B. Tech. Project Report

MATLAB SIMULATIONS OF RF/MICROWAVE CIRCUITS

Sumit Gautam

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology,
Jaipur, Rajasthan, INDIA

under the guidance of

Prof. Raghuvir Tomar

August 2012 – April 2013

submitted in partial fulfillment of
the requirements to obtain the degree of
Bachelor of Technology (Hons.)
in
Electronics and Communication Engineering

CERTIFICATE

I hereby certify that the work that is being presented in the B.Tech. (Honors) Project Report entitled **"MATLAB simulations of RF/Microwave Circuits",** in partial fulfillment of the requirements for the award of the **Bachelor of Technology (Honors) in Electronics & Communication Engineering** and submitted to the Department of Electronics & Communication Engineering of The LNM Institute of Information Technology, Jaipur, Rajasthan is an authentic record of my own work carried out during a period from August 2012 to April 2013 under the supervision of **Prof. Raghuvir Tomar, E&CE Department**.

The matter presented in this report has not been submitted by me for the award of any other degree elsewhere.

(Signature of Candidate)
SUMIT GAUTAM
Y09UC285

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Signature of Supervisor)

Date:

Prof. Raghuvir Tomar
Project Supervisor
Electronics & Communication Engineering Department
The LNM Institute of Information Technology,
Jaipur, Rajasthan

ABSTRACT

In this project report, the main aim is to provide efficient and complete MATLAB codes for different topics related to Microwave Engineering. The MATLAB codes can be used to estimate values usable for the designing of various components used in Microwaves like: Transmission lines, waveguides etc. This can be useful to the manufacturers since the codes provide very precise values for different input parameters.

The project report is made in such a way that the reader could understand basic concepts of Microwave Engineering and could get to understand the topics better by executing the MATLAB codes for each topic. For efficiency, MATLAB codes are connected via MATLAB Graphic User Interface making it simple for the user to execute various topics.

The MATLAB GUI makes the project of great value to the user and could be used by manufacturers as a tool for getting precise and adequate results for developing various components used in Microwave Engineering. This project could be developed further and could possibly prove to be very useful to the engineers in the field of Microwave Engineering.

ACKNOWLEDGEMENT

First and foremost, I would like to express sincere gratitude and thank my guide, **Prof. R. Tomar**, for having suggested the topic of my project and for his constant support and guidance, without which I would not have been able to attempt this project.

SUMIT GAUTAM

CONTENTS

CHAPTER 1 – INTRODUCTION

Alternating current signals with frequencies between 300 MHz and 300 GHz (corresponding to the wavelength from ɹ = 1m and ɹ = 1mm respectively) are termed as **Microwaves**. Maxwell's Equations are an integral part of microwave engineering. There are many important applications of Microwaves mainly comprising of terms like: antenna gain, more bandwidth, non-bent nature (microwaves travel in the line of sight), preferred for RADAR systems. Other areas includes: heating methods, treatment, basic science, remote sensing and medical diagnostics.

As discussed earlier, Maxwell's equations are very important in Microwaves as relationship between the Electric and Magnetic fields are provided. The general form of time-varying Maxwell's Equations written in "point" or differential form are as follows:

$$\nabla \times \bar{\mathcal{E}} = \frac{-\partial \bar{\mathcal{B}}}{\partial t} - \bar{\mathcal{M}}$$

$$\nabla \times \bar{\mathcal{H}} = \frac{\partial \bar{\mathcal{D}}}{\partial t} + \bar{\mathcal{J}}$$

$$\nabla \cdot \bar{\mathcal{D}} = \rho$$

$$\nabla \cdot \bar{\mathcal{B}} = 0$$

$\bar{\mathcal{E}}$ is the electric field intensity, in V/m.
$\bar{\mathcal{H}}$ is the magnetic field intensity, in A/m.
$\bar{\mathcal{D}}$ is the electric flux density, in Coul/m^2.
$\bar{\mathcal{B}}$ is the magnetic flux density, in Wb/m^2.
$\bar{\mathcal{M}}$ is the (fictitious) magnetic current density, in V/m^2.
$\bar{\mathcal{J}}$ is the electric current density, in A/m^2.
ρ is the electric charge density, in Coul/m^3.

Solutions of various phenomenon such as: wave equation and basic plane wave, plane waves in lossy medium, plane waves in good conductor (skin depth), general plane wave solutions, energy and power etc can be determined by the four Maxwell's Equation stated above.

Network parameters, in the Two-Port Networks, are also important in Microwaves Engineering. The network parameters includes: the Impedance Parameters (Z-Parameters), Admittance Parameters (Y-Parameters), Transmission Parameters (ABCD-Parameters) and Scattering Parameters (S-Parameters). These parameters are inter-convertible and one can easily reduce a two-port network circuit into any one of these parameters.

Transmission lines and Waveguides are used to carry the Microwaves from one point to the other. Electrical and electronic components can be either lumped or distributed. Lumped components are those whose electrical properties do not change appreciably as we move along the length, width, and height of the component. Some examples of lumped components are resistors, capacitors, and inductors used at low frequencies. Distributed components are those whose electrical properties change along the length, width, and height of the component. Transmission lines are one distinct example of a distributed component.

CHAPTER 2 – ENGINEERING ELECTROMAGNETICS

Engineering Electromagnetics is the part of Microwaves Engineering via which we head towards Microwave Network analysis. This chapter deals with the introduction to the circuit theory consisting of the Z-parameters, Y-parameters, ABCD-parameters and S-parameters. The MATLAB codes will be followed after each topic is discussed thoroughly.

This segment also deals with the interconnections of the two port networks simulations. Logarithmic scales are also implemented for further help. In this way, we would be able to get precise readings of the parameters along with various interconnections and conversions from one scale of measurement to other.

2.1 Two Port Networks

2.1.1 *Various Two-Port Networks*

The following defines various two-port networks and their inter-conversions.

1. Impedance (Z) Parameters

The impedance or Z parameters are meant to relate the input and output voltages V1 and V2 to the input and output currents I1 and I2 and, like Y parameters, are preferred for frequencies below 1MHz. Mathematically, the Z parameters are defined by the following two equations.

$$V_1 = Z_{11}I_1 + Z_{12}I_2$$
$$V_2 = Z_{21}I_1 + Z_{22}I_2$$

Z-Parameters from above equations can be described as follows:

$$Z_{11} = \left.\frac{V_1}{I_1}\right|_{I2=0} \qquad Z_{12} = \left.\frac{V_1}{I_2}\right|_{I1=0} \qquad Z_{21} = \left.\frac{V_2}{I_1}\right|_{I2=0} \qquad Z_{22} = \left.\frac{V_2}{I_2}\right|_{I1=0}$$

2. Admittance (Y) Parameters

The admittance or Y parameters relate the input and output currents I1 and I2 to the input and output voltages V1 and V2. Mathematically speaking,

$$I_1 = Y_{11}V_1 + Y_{12}V_2$$
$$I_2 = Y_{21}V_1 + Y_{22}V_2$$

Y-Parameters from above equations can be described as:

$$Y_{11} = \left.\frac{I_1}{V_1}\right|_{V2=0} \qquad Y_{12} = \left.\frac{I_1}{V_2}\right|_{V1=0} \qquad Y_{21} = \left.\frac{I_2}{V_1}\right|_{V2=0} \qquad Y_{22} = \left.\frac{I_2}{V_2}\right|_{V1=0}$$

3. Transmission (ABCD) Parameters

ABCD parameters express input port voltage V1 and input current I1 as functions of output port voltage V2 and output current I2. These parameters are defined by means of the following two equations.

$$V_1 = AV_2 - BI_2$$
$$I_1 = CV_2 - DI_2$$

ABCD-Parameters from above equations can be described as:

$$A = \left.\frac{V_1}{V_2}\right|_{I2=0} \qquad B = -\left.\frac{V_1}{I_2}\right|_{V2=0} \qquad C = \left.\frac{I_1}{V_2}\right|_{I2=0} \qquad D = -\left.\frac{I_1}{V_2}\right|_{I2=0}$$

4. Scattering (S) Parameters

Scattering or S-parameters are based on incident and reflected powers and are the most unambigiously-defined parameters for RF and microwave frequencies. These parameters are defined by means of the following two equations.

$$b_1 = S_{11}a_1 + S_{12}a_2$$
$$b_2 = S_{21}a_1 + S_{22}a_2$$

where a1 and a2 are quantities representing the two incident signals and b1 and b2 are quantities representing the two reflected signals.

S-Parameters from the above equations can be defined as:

$$S_{11} = \left.\frac{b_1}{a_1}\right|_{a2=0} \qquad S_{12} = \left.\frac{b_1}{a_2}\right|_{a1=0} \qquad S_{21} = \left.\frac{b_2}{a_1}\right|_{a2=0} \qquad S_{22} = \left.\frac{b_2}{a_2}\right|_{a1=0}$$

Relationships between various parameters are as follows:

Conversions Between Two-Port Network Parameters

	S	Z	Y	$ABCD$
S_{11}	S_{11}	$\dfrac{(Z_{11}-Z_0)(Z_{22}+Z_0)-Z_{12}Z_{21}}{\Delta Z}$	$\dfrac{(Y_0-Y_{11})(Y_0+Y_{22})+Y_{12}Y_{21}}{\Delta Y}$	$\dfrac{A+B/Z_0-CZ_0-D}{A+B/Z_0+CZ_0+D}$
S_{12}	S_{12}	$\dfrac{2Z_{12}Z_0}{\Delta Z}$	$\dfrac{-2Y_{12}Y_0}{\Delta Y}$	$\dfrac{2(AD-BC)}{A+B/Z_0+CZ_0+D}$
S_{21}	S_{21}	$\dfrac{2Z_{21}Z_0}{\Delta Z}$	$\dfrac{-2Y_{21}Y_0}{\Delta Y}$	$\dfrac{2}{A+B/Z_0+CZ_0+D}$
S_{22}	S_{22}	$\dfrac{(Z_{11}+Z_0)(Z_{22}-Z_0)-Z_{12}Z_{21}}{\Delta Z}$	$\dfrac{(Y_0+Y_{11})(Y_0-Y_{22})+Y_{12}Y_{21}}{\Delta Y}$	$\dfrac{-A+B/Z_0-CZ_0+D}{A+B/Z_0+CZ_0+D}$
Z_{11}	$Z_0\dfrac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$	Z_{11}	$\dfrac{Y_{22}}{\lvert Y\rvert}$	$\dfrac{A}{C}$
Z_{12}	$Z_0\dfrac{2S_{12}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$	Z_{12}	$\dfrac{-Y_{12}}{\lvert Y\rvert}$	$\dfrac{AD-BC}{C}$
Z_{21}	$Z_0\dfrac{2S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$	Z_{21}	$\dfrac{-Y_{21}}{\lvert Y\rvert}$	$\dfrac{1}{C}$
Z_{22}	$Z_0\dfrac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$	Z_{22}	$\dfrac{Y_{11}}{\lvert Y\rvert}$	$\dfrac{D}{C}$
Y_{11}	$Y_0\dfrac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$	$\dfrac{Z_{22}}{\lvert Z\rvert}$	Y_{11}	$\dfrac{D}{B}$
Y_{12}	$Y_0\dfrac{-2S_{12}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$	$\dfrac{-Z_{12}}{\lvert Z\rvert}$	Y_{12}	$\dfrac{BC-AD}{B}$
Y_{21}	$Y_0\dfrac{-2S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$	$\dfrac{-Z_{21}}{\lvert Z\rvert}$	Y_{21}	$\dfrac{-1}{B}$
Y_{22}	$Y_0\dfrac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$	$\dfrac{Z_{11}}{\lvert Z\rvert}$	Y_{22}	$\dfrac{A}{B}$
A	$\dfrac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{2S_{21}}$	$\dfrac{Z_{11}}{Z_{21}}$	$\dfrac{-Y_{22}}{Y_{21}}$	A
B	$Z_0\dfrac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{2S_{21}}$	$\dfrac{\lvert Z\rvert}{Z_{21}}$	$\dfrac{-1}{Y_{21}}$	B
C	$\dfrac{1}{Z_0}\dfrac{(1-S_{11})(1-S_{22})-S_{12}S_{21}}{2S_{21}}$	$\dfrac{1}{Z_{21}}$	$\dfrac{-\lvert Y\rvert}{Y_{21}}$	C
D	$\dfrac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{2S_{21}}$	$\dfrac{Z_{22}}{Z_{21}}$	$\dfrac{-Y_{11}}{Y_{21}}$	D

$|Z| = Z_{11}Z_{22} - Z_{12}Z_{21}$; $|Y| = Y_{11}Y_{22} - Y_{12}Y_{21}$; $\Delta Y = (Y_{11}+Y_0)(Y_{22}+Y_0) - Y_{12}Y_{21}$; $\Delta Z = (Z_{11}+Z_0)(Z_{22}+Z_0) - Z_{12}Z_{21}$; $Y_0 = 1/Z_0$

The MATLAB implementation of the described networks is as follows:

<u>Two-port Analysis: (Code)</u>

```
clear all;
clear;
clc;

j=sqrt(-1);
q=input('            This program calulates Z,Y,ABCD and S parameters for a given circuit.\nPlease input the network type(Tee:t and Pi:p):','s');

%Input the Impedance types

k1=input('Enter the impedance type of Z1 <r/l/c>:','s');
if(k1=='r')
    R1=input('Enter the value of Resistance (in ohms) at Z1=','s');
```

```
end
if (k1=='l')
    R1=input('Enter the value of Inductance (in Henry) at Z1=','s');
end
if (k1=='c')
    R1=input('Enter the value of Capacitance (in Farad) at Z1=','s');
end

k2=input('Enter the impedance type of Z2 <r/l/c>:','s');
if(k2=='r')
    R2=input('Enter the value of Resistance (in ohms) at Z2=','s');
end
if (k2=='l')
    R2=input('Enter the value of Inductance (in Henry) at Z2=','s');
end
if (k2=='c')
    R2=input('Enter the value of Capacitance (in Farad) at Z2=','s');
end

k3=input('Enter the impedance type of Z3 <r/l/c>:','s');
if(k3=='r')
    R3=input('Enter the value of Resistance (in ohms) at Z3=','s');
end
if (k3=='l')
    R3=input('Enter the value of Inductance (in Henry) at Z3=','s');
end
if (k3=='c')
    R3=input('Enter the value of Capacitance (in Farad) at Z3=','s');
end

R1=str2num(R1);
R2=str2num(R2);
R3=str2num(R3);

%Input Frequency

if (k1=='l' || k1=='c' || k2=='l' || k2=='c' || k3=='l' || k3=='c')
    f=input('Enter the frequency of the circuit(in Hertz):','s');
    f=str2num(f);
end


%Calculation of Z1,Z2 and Z3

if (k1=='r')
    Z1=R1;
end
if (k1=='l')
    Z1=j*2*pi*f*R1;
end
if (k1=='c')
    Z1=-j/2*pi*f*R1;
end

if (k2=='r')
```

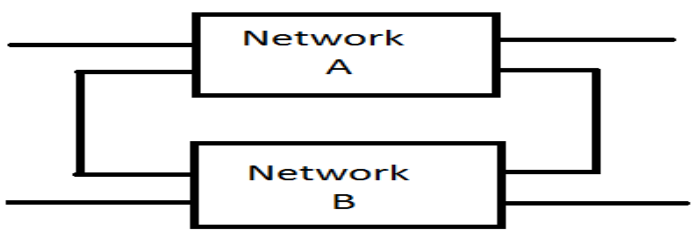
```
    Z2=R2;
end
if (k2=='l')
    Z2=j*2*pi*f*R2;
end
if (k2=='c')
    Z2=-j/2*pi*f*R2;
end

if (k3=='r')
    Z3=R3;
end
if (k3=='l')
    Z3=j*2*pi*f*R3;
end
if (k3=='c')
    Z3=-j/2*pi*f*R3;
end

%Calculation of parameters
d='y';
while(d=='y')
clc;
l=input('Please select the type of parameters you want to
compute.\n1.Impedance parameters (Z-parameters).\n2.Admittance
parameters (Y-parameters).\n3.Transmission parameters (ABCD-
parameters).\n4.Scattering parameters (S-parameters).\nPlease enter
your choice:','s');
l=str2num(l);

if (q=='t')
    Z=[Z1+Z2 Z2
        Z2 Z2+Z3];
end

if (q=='p')
    Z=[(Z2*(Z1+Z3))/(Z1+Z2+Z3) Z2*Z3/(Z1+Z2+Z3)
        Z2*Z3/(Z1+Z2+Z3) Z3*(Z1+Z2)/(Z1+Z2+Z3)];
end
if(l==1)
    display('The Impedance Matrix is:');
    display(Z);
    if(Z(2)==Z(3))
        display('This is a Reciprocal Network.');
    end
    TF=Z(3)/Z(1);
    Zin=Z(1);
    display('Transfer Function of the Network :');
    display(TF);
    display('Input Impedance of the Network :');display(Zin);
end
if(l==2)
    display('The Admittance Matrix is:');
    Y=[ Z(4)/det(Z) -Z(2)/det(Z)
        -Z(3)/det(Z) Z(1)/det(Z)]
    if(Y(2)==Y(3))
```

```
        display('This is a Reciprocal Network');
    end
end
if(l==3)
    display('The Transmission Matrix is:');
    ABCD=[ Z(1)/Z(3) det(Z)/Z(3)
           1/Z(3)    Z(4)/Z(3)]
    if(ABCD(1)*ABCD(4)-ABCD(2)*ABCD(3)==1)
        display('This is a Reciprocal Network');
    end
end
if(l==4)
    Z0=input('Please enter the value of Z0 (in Ohms):');
    display('The Scattering Matrix is:');
    delz=(Z(1)+Z0)*(Z(4)+Z0);
    S=[((Z(1)-Z0)*(Z(4)+Z0)-Z(2)*Z(3))/delz 2*Z(2)*Z0/delz
        2*Z(2)*Z0/delz ((Z(1)+Z0)*(Z(4)-Z0)-Z(2)*Z(3))/delz]
end

d=input('Do you want to continue? <y/n>:','s');
end
```

2.1.2 *Two-Port Interconnections*

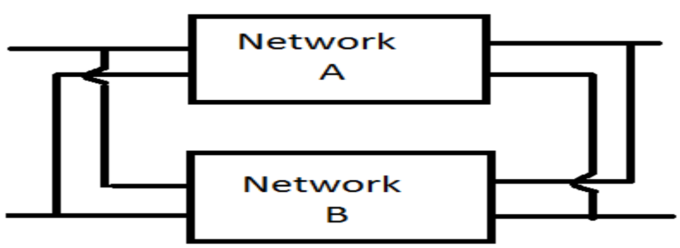
Two-Port interconnections may be described as follows.

(a) Series Connection:



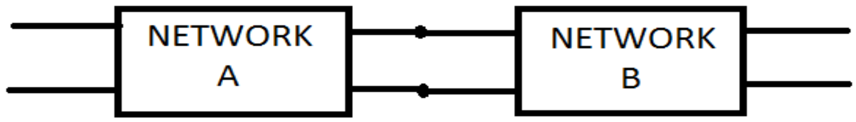
$$[Z]_{overall} = [Z]^A + [Z]^B$$

(b) Parallel Connection:



$$[Y]_{overall} = [Y]^A + [Y]^B$$

(c) Cascade Connection



$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{\text{Overall}} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{A} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{B}$$

The MATLAB code of above is as follows:

<u>Two-Port Interconnections: (Code)</u>

```
clear all;
clear;
clc;

j=sqrt(-1);
q_A=input('             This program finds equivalent parameter matrix for an interconnections of TWO 2-port Networks.\nPlease input the network type of NETWORK A(Tee:t and Pi:p):','s');

%Input the Impedance types of Network A

k1_A=input('Enter the impedance type of Z1 <r/l/c>:','s');
if(k1_A=='r')
    R1_A=input('Enter the value of Resistance (in ohms) at Z1=','s');
end
if (k1_A=='l')
    R1_A=input('Enter the value of Inductance (in Henry) at Z1=','s');
end
if (k1_A=='c')
    R1_A=input('Enter the value of Capacitance (in Farad) at Z1=','s');
end

k2_A=input('Enter the impedance type of Z2 <r/l/c>:','s');
if(k2_A=='r')
    R2_A=input('Enter the value of Resistance (in ohms) at Z2=','s');
end
if (k2_A=='l')
    R2_A=input('Enter the value of Inductance (in Henry) at Z2=','s');
end
if (k2_A=='c')
    R2_A=input('Enter the value of Capacitance (in Farad) at Z2=','s');
end

k3_A=input('Enter the impedance type of Z3 <r/l/c>:','s');
if(k3_A=='r')
    R3_A=input('Enter the value of Resistance (in ohms) at Z3=','s');
end
if (k3_A=='l')
```

```
    R3_A=input('Enter the value of Inductance (in Henry) at Z3=','s');
end
if (k3_A=='c')
    R3_A=input('Enter the value of Capacitance (in Farad) at Z3=','s');
end

R1_A=str2num(R1_A);
R2_A=str2num(R2_A);
R3_A=str2num(R3_A);

%Input Frequency

if (k1_A=='l' || k1_A=='c' || k2_A=='l' || k2_A=='c' || k3_A=='l' ||
k3_A=='c')
    f_A=input('Enter the frequency of the circuit(in Hertz):','s');
    f_A=str2num(f_A);
end



q_B=input('Please input the network type of NETWORK B(Tee:t and
Pi:p):','s');

%Input the Impedance types of NETWORK B

k1_B=input('Enter the impedance type of Z1 <r/l/c>:','s');
if(k1_B=='r')
    R1_B=input('Enter the value of Resistance (in ohms) at Z1=','s');
end
if (k1_B=='l')
    R1_B=input('Enter the value of Inductance (in Henry) at Z1=','s');
end
if (k1_B=='c')
    R1_B=input('Enter the value of Capacitance (in Farad) at Z1=','s');
end

k2_B=input('Enter the impedance type of Z2 <r/l/c>:','s');
if(k2_B=='r')
    R2_B=input('Enter the value of Resistance (in ohms) at Z2=','s');
end
if (k2_B=='l')
    R2_B=input('Enter the value of Inductance (in Henry) at Z2=','s');
end
if (k2_B=='c')
    R2_B=input('Enter the value of Capacitance (in Farad) at Z2=','s');
end

k3_B=input('Enter the impedance type of Z3 <r/l/c>:','s');
if(k3_B=='r')
    R3_B=input('Enter the value of Resistance (in ohms) at Z3=','s');
end
if (k3_B=='l')
    R3_B=input('Enter the value of Inductance (in Henry) at Z3=','s');
end
if (k3_B=='c')
```

```
    R3_B=input('Enter the value of Capacitance (in Farad) at Z3=','s');
end

R1_B=str2num(R1_B);
R2_B=str2num(R2_B);
R3_B=str2num(R3_B);

%Input Frequency
if (k1_B=='l' || k1_B=='c' || k2_B=='l' || k2_B=='c' || k3_B=='l' ||
k3_B=='c')
    f_B=input('Enter the frequency of the circuit(in Hertz):','s');
    f_B=str2num(f_B);
end

%Calculation of Z1,Z2,Z3 AND NETWORK PARAMETERS OF NETWORK A
 if (k1_A=='r')
    Z1_A=R1_A;
end
if (k1_A=='l')
    Z1_A=j*2*pi*f_A*R1_A;
end
if (k1_A=='c')
    Z1_A=-j/2*pi*f_A*R1_A;
end

if (k2_A=='r')
    Z2_A=R2_A;
end
if (k2_A=='l')
    Z2_A=j*2*pi*f_A*R2_A;
end
if (k2_A=='c')
    Z2_A=-j/2*pi*f_A*R2_A;
end

if (k3_A=='r')
    Z3_A=R3_A;
end
if (k3_A=='l')
    Z3_A=j*2*pi*f_A*R3_A;
end
if (k3_A=='c')
    Z3_A=-j/2*pi*f_A*R3_A;
end

if (q_A=='t')
    Z_A=[Z1_A+Z2_A Z2_A
        Z2_A Z2_A+Z3_A];
end

if (q_A=='p')
    Z_A=[(Z2_A*(Z1_A+Z3_A))/(Z1_A+Z2_A+Z3_A) Z2_A*Z3_A/(Z1_A+Z2_A+Z3_A)
        Z2_A*Z3_A/(Z1_A+Z2_A+Z3_A) Z3_A*(Z1_A+Z2_A)/(Z1_A+Z2_A+Z3_A)];
end
    Y_A=[ Z_A(4)/det(Z_A) -Z_A(2)/det(Z_A)
```

```
        -Z_A(3)/det(Z_A) Z_A(1)/det(Z_A)];

    ABCD_A=[ Z_A(1)/Z_A(3) det(Z_A)/Z_A(3)
           1/Z_A(3)    Z_A(4)/Z_A(3)];

%Calculation of Z1,Z2,Z3 AND NETWORK PARAMETERS OF NETWORK B

if (k1_B=='r')
    Z1_B=R1_B;
end
if (k1_B=='l')
    Z1_B=j*2*pi*f_B*R1_B;
end
if (k1_B=='c')
    Z1_B=-j/2*pi*f_B*R1_B;
end

if (k2_B=='r')
    Z2_B=R2_B;
end
if (k2_B=='l')
    Z2_B=j*2*pi*f_B*R2_B;
end
if (k2_B=='c')
    Z2_B=-j/2*pi*f_B*R2_B;
end

if (k3_B=='r')
    Z3_B=R3_B;
end
if (k3_B=='l')
    Z3_B=j*2*pi*f_B*R3_B;
end
if (k3_B=='c')
    Z3_B=-j/2*pi*f_B*R3_B;
end

if (q_B=='t')
    Z_B=[Z1_B+Z2_B Z2_B
        Z2_B Z2_B+Z3_B];
end

if (q_B=='p')
    Z_B=[(Z2_B*(Z1_B+Z3_B))/(Z1_B+Z2_B+Z3_B) Z2_B*Z3_B/(Z1_B+Z2_B+Z3_B)
        Z2_B*Z3_B/(Z1_B+Z2_B+Z3_B) Z3_B*(Z1_B+Z2_B)/(Z1_B+Z2_B+Z3_B)];
end
    Y_B=[ Z_B(4)/det(Z_B) -Z_B(2)/det(Z_B)
        -Z_B(3)/det(Z_B) Z_B(1)/det(Z_B)];

    ABCD_B=[ Z_B(1)/Z_B(3) det(Z_B)/Z_B(3)
           1/Z_B(3)    Z_B(4)/Z_B(3)];
u='y';
while(u=='y')
clc;
```

```
p=input('Select the interconnection of your type:\n 1.Series
Interconnection.\n 2.Parallel Interconnection.\n 3.Cascade
Interconnection.\n Your Choice Please:');
display('The Matrix for desired interconnection is:');
if(p==1)
    Zoverall=Z_A+Z_B
end
if(p==2)
    Yoverall=Y_A+Y_B
end
if(p==3)
    ABCDoverall=ABCD_A*ABCD_B
end

u=input('Do you want to continue? <y/n>:','s');
end
```

2.1.3 *Logarithmic Scales*

A few key terms used in this section are as follows:

- Bell (B) = $\log(P_2/P_1) = 2\log(V_2/V_1)$
- Decibell (dB) = $10\log(P_2/P_1) = 20\log(V_2/V_1)$
- Neper (Np) = $\ln(V_2/V_1)$
- dB=8.686 Np
- dB_m=10log(P<in Watts>/1mW)
- dB_w=10log(P<in Watts>/1W)

The MATLAB code to demonstrate the above is as follows:

Logarithmic Scales: (Code)

```
clear all;
clc;
w='y';
while(w=='y')
    clc;
k=input('                    Main-Menu\n 1.Find dB value of a 2-Port
Network.\n 2.Find Neper value of a 2-Port Network.\n 3.Convert dB value
to Neper and vice-versa.\n 4.Find dBm value for a given value of
Power(in Watts).\n 5.Find Power(in Watts) for a given dBm value.\n
6.Find dBw value for a given value of Power(in Watts).\n 7.Find
Power(in Watts) for a given dBw value.\n 8.Convert a value from dBw to
dBm.\n Please enter your choice:');
clc;
if(k==1)
    display('Find dB value of a 2-Port Network.');
    V1=input('\nEnter the value of V1(in Volts):');
    V2=input('\nEnter the value of V2(in Volts):');
    dB=20*(log(V2/V1)/2.303);
    display('The Value in dB:');display(dB);
end
if(k==2)
    display('Find Neper value of a 2-Port Network.');
```

```
    V1=input('\nEnter the value of V1(in Volts):');
    V2=input('\nEnter the value of V2(in Volts):');
    Np=log(V2/V1);
    display('The Value in Neper:');display(Np);
end
if(k==3)
    display('Convert dB value to Neper and vice-versa.');
    s=input('a.Convert dB to Neper.\n b.Convert Neper to dB.\n Your
Choice:');
    if(s=='a')
        Np=input('Enter value in Neper:');
        dB=8.686*Np;
        display('The value in dB:');dislpay(dB);
    end
    if(s=='b')
        dB=input('Enter value in dB:');
        Np=dB/8.686;
        display('The value in Neper:');dislpay(Np);
    end
end
if(k==4)
    display('Find dBm value for a given value of Power(in Watts).');
    P=input('Enter the value of Power(in Watts):');
    dBm=10*(log(P/0.001)/2.303);
    display('The value in dBm:');display(dBm);
end
if(k==5)
    display('Find Power(in Watts) for a given dBm value.');
    dBm=input('Enter the value of dBm:');
    P=power(10,(dBm/10))*0.001;
    display('The value of Power(in Watts):');display(P);
end
if(k==6)
    display('Find dBw value for a given value of Power(in Watts).');
    P=input('Enter the value of Power(in Watts):');
    dBw=(10*log(P/1))/2.303;
    display('The value in dBw:');display(dBw);
end
if(k==7)
    display('Find Power(in Watts) for a given dBw value.');
    dBw=input('Enter the value of dBw:');
    P=power(10,(dBw/10))*1;
    display('The value of Power(in Watts):');display(P);
end
if(k==8)
    display('Convert a value from dBw to dBm.');
    dBw=input('Enter value in dBw:');
    dBm=30+dBw;
    display('The value in dBm:');display(dBm);
end
w=input('Do you want to continue <y/n>:','s');
end
```

2.2 Transmission Lines

Transmission lines are used to carry electromagnetic energy from one point in physical space to another. Examples of transmission lines are: Two-wire lines, Multi-wire lines, Waveguides, Coaxial lines, Optical Fibre etc.

2.2.1 *Lossless Transmission Line*

For a lossless transmission line, we have : R=0 ; G=0; α=0; β=ω*sqrt(LC) and Z_o=sqrt(L/C). The Equations for V(x) and I(x) may be written as:

$V(x)=V^{+}e^{-j\beta x}+V^{-}e^{j\beta x}$

$I(x)=(1/Zo)*([V^{+}e^{-j\beta x}-V^{-}e^{j\beta x}]$

Here, we find Z_{in}(Input Impedance),ρ_l(Volatge reflection coefficient at x=l),ρ_0(Volatge reflection coefficient at x=0),VSWR and First Voltage and First Current Maxima for a given Z_o (Characteristic impedance) and Z_l(complex load impedance)

The MATLAB code demonstrating the above phenomenon is as follows:

<u>Lossless Transmission Line: (Code)</u>

```
clear all;
clc;

display('     This program finds: Zin,Pl,P0,VSWR and First Voltage and 
First Current Maxima ');
j=sqrt(-1);
Z0=input('Enter value of characteristic Impedance (in Ohms):');
display('Please enter complex value for load impedance:');
Zr=input('Enter real part of Zl(in ohms):');
Zc=input('Enter imaginary part of Zl(in ohms):');
Zl=Zr+j*Zc;
Lbyz=input('Enter the value of electric length:');
BL=2*pi*(Lbyz);
q='y';
while(q=='y')
clc;
display('                 MAIN MENU');
k=input('1. Find Input Impedance (Zin) for the given inputs.\n2. Find 
Reflection Coefficient  at the load(Pl) for the given inputs.\n3. Find 
Reflection Coefficient  at the input(Po) for the given inputs.\n4. Find 
the VSWR for the given inputs.\n5. Find the locations of the first 
voltage maximum and the first current maximum.\n\n Your Choice 
Please:');
if(k==1)
    display(' The Value of Input Impedance (Zin):');
    Zin= Z0*(Zl+j*Z0*tan(BL))/(Z0+j*Zl*tan(BL))
end
if(k==2)
    display(' The value of Reflection Coefficient at the load (Pl):');
    Pl=(Zl-Z0)/(Zl+Z0)
end
if(k==3)
    display(' The value of Reflection Coefficient at the input (P0):');
```

```
    P0=((Zl-Z0)/(Zl+Z0))*exp(-2*j*BL)
end
if(k==4)
    display(' The value of VSWR:');
    Pl=(Zl-Z0)/(Zl+Z0);
    VSWR=(1+abs(Pl))/(1-abs(Pl))
end
if(k==5)
    display(' The values of locations of the first voltage maximum and
the first current maximum:');
    P0=((Zl-Z0)/(Zl+Z0))*exp(-2*j*BL);
    a=angle(P0);
    Xvmax=(2*pi-(a+2*pi))/4*pi;
    Ximax=(3*pi-(a+2*pi))/4*pi;
    display(' The first voltage maximum = lambda * ');display(Xvmax);
    display(' The first current maximum = lambda * ');display(Ximax);
end
q=input('Do you want to continue <y/n> :','s');
end
```

2.2.2 *Lossless Dielectric*

In this case, for a given value of wavelength, frequency and the rate at which the field amplitude decays inside the material, we can calculate the Complex Permittivity, Complex Dielectric Constant and the Loss Tangent of the material.

The MATLAB code demonstrating above concept is as follows:

Lossless Dielectric: (Code)

```
%This Program Calculates the Complex Permittivity ,the Complex
%Dielectric Constant and the Loss-Tangent of the material .

clear all;
clc;

j=sqrt(-1);
f=input('This Program Calculates the Complex Permittivity ,the Complex
Dielectric Constant and the Loss-Tangent of the material.\nEnter the
value of frequency(in GHz):');
ld=input('Enter the value of wavelength of propagation inside the
material (in cms):');
ad=input('Enter the value at which the rate of field amplitude decay
inside the material (in dB/m):');
clc;
display('The value of a (in Np/m) and b (in rad/m):');
a=ad/8.686
b=(2*pi)/(ld/100)
c=3*power(10,8);
Er_1= (a*a-b*b)/-((2*pi*f*power(10,9)/c)*(2*pi*f*power(10,9)/c))
Er_2= (2*a*b)/((2*pi*f*power(10,9)/c)*(2*pi*f*power(10,9)/c))
display('Complex Dielectric Constant of the Material:');
Er=Er_1+j*Er_2
```

```
tandel=(Er_2/Er_1);
display('Loss Tangent:');display(tandel);
```

2.2.3 *Skin Depth*

In many practical cases, there is loss or attenuation due to good conductors. In such a case, the conductive current is much greater than the displacement current (i.e. σ = ωϵ). In general, most of the metals are good conductors. Therefore, in terms of complex ϵ, the equivalent condition becomes: ϵ'' >> ϵ'. Approximation to the propagation constant can be adequately given as:

$$\gamma = \alpha + j\beta \simeq j\omega\sqrt{\mu\epsilon}\sqrt{\frac{\sigma}{j\omega\epsilon}} = (1+j)\sqrt{\frac{\omega\mu\sigma}{2}}$$

The skin depth or the characteristic depth of penetration can be defined as:

$$\delta_s = \frac{1}{\alpha} = \sqrt{\frac{2}{\omega\mu\sigma}}$$

The MATLAB code for Skin-Depth described is as follows:

<u>Skin Depth: (Code)</u>

```
%This Program Calculates the Skin-Depth of given materials.

clear all;
clc;

k='y';
while(k=='y')
    clc;
    sigma=input('          This Program Calculates the Skin-Depth of
given materials.\nPlease enter the conductivity value of material (X
10^7 S/m):');
    f=input('Enter the frequency at which skin-depth needs to be
calculated (in Hz):');
    w=2*pi*f;
    uo=4*pi*power(10,-7);
    display('Skin Depth of the Material (in cm):')
    delta=sqrt(2/(uo*w*sigma*power(10,7)))*100
    k=input('Do you want to continue <y/n>:','s');
end
```

2.3 Rectangular Waveguide

2.3.1 *Rectangular Waveguide*

A rectangular waveguide is a hollow piece of perfectly conducting metal which has a rectangular cross-section. This type of transmission structure does not support a TEM mode of transmission. However, TE and TM modes are possible.

As shown in Figure 2.3.1, the larger dimension in the cross-section of the waveguide is labeled 'a' and is assumed to be along the x-axis. The narrower dimension is labeled 'b' and is assumed to be along the y-axis. The direction of propagation is assumed to be along the z-axis. Also, unless otherwise stated, the medium inside the waveguide is assumed to be air (i.e., dielectric constant=relative permeability=1, and electrical conductivity=0).

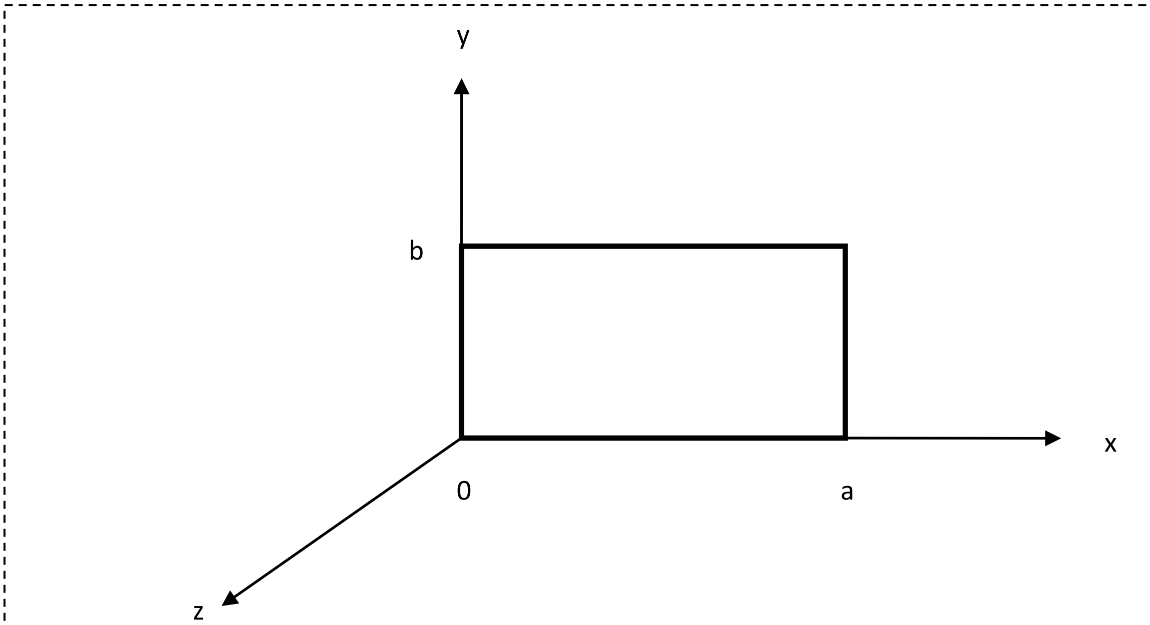


Figure 2.3.1 The cross-section of a rectangular waveguide

<u>Rectangular Waveguide: (Code)</u>

```
clear all;
clc;

display('        This program calculates the cut-off frequencies at
different modes');
i=[0 0 1 1 2 2 0 1 2 3 3 3 0 1 2 3];
j=[0 1 0 1 0 1 2 2 2 0 1 2 3 3 3 3];
a=input('Enter the value of a (in inches):');
b=input('Enter the value of b (in inches):');
c=3*power(10,10);
a=2.54*a;b=2.54*b;
for k=1:16
    fc_t(k)=(c/2)*sqrt((i(k)*i(k))/(a*a) + (j(k)*j(k))/(b*b));
end
```

```
fc=sort(fc_t);
clc;
%display(fc_t);
%display(fc);
o=['  -- '
   '  DM '
   '1HOM '
   '2HOM '
   '3HOM '
   '4HOM '
   '5HOM '
   '6HOM '
   '7HOM '
   '8HOM '
   '9HOM '
   '10HOM'
   '11HOM'
   '12HOM'
   '13HOM'
   '14HOM'];
for k=1:16
    for l=1:16
        if(fc_t(l)==fc(k))
            m(k)=i(l);
            n(k)=j(l);
        end
    end
end
clc;
disp('       m          n    fc(in Ghz)');
pay=[m
    n
    fc*power(10,-10)];
payt=pay';
disp(payt);
```

2.3.2 *WR 90 WG*

For a WR90 WG (a=22.86 mm and b=10.16 mm), we calculate the following operating parameters at a given input frequency: Guided Wavelength, Intrinsic Impedance(Z_w),Phase Velocity(V_p) and Group Velocity(V_g).Also calculate the Power Flow for TE_{10} mode.

The MATLAB code for above is as follows:

WR 90 WG: (Code)

```
% For a WR90 WG ( a=22.86 mm and b=10.16 mm), calculate the following
% operating parameters at a given input frequency: Guided Wavelength,
% Intrinsic Impedance(Zw),Phase Velocity(Vp) and Group Velocity(Vg)
% Also calculate the Power Flow for TE(1,0) mode

clear all;
clc;
a=2.286; b=1.016;
c=3*power(10,10);
```

```
fc=(c/2)*sqrt(1/(a*a));
q='y';
while(q=='y')
clc;
f=input('Enter the desired frequency (in GHz):');
f=f*power(10,9);
w=2*pi*f;
ko=w/c;
kz=sqrt((ko*ko)-((pi*pi)/(a*a)));

lambda_g=2*pi/kz;
Zw=120*pi/sqrt(1-((fc*fc)/(f*f)));
Vp=c/sqrt(1-((fc*fc)/(f*f)));
Vg=c*c/Vp;
P10=((a*b)/4)*power(3*power(10,4),2)/Zw;

display('Guided Wavelength (in cms):');
disp(lambda_g);
display('Intrinsic Impedance(Zw) (in ohms):');
disp(Zw);
display('Phase Velocity(Vp) (in cm/sec):');
disp(Vp);
display('Group Velocity(Vg) (in cm/sec):');
disp(Vg);
display('Power in TE(1,0) mode (in Watts):');
disp(P10);

q=input('Do you want to continue <y/n>:','s');
end
```

2.4 Circular Waveguide

2.4.1 *Circular Waveguide*

A circular waveguide is a hollow piece of perfectly conducting metal which has a circular cross-section. This type of transmission structure does not support a TEM mode of transmission. However, TE and TM modes are possible.

As shown in Figure 2.4.1, the radius of the waveguide is denoted by 'a'. The direction of propagation is assumed to be along the z-axis, and the cylindrical coordinate system (ρ,ϕ,z) is employed for analysis. The point of observation is denoted by P as shown. Also, unless otherwise stated, the medium inside the waveguide is assumed to be air (i.e., dielectric constant=relative permeability=1, and electrical conductivity=0).

Please also be aware that one is interested only in fields inside the waveguide, i.e., for 0<= ρ <=a. The field components outside the waveguide will be zero because of the perfect electrical conductor present at ρ=a.

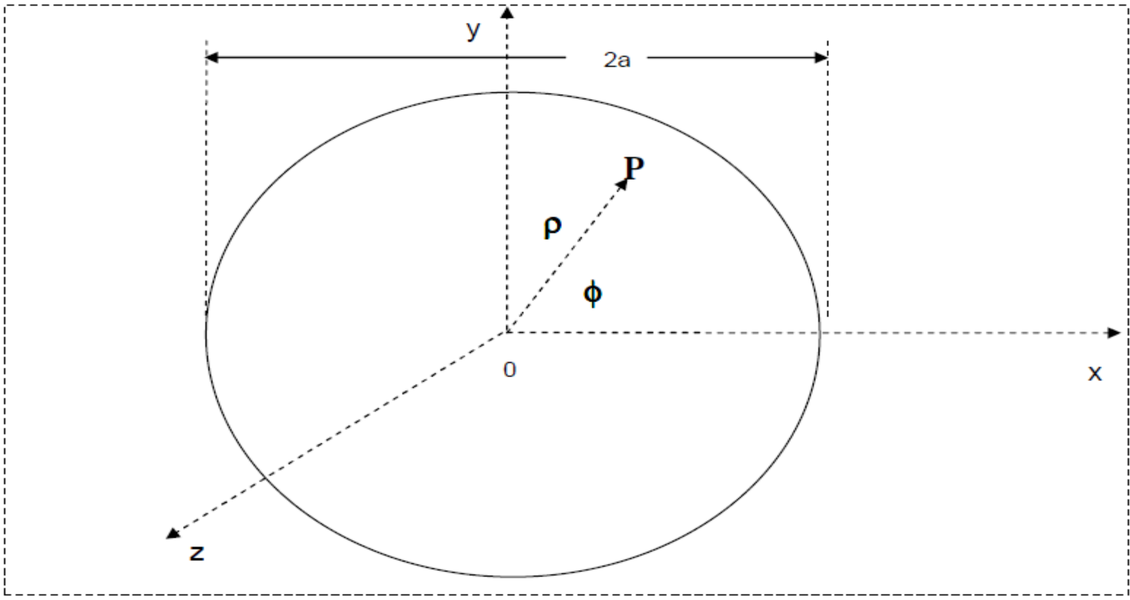


Figure 2.4.1 The cross-section of a circular waveguide

MATLAB code for basic circular waveguide is as follows:

Circular Waveguide: (Code)

```
% Circular Waveguide : For TE(1,1) mode; Determine: 1>Cut-off frequency
% 2>Guided Wavelength for an operating frequency 3>Wave Impedance
4>Phase
% Velocity (Vp) 5>Group Velocity (Vg)

clear all;
clc;

c=3*power(10,10);
q='y';
while(q=='y');
display('            Main-Menu');
s=input('Determine: Cut-off frequency,Guided Wavelength for an
operating frequency,Wave Impedance,Phase Velocity (Vp) and Group
Velocity (Vg) for:\n1. TE(1,1) Mode in air medium.\n2. TE(1,1) Mode in
dielectric medium.\n3. TM(0,1) Mode in air medium.\n4. TM(0,1) Mode in
dielectric medium.\nYour Choice Please:');
if(s==1)
% TE(1,1) Mode in air medium.

a=input('Enter value of a <radius> (in cms):');
f=input('Enter the value of operating frequency(in GHz):');
f=f*power(10,9);

fc=(1.841*c)/(2*pi*a);
w=2*pi*f;
kz=(2*pi/c)*sqrt(power(f,2)-power(fc,2));
Wg=2*pi/kz;
Zw=120*pi/(sqrt(1-(power((fc/f),2))));
Vp=w/kz;
```

```
Vg=c*c/Vp;
end

if(s==2)
% TE(1,1) Mode in a dielectric medium.

a=input('Enter value of a <radius> (in cms):');
Er=input('Enter the value of dielectric constant:');
f=input('Enter the value of operating frequency(in GHz):');
f=f*power(10,9);

fc=(1.841*c)/(2*pi*a*sqrt(Er));
w=2*pi*f;
kz=(2*pi*f/c)*sqrt(Er-power((fc/f),2));
Wg=2*pi/kz;
Zw=120*pi/(sqrt(Er-(power((fc/f),2))));
Vp=w/kz;
Vg=c*c/Vp;
end

if(s==3)
% TM(0,1) Mode in air medium.

a=input('Enter value of a <radius> (in cms):');
f=input('Enter the value of operating frequency(in GHz):');
f=f*power(10,9);

fc=(2.405*c)/(2*pi*a);
w=2*pi*f;
kz=(2*pi/c)*sqrt(power(f,2)-power(fc,2));
Wg=2*pi/kz;
Zw=120*pi*(sqrt(1-(power((fc/f),2))));
Vp=w/kz;
Vg=c*c/Vp;
end

if(s==4)
% TM(0,1) Mode in a dielectric medium.

a=input('Enter value of a <radius> (in cms):');
Er=input('Enter the value of dielectric constant:');
f=input('Enter the value of operating frequency(in GHz):');
f=f*power(10,9);

fc=(2.405*c)/(2*pi*a*sqrt(Er));
w=2*pi*f;
kz=(2*pi*f/c)*sqrt(Er-power((fc/f),2));
Wg=2*pi/kz;
Zw=120*pi*(sqrt(Er-(power((fc/f),2))));
Vp=w/kz;
Vg=c*c/Vp;
end

clc;
display('Cut-off frequency (in Hz):');disp(fc);
```

```
display('Guided Wavelength (in cms):');disp(Wg);
display('Wave Impedance (in Ohms):');disp(Zw);
display('Phase Velocity (Vp) (in cm/sec):');disp(Vp);
display('Group Velocity (Vg) (in cm/sec):');disp(Vg);

q=input('Do you want to continue <y/n>:','s');
end
```

2.4.2 *Circular Waveguide Safety Margin*

For a circular waveguide for the X-band, assume standard size to be '2a' inches diameter. We convert 2a (inches) to 2a (centimeters). Now,

1. For no safety margin: Calculate f_{cTE11} and f_{cTM01}. Calculate the safety margin as f_{cTM01} - f_{cTE11}.
2. For 20% safety margin: Calculate f_{cTE11} and f_{cTM01}. Now, $f_{cTE11new}$ = (f_{cTE11} + 10% of f_{cTE11}) and $f_{cTM01new}$ = (f_{cTM01} – 10% of f_{cTM01}).Calculate the safety margin as $f_{cTM01new}$ - $f_{cTE11new}$.

The MATLAB code for above is as follows:

<u>Circular Waveguide Safety Margin: (Code)</u>

```
% For Circular Waveguide (input a): Calculate the bandwidth of the
waveguide,
% assuming : (i) no safety margin and (ii) 20% safety margin. (Air
Media)

clear all;
clc;
q='y'
while(q=='y')
clc;
c=3*power(10,10);
a=input('Enter value of a <radius> (in cms):');

% TE(1,1) Mode in air medium.
fc_te_n=(1.841*c)/(2*pi*a);

% TM(0,1) Mode in air medium.
fc_tm_n=(2.405*c)/(2*pi*a);

BW_n=fc_tm_n-fc_te_n;
display('(i) Bandwidth with no safety margin (in Hz):');
disp(BW_n);

fc_te_2=fc_te_n+0.1*fc_te_n;
fc_tm_2=fc_tm_n-0.1*fc_tm_n;
BW_2=fc_tm_2-fc_te_2;
display('(ii) Bandwidth with 20% safety margin (in Hz):');
disp(BW_2);
q=input('Do you want to continue <y/n>:','s');
end
```

2.4.3 *Total Loss in Circular Waveguide*

Losses in a circular waveguide can be described as follows:

$\alpha_c=[R_s/(ak\eta K_z)]*(k_c^2+k^2/P_{11}'^2-1)$ Np/m

for TE_{11} mode

R_s=Surface Resistance = $\mathrm{sqrt}(\omega\mu_o/2\sigma)$

$\alpha_d=[\,k^2 \tan\delta/2k_z\,]$ (Plumbing)

The MATLAB code for above is as follows:

<u>Total Loss in Circular Waveguide: (Code)</u>

```
% Overall Loss in a Circular Waveguide

clear all;
clc;

Er=input('Enter the value of dielectric constant:');
a=input('Enter the value of radius <a> (in cms):');
f=input('Enter the value of operating frequency (in GHz):');
L=input('Enter the value of the length of the waveguide (in cms):');
tandel=input('Enter the value of loss tangent <tan (delta)> :');
sigma=input('Enter the value of conductivity of the material (X10^7
S/m):');
sigma=sigma*power(10,7);

%Definitions
f=f*power(10,9);
c=3*power(10,10);
w=2*pi*f;
uo=4*pi*power(10,-7);

%Calculations
Rs=sqrt((w*uo)/(2*sigma));
k=w*sqrt(Er)/c;
kc=1.8412/a;
kz=sqrt(power(k,2)-power(kc,2));
n=120*pi/sqrt(Er);

ac=(Rs/(a*k*n*kz))*(power(kc,2)+(power(k,2)/(power(1.8412,2)-1)));

ad=(power(k,2)*tandel)/(2*kz);

alpha=(ac+ad)*L;

display('Total loss in Nepers:');disp(alpha);
alpha_dB=alpha*8.686;
display('Total loss in dB:');disp(alpha_dB);
```

2.5 Coaxial Lines

2.5.1 *Coaxial Lines*

A coaxial transmission line is made up of two concentric conductors, each having a circular cross-section. The inner conductor is assumed to be a solid one, whereas the outer conductor is assumed to be hollow. The space between the two conductors is assumed to be filled with a dielectric material whose dielectric constant is ε_r . This material may be magnetic also, depending on the intended application. In some cases, the material may be air as well.

The coaxial line supports a TEM mode of transmission. It can also support TE and TM modes although those modes are generally avoided in practice.

As shown in Figure 2.5, the radius of the inner conductor is denoted by 'a' and that of the outer conductor is denoted by 'b'. . The direction of propagation is assumed to be along the z-axis, and the cylindrical coordinate system (ρ,ϕ,z) is employed for analysis. The point of observation is denoted by P. Also, unless otherwise stated, the medium inside the coaxial line is assumed to have electrical conductivity=0.

Please be aware that we are interested only in the range $a \le \rho \le b$., since fields outside the coaxial line will be zero because of the presence of a perfect electrical conductor at $\rho=b$. Fields inside the solid inner conductor will also be zero.

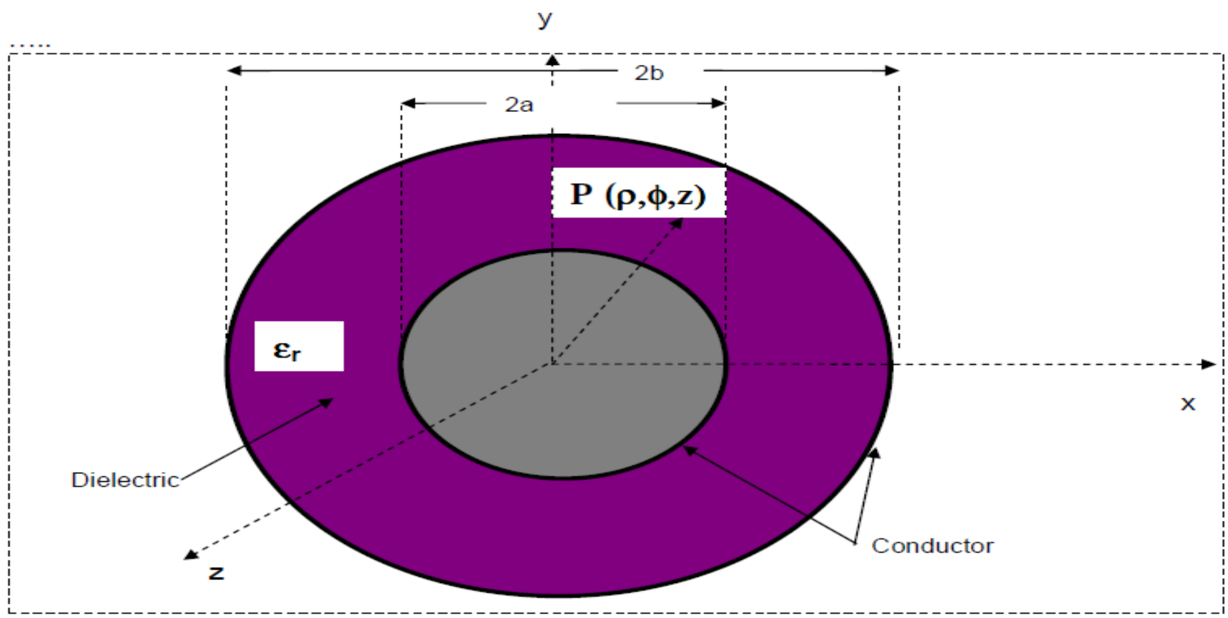


Figure 2.5 The cross-section of a coaxial lines

MATLAB code for calculating the Outer Radius of the coaxial line is as follows:

<u>Outer Radius of Coaxial Cable: (Cable)</u>

```
% To calculate the outer radius of the coaxial cable

clear all;
clc;
```

```
k=input('             Main-Menu\nCalculate the outer radius of coaxial
line:\n1.Without Dielectric Mediun.\n2.With Lossless Dielectric
Medium.\nYour Choice Please:');
display('             Output');
if(k==1)
    Zo=input('Enter the value of characteristic impedance (in ohms):');
    a=input('Enter the value of inner radius a (in cms):');
    b=exp(Zo/60)*a;
end
if(k==2)
    Zo=input('Enter the value of characteristic impedance (in ohms):');
    Er=input('Enter the value of Lossless Dielectric Constant:');
    a=input('Enter the value of inner radius a (in cms):');
    b=exp((Zo*sqrt(Er))/60)*a;
end

display('Value of outer radius of the coaxial cable (in cms):')
disp(b);

b=b*1000/2.54;
display('Value of outer radius of the coaxial cable (in mil):')
disp(b);
```

2.5.2 *RG-142*

Consider a piece of RG-142 Coaxial cable. Calculate 1>Z_o 2>$Y=\alpha+j\beta$ and 3>V_p and V_g of the cable in its dominant mode ; with all possible inputs.

The MATLAB code for above is as follows:

<u>Coaxial Cable RG-142: (Code)</u>

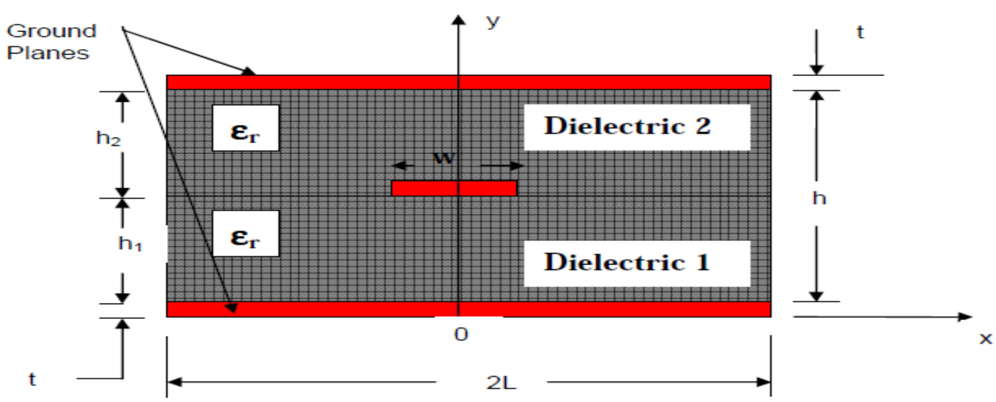
```
% Consider a piece of RG-142 Coaxial cable.Calculate 1>Zo 2>Y=a+jB 3>Vp
and
% Vg of the cable in its dominant mode ; with all possible inputs

clear all;
clc;

a=input('Enter the value of inner radius (in mil):');
b=input('Enter the value of outer radius (in mil):');
Er=input('Enter the value of Dielectric Constant:');
f=input('Enter the value of frequency (in GHz):');
f=f*power(10,9);
c=3*power(10,8);

Zo=(60/sqrt(Er))*log(b/a);

Y=(2*pi*f*sqrt(Er))/c;

Vp=c/sqrt(Er);
Vg=Vp;
```

```
display('The value of Characteristic Impedance Zo (in ohms):')
disp(Zo);

display('Propagation Constant (in rad/m):')
disp(Y);

display('Phase Velocity (in m/sec):')
disp(Vp);

display('Group Velocity (in m/sec):')
disp(Vg);
```

2.6 Microwave Integrated Circuits (MICs)

The Microwave Integrated Circuit (MIC) lines are a very useful alternative to coaxial lines and waveguides. The two main advantages of the MIC lines are i) miniaturized design and ii) lower fabrication cost in large numbers.
MIC lines are typically fabricated by using photolithography techniques to print the desired conductor pattern on a dielectric substrate. In many applications, multi-substrate designs are also used. .
The five most commonly used MIC lines are a) stripline, b) microstrip, c) slotline, d) coplanar waveguide (CPW) and e) coplanar strips (CPS).

2.6.1 *Semi-empirical formulas for "analyzing" a STRIPLINE*



Semi-empirical formulas for "analyzing" a STRIPLINE are as follows:

V_p=c/sqrt(ϵ_r)

Z_o=(30π/sqrt(ϵ_r)) * (1/((w_e/b)+0.441))

The effective width, w_e, of the centre conductor is given by:
$w_e/b = w/b$ for $w/b > 0.35$
$w_e/b = w/b - (0.35 - (w/b))^2$ for $w/b <= 0.35$
Thickness of the conductor, t, has been assumed to be zero which is sufficiently accurate approximation to reality.

Objective of Analysis: For given w, ϵ_r and h, Find Z_o and V_p.

The MATLAB Code for above is as follows:

<u>Semi-Empirical Formulae for "Analyising" a Stripline: (Code)</u>

```
clear all;
clc;

c=3*power(10,8);
h=input('Enter the substrate thickness (in mills):');
b=2*h;
w=input('Enter the width (in mills):');
Er=input('Enter the value of dielectric constant:');
if(w/b>0.35)
    web=w/b;
end

if(w/b<=0.35)
    web = w/b- power((0.35-w/b),2);
end

vp=c/sqrt(Er);
Zo=((30*pi)/sqrt(Er))*(1/(web+0.441));

disp('The phase velocity inside the Stripline (in m/sec)=')
disp(vp);

disp('The characteristic impedance of the Stripline (in ohms)=')
disp(Zo);
```

2.6.2 *Semi-empirical formulas for "designing" a STRIPLINE*

Objective: For a given ϵ_r and Z_o, find w for a given h.

Formulae involved are as follows:

w/b=x	for	$\text{sqrt}(\epsilon_r)*Z_o<120$
w/b=0.85-sqrt(0.6-x)	for	$\text{sqrt}(\epsilon_r)*Z_o>120$

where x= (30π/ sqrt(ϵ_r)) - 0.441

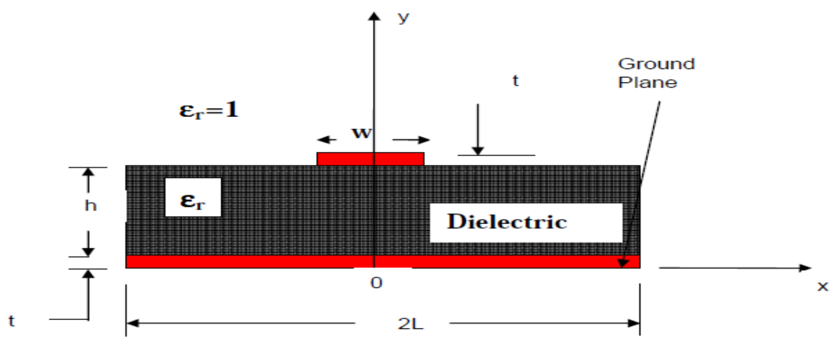
The MATLAB code demonstrating above concept is as follows:

<u>Semi-Empirical Formulae for "Designing" a Stripline: (Code)</u>

```
clear all;
clc;

Zo=input('Enter the value of characteristic impedance(in
Ohms):');
Er=input('Enter the value of dielectric constant:');
```

```
h=input('Enter the value of substrate thickness (in mills):');
b=2*h;

x= (30*pi)/(sqrt(Er)*Zo)-0.441;
if((sqrt(Er)*Zo)<120)
    w=x*b;
end
if((sqrt(Er)*Zo)>120)
    w=(0.85-sqrt(0.6-x))*b;
end

disp('The value of w (in mil)=')
disp(w);
```

2.6.3 *Semi-empirical formulas for "analyzing" a MICROSTRIP*



We define: $v_p = c/\mathrm{sqrt}(\epsilon_{r(eff)})$

where $\epsilon_{r(eff)}$ is known as the EFFECTIVE dielectric constant.

The semi-empirical formula for $\epsilon_{r(eff)}$ follows:

$\epsilon_{r(eff)} = (\epsilon_r + 1)/2 + ((\epsilon_r - 1)/2) * 1/\mathrm{sqrt}(1+12h/w)$

$Z_o = (60/\mathrm{sqrt}(\epsilon_{r(eff)})) * \ln(8h/w + w/4h)$ for $(w/h) <= 1$

$Z_o = 120\pi / (\mathrm{sqrt}(\epsilon_{r(eff)}) * [w/h + 1.393 + 0.667 \ln\{w/h + 1.44\}])$ for $(w/h) >= 1$

The MATLAB code demonstrating above concept is as follows:

<u>Semi-Empirical Formulae for "Analyising" a MICROSTRIP line: (Code)</u>

```
clc;
clear all;

display('This Program calculates the effective dielectric
constant and Characteristic impedance of a microstrip line');

Er=input('Enter the value of dielectric constant:');
h=input('Enter the value of height of the Microstrip line h(in
mil):');
w=input('Enter the value of width of the Microstrip Line w(in
mil):');
Ereff=((Er+1)/2)+((Er-1)/2)*(1/sqrt(1+12*(h/w)));
```

```
if(w/h<=1)
    Zo=(60/sqrt(Ereff))*log(8*(h/w)+(w/(4*h)));
end
if(w/h>1)
Zo=(120*pi)/(sqrt(Ereff)*((w/h)+1.393+0.667*log((w/h)+1.414)));
end

disp('The value of Effective Dielectric constant:')
display(Ereff);
disp('The value of Characteristic Impedance:')
display(Zo);
```

2.6.4 *Semi-empirical formulas for "designing" a MICROSTRIP*

Aspect ratio is defined as : (w/h)

The Semi-empirical formulas for "designing" a microstrip are as follows:

w/h=8eA/(e2A-2) for (w/h)>2

w/h= (2/ π)*[B-1-ln(2B-1)+(ϵ_r -1)/2 ϵ_r {ln(B-1)+0.39-0.61/ ϵ_r }] for (w/h)>2

where A=(Z0/60)*sqrt((ϵ_r +1)/2) + (ϵ_r -1)/ (ϵ_r +1) * (0.23+0.11/ϵ_r)

and B=377 π/(2*Z_o *sqrt(ϵ_r))

The MATLAB code demonstrating above concept is as follows:

<u>Semi-Empirical Formulae for "Designing" a Stripline: (Code)</u>

```
clc;
clear all;

Er=input('Enter the value of dielcetric constant:');
Zo=input('Enter the value of characteristic impedance (in
ohms):');
h=input('Enter the height of the Microstrip line (in mil):');

A=(Zo/60)*sqrt((Er+1)/2) + ((Er-1)/(Er+1))*(0.23+(0.11/Er));
B=(377*pi)/(2*Zo*sqrt(Er));

AR1=(8*exp(A))/(exp(2*A)-2);
AR2=(2/pi)*(B-1-log(2*B-1)+((Er-1)/(2*Er))*(log(B-1)+0.39-
(0.61/Er)));
if(AR1<=2)
    w=AR1*h;
end

if(AR2>2)
    w=AR2*h;
end

disp('The width of the Microstrip line (in mil):')
disp(w);
```

CHAPTER 3 – MICROWAVE NETWORK ANALYSIS

In this chapter, we will deal with Impedance Matching and Tuning, Microwave Resonators, Microwave Filters, Microwave Amplifier Design and Oscillators & Mixers.

Impedance Matching and Tuning consists of the topics: Matching with Lumped Elements (L-Network), Single Stub, Double stub and the Quarter Wave Transformer.

3.1 Impedance Matching and Tuning

Impedance matching or tuning of the circuit is important for the following reasons:

- Maximum power is delivered when the load is matched to the line (assuming the generator is matched), and power loss in the feed line is minimized.
- Impedance matching sensitive receiver components (antenna, low-noise amplifier, etc.) improves the signal-to-noise ratio of the system.
- Impedance matching in a power distribution network (such as an antenna array feed network) will reduce amplitude and phases errors.

Factors that may be important in the selection of a particular matching network include the following:

- *Complexity*- As with most engineering solutions, the simplest design that satisfies the required specifications is generally the most preferable. A simpler matching network is usually cheaper, more reliable, and less lossy than a more complex design
- *Bandwidth*- Any type of matching networks can ideally give a perfect(zero reflection) at a single frequency. In many applications, however, it is desirable to match a load over a band of frequency. There are several ways of doing this with, of course, a corresponding increase in complexity.
- *Implementation*- Depending on the type of transmission line or waveguide being used, one type of matching network may be preferable compared to another. For example, tuning stubs are much easier to implement waveguide than are multisection quarter-wave transformers
- *Adjustability*- In some applications the matching network may require adjustment to match a variable load impedance. Some types of matching network are more amenable than others in this regard.

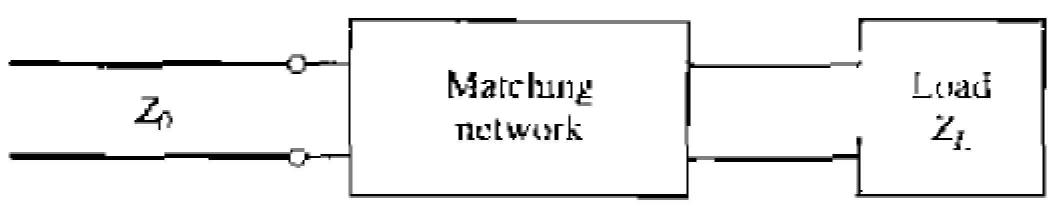


Figure 3.1 : A lossless network matching an arbitrary load impedance to a transmission line.

3.1.1 *Matching with Lumped Elements (L-Networks)*

Consider the following diagram

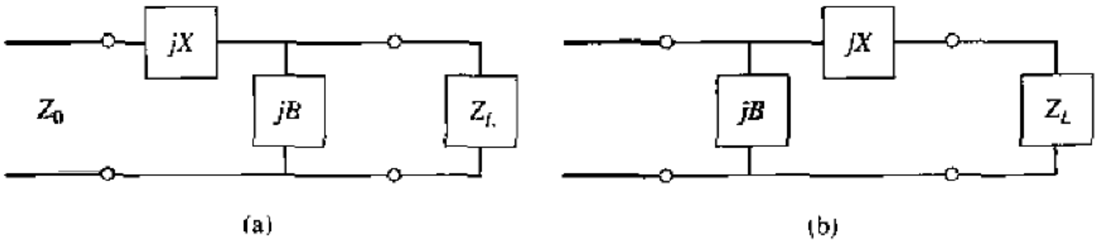


Figure 3.1.1: L-Section Matching Networks.(a) Network for Z_L inside 1+jx circle (b) Network for Z_L outside 1+jx circle.

Analytic solution to the above circuits may be given as:

For Fig. 3.1.1 (a), the following are derived (when $R_L>Z_o$):

$B=(X_L+/-\text{sqrt}(R_L/Z_o)*\text{sqrt}(R_L^2+X_L^2-Z_oR_L))/(R_L^2+X_L^2)$

$X=1/B+(X_LZ_o/R_L)-(Z_o/BR_L)$

For Fig. 3.1.1 (b), the following are derived (when $R_L<Z_o$):

$X=+/-\ \text{sqrt}(R_L(Z_o-R_L))-X_L$

$B=+/-\ \text{sqrt}((Z_o-R_L)/R_L)/Z_o$

The MATLAB code demonstrating the above concept is as follows:

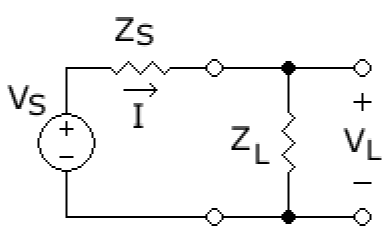
Matching with the lumped elements: (Code)

```
% Design an L-section matching network to match a series RC load
with an
% impedance Zl=Rl-jXl ohms, to a Zo ohms line, at frequency of
500 MHz.

clc;
clear all;

j=sqrt(-1);

Rl=input('Enter the real value of load impedance(in ohms):');
Xl=input('Enter the imaginary value of load impedance(in
ohms):');

Zl=Rl+j*Xl;

Zo=input('Enter the value of input impedance(in ohms):');

f=input('Enter the frequency (in MHz):');

if(Rl>Zo)
```

```
    B1=(Xl+sqrt((Rl/Zo))*sqrt(Rl*Rl+Xl*Xl-
Zo*Rl))/(Rl*Rl+Xl*Xl);b1=B1*Zo;
    B2=(Xl-sqrt((Rl/Zo))*sqrt(Rl*Rl+Xl*Xl-
Zo*Rl))/(Rl*Rl+Xl*Xl);b2=B2*Zo;
    X1=(1/B1)+((Xl*Zo)/Rl)-(Zo/(B1*Rl));x1=X1/Zo;
    X2=(1/B2)+((Xl*Zo)/Rl)-(Zo/(B2*Rl));x2=X2/Zo;
end

if(Rl<Zo)
    X1=sqrt(Rl*(Zo-Rl))-Xl;x1=X1/Zo;
    X2=-sqrt(Rl*(Zo-Rl))-Xl;x2=X2/Zo;
    B1=sqrt((Zo-Rl)/Rl)/Zo;b1=B1*Zo;
    B2=-sqrt((Zo-Rl)/Rl)/Zo;b2=B2*Zo;
end

display('The values of first possible network:');
disp('The value of Capacitance(in pF)=')
C1=b1/(2*pi*f*Zo)*power(10,6);
disp(C1);
disp('The value of Inductance(in nH)=')
L1=(x1*Zo)/(2*pi*f)*power(10,3);
disp(L1);
display('The values of second possible network:');
disp('The value of Capacitance(in pF)=')
C2=-1/(2*pi*f*x2*Zo)*power(10,6);
disp(C2);
disp('The value of Inductance(in nH)=')
L2=-Zo/(2*pi*f*b2)*power(10,3);
disp(L2);
```

3.1.2 *Simple Resistive Matching*

In electronics, **impedance matching** is the practice of designing the input impedance of an electrical load (or the output impedance of its corresponding signal source) to maximize the power transfer or minimize reflections from the load.



In the case of a complex source impedance Z_S and load impedance Z_L, maximum power transfer is obtained when

$$Z_S = Z_L^*$$

where * indicates the complex conjugates. Minimum reflection is obtained when

$$Z_S = Z_L$$

The concept of impedance matching was originally developed for electrical engineering, but can be applied to any other field where a form of energy (not necessarily electrical) is transferred between a source and a load. An alternative to impedance matching is impedance bridging, where the load impedance is chosen to be much larger than the source impedance and maximizing voltage transfer (rather than power) is the goal.

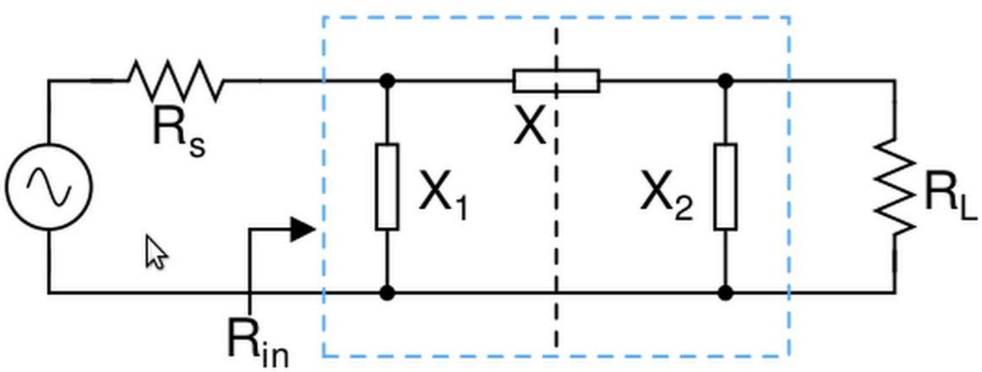
The MATLAB code demonstrating the above concept is as follows:

Simple Resistive Matching: (Code)

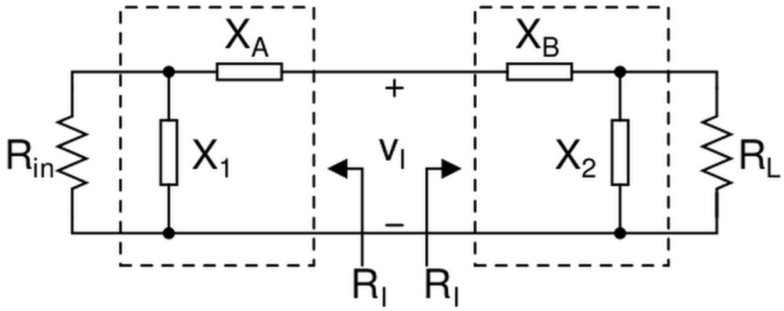
```
clear all;
clc;

R0=input('Enter the characteristic impedance of the line\n');
RL=input('Enter the load impedance , RL\n');
if R0>RL
RM=R0-RL;
Effic=RL/R0;
fprintf('RM=%d ',RM);
fprintf('Efficiency=%d\n',Effic);
fprintf('Reflection coefficient =0 , this was our
aim..!!!!\n\n\n');
else
fprintf('R0 should be greater than RL');
end
```

3.1.3 *π Matching(Pi Matching)*

A pi-match network is shown in the figure below. Two back-to-back connected l-match circuits form a π-match network. The additional element in π-match, compared with L-macth, allows independently setting impedance transformation ratio(R_L/R_{in}) and Q-factor of the circuit.



In the simplified figure shown below, if v_I is the voltage at the input of L-match circuit, and R_I is the impedance seen looking into each L-match circuit, then current flowing through inductor of each L-match circuit is $I_L = \frac{v_L}{R_I}$.



The MATLAB code demonstrating the above concept is as follows:

Π Matching: (Code)

```
% pie matching

clc;
clear all;

R0=input('Enter the characteristic impedance of the line:');
K=input('\nEnter the impedance ratio , K :');
RL=input('\nEnter the output load (output char. imped.) , RL:');
a=K*K;
b=K*RL+K*K*RL-R0*K*K-R0*K;
c= -2*RL*R0*K - RL*R0;
D= sqrt(b*b-4*a*c);
R1=(-b+D)/(2*a);
fprintf('\nR1=%d ',R1);
fprintf('\nR2 = %d\n', (K*R1));
```

3.1.4 *Using Reactive Elements with First Element as Series Element*

The MATLAB code demonstrating the above concept is as follows:

Using Reactive Elements with First Element as Series Element: (Code)

```
% LUMPED with first element in series

clc;
clear all;

R0=input('Enter the characteristic impedance of the line,R0:');
RL=input('\nEnter the real part of output load , RL:');
XL=input('\nEnter the imaginary part of output load , XL:');
if ((RL*RL + XL*XL) > (R0*RL))
```

```
a= RL*RL + XL*XL;
D= sqrt((a-R0*RL)*(RL/R0));
B1=(XL+D)/(a);
B2=(XL-D)/(a);
X1=(1/B1) + (XL*R0)/RL -R0/(B1*RL);
X2=(1/B2) + (XL*R0)/RL -R0/(B2*RL);
fprintf('B1=%d , ',B1);
fprintf('X1=%d \n',X1);
fprintf('B2=%d, ',B2);
fprintf('X2=%d \n',X2);
end
```

3.1.5 *Using Reactive Elements with First Element as Parallel Element*

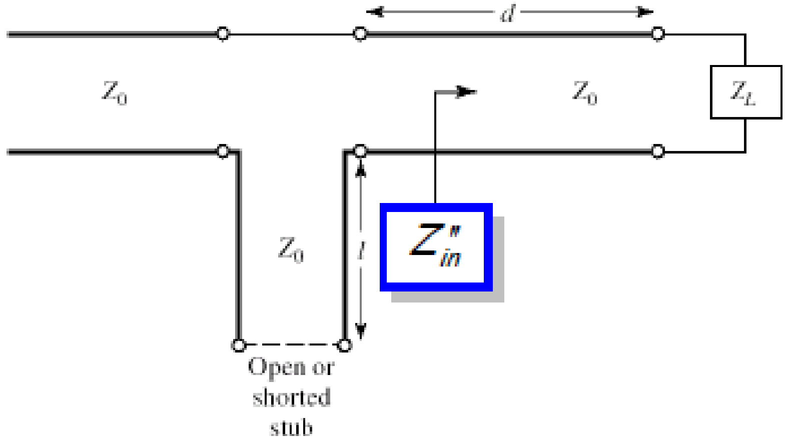
The MATLAB code demonstrating the above concept is as follows:

Using Reactive Elements with First Element as Parallel Element: (Code)

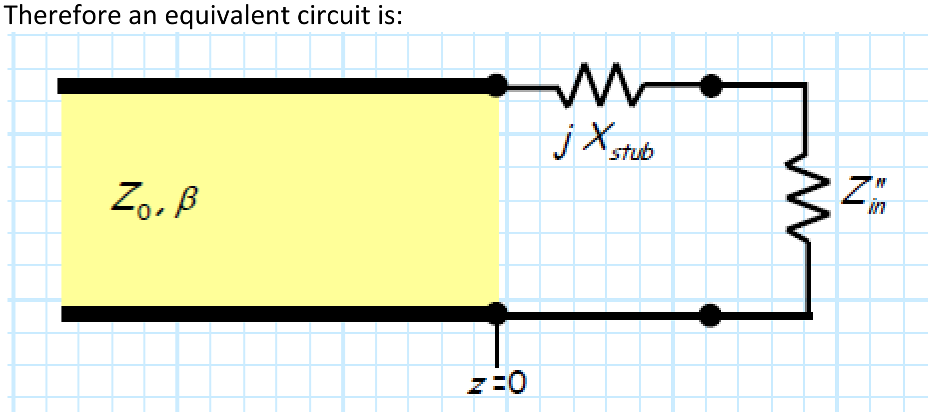
```
% LUMPED with first element in PARALLEL

clc;
clear all;

R0=input('Enter the characteristic impedance of the line,RO:');
RL=input('\nEnter the real part of output load , RL:');
XL=input('\nEnter the imaginary part of output load , XL:');

if (R0 > RL)
    X1=sqrt(RL*(R0-RL))-XL;
    X2=-sqrt(RL*(R0-RL))-XL;
    B1=sqrt(R0/RL-1)/(R0);
    B2=-sqrt(R0/RL-1)/(R0);
    fprintf('B1=%d , ',B1);
    fprintf('X1=%d \n',X1);
    fprintf('B2=%d ',B2);
    fprintf('X2=%d \n',X2);
end

if (R0<RL)
    fprintf('Characteristic Impedance should be greater than Load
Impedance\n');
end
```

3.1.6 *Impedance Matching Using Series Stub*

Consider the following transmission line structure, with series stub:



Therefore an equivalent circuit is:



Where of course:

$$Z_{in}'' = Z_0[(Z_L + jZ_0 \tan \beta d)/ (Z_0 + jZ_L \tan \beta d)]$$

And the **reactance** jX_{stub} is either:

$jX_{stub} = -jZ_0 \cot \beta l$ for an open-circuit stub
$jX_{stub} = jZ_0 \cot \beta l$ for an short-circuit stub

Therefore, for the matched circuit, we require:

$$jX_{stub} + Z_{in}'' = Z_0$$

i.e. $\mathrm{Re}\{ Z_{in}''\} = Z_0$
and $\mathrm{Im}\{ jX_{stub} + Z_{in}''\} = 0 \;\Rightarrow\; X_{stub} = -X_{in}''$
where $X_{in}'' = \mathrm{Im}\{ Z_{in}''\}$

Not the design parameters for this stub tuner are transmission line lengths d & l.
More specifically, we :
1> Set d such that $\mathrm{Re}\{ Z_{in}''\} = Z_0$.
2>Then set l such that $X_{stub} = -X_{in}''$.

The MATLAB code demonstrating the above concept is as follows:

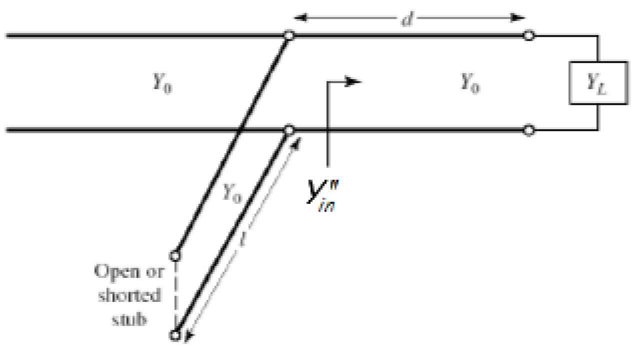
Impedance Matching Using Series Stub: (Code)

```
%Stub connection in SERIES

clear all;
clc;

R0=input('Enter the characteristic impedance of the line,RO:');
Y0=1/R0;
RL=input('\nEnter the real part of output load , RL:');
XL=input('\nEnter the imaginary part of output load , XL:');

GL=RL/(RL*RL+XL*XL);
BL=-XL/(RL*RL+XL*XL);

F=input('\nEnter the FREQUENCY , F(in GHz):');
OC=input('Select stub type(0/1) \n(0)Open circuited\n(1)Short
circuited\n');

LAM=0.3/F;
A= Y0*(GL-Y0);
B=-2*BL*Y0;
C=Y0*GL-GL*GL-BL*GL;

T1=(-B+sqrt(B*B-4*A*C))/(2*A);
T2=(-B-sqrt(B*B-4*A*C))/(2*A);

D1=LAM/(2*pi)*atan(T1);
D2=LAM/(2*pi)*atan(T2);
if (D1<0)
D1=D1+ LAM/2;
end

if (D2<0)
D2=D2+ LAM/2;
end

X1=(T1*GL*GL-(Y0-BL*T1)*(BL+Y0*T1)) / (Y0*(GL*GL +
power((BL+Y0*T1),2)));
X2= (T2*GL^2-(Y0-BL*T2)*(BL+Y0*T2)) / (Y0*(GL*GL +
power((BL+Y0*T2),2)));

L1=(-2*OC+1)*LAM/(2*pi)*atan(power((Y0/X1),(-2*OC+1)));
L2=(-2*OC+1)*LAM/(2*pi)*atan(power((Y0/X2),(-2*OC+1)));

if (L1<0)
L1=L1+ LAM/2;
end

if (L2<0)
L2=L2+ LAM/2;
end
```

```
fprintf('D1=%d m , ',D1);
fprintf('T1=%d \n',T1);

fprintf('D2=%d m ',D2);
fprintf('T2=%d \n ',T2);

fprintf('X1=%d ohm , ',X1);
fprintf('L1=%d m\n',L1);

fprintf('X2=%d ohm ',X2);
fprintf('L2=%d m\n ',L2);
fprintf('\n ALL UNITS IN S.I. \n');
```

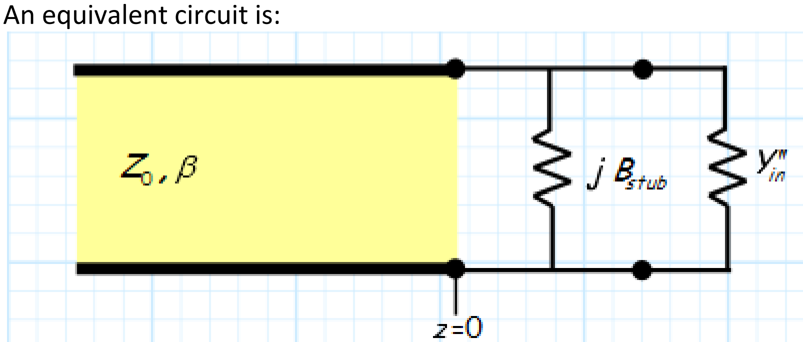
3.1.7 *Impedance Matching Using Parallel Stub*

Consider the following transmission line structure, with a shunt-stub:



The two design parameters for this matching network are lengths l and d.

An equivalent circuit is:



Where of course:

$$Y_{in}'' = Y_0[(Y_L + jY_0 \tan \beta d) / (Y_0 + jY_L \tan \beta d)]$$

And the **reactance** jB_{stub} is either:

$jB_{stub} = -jY_0 \tan \beta l$	for an open-circuit stub
$jB_{stub} = jY_0 \cot \beta l$	for an short-circuit stub

Therefore, for the matched circuit, we require:

$$jB_{stub} + Y_{in}'' = Y_0$$

i.e. $\mathrm{Re}\{ Y_{in}'' \} = Y_0$

and $\mathrm{Im}\{ jB_{stub} + Y_{in}'' \} = 0 \Rightarrow B_{stub} = -B_{in}''$

where $B_{in}'' = \mathrm{Im}\{ Y_{in}'' \}$

Not the design parameters for this stub tuner are transmission line lengths d & l. More specifically, we :

1> Set d such that $\mathrm{Re}\{ Y_{in}'' \} = Y_0$.

2>Then set l such that $B_{stub} = -B_{in}''$.

The MATLAB code demonstrating the above concept is as follows:

Impedance Matching Using Parallel Stub: (Code)

```
%Stub connection in PARALLEL
clc;
clear all;

R0=input('Enter the characteristic impedance of the line,RO:');
Y0=1/R0;
RL=input('\nEnter the real part of output load , RL:');
XL=input('\nEnter the imaginary part of output load , XL:');
F=input('\nEnter the FREQUENCY , F(in GHz):');
OC=input('Select stub type(0/1)\n(0)Open circuited\n(1)Short
circuited\n');

LAM=0.3/F;

A= R0*(RL-R0);
B=-2*XL*R0;
C=R0*RL-RL*RL-XL*XL;

T1=(-B+sqrt(B*B-4*A*C))/(2*A);
T2=(-B-sqrt(B*B-4*A*C))/(2*A);

D1=LAM/(2*pi)*atan(T1);
D2=LAM/(2*pi)*atan(T2);

if D1<0
D1=D1+ LAM/2;
end

if D2<0
D2=D2+ LAM/2;
end

B1=(T1*RL*RL-(R0-XL*T1)*(XL+R0*T1)) / (R0*(RL*RL +
power((XL+R0*T1),2)));
B2= (T2*RL*RL-(R0-XL*T2)*(XL+R0*T2)) / (R0*(RL*RL +
power((XL+R0*T2),2)));
```

```
B_STUB1=-B1;
B_STUB2=-B2;

L1=(2*OC-1)*LAM/(2*pi)*atan(power((Y0/B1),(2*OC-1)));
L2=(2*OC-1)*LAM/(2*pi)*atan(power((Y0/B2),(2*OC-1)));

if L1<0
L1=L1+ LAM/2;
end

if L2<0
L2=L2+ LAM/2;
end

fprintf('D1=%d m , ',D1);
fprintf('T1=%d \n',T1);

fprintf('D2=%d m',D2);
fprintf('T2=%d \n',T2);

fprintf('B_STUB1=%d mho , ',-B1);
fprintf('L1=%d m\n',L1);

fprintf('B_STUB2=%d mho',-B2);
fprintf('L2=%d m\n',L2);

fprintf('\n ALL UNITS IN S.I. \n');
```

3.1.8 *Impedance Transformer of any Frequency*

The MATLAB code demonstrating the above concept is as follows:

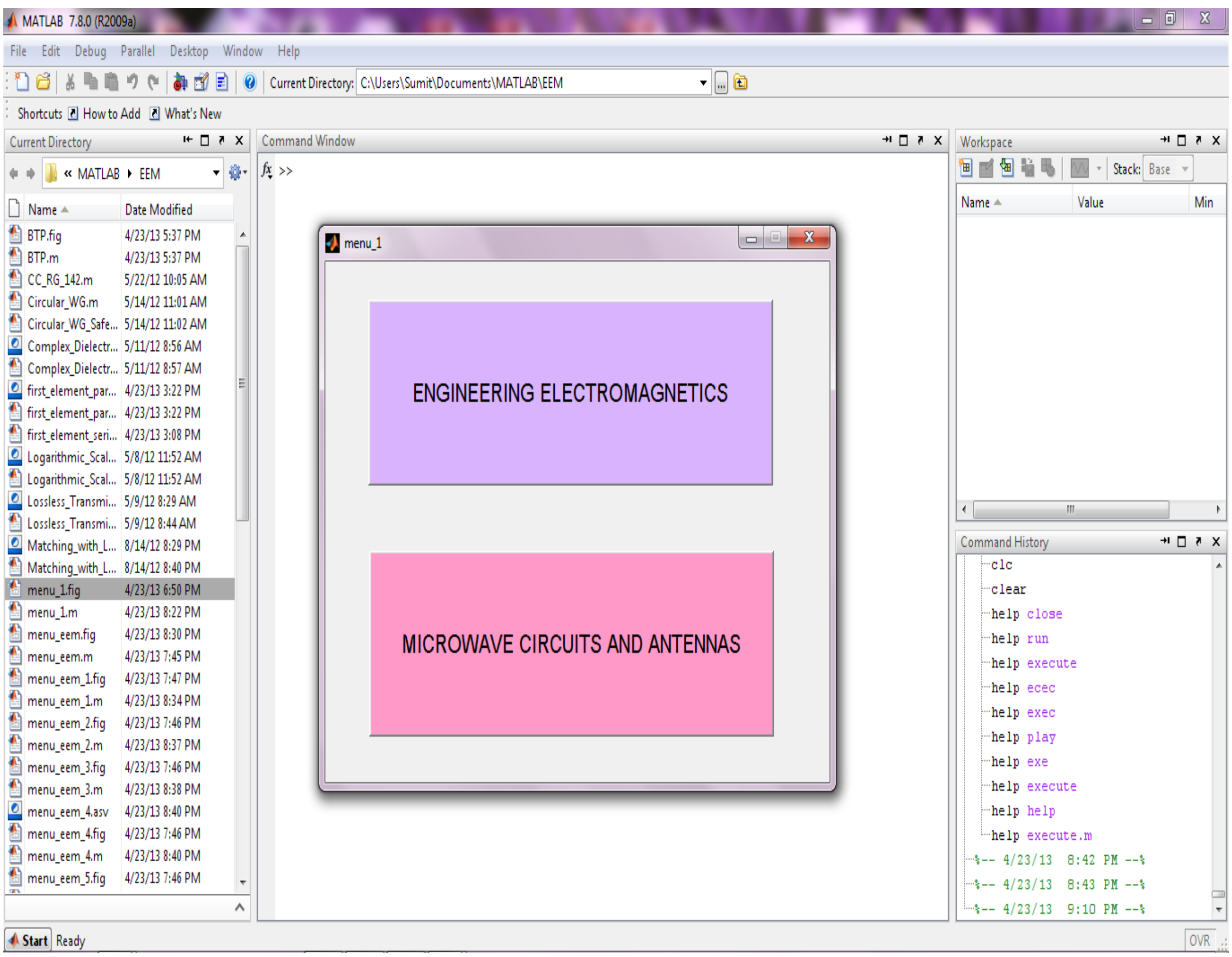
Impedance Transformer of any Frequency: (Code)

```
% Transformer

clc;
clear all;

R0=input('Enter the characteristic impedance of the line:');
RL=input('\nEnter the load impedance , RL:');
F=input('\nEnter the Frequency:');
V=input('\nEnter the VSWR of the line:');
Z1=sqrt(RL*R0);
Pm = (V-1)/(V+1);
Q = acos((Pm/sqrt(1-Pm*Pm))*(2*sqrt(R0*RL)/abs(RL-R0)));
FB= (2-4*Q/pi)*100;
fprintf('Characteristic impedance of the matching section=%d
\n',Z1);
fprintf('Q=%d \n',Q);
fprintf('Fractional Bandwidth=%d\n',FB);
```

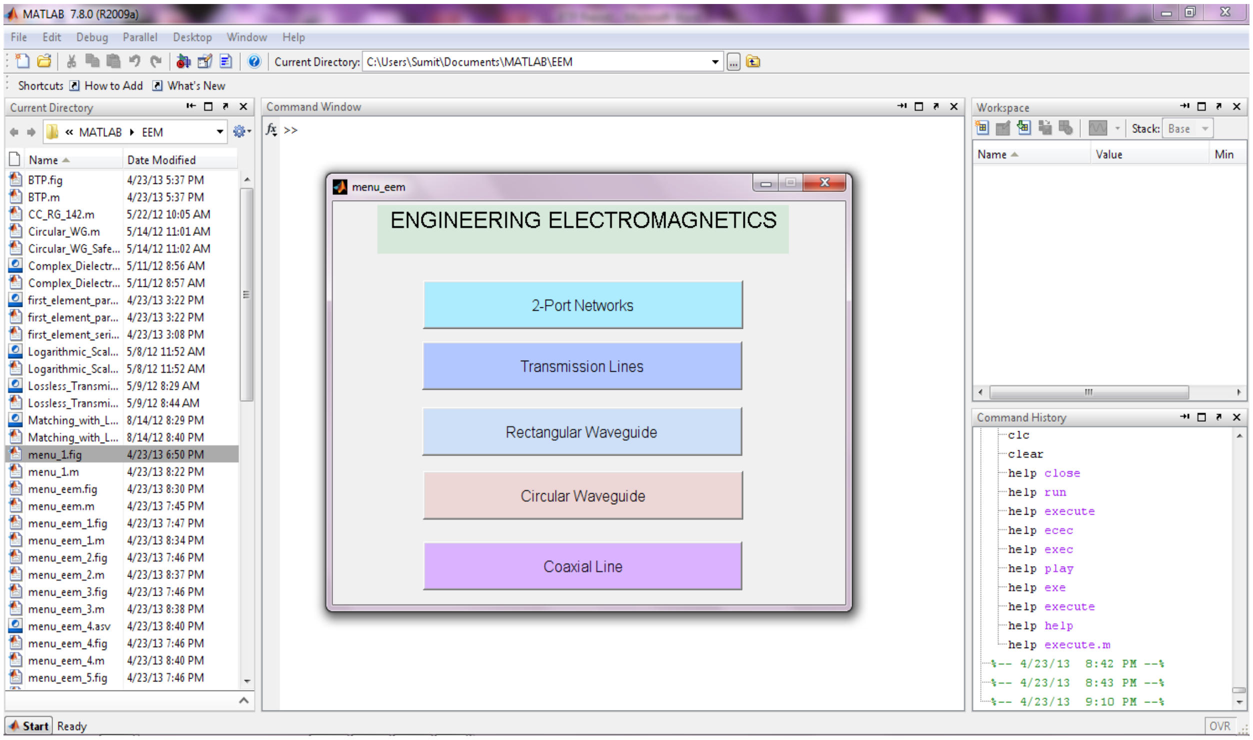
CHAPTER 4 – HOW TO EXECUTE THE PROJECT USING MATLAB GUI

A Graphic User Interface has been made with the help of MATLAB to execute the project easily. There is a BTP folder been provided with the project which contains all the MATLAB codes as provided in earlier parts of this report. To easily explore the project, one needs to perform the following tasks:

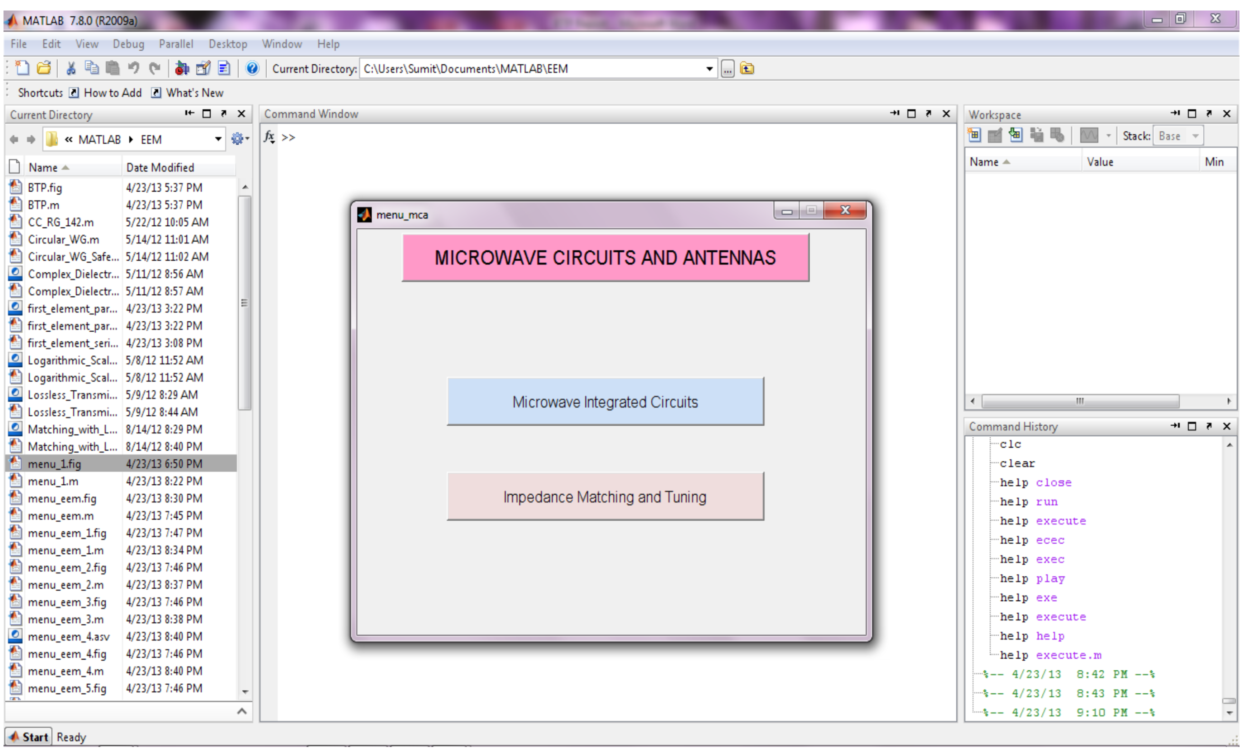
1. The user need to run the file named “menu_1.fig” from the MATLAB command prompt window with same path in which all codes are stored and this menu will be displayed:



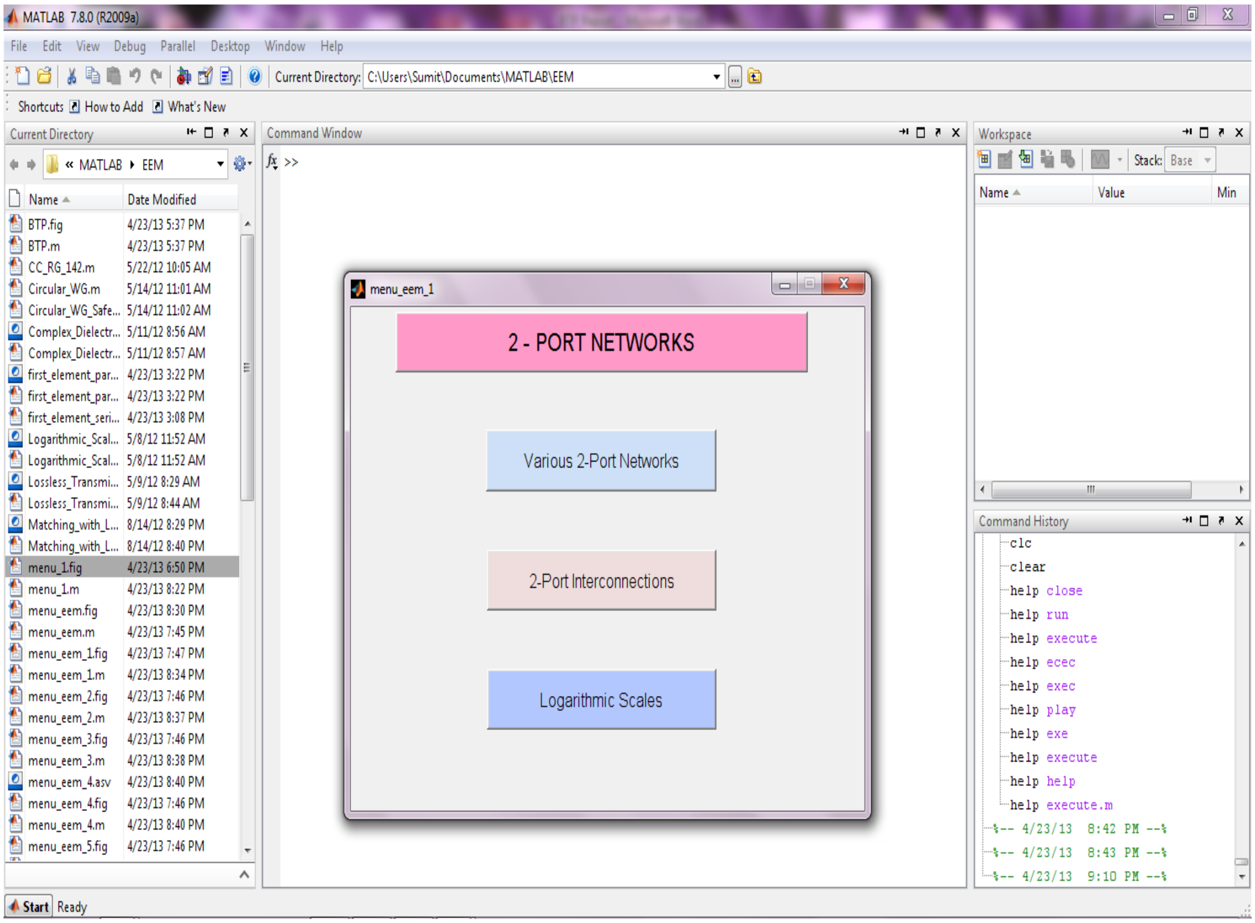
2. If the User clicks on “ENGINEERING ELECTROMAGNETICS”, the following is displayed:



3. Similarly, if the User clicks on “MICROWAVE CIRCUITS AND ANTENNAS”, the following is displayed:



4. On going futher, inside to the sub-topics, on final stage, the '.m' file of the desired topic will be open. This file can be run and the required output can be generated. For example, in the figure below, if the user clicks on "Various 2-Port Networks", a MATLAB file will be opened. The user should thus run the file to get the desired output.



CHAPTER 5 – CONCLUSIONS

Microwave engineering pertains to the study and design of microwave circuits, components, and systems. Fundamental principles are applied to analysis, design and measurement techniques in this field. The short wavelengths involved distinguish this discipline from Electronic engineering. This is because there are different interactions with circuits, transmissions and propagation characteristics at microwave frequencies.

Some theories and devices that pertain to this field are antennas, radar, transmission lines, space based systems (remote sensing), measurements, microwave radiation hazards and safety measures. The microwave engineering discipline has become relevant as the microwave domain moves into the commercial sector, and no longer only applicable to 20^{th} and 21^{st} century military technologies. Inexpensive components and digital communications in the microwave domain have opened up areas pertinent to this discipline. Some of these areas are radar, satellite, wireless radio, optical communication, faster computer circuits, and collision avoidance radar.

Microwave engineering today lays the foundations for tomorrow's infrastructure in literally every aspect of life. Securing a good position in tomorrow's global economy requires the qualified human resources, which should be brought in from overseas while the industry, government and academy act to increase the number of graduates to meet the demand.

Today, the majority of applications of microwaves are related to radar and communication systems. Radar systems are used for detecting and locating targets and for air traffic control systems, missile tracking radars, automobile collision avoidance systems, weather prediction, motion detectors, and a wide variety of remote sensing systems.

Owing to the demands in the Microwave Engineering field, this project is very worthy considering the future of Microwave Engineering. It can be amended from time to time as per requirements. However, it is not a complete package of what would be required as per industrial norms. More important topics need to be added to make this project worth for the industrial usage.

BIBLIOGRAPHY

[1] Lecture Notes provided in the course "**Engineering Electromagnetics**" by *Prof. R. Tomar*.

[2] Text Book – "**Microwave Engineering**" by *David Pozar*.

[3] Text Book – "**EM Waves and Radiating Systems**" by *E.C.Jordan and K.G.Balmain*.

[4] Text Book – "**Microwave Engineering**" by *Annapurna Das and Sisir K Das*.

[5] http://www.electromagneticcareers.com/Index.aspx?ItemID=120

[6] **Google Search Engine** (http://www.google.com/) (For searching key terms)

[7] http://www.ittc.ku.edu/~jstiles/723/handouts/

[8] **Wikipedia** (http://en.wikipedia.org/)

[9] http://www.nitt.edu/

[10] http://www.madmadscientist.com/html/Theory.htm

[11] http://home.sandiego.edu/~ekim/e194rfs01/

[12] http://webee.technion.ac.il/people/schachter/Teaching/Microwaves%202011-2012-locked.pdf