9066	0.6769	0.4855	0.3325	0.9145	0.3600	0.1922
185.0244	111.7432	82.8307	66.7305	5.0340	136.2876	90.1165
61.7791	65.7842	71.8230	78.6520	66.6357	65.0384	68.3909

Columns 22 through 28

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
70.0000	70.0000	70.0000	90.0000	90.0000	90.0000	90.0000
80.0000	200.0000	500.0000	30.0000	60.0000	90.0000	300.0000
13.2627	11.3093	7.7383	13.8442	13.1552	12.4310	6.3745
0.0063	0.0094	0.0248	0.0064	0.0080	0.0097	0.0213
1.8584	1.5995	1.0749	2.4179	2.2349	2.0586	1.0123
3.4892	2.6174	4.8419	3.4468	3.7625	4.0403	4.9209
-3.2833	-2.1580	-6.5218	-3.0304	-3.6347	-4.1715	-6.0402
0.0488	-0.2808	0.6029	-1.0538	-0.7201	-0.4271	0.4859
67.7916	35.4038	5.8536	127.0123	91.2764	96.0007	23.7164
73.1136	95.1170	86.7295	70.1120	75.5835	85.1464	91.1735

Columns 29 through 35

900.0000	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
360.0000	360.0000	360.0000	360.0000	360.0000	360.0000	360.0000
10.0000	10.0000	10.0000	10.0000	10.0000	30.0000	30.0000
7.0000	10.0000	30.0000	60.0000	90.0000	20.0000	50.0000
14.4258	14.4140	14.2724	13.8549	13.1911	15.1308	14.2001
0.0051	0.0051	0.0052	0.0054	0.0056	0.0046	0.0043
1.6570	1.6234	1.4227	1.1991	1.0682	1.5394	1.3707
2.9855	3.0660	3.5624	4.1763	4.6334	4.0315	3.9955
-0.5888	-0.7776	-1.9447	-3.3961	-4.4885	-3.6417	-3.8458
-1.7114	-1.5832	-0.7848	0.2304	1.0265	0.4220	0.8102
19.9502	13.6716	27.9442	19.6615	15.0107	117.4703	79.2965
22.7010	30.6612	11.5182	14.4526	16.9895	42.4843	47.9708

REFERENCES

1. Meier, Paul J., Integrated Finline Components, IEEE Transactions Microwave Theory and Techniques, vol. MTT-22, pp. 1209-1216, December 1974.

2. Knorr, J.B., and Shada, P.M., Millimeter-wave Fin-line Characteristics, IEEE Transactions Microwave Theory and Technique, vol. MTT-28, pp. 737-742, Jul 1980.

3. Knorr, J.B., and Deal, J.C., Scattering Coefficients of an Inductive Strip in Fin-line: Theory and Experiment, IEEE Transactions Microwave Theory and Technique, vol. MTT-33, pp. 1011-1017, Oct 1985.

4. Deal, J.C., Numerical Computation of the Scattering Coefficients of an Inductive Strip in Fin-line, M.S. Thesis, Naval Postgraduate School, Monterey, CA., Mar 1984.

5. Morua, Michael L., A Circuit Model for an Inductive Strip in Homogeneous Finline, M.S. Thesis, Naval Postgraduate School, Monterey, CA., Jun 1990.

6. Grohsmeyer, Janeen, Analytical Expressions for Impedance and Wavelength in Inhomogeneous Finline. Report presented for EC2990, Naval Postgraduate School, Dec 1990.

7. Pozar, David M., Microwave Engineering, Addison-Wesley Publishing Company, 1990.

8. Knorr, J.B., A CAD Model for the Inductive Strip in Finline, Technical Report NPS 62-88-013, Naval Postgraduate School, Monterey, CA., Aug 1988.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5002	1
4.	Professor Jeffery B. Knorr, Code EC/Ko Naval Postgraduate School Monterey, CA 93943-5002	2
5.	Professor David Jenn, Code Ec/Jn Naval Postgraduate School Monterey, CA 93943-5002	1
6.	Commandant (G-PO-2) U.S. Coast Guard Washington, D.C. 90593	2
7.	Commanding Officer USCG Electronics Engineering Center (Attn: LT M.R. Linzey) P.O. Box 60 Wildwood, NJ 08260-0060	2