UNIVERSITY OF NAIROBI
DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

A VECTOR NETWORK ANALYZER FOR LOW FREQUENCY APPLICATIONS

PROJECT INDEX: PROJ-017
NAME: CHESIR ELISHA KIMUTAI
REG. NUMBER: F17/1370/2010

SUPERVISOR: DR. W.MWEMA
EXAMINER:

Project report submitted in partial fulfilment of the requirement for the award the degree of Bachelor of Science in Electrical and Electronic Engineering of University of Nairobi

Date of Submission: APRIL, 2015
DECLARATION OF ORIGINALITY

NAME OF THE STUDENT: CHESIR ELISHA KIMUTAI
REGISTRATION NUMBER: F17/1370/2010
COLLEGE: ARCHITECTURE AND ENGINEERING
FACULTY/SCHOOL/INSTITUTE: ENGINEERING
DEPARTMENT: ELECTRICAL AND INFORMATION
COURSE NAME: BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING
TITLE OF WORK: VECTOR NETWORK ANALYZER FOR LOW FREQUENCY APPLICATIONS

1) I understand what plagiarism is and I am aware of the university policy in this regard.

2) I declare that this my final year report is my original work and has not been submitted elsewhere for examination, award of degree or publication. Where other people’s work has been used, this has properly been acknowledged and referenced according to the University of Nairobi’s requirements.

3) I have not sought or used the services of any professional agencies to produce his work.

4) I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.

5) I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with university anti-plagiarism policy.

Signature: ……………………………………………………………………………………………………………………………………………………………

Date: ……………………………………………………………………………………………………………………………………………………………
DEDICATION

I dedicate this project to my parents-they have been a great pillar in my life, my siblings Francisco, Otilia and Emma to all my friends who made my time in school a joyous happening.
ACKNOWLEDGEMENTS:
I would like to express a deep sense of gratitude and appreciation to all those who made it possible for me to complete this project. Special thanks to my supervisor, Dr. Wilfred N. Mwema, Department of Electrical & Electronics Engineering, University of Nairobi. Without his wise counsel and able guidance, it would have been impossible to complete this project. Thanks Sir for all your moral support and your ideas!

Last but not least, I thank God for giving me a great family, friends and great teachers in all respect of life, for allowing me to share all these experiences with them and giving me peace and strength to tackle all the difficulties encountered throughout the years and during this project.
# Table of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>General two port Network</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Semiconductor parameter analyzer displaying the h values of a 2N3904 BJT</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>System model</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>The block diagram of the design</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>The schematic diagram of the signal conditioner</td>
<td>20</td>
</tr>
<tr>
<td>3.3</td>
<td>Breadboard implementation</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Output of the signal conditioner at different amplitudes</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>Bode plot without the DUT</td>
<td>23</td>
</tr>
<tr>
<td>4.3(a)</td>
<td>Bode plot with the output signal altered using the PSD method</td>
<td>24</td>
</tr>
<tr>
<td>4.3(b)</td>
<td>Bode plot with the output signal altered using the direct division method</td>
<td>24</td>
</tr>
<tr>
<td>4.4</td>
<td>VNA in use to characterize the DUT (a loudspeaker system)</td>
<td>25</td>
</tr>
<tr>
<td>4.5(a)</td>
<td>Frequency response of the tweeter system</td>
<td>26</td>
</tr>
<tr>
<td>4.5(a)</td>
<td>Frequency response of the bass system</td>
<td>26</td>
</tr>
</tbody>
</table>
# Table of contents

DECLARATION OF ORIGINALITY.................................................................................................................. i

DEDICATION.................................................................................................................................................. ii

ACKNOWLEDGEMENTS................................................................................................................................. iii

TABLE OF FIGURES....................................................................................................................................... iv

GLOSSARY AND ACRONYMS........................................................................................................................ vii

SYMBOLS AND DEFINITIONS..................................................................................................................... viii

ABSTRACT........................................................................................................................................................ ix

CHAPTER 1: INTRODUCTION.......................................................................................................................... 1

1.1. Background............................................................................................................................................... 1

1.2. Problem statement................................................................................................................................... 1

1.3. Objectives................................................................................................................................................ 2

1.3.1. Overall objectives................................................................................................................................. 2

1.3.2. Specific objectives................................................................................................................................. 2

1.4. Design approach...................................................................................................................................... 2

1.5. Scope of the project................................................................................................................................. 3

1.6. Relevance of the project......................................................................................................................... 3

1.7. Report layout .......................................................................................................................................... 4

CHAPTER 2..................................................................................................................................................... 5

2.1. Literature review...................................................................................................................................... 5

2.1.1. Two-port network representations.................................................................................................... 5

2.1.1.1. Z-parameters................................................................................................................................. 6

2.1.1.2. Y-parameters................................................................................................................................. 6

2.1.1.3. H-parameters................................................................................................................................. 7

2.1.1.4. Transmission parameters............................................................................................................... 8

2.1.2. Setbacks.............................................................................................................................................. 10

2.2. Theory and background......................................................................................................................... 10

2.2.1. Building blocks of a VNA.................................................................................................................. 11

2.2.3. MATLAB application in measurement............................................................................................. 11

2.2.4. The transfer function concept............................................................................................................ 13
2.2.5. Signal conditioner........................................................................................................16

CHAPTER 3: DESIGN AND IMPLEMENTATION.........................................................................18

3.1. Design consideration........................................................................................................18
3.2. System building blocks....................................................................................................18
   3.2.1. Swept frequency sine wave generator.......................................................................19
   3.2.2. Signal conditioner......................................................................................................20
   3.2.3. Signal analyzer........................................................................................................21
3.3. Testing the VNA................................................................................................................21

CHAPTER 4: RESULTS AND DISCUSSION..............................................................................22

4.1. Results.............................................................................................................................22
4.2. Discussion.........................................................................................................................27

CHAPTER 5: CONCLUSION....................................................................................................28

5.1.........................................................................................................................................28
5.2.........................................................................................................................................28
5.3.........................................................................................................................................28

References..................................................................................................................................29

Appendix..................................................................................................................................30
GLOSSARY AND ACRONYMS

RF - Radio Frequency
VNA - Vector Network Analyzer
DUT - Device under Test
PC - Personal Computer
LO - Local oscillator
TF - Transfer Function
R & D - Research and Development

MATLAB- MATLAB is a high performance language for technical computing. It integrates computing, visualization, and programming.
SYMBOLS AND DEFINITIONS

S-Parameters - Scattering Parameters
PSD - Power Spectral Density
FFT - Fast Fourier Transform
DFT - Discrete Fourier Transform
DTFT - Discrete Time Fourier Transform
LPF - Low Pass Filter
HPF - High Pass Filter
$w$ - frequency (rads/s)
$x$ - Information vector
ABSTRACT

In building more complex electrical systems, the electrical performance of circuits and components used need to be predetermined, designers need to accurately characterize these networks. This project proposes the design and realization of a vector network analyzer for low frequency applications.

A high-performance language for technical computing, MATLAB which integrates computation, programming, and visualization has been used to realize this project basing on the concept of computing the rough estimate of a system transfer function by comparing the power, at different frequencies, of a signal before and after being input to the device under test. It is based on the fact that an input at a given frequency for a linear for a system result in an output of the same frequency.

The experimental part of the project proves it is possible to build an efficient MATLAB based vector network analyzer cheaply that provides a better visualization and more flexible and comprehensive information than the traditional methods.
CHAPTER 1: INTRODUCTION

1.1. Background

There are several reasons as to why components need to be tested. Most components are used to build more complicated systems. For instance, there are amplifiers in most transceivers to boost LO (local oscillator) power to mixers, as well as filters for signal harmonics removal. Most are times, Research & Development engineers are required to measure effects of these components to determine their simulation models and actual hardware prototypes. In order to produce components, manufacturers ought to measure the performance of these products to ensure they provide the exact specifications for each. This way will know the behavior of a particular component in their application.

When electrical systems are used in conveying low-frequency signals containing information, the main point of concern is ensuring maximum efficiency as well as minimized distortion in transmission. In case of any unexpected response of electrical networks, vector –correction to eliminate system errors and improve accuracy in measurement. This process will demand that the phase and magnitude data be obtained to build a proper error model. To measure the electrical performance of circuits and components used in complex systems, designers need to accurately characterize these two port networks such as amplifiers and filters using network analyzers in a method termed as network analysis basing on their effect on amplitude and phase of signals whose characteristics are known.

1.2. Problem statement

In most cases, only information on magnitude is sufficient for analysis. For example, to obtain the gain of a given amplifier or the band specification of a filter only the magnitude data will be needed. However, phase measurement is an element critical to network analysis. Sufficient of device and networks characterization requires the measurement of phase and magnitude as well. This is fundamental in developing electric circuit models for simulation and to match circuits based on the normal techniques of conjugate matching. Moreover, time-domain characterization requires magnitude and phase data. The inverse transforms can then be done on the Fourier transform of the transfer functions of these networks and components. Phase information is also necessary to improve accuracy of measurement as well as in performing vector error correction.

There are several ways in which network analysis have been done in the past or is being done presently. Usually characterization of low frequency devices or networks is mainly based on measuring the H, Y and Z parameters. To compute a set of electrical parameters of linear 2-port networks, measurements are always done under various conditions (open-circuit and short-circuit conditions.) and the total voltage and current at the input and output ports of a device of a network are measured. These parameters can then be used to give a complete description on the electrical behavior of the device when placed between a source and a load.

However this methods are time consuming, tedious and involve a lot of measurement and calculation. Moreover, several equipment are involved in performing the task. This reduces the effectiveness of the process because the accuracy of taking measurements is compromised as
several readings from different equipments will MATLAB have to be taken and compiled before establishing the behavior of the DUT.

The network analyzer developed here takes in consideration the fact that the output any system is defined by a transfer function and applies this concepts in obtaining the transfer function of a DUT from a math program and plotting its characteristics using the appropriate commands. These method provides a better visualization of the characteristics of a device as the information can be processed easily and presented in a way considered best for analysis and drawing of conclusions.

This method allows the use of a single equipment (in this case a P.C with an installed program) and a simple preconditioning circuit to protect the components of the PC involved in the measurement. This makes it extremely accurate, fast and less complex method to use in network analysis surpassing the traditional methods where a lot of equipments and calculations are involved in obtaining the device transfer characteristics.

1.3. Objectives

1.3.1. Overall objective

The main aim of this project is to develop a functional vector network analyzer for low-frequency applications.

1.3.2. Specific objectives

The VNA should be able to perform the following:

- Generate a swept-frequency and swept-power test signals within the MATLAB space.
- Input the test signal to the DUT and obtain the output into the MATLAB space.
- Compute the transfer function of the DUT and plot its frequency response.

1.3.3. Design approach

The necessity for efficient power transmission is one of the major reasons for the utilization of transmission lines at higher frequencies. At audio frequencies signals have much larger wavelengths and therefore a simple wire is adequate for power conduction. The resistance of the wire is analytically low and has minor effect on this category of signals. No matter where the measurements are made, the transit voltage and current are relatively the same. Therefore there is no need to consider scattering parameters (S-parameters) instead the two port DUT is input with a swept-frequency signal and the output is taken for analysis by comparing it with the input to obtain the device transfer function.

The main factors taken in consideration include:

- Designing a low-cost VNA
- Efficiency
- Simplicity
This project targets to design a VNA that utilizes the MATLAB in generation of test signal as well as analysis of the output signal. It also incorporates a pre-conditioning circuit just for the purpose of device protection for instance in case where the test signal is amplified beyond levels that can be tolerated by the components involved in measurement.

1.4. Scope of the project

There exists different types of networks and various mechanisms of establishing their behavior. This project is built for the purpose of characterizing two port networks basing their response to low frequency signals. It entails the use of MATLAB data acquisition (DAQ) as well as signal processing units with a simple signal conditioning components.

The project majors on establishing a cheap, efficient and effective way of characterizing low frequency network applications as well as demonstrating its functioning by designing and implementing a simple VNA by generating a MATLAB code that can address the PC sound card which is simply a DAC and ADC at the same time and interfaces the analogue signals with the PC which can only process digital signals.

This project covers only the vector measuring instrument and a simple illustration on its functioning. How the system is used or applied later on is beyond the scope of this project.

1.5. Relevance of the project

Network analyzers in low frequencies are particularly used in measuring linear characteristics, to measure device magnitude, phase, and impedance this done by measuring their effect on the amplitude and phase of swept-frequency and swept-power test signals components. This is important for various reasons. First of all, to fully define the characteristics of a linear network and ensure distortion-free transmission both measurements are required.

Moreover, time-domain characterization requires magnitude and phase information in order to perform an inverse-Fourier transform which is necessary in studying the time domain characteristics of a given network. Vector–correction to eliminate system errors and improve accuracy in measurement will demand that the phase and magnitude data be obtained in order to build a proper error model. Capability to perform phase-measurement is equally important even when making scalar measurements in order to achieve high levels of accuracy.

Important application of VNAs exist in signal transmission, filters and amplifiers. In any communications system, the effects and consequences of signals getting distorted need to be considered. Many are times we highly regard the distortion occurring due to the nonlinear properties (for instance, when desired carrier signals introduce products of intermodulation), signal distortion can also be introduced by purely linear systems. The signal time waveform can be changed by linear systems for as they pass through them their amplitude and phase of the spectral (frequency) components that make up the signal are altered.
1.6. Report layout

The following section briefly describes the outline of each chapter:

➢ The two sections contained in chapter 2 are:

• Literature review
The various possible applications of the network analyzer.

• Theoretical Background
A case study as well as the concepts necessary to understand the fundamental principles observed in the project are discussed here.

➢ In Chapter 3 the design and implementation are presented.
Several concepts have been included, namely:

• Block diagrams used giving a general view of the project
• Characterizing given devices such as amplifiers and filters
  ➢ In chapter 4 project analysis and results have been outlined
  ➢ In chapter 5 conclusions are drawn and recommendations have been made
CHAPTER 2:

2.1. Literature review

At low frequencies the wavelength is far much greater as compared to the wire length therefore current can still flow along this wires and still maintain a high efficiency power transfer. The measured voltage and current does not depend on the position along the wire.

There are several methods used to characterize networks at low frequencies. In this chapter the most of the common methods of analysis have been reviewed. This is the use of Z, Y and H parameters. It is from this information that we can obtain the transfer characteristics of a DUT.

In this section, the describing equations for the various two-port network representations are given.

2.1.1. Two-port network representations [1, pp. 174-197] [1]

A general two-port network is shown in Figure below.

![Figure 2.1. General Two-Port Network](image)

$I_1$ and $V_1$ are input current and voltage, respectively. Also, $I_2$ and $V_2$ are output current and voltage, respectively. It is assumed that the linear two-port circuit contains no independent sources of energy and that the circuit is initially at rest (no stored energy). Furthermore, any controlled sources within the linear two-port circuit cannot depend on variables that are outside the circuit.
2.1.1.1. Z-parameters

A two-port network can be described by z-parameters as

\[ V_1 = z_{11}I_1 + z_{12}I_2 \] .............................. (1)

\[ V_2 = z_{21}I_1 + z_{22}I_2 \] .............................. (2)

In matrix form, the above equation can be rewritten as

\[
\begin{bmatrix}
V_1 \\
V_2
\end{bmatrix} =
\begin{bmatrix}
z_{11} & z_{12} \\
z_{21} & z_{22}
\end{bmatrix}
\begin{bmatrix}
I_1 \\
I_2
\end{bmatrix}
\]

The z-parameter can be found as follows

\[ z_{11} = \frac{V_1}{I_1} \bigg|_{I_2=0} \]

\[ z_{12} = \frac{V_1}{I_2} \bigg|_{I_1=0} \]

\[ z_{21} = \frac{V_2}{I_1} \bigg|_{I_2=0} \]

\[ z_{22} = \frac{V_2}{I_2} \bigg|_{I_1=0} \]

The z-parameters are also called open-circuit impedance parameters since they are obtained while the output side of the network is kept open.

2.1.1.2. Y-parameters

A two-port network can also be represented using y-parameters. The describing equations are

\[ I_1 = y_{11}V_1 + y_{12}V_2 \]

\[ I_2 = y_{21}V_1 + y_{22}V_2 \]

Where,

\[ V_1 \] and \[ V_2 \] are independent variables and

\[ I_1 \] and \[ I_2 \] are dependent variables.

In matrix form, the above equations can be rewritten as
\[
\begin{bmatrix}
I_1 \\
I_2
\end{bmatrix} =
\begin{bmatrix}
y_{11} & y_{12} \\
y_{21} & y_{22}
\end{bmatrix}
\begin{bmatrix}
V_1 \\
V_2
\end{bmatrix}
\]

The y-parameters can be found as follows:

\[y_{11} = \frac{I_1}{V_1} \bigg|_{V_2=0}\]

\[y_{12} = \frac{I_1}{V_2} \bigg|_{V_1=0}\]

\[y_{21} = \frac{I_2}{V_1} \bigg|_{V_2=0}\]

\[y_{22} = \frac{I_2}{V_2} \bigg|_{V_1=0}\]

The y-parameters are also called short-circuit admittance parameters. They are obtained as a ratio of current and voltage and the parameters are found by short-circuiting port 2 \((V_2 = 0)\) or port 1 \((V_1 = 0)\). The following two examples show how to obtain the y-parameters of simple circuits.

**2.1.1.3. H-parameters**

A two-port network can be represented using the h-parameters. The describing equations for the h-parameters are

\[V_1 = h_{11}I_1 + h_{12}V_2\]

\[I_2 = h_{21}I_1 + h_{22}V_2\]

Where,

\(I_1\) And \(V_2\) are independent variables and

\(V_1\) And \(I_2\) are dependent variables.

In matrix form, the above two equations become

\[
\begin{bmatrix}
V_1 \\
I_2
\end{bmatrix} =
\begin{bmatrix}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
I_1 \\
V_2
\end{bmatrix}
\]
The h-parameters can be found as follows:

\[ h_{11} = \left. \frac{V_1}{I_1} \right|_{I_2 = 0} \]

\[ h_{12} = \left. \frac{V_1}{V_2} \right|_{I_1 = 0} \]

\[ h_{21} = \left. \frac{I_2}{I_1} \right|_{V_2 = 0} \]

\[ h_{22} = \left. \frac{I_2}{V_2} \right|_{I_1 = 0} \]

The h-parameters are also called hybrid parameters since they contain both open-circuit parameters \((I_1 = 0)\) and short-circuit parameters \((V_2 = 0)\). The h-parameters of a bipolar junction transistor are determined in the following.

2.1.1.4. Transmission parameters

A two-port network can be described by transmission parameters. The describing equations are

\[ V_1 = a_{11}V_2 - a_{12}I_2 \]

\[ I_1 = a_{21}V_2 - a_{22}I_2 \]

Where,

\(V_2\) and \(I_2\) are independent variables and

\(V_1\) and \(I_1\) are dependent variables.

In matrix form, the above two equations can be rewritten as

\[
\begin{bmatrix}
V_1 \\
I_1
\end{bmatrix} =
\begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}
\begin{bmatrix}
V_2 \\
I_2
\end{bmatrix}
\]

The transmission parameters can be found as:
The transmission parameters express the primary (sending end) variables $V_1$ and $I_1$ in terms of the secondary (receiving end) variables $V_2$ and $I_2$. The negative of $I_2$ is used to allow the current to enter the load at the receiving end. Examples 7.5 and 7.6 show some techniques for obtaining the transmission parameters of impedance and admittance networks.

All these parameters are inter-related and once a given set has been established, they can be converted to another as desired for analysis. The network gain, input and output impedance and other characteristics will be easily calculated.

The usefulness of the different methods of description comes clearly into evidence when the problem of synthesizing or designing network filters, matching networks, wave shaping networks and many others.

In synthesizing a network for a specific application, it is often convenient to break down a complicated problem into several parts to ease analysis.

This is one of the most common methods of establishing the signal transfer behavior of electrical networks. As an example, parameter values for bipolar transistors are commonly quoted in terms of $h$ parameters which are typically measured with the emitter grounded also known as the common-emitter configuration. Semiconductor parameter analyzer is one of the many types of instruments employed to obtain the $h$ parameters of a particular transistor. It is from these $h$ parameters that the scientist is able to know the transistor specifications. A snapshot of a Semiconductor parameter analyzer displaying the $h$ values of a 2N3904 BJT is shown in fig. 2.2. below [1]
2.1.2. Setbacks

One of the major disadvantages of this method in network analysis is that active devices must first be replaced by their equivalent passive circuitry containing only three or four impedances. The validity of the equivalent networks may be restricted to small amplitudes or perhaps limited range of frequencies this circuit will also be just a linear approximation of a non-linear circuit.

2.2. Theory and background

There are two basic types of network analyzers are

a. Scalar network analyzer (SNA) to measures amplitude properties only
b. Vector network analyzer (VNA) that measures both amplitude and phase properties

SNAs are easily implemented but do not give adequate information concerning system response unless the purpose for measurement was purely concerned with amplitude effects on the signal therefore VNAs supersede SNAs as network analyzers.
2.2.1. Building blocks of a VNA

- **Test signal generator**
  The network analyzer requires a test signal in order to be capable of establishing network behavior. Most VNAs in the past did not have an in-built signal source but nowadays modern VNAs have at least one generator incorporated in the system. In fact high level VNAs have more than one signal generator to make them effectively characterize different categories of networks. A test signal should have varying but known parameters. This way it will subject the DUT into different input conditions and enable complete network characterization according to the DUT effects on the signal at various conditions.

- **Test set**
  It routes the test signal to the DUT and from the DUT to the receiver. At the same time, it takes the reference (uninterrupted input signal) through a delay channel. The reference signal is necessary for it can be compared to the signal getting out of the DUT in order to determine any changes to its original parameters. It does this by splitting off the test signal at the input stage. However in a case where a computer program is used to generate this input signal, the reference will be readily available at the program workspace. This will ease measurement as well as reduce any errors that would have encountered during propagation.

- **Receiver**
  This is a crucial component of the VNA. It takes in the reference as well as the output from the DUT. It’s from this point that signal phase and magnitude measurements are done. Modern VNAs have multi-port receiver that make it possible to compare the characteristics of different DUTs at the same time. This however is made much easier when a computer program is used because the unique outputs of the various DUTs in the research can be channeled, saved and processed conveniently, effectively and efficiently from the PC program workspace using one port through at different instance of time.

- **Display**
  An interface is inevitable. The measurements are useless if they cannot be visualized. This display point is needed both when performing signal conditions as well as when reading off the frequency response of the DUT graphically.

2.2.3. MATLAB application in measurement.

In this project most of the functions above have been confined to a computer software (MATLAB). In this experiment the software makes use of the functions built into MATLAB’S data acquisition toolbox. The following section seeks to elaborate on the concept used.

Audio (low frequency) signals can be stored in a computer in different formats. For instance, .wav and .mp3 formats. These files can be read by the sound-playing programs and decoded so that sound can be produced from a computer’s sound devices. So these signals are actually represented as vectors. The computer is able to perform all these functions through DAC and ADC conversions done by the computer’s sound card. Real world signals occur as continuous...
time continuous amplitude (analogue) signals and the only way that a computer can store and process these signals is by digitizing them (converting into binary format). The sound card therefore acts as an ADC and vice versa.

In a linear system, the frequency response function can be obtained by using sinusoidal inputs. However, some inputs - including random signals - can be expressed as a sum or integral of sinusoids. Finite length signals can be expressed as a sum of sinusoids, for example. These signals can then be used to compute the frequency response function, and once you have the frequency response function you can use Bode plots to determine a numerical transfer function.

Using more than one time response will let you check things like linearity of the system or whether the system varies in time, however getting time response data into a computer for analysis is easily automated.

There are ways in which we can get audio signals into MATLAB, including:

- Converting an external audio file into a MATLAB vector, e.g. using commands such as aures and wavread for sound for files already stored in the .wav format.
- For a sound signal already existing as a MATLAB vector, the use of the load command can get the signal into MATLAB workspace.
- We can as well create a vector in MATLAB using the normal mathematical functions or the in-build MATLAB functions to create special signals such as chirp signal.
- Also the wavrecord function is used in MATLAB to record sound from the audio input of your sound card.

It is also good to note that signal vectors available in the MATLAB workspace can be saved as a .wav file in the PC for further processing.

In this project MATLAB has been used exclusively in processing signals in order to be able to obtain DUT transfer characteristics. MATLAB is a very useful tool in order to understand the basic properties of discrete signals and digital filters. In MATLAB it is easy to make calculations, listen to signals and plot them in both the time and frequency domain.

2.2.4. The transfer function concept

Given a general linear two port network, N without any initial energy storage we assume a sinusoidal forcing and response functions, arbitrary taken to be voltages. Letting the input voltage be simply \( V_i(t) = A\cos(w_x t + \Theta) \) and the output can be described as \( V_o(t) = B\cos(w_x t + \Theta) \) where both the magnitudes and phases are the functions of angular frequency \( w_x \). In phasor form, the forcing and response functions can be written as

\[
V_i = A e^{j\Theta} \quad \text{And} \quad V_o = B e^{j\phi}
\]

The ratio of the phasor response to the forcing function is a complex number that is a function of \( w_x \).
\[ V_0/V_i = G(w_x) = (B/A) e^{j(\phi - \Theta)} \]

Where \( B/A \) is the amplitude of \( G \) and \( (\phi - \Theta) \) is its phase angle. This transfer function \( G(w_x) \) could be obtained experimentally by varying \( w_x \) over a large range of values and measuring the amplitude \( B/A \) and phase \( (\phi - \Theta) \) for each value of \( w_x \). Plotting each of these parameters as a function of frequency, the resultant pair of curves would completely describe the transfer function.

\[ V_i(t) = A \cos(w_x t + \phi) \]
\[ V_0(t) = B \cos(w_x + \phi) \]

**Figure 2.3.**

However, signals present in a PC can only exist in digital as opposed to continuous time, continuous amplitude format. The problem of determining the output of a physical system in terms of the input and the impulse response was solved by using the convolution integral and initially working with time domain. The input, the output and the impulse response were all in time domain. Subsequently, a more convenient way to perform such operations was in frequency domain as the Laplace transform of the convolution of two functions is simply the product of each function in the frequency domain. Along the same lines, we find the same is true when working with Fourier transforms. Z-transform method is an operational method that is powerful when working with discrete-time systems.

The frequency response,

\[ H(e^{j\omega T}) \]

For, \( z = e^{j\omega T} \)

Where \( \omega \) continuous radian frequency and \( T \) is the sampling period, can be obtained in the following ways:

1. Using direct division of the input and output signals at the DUT [3]

\[ H(z) = B(z)/A(z) \]

for,

\( H(z) \) : transfer function
\( B(z) \): output

\( A(z) \): Input, And

\[ z = e^{j\omega T} \]

\[
H(e^{j\omega_k T}) = \frac{B(e^{j\omega_k T})}{A(e^{j\omega_k T})}, \quad e^{j\omega_k T} \triangleq e^{j2\pi k/N_s}, \quad k = 0, 1, 2, \ldots, N_s - 1,
\]

Where \( N_s \) is the desired number of spectral samples around the unit circle in the \( z \) plane. This is the same thing as the sampled DTFT of: \( a \) divided by the sampled DTFT of: \( b \)

\[
H(e^{j\omega_k T}) = \frac{\text{DTFT}_{\omega_k T}(b)}{\text{DTFT}_{\omega_k T}(a)}
\]

The uniformly sampled DTFT has its own name: the discrete Fourier transform (DFT). Hence, we can write

\[
H(e^{j\omega_k T}) = \frac{\text{DFT}_{\omega_k T}(b)}{\text{DFT}_{\omega_k T}(a)}, \quad k = 0, 1, 2, \ldots, N_s - 1
\]

\[
\text{DFT}_{\omega_k T}(x) \triangleq \sum_{n=0}^{N_s-1} x(n)e^{-j\omega_k nT}
\]

Where,

\[ \omega_k \triangleq 2\pi f_s k / N_s \]

And the sampling frequency is:
\[ f_s = 1/T \]

This (DFT) can easily be computed by a computer algorithm tagged FFT (Fast Fourier transform). If \( x(t) \) is a periodic function with period \( T \), then \( x(t) \) may be written as infinite Fourier series.

In order to perform the Fourier transform above by digital computer, it is convenient to sample the arbitrary function \( x(t) \) at a series of regularly spaced times to form a series of discrete values which we write as \( \{x_m\} \), where \( x_m = x(m\Delta) \) and \( m = 0, 1, \ldots, N - 1 \). It can then be shown that if we imagine our set of \( N \) samples is actually a periodic repetition of these \( N \) samples, we can find an invertible discrete transform which approximates the original continuous signal.

II. Using the power spectral density [3] [4]

The transfer behavior can also be computed by first obtaining the power spectral density of the two signals. The transfer function is then obtained as the quotients of the cross power spectral density (\( P_{yx} \)) of \( x \) and \( y \) and the PSD, \( (P_{xx}) \) of \( x \).

\[ T_{xy}(f) = \frac{P_{yx}(f)}{P_{xx}(f)} \]

\([Pxx,w] = \text{pwelch}(x)\) estimates the power spectral density \( P_{xx} \) of the input signal vector \( x \) using Welch's method. Welch's method splits the data into overlapping segments, computes modified periodograms of the overlapping segments, and averages the resulting periodograms to produce the power spectral density estimate.

\( \text{pwelch} \) function calculates the power spectral density using Welch's method.

Generally one of the ways to estimate the PSD of a process is to compute the discrete-time Fourier transform of the samples of this process using FFT algorithm and appropriately scale the magnitude squared of the result. This estimate is known as the periodogram.

The periodogram estimate of the PSD of a length-L signal \( x_L[n] \) is

\[ P_{xx}(f) = \frac{1}{LF_s} \left| \sum_{n=0}^{L-1} x_L(n) e^{-j2\pi fn/F_s} \right|^2 \]

Where \( F_s \) is the sampling frequency.

Practically, the actual computation of \( P_{xx}(f) \) employs an FFT and can only be performed at a finite number of frequency points. Implementations of the periodogram method mostly compute the \( N \)-point PSD estimate at the frequencies
\[ f_k = \frac{kF_s}{N} \quad k = 0, 1, \ldots, N - 1 \]

There are cases where the computation of the periodogram using an FFT algorithm is more efficient, that is, when the number of frequencies is a power of two. Therefore, it is not uncommon to pad the input signal with zeros to extend its length to a power of two.

\[ n(H) \triangleq \frac{1}{N_s} \sum_{k=0}^{N_s-1} H(e^{j\omega_k T}) e^{j\omega_k nT} \]

We can thereafter specify sampling frequencies for our obtained signals in order to able to process them according to our needs. However failure to do this will lead to the MATLAB program using the default values which most of the time may not give us the correct results.

2.2.5. Signal conditioner

It is also important to note that depending on the computer being used, the sound card will have limitations on the sampling frequencies that they can support. A general knowledge is that most sound cards do not support sampling frequencies of less than 5-10kHz or greater than 44.1kHz. They also have amplitude limit [-1, 1] in that signals whose amplitude deviate with a great extend from this value will be attenuated hence getting distorted or worse still may blow off the sound card hence creating unnecessary damages to the PC. Basing on the DUT and the sound card specifications of the PC under use, a signal preconditioning circuit is a must.

A signal conditioner prepares a signal for use by another component. The input to a signal conditioner is usually the output from a sensor (or primary element). The operations performed by a signal conditioner include isolation, impedance conversion, noise reduction, amplification linearization and conversion.

As a way to condition the input swept frequency wave, we introduce an attenuator in order to ensure that the wave is not pre-amplified before getting to the DUT. This way, we will make sure that any changes in amplitude will be purely attributed to the effects of the DUT.

However, other errors maybe introduced in the process which include new frequency components that may necessitate the use of filters.

The effects above necessitate the introduction of a low pass filter before introducing the signal through the DUT for analysis in order to ensure that the frequency components remain within the scope of the project, that is, below 20 kHz.
Once the test set has been assembled, the DUT is included in the VNA and signal conditioning is performed to protect the PC soundcard as well as ensure minimal or zero signal distortion. The immediate signal just before it gets to the DUT is taken into the MATLAB workspace. The DUT output is also taken into the MATLAB workspace as well. It is from this point that the transfer function is computed using the `tfestimate` MATLAB function: $T_{xy} = \text{tfestimate}(x,y)$ or by direct division of the Fourier transforms of the input and outputs of the DUT. $T_{xy} = \text{tfestimate}(x,y)$ will finds an estimate of transfer function, $T_{xy}$ given input signal vector $x$ and output signal vector $y$. $x$ and $y$ must are vectors and must be of the same length. The relationship between the input $x$ and output $y$ is modeled by the linear, time-invariant transfer function $T_{xy}$. 
CHAPTER 3: DESIGN AND IMPLEMENTATION

The design and fabrication process of this project basing on the theory and background studies and the reasons on the choice of methods and procedures used to address the problem statement are covered in this chapter.

3.1. Design consideration

In regard to the design problem in hand, it is required that the changes on known characteristics of a signal after passing through a DUT be studied in order to obtain the transfer behavior of the DUT. Therefore a signal generator is an essential part of the VNA. To be able to study the frequency response of the DUT, it is necessary to have a reference signal whose frequency changes with time and therefore a swept frequency sine wave has been used as a trigger signal. As part of the measurement requirements a signal display and analyzer are inevitable as the changes to the reference signal need to be obtained and analyzed. All these functions can be one conveniently using MATLAB application software. Signal generation and analysis is done from MATLAB work space. The sound card acts as the DAC and ADC and interfaces the digital and analogues forms of the signal as they occur in generation and analysis as well as when they are input and obtained from the DUT.

It is in this regard that a simple signal conditioner (passive) is required in order to protect the PC sound card in cases where the signal to be analyzed is way over and above the signal input limits of the sound card being used.

3.2. System building blocks

The block diagram of the design is as shown below:

Figure 3.1.
3.2.1. Swept frequency sine wave generator

To determine the frequency response a swept sine wave is fed into the DUT and the unit’s output is measured. The presence of a data-acquisition system made it possible and easy to program the system’s analog output into generating a swept sine wave.

Despite the programming languages having an option pre-established swept sine function where a start frequency, stop frequency, and a sweep time is entered to effectively characterize the system the swept sine wave had to be generated point by point, and have the data send to the data-acquisition system’s driver in order to load it into the DUT.

To produce the swept sine wave signal, the code is written to generate an array of numbers which represent the swept sine’s amplitude at each sampling point of the waveform. As compared to when programming a single frequency sine wave, the frequency change from one point to the next accounted for using the MATLAB loop commands or the equation stated below. This way, for any addition of a successive point, the array will cover a bigger portion of a cycle.

The form of the equation to generates the array is

\[ y(i) = A \sin \left( \frac{\omega_i^2}{2} + b_i \right) \]

Where:

\( y \), is the varying signal

\( i \), an integer

\( A \), The peak voltage amplitude and

\( a \) And \( b \) are variables:

\[ a = \frac{2\pi(f_2 - f_1)}{n} \]

\[ b = 2\pi f_1 \]

Where:

\( n = \) number of samples

\( f_1 = \) normalized initial (lowest) frequency
\[ f_2 = \text{normalized highest frequency} \]

\[ f_1 \text{ and } f_2 \] are variables expressed in units of cycles per sample. The value of \( f_1 \) is obtained by dividing the lowest frequency of the sine wave in hertz by the sample rate and the \( f_2 \) is similarly obtained by dividing the highest frequency component of the sine wave by the sample rate.

### 3.2.2. Signal conditioner

To prepare the signal for analysis and at the same time protecting the PC soundcard, a variable voltage divider has been utilized before the signal gets into the DUT to make sure that the best test signal conditions are applied to the network and after the signal gets out of the DUT to make sure that the signal at the output of the DUT does not destroy the PC sound card in a case where the DUT causes amplification on the input signal. Moreover, switching diodes have been utilized in order to make sure that neither the input to the DUT or DAQ system exceeds the 0.7v voltage level which is a forward bias of the diode and at the same time allow the system to operate effectively at medium frequencies.

The schematic diagram of the conditioner is as shown below:

![Schematic Diagram](image)

**Figure 3.2**

Two 10k potentiometers, two 10k resistors and four IN1158 were used.

The bread board implementation is as shown below:

![Bread Board Implementation](image)

**Figure 3.2**

To get exact results, appropriate computations are made to establish and compensate for the changes occurring during signal conditioning. This necessitates a digital multimedia to measure
the potentiometer resistance in order to calculate the voltage division using the formula:

\[ V_i = \frac{R_2}{R_1+R_2} V_0 \]

3.2.3. Signal analyzer

The signal analyzer works to estimate the transfer function from the time series data. In this project the estimate of the transfer function between the variables in the experimental data set was obtained from MATLAB directly by having the command `tfestimate` compute a transfer function estimate (That uses the PSD method):

\[ H_{est}(j\omega) = \frac{V_{out}(j\omega)}{V_{in}(j\omega)} \]

The form of the specific command was:

```matlab
[EstH, EstF] = tfestimate(Vin, Vout, [], [], [], Fs)
EstMag = abs(EstH);
EstPhase = angle(EstH);
EstOmega = EstF*2*pi;
```

A magnitude and a phase plot was then generated using MATLAB commands for the experimentally determined transfer function. MATLAB's `tfestimate` being expected to produce a numerical estimate of the phase and magnitude of a transfer function for a given input signal and an output signal. This plot gives the possibility to visualize the transfer behavior of the DUT.

3.3. Testing the VNA

To investigate the performance of the designed a loudspeaker drive system with separate high frequency amplifier (tweeter system) and a low frequency amplifier unit(bass system). The experiment was to see that the VNA could show that the two units were high and low pass filters respectively as well as show the phase shift degree if any.
CHAPTER 4: RESULTS AND DISCUSSION

4.1. Results

The figures (b to e) below show the output of the signal conditioner to the input signal shown in figure (a) varied to obtain suitable magnitude on the input signal without exceeding the 0.7v voltage amplitude.

![Output waveforms](image1.png)

![Output waveforms](image2.png)

![Output waveforms](image3.png)

![Output waveforms](image4.png)

![Output waveforms](image5.png)

*Figure 4.1 (a-e)*
Figure 4.2 below shows the bode plot obtained by outputting out test signal form the audio jack and taking it back through the microphone into the MATLAB work space for analysis. It can be noted that there was no significant change in magnitude and phase over the frequency ranges used except for the extreme low frequency end which can be attributed to the poor sampling by the PC sound card at such frequencies.

![Bode Plot](image)

**Figure 4.2**

By altering the magnitude of the signal for frequencies greater than 3000Hz the frequency response plot obtained is shown in the figure 4.3 (a) and (b) below. Figure a is obtained by computing the transfer function using the power spectrum analysis of output and reference signal while figure b is obtained by computing the FFT transforms of the two signals and performing direct division both of which give the estimate of the transfer characteristics between the two signals. In this project computation of transfer function has been done entirely by performing power spectrum analysis. A process that has been simplified by MATLAB algorithms and has been addressed using the command:

```matlab
[EstH, EstF] = tfestimate(Vin, Vout, [], [], [], Fs)
EstMag = abs(EstH);
EstPhase = angle(EstH);
EstOmega = EstF*2*pi;
```
Figure 4.3(a)

Figure 4.3(b)
Figure 4.4 below shows the VNA being used to visualize the electrical characteristics of our DUT (a loudspeaker system-motherboard).

Figure 4.4 (a)

Figure 4.4(b)
The figure 4.5 (a) and (b) below shows the frequency response of the tweeter and bass system respectively.

**Figure 4.5 (a)**

**Figure 4.5 (b)**
4.2. Discussion
Despite the measurement limits dictated by the lowest possible signal frequency to be sampled by the soundcard, from the experiment, the characteristics of the loud speaker driver have been obtained using the designed VNA. It is seen from the magnitude vs. frequency plot that the tweeters amplify only higher frequencies (greater than 300Hz) and attenuates lower frequencies while the base amplifier attenuates frequencies to the lowest end but attenuates those greater than 500Hz. It can also be seen from the phase vs. frequency that the amplifier has no phase shift when the output is referred to the input. However the plots are not as clear as they ought to be. This can be attributed to poor distribution of frequency components to the test signal, the noise introduced by the amplifying system as well as poor signal sampling by the soundcard especially when transmitting low frequency signals.
CHAPTER 5: CONCLUSION

5.1. Recommendation for further work.

To record accurate results I propose that the signal conditioner should be precise. Any attenuation or amplification factors subjected to the DUT output signal should be pre-set so that exact deviation of amplitude vs. frequency plot from the correct point should be known with ease.

There is a lot of noise involved when computing the system transfer function. It is therefore necessary to find a way to obtain consumable plots. One way is tabulating the transfer function values and statistically performing curve fitting on them. The estimated equation can then be plotted for any frequencies between 0 and 20 KHz.

Some parts of this project had to be rescoped due to time constraints. To improve on this in the future, the person involved in any project should have adequate time for research and development so that no stone is left unturned.

5.2. Difficulties encountered

As predicted during in the theory, obtaining accurate DUT characteristics using this method for very low frequencies in the range of 0 to 3 KHz proved difficult despite efforts to cover all frequency ranges in the test signal. Also, establishing the transfer function of two different signals brought about the need to match the matrix size of both the sent and received signals as referred to the MATLAB workspace which was challenging considering the different sampling frequencies. Besides, computing the transfer function directly by comparing the input and output signals never gave clear DUT characteristics because the amplitude of the test signal was rather small. This is because the data acquisition system limits signals to a magnitude of 1V to protect the PC soundcard.

Time constraint limited the research scope of this project. There was no enough time to fully advance most of the concepts used in the design.

5.3. Conclusion

The main objective of this project, to design and implement a vector network analyzer for low frequency applications was met despite the inaccuracies due to the rough estimation of transfer characteristics. Demonstrations on how to develop a fully functional VNA for low frequency applications given only a PC have been illustrated from the design stage through implementation to the point where the VNA has been tested to prove its functionality.
References


   http://www.facstaff.bucknell.edu/mastascu/ectrolhtml/Model/Model5.html


APPENDIX

Creating a swept-frequency sinusoidal signal

Fs=44100;
t=0:1/Fs:2;
num=length(t);
y=zeros(1,num);

%y=y(:,);
%size(y)
%break

f1=10;
f2=50;
f3=300;
f4=800;
f5=1000;
f6=3000;
f7=8000;
f8=12000;
f9=16000;
f10=18000;

%T=1/fs;
%y=[];
for a=1:num;
    if a<=num/10;
        y(a)=0.7*sin(2*pi*f1*(t(a)+2.9535));
    else if a<=2*num/10;
        y(a)=0.7*sin(2*pi*f2*(t(a)+2.9535));
    else if a<=3*num/10;
        y(a)=0.7*sin(2*pi*f3*(t(a)+2.9535));
    else if a<=4*num/10;
        y(a)=0.7*sin(2*pi*f4*(t(a)+2.9535));
    else if a<=5*num/10;
        y(a)=0.7*sin(2*pi*f5*(t(a)+2.9535));
    else if a<=6*num/10;
        y(a)=0.7*sin(2*pi*f6*(t(a)+2.9535));
    else if a<=7*num/10;
        y(a)=0.7*sin(2*pi*f7*(t(a)+2.9535));
    else if a<=8*num/10;
        y(a)=0.7*sin(2*pi*f8*(t(a)+2.9535));
    else if a<=9*num/10;
        y(a)=0.7*sin(2*pi*f9*(t(a)+2.9535));
    else if a<=num;
        y(a)=0.7*sin(2*pi*f10*(t(a)+2.9535));
    end
end
end
end
end
end
end
end
end
end
end

end
wavwrite(y,'C:\Users\MASSIR\Documents\MATLAB\Good2');
%wavplay(y,Fs)
size(y)
figure
plot(t,y)
break
display(t)
l=length(y);
%l=8000;
N=2^nextpow2(l);

Y=fft(y,N)/l;
Y=Y(:);
display(Y)
break
size(Y)
%break
magnitude=2*abs(Y(1:N/2+1));
display(magnitude)

phase=unwrap(angle(Y(1:N/2+1)));
F=(Fs/2*linspace(0,1,N/2+1));
figure
plot(F,magnitude);
title('Magnitude spectrum');
xlabel('F[Hz]');
ylabel('Magnitude');
figure
plot(F,phase);
title('Phase spectrum');
xlabel('F[Hz]');
ylabel('Phase [degrees]');

Sending and receiving data to and from the DUT through the soundcard
if (~isempty(daqfind))
    stop(daqfind)
end
ao = analogoutput('winsound', 0);
addchannel(ao,1);
%load handel
[y, Fs, nbits] = wavread('Good.wav');
set(ao, 'StandardSampleRates', 'Off')
set(ao, 'SampleRate', Fs);
Fs = 440;
data = [y y];
putdata(ao, data);
startindex = 1;
increment = 500;
start(ao);
while isrunning(ao)
    while (ao.SamplesOutput < startindex + increment -1), end
    try
        x = ao.SamplesOutput;
        subplot(3,1,1);
        plot(y(x:x+increment-1));
        xlabel('time(s)');
        ylabel('signal(v)');
        title('Data sent using audiojack for 8 seconds');
        grid on;
        set(gca, 'YLim', [-0.8 0.8], 'XLim', [1 increment])
        drawnow;
    end
end
N = 64;
X = abs(fft(data,N));
%X = fftshift(X);
F = [-N/2:N/2-1];
P = unwrap(angle(data))
subplot(3,1,2);
plot(F, X),
xlabel('frequency(Hz)')
ylabel('Magnitude(dB)')
title('Magnitude plot')
grid on
subplot(3,1,3);
F = (0:length(data)-1)/length(data)*100;
plot(F, P),
delete(ao);
if (~isempty(daqfind))
    stop(daqfind)
end

duration = 10;
ai = analoginput('winsound');
addchannel(ai, 1);
ai
sampleRate = get(ai, 'SampleRate')
get(ai, 'SamplesPerTrigger')
requiredSamples = floor(sampleRate * duration);
set(ai, 'SamplesPerTrigger', requiredSamples);
waitTime = duration * 1.1 + 0.5
start(ai)
tic
wait(ai, waitTime);
toc
[data, time] = getdata(ai);
figure;
subplot(3,1,1);
plot(time, data);
xlabel('Time (s)');
ylabel('Signal (Volts)');
title('Data Acquired using Microphone for 10 seconds');
grid on;

y = fft(data); % Compute DFT of x
m = abs(data); % Magnitude
p = unwrap(angle(data)); % Phase
f = (0:length(y)-1)*100/length(y); % Frequency vector
subplot(3,1,2)
plot(f,m)
xlabel('Frequency (Hz)');
ylabel('Magnitude (DB)');
title('Magnitude plot')
grid on;
subplot(3,1,3)
plot(f,p)
xlabel('Frequency (arb.)')
ylabel('Phase (rad)')

Plotting the DUT transfer function
if (~isempty(daqfind))
    stop(daqfind)
end
ao = analogoutput('winsound', 0);
addchannel(ao, 1);
%load handel
[y, Fs, nbits] = wavread('Good.wav');
set(ao, 'StandardSampleRates', 'Off')
set(ao, 'SampleRate', Fs);
%Fs=440;
data = y;
%wavplay data
putdata(ao, data);
startindex = 1;
increment = 500;
start(ao);
%break
%duration = 12.500125;
a = analoginput('winsound');
addchannel(a, 1);
a;
sampleRate = get(a, 'SampleRate');
get(a, 'SamplesPerTrigger');
requiredSamples = floor(sampleRate * duration);
set(a, 'SamplesPerTrigger', requiredSamples);
waitTime = duration * 1.1 + 0.5;
start(a);
tic;
wait(a, waitTime);
toc;
[data, time] = getdata(a);
x = data;
wavwrite(x, 'C:\Users\MASSIR\Documents\MATLAB\output');
x = x(:);
[y, Fs, nb] = wavread('Good2.wav');
Y = fft(y);
X = fft(x);
size(x);
[EstH, EstF] = tfestimate(y, x, [], [], [], Fs);
EstMag = abs(EstH);
EstPhase1 = (angle(EstH));
EstPhase = fix(EstPhase1/100)*100;
EstOmega = EstF*2*pi;
figure(1); clf
% Magnitude plot on top
subplot(2, 1, 1);
semilogx(EstOmega, 20*log10(EstMag), 'k-');
xlabel('\omega, rad/s')
ylabel('|H|, dB')
grid on
% Phase plot on bottom
subplot(2, 1, 2);
semilogx(EstOmega, EstPhase, 'k-');
xlabel('\omega, rad/s')
ylabel('\angle H, rad')
grid on