SREE VIDYANIKETHAN ENGINEERING COLLE

(An autonomous institution affiliated to JNTUA, Anantapuramu)

SreeSainath Nagar, Tirupati – 517 102.

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Lab Manual

SIGNALS AND NETWORKS LAB (16BT30232)

(II B. Tech., I-Semester, EEE)

Prepared by

Dr. S Farook, Associate Professor

Mr. B. Subba Reddy, Assistant Professor

Department of EEE

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

Sree Sainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Laboratory manual

Signals and Networks Lab

(16BT30232)

II B. Tech. I Semester

Electrical and Electronics Engineering

Department of Electrical and Electronics Engineering

Sree Vidyanikethan Engineering College

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

Sree Sainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

16BT30232: Signals and Networks Lab

Contents

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

Sree Sainath Nagar, Tirupati – 517 102

VISION

To be one of the Nation's premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

- To foster intellectual curiosity, pursuit and dissemination of knowledge.
- To explore students' potential through academic freedom and integrity.
- To promote technical mastery and nurture skilled professionals to face competition in ever increasing complex world.

QUALITY POLICY

Sree Vidyanikethan Engineering College strives to establish a system of Quality Assurance to continuously address, monitor and evaluate the quality of education offered to students, thus promoting effective teaching processes for the benefit of students and making the College a Centre of Excellence for Engineering and Technological studies.

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

Sree Sainath Nagar, Tirupati – 517 102

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

VISION

To become the Nation's premiere centre of excellence in electrical engineering through teaching, training, research and innovation to create competent engineering professionals with values and ethics.

MISSION

- Department of Electrical Engineering strives to create human resources in Electrical Engineering to contribute to the nation development and improve the quality of life.
- Imparting Knowledge through implementing modern curriculum, academic flexibility and learner centric teaching methods in Electrical Engineering
- Inspiring students for aptitude to research and innovation by exposing them to industry and societal needs to creating solutions for contemporary problems
- Honing technical and soft skills for enhanced learning outcomes and employability of students with diverse background through comprehensive training methodologies
- Inculcate values and ethics among students for a holistic engineering professional practice.

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Signals and Networks Lab

(16BT30232)

Program Educational Objectives

Within few years of graduation, graduates will:

PEO 1	have enrolled in academic program in the disciplines of electrical engineering and multidisciplinary areas.
PEO 2	become entrepreneurs or be employed as productive and valued engineers in reputed industries.
PEO 3	engage in lifelong learning, career enhancement and adopt to changing professional and societal needs.

Program Outcomes - B.Tech – EEE

On successful completion of the program, engineering graduates will be able to:

PO 1	**Engineering knowledge**	Apply the knowledge of mathematics, science, engineering fundamentals, and concepts of engineering to the solution of complex engineering problems.
PO 2	**Problem analysis**	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO 3	**Design/development of solutions**	Apply the knowledge of mathematics, science, engineering fundamentals, and concepts of engineering to the solution of complex engineering problems.
PO 4	**Conduct investigations of complex problems**	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5	**Modern tool usage**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO 6	**The engineer and society**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	**Environment and sustainability**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
PO 8	**Ethics**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work**	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**Communication**	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	**Project management and finance**	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**Life-long learning**	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes - B.Tech – EEE

On successful completion of the program, engineering graduates will:

PSO 1	Demonstrate knowledge of Electrical and Electronic circuits, Electrical Machines, Power Systems, Control Systems, and Power Electronics for solving problems in electrical and electronics engineering.
PSO 2	Analyze, design, test and maintain electrical systems to meet the specific needs of the Industry and society.
PSO 3	Conduct investigations to address complex engineering problems in the areas of Electrical Machines, Power Systems, Control Systems and Power Electronics.
PSO 4	Apply appropriate techniques, resources and modern tools to provide solutions for problems related to electrical and electronics engineering.

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Signals and Networks Lab

(16BT30232)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	*50*	*100*	*0*	*0*	*3*	*2*

Course description:

Experimentation on Signals and systems; Transient analysis; Two-port network parameters and passive filters.

Course outcomes:

On successful completion of the course, students will be able to:

- CO1. demonstrate knowledge on signals, transients, two port networks & filters and their experimental implementation.
- CO2. analyze and relate the experimental observations & measurements for validation.
- CO3. design a suitable experimental/simulation procedure for practical investigations on signals, systems and networks.
- CO4. demonstrate skills in evaluating various parameters and interpret the observations to provide feasible solutions.
- CO5. select appropriate technique for experimental investigations, analysis and interpretation of signals and networks.
- CO6. apply the conceptual knowledge of signals, transients, filters and twoport network models in relevance to industry and society.
- CO7. commit to ethical principles and standards while exercising the practical investigations on signals and networks.
- CO8. work individually or in a group in the field of signals and networks.
- CO9. communicate effectively in verbal and written form in signals and networks domain.

Signals and Networks Lab (16BT30232)
CO, PO and PSOs mapping

Course Code Course Title	CO's	Course Outcomes	PROGRAM OUTCOMES												PROGRAM SPECIFIC OUTCOMES			
			PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
Signals and Networks Lab (16BT30232)	**CO1**	demonstrate knowledge on signals, transients, two port networks & filters and their experimental implementation.	**H**												**H**			
	CO2	analyze and relate the experimental observations & measurements for validation.	**M**	**H**												**H**		
	CO3	design a suitable experimental/simulation procedure for practical investigations on signals, systems and networks.	**L**	**M**	**H**											**H**		
	CO4	demonstrate skills in evaluating various parameters and interpret the observations to provide feasible solutions.	**M**	**M**	**M**	**H**											**H**	
	CO5	select appropriate technique for experimental investigations, analysis and interpretation of signals and networks.	**L**	**M**	**M**		**H**											**H**
	CO6	apply the conceptual knowledge of signals, transients, filters and twoport network models in relevance to industry and society.			**M**	**M**		**H**										
	CO7	commit to ethical principles and standards while exercising the practical investigations on signals and networks.			**M**	**M**			**H**									
	CO8	work individually or in a group in the field of signals and networks.				**M**		**M**			**H**							
	CO9	communicate effectively in verbal and written form in signals and networks domain						**L**				**H**						

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Signals and Networks Lab

(16BT30232)

Int. Marks	Ext. Marks	Total Marks		L	T	P	C
50	*50*	*100*		*0*	*0*	*3*	*2*

CO ASSESMENT

<table>
<tr><td rowspan="3">Signals and Networks Lab (16BT30232)</td><td>External 50</td><td colspan="2">Semester-end Lab Examination for 3 hours duration (External evaluation)</td><td>50 marks are allotted for laboratory/drawing examination during semester-end.</td></tr>
<tr><td rowspan="2">Internal 50</td><td>30</td><td>Day-to-Day evaluation for Performance in laboratory experiments and Record. (Internal evaluation).</td><td rowspan="2">Two laboratory examinations, which include Day-to-Day evaluation and Practical test, each for 50 marks are to be evaluated. For a total of 50 marks 75% of better one of the two and 25% of the other one are added and finalized.
Laboratory examination-I: Shall be conducted just before I mid-term examinations.
Laboratory examination-II: Shall be conducted just before II midterm examinations.</td></tr>
<tr><td>20</td><td>Practical test (Internal evaluation).</td></tr>
</table>

SREE VIDYANIKETHANENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

RUBRICS FOR SIGNALS AND NETWORKS LAB (16BT30232)

Course Outcome On successful completion of the course, students will be able to		Poor (0 - 1) Mark	Average (2 - 3) Marks	Excellent (4-5) Marks
CO1	**Demonstrate knowledge on** • **Various types of signals, systems and their properties.** • **Transient behaviour of various circuit components** • **Various types of filters, their realization and characteristics.**	Unable to demonstrate knowledge on signal, systems, networks and filters for various investigations.	Able to demonstrate knowledge on signal, systems, networks and filters for various investigations.	Able to demonstrate knowledge on signal, systems, networks and filters for various investigations and perform in-depth analysis.
CO2	**analyze the performance of systems for various signals, Two port networks and Filters.**	Unable to analyse the performance of signal, systems, networks and filters.	Able to analyse the performance of signal, systems, networks and filters up to some extent.	Able to analyse the performance of signal, systems, networks and filters completely and provide valid justifications.
CO3	**design suitable filter circuits, Two port networks to meet the design specifications.**	Unable to design the experimental circuit.	Able to design some parameters of the circuit.	Able to design the experimental circuit based on the given design speciation.
CO4	**interpret and synthesize the data obtained from experimentation on various systems and networks to provide valid conclusions.**	Unable to interpret and synthesize the data obtained from experimentation on signals & systems, networks and filters.	Able to interpret and synthesize the data obtained from experimentation on signals & systems, networks and filters to some extent.	Able to interpret and synthesize the data obtained from experimentation on signals & systems, networks and filters.
CO5	**select and apply appropriate technique for testing and validation of different systems and networks for domestic and industrial applications.**	Unable to select and apply appropriate tools, meters and methods for valid investigations.	Able to select and apply appropriate tools, meters and methods for valid investigations. to some extent.	Able to select and apply appropriate tools, meters and methods for valid investigations.

CO6	**apply the conceptual knowledge of signals, transients, filters and two port network models in relevance to industry and society.**	Unable to apply the conceptual knowledge of signals & systems, Networks and filters in relevance to industry and society.	Able to apply the conceptual knowledge of signals & systems, Networks and filters to industry and society to some extent.	Able to apply the conceptual knowledge of signals & systems, Networks and filters to industry and society.
CO7	**commit to ethical principles and standards while exercising the practical investigations on signals and networks.**	Unable to follow ethical principles and standards.	Able to follow ethical principles and standards to some extent.	Able to follow ethical principles and standards.
CO8	**work individually or in a group in the field of signals and networks.**	Unable to work individually or in a group	Occasionally work individually or in a group	Able to work and execute the problem individually as well as in a group.
CO9	**communicate effectively in verbal and written form in signals and networks domain.**	Unable to communicate effectively in verbal and written form on the investigations, analysis and interpretations.	Able to communicate effectively in verbal and written form on the investigations, analysis and interpretations to some extent.	Able to communicate effectively in verbal and written form on the investigations, analysis and interpretations.

Faculty In-charge

Chairman, BoS / HOD, EEE

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Year and Semester	**II B.Tech. - I Semester**	Roll No :	
Name of the Laboratory	**Signals and Networks**	Course Code	**16BT30232**

Day-to-Day Evaluation: 30 Marks

S. No	Date	Experiment Name	CO1 Knowledge 5 M	CO2 Analysis 5 M	CO3 Programming/Design 5 M	CO4 Evaluating Skills 5 M	CO5 Modern tools 5 M	CO6 Societal needs 5 M	CO7 Ethics and standards 5 M	CO8 Team work 5 M	CO9 Report writing 5 M	TOTAL (45 M) 45M	TOTAL (30 M) 30M	Signature of the Faculty
1		Introduction to MATLAB basics and Programming -I												
2		Introduction to MATLAB basics and Programming -II												
3		Generation of continuous time signals.												
4		Basic operations on the signals.												
5		Systems and their properties.												
6		Convolution of signals.												
7		Transformation of signals into time and frequency domains.												
8		Transient response of RL circuit and applications.												
9		Transient response of RC circuit and applications.												
10		Transient response of RLC circuit and applications.												
11		Determination of Open circuit and Short circuit parameters in isolated and interconnected networks.												
12		Determination of ABCD and Hybrid parameters in isolated and interconnected networks.												
13		Design, analysis and application of Low pass and High pass filters.												
14		Design, analysis and application of Band Pass and stop filters.												

Faculty in-charge

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Signals and Networks Lab

(16BT30232)

INTERNAL EXAM EVALUATION: (20 Marks)

Each student has to conduct a suitable experiment for the task assigned to him/her individually similar to end semester external examination. The performance of the student will be evaluated as follows:

COs	Assessment Parameter	Marks (20 M)
CO1	demonstrate knowledge on signals, transients, two port networks & filters and their experimental implementation.	**3 M**
CO2	analyze and relate the experimental observations & measurements for validation.	**2 M**
CO3	design a suitable experimental/simulation procedure for practical investigations on signals, systems and networks.	**3 M**
CO4	demonstrate skills in evaluating various parameters and interpret the observations to provide feasible solutions.	**2 M**
CO5	select appropriate technique for experimental investigations, analysis and interpretation of signals and networks.	**2 M**
CO6	apply the conceptual knowledge of signals, transients, filters and twoport network models in relevance to industry and society.	**2 M**
CO7	commit to ethical principles and standards while exercising the practical investigations on signals and networks.	**2 M**
CO8	work individually or in a group in the field of signals and networks.	**2 M**
CO9	communicate effectively in verbal and written form in signals and networks domain.	**2 M**

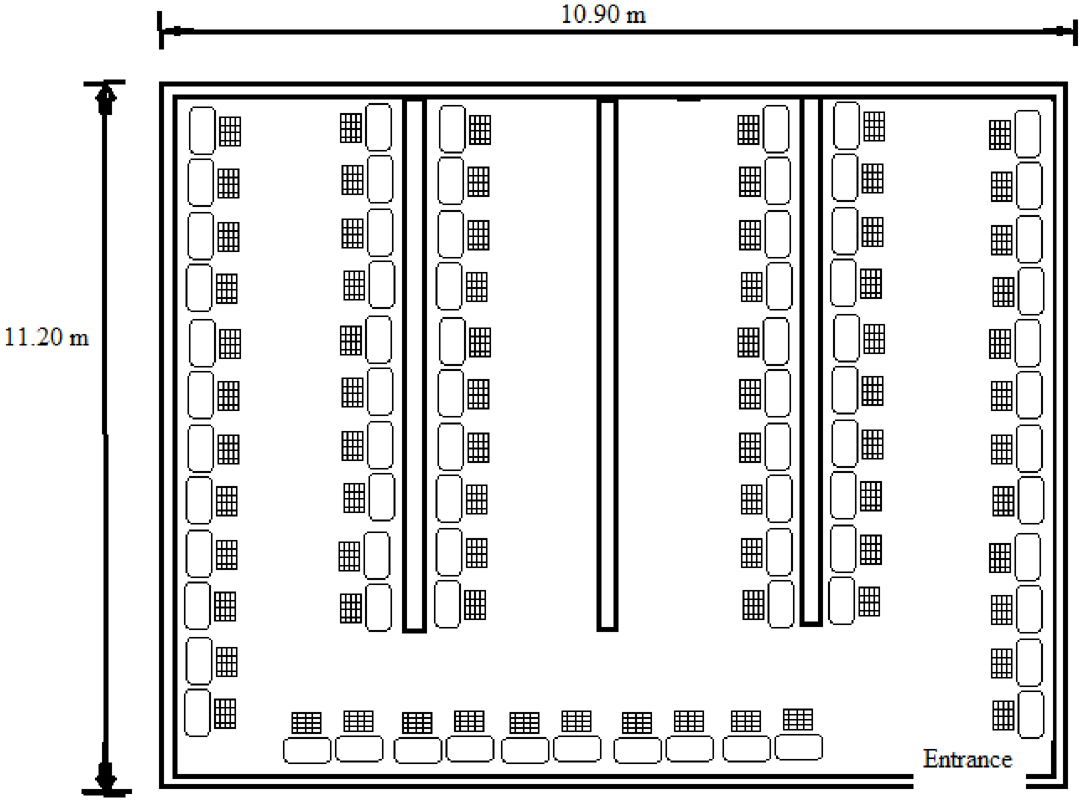
Sree Vidyanikethan Engineering College

Autonomous

Sree Sainath Nagar, Tirupathi-517102

Electrical and Electronics Engineering Departet

Signals and Networks Lab Layout



Laboratory equipment	**Laboratory details** **Room No. 423**	
No. of systems: 75	**Total cost of equipment**	Rs. 28,26,415.00 /-
Configuration	**Lab area**	120 Sq-mts
Processor: Intel - i3 **RAM:** 4 Gb **Hard disk:** 1 Tb	**Lab-in-charge**	Dr. S. Farook
	Lab technician	Mr. K. Mohan babu

Lab-in-charge **HOD**

GENERAL INSTRUCTIONS

1. Shoes shall be worn that provide full coverage of the feet, and appropriate personal clothing shall be worn in laboratories.
2. Students shall be familiar with the locations and operation of safety and emergency equipment such as, emergency power off, emergency telephones, and emergency exits.
3. Do not displace or remove laboratory equipment without instructor or technician authorization.
4. Never open or remove cover of equipment in the laboratories without instructor authorization.
5. No power laboratory should be performed without laboratory instructor present
6. Before equipment is made live, circuit connections and layout should be checked by the instructor
7. Never make any changes to circuits without first isolating the circuit by switching off and removing connections to supplies.
8. Voltages above 50 V RMS AC and 50 V DC are always dangerous. Extra precautions should be considered as voltage levels are increased.
9. Be familiar with the locations and operation of safety and emergency equipment such as emergency power off in your lab
10. Remove metal bracelets or watchstraps.
11. Use extension cords only when necessary and only on a temporary basis.
12. Do not use damaged cords, cords that become hot, or cords with exposed wiring. Inform the instructor about damaged cords.
13. Know the correct handling procedures for batteries, cells, capacitors, inductors and other high energy-storage devices.
14. If for a special reason, it must be left on, a barrier and a warning notice are required.
15. Equipment found to be faulty in any way should be reported immediately and not used until it is inspected and declared safe.

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Autonomous)

SreeSainath Nagar, Tirupati – 517 102

Department of Electrical and Electronics Engineering

Year and Semester	**II B. Tech. I Semester**
Branch	**Electrical and Electronics Engineering**
Name of the Laboratory	**Signals and Networks Lab (16BT30232)**

List of Experiments: Any TEN experiments are to be conducted

1. Generation of continuous time signals.
2. Basic operations on the signals.
3. Systems and their properties.
4. Convolution of signals.
5. Transformation of signals into time and frequency domains.
6. Transient response of RL circuit and applications.
7. Transient response of RC circuit and applications.
8. Transient response of RLC circuit and applications.
9. Determination of Open circuit and Short circuit parameters in isolated and interconnected networks.
10. Determination of ABCD and Hybrid parameters in isolated and interconnected networks.
11. Design, analysis and application of Low pass and High pass filters.
12. Design, analysis and application of Band Pass and Band stop filters.

Introduction to MATLAB

What is MATLAB?

A high-performance language for technical computing.
Typical uses of MATLAB:

- Mathematical computations.
- Algorithmic development.
- Model prototyping (prior to complex model development).
- Data analysis and exploration of data (visualization).
- Scientific and engineering graphics for presentation.
- Complex analysis using MATLAB toolboxes (i.e., statistics,
- neural networks, fuzzy logic, H-infinity control, economics, etc.) .

Why is MATLAB

Because it simplifies the analysis of mathematical models .

- It frees you from coding in high-level languages (saves a lot of time - with some computational speed penalties)
- Provides an extensible programming/visualization environment.
- Provides professional looking graphs.
- Provide a lot of toolbox that help me.
- MATLAB is usually faster than Mathematica and Maple in numeric intensive tasks.
- MATLAB has more textbooks than other packages combined (350+ books). Perhaps this speaks on the acceptance by the user community.

Introduction:

The name MATLAB stands for **MAT**rix **LAB**oratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

- MATLAB is a high-performance language for technical computing. It integrates *computation*, *visualization*, and *programming* environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated *data structures*, contains built-in editing and *debugging tools*, and supports *object-oriented programming*

- MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems.
- MATLAB is an interactive system whose basic data element is an *array* that does not require dimensioning.
- It has powerful *built-in* routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as *toolbox*. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, Neural networks, Fuzzy logic, communications and various fields of applied science and engineering.

Starting MATLAB

One can enter MATLAB by double-clicking on the MATLAB shortcut *icon* on Windows desktop. Upon starting of MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains *other* windows. The major tools within or accessible from the desktop are:

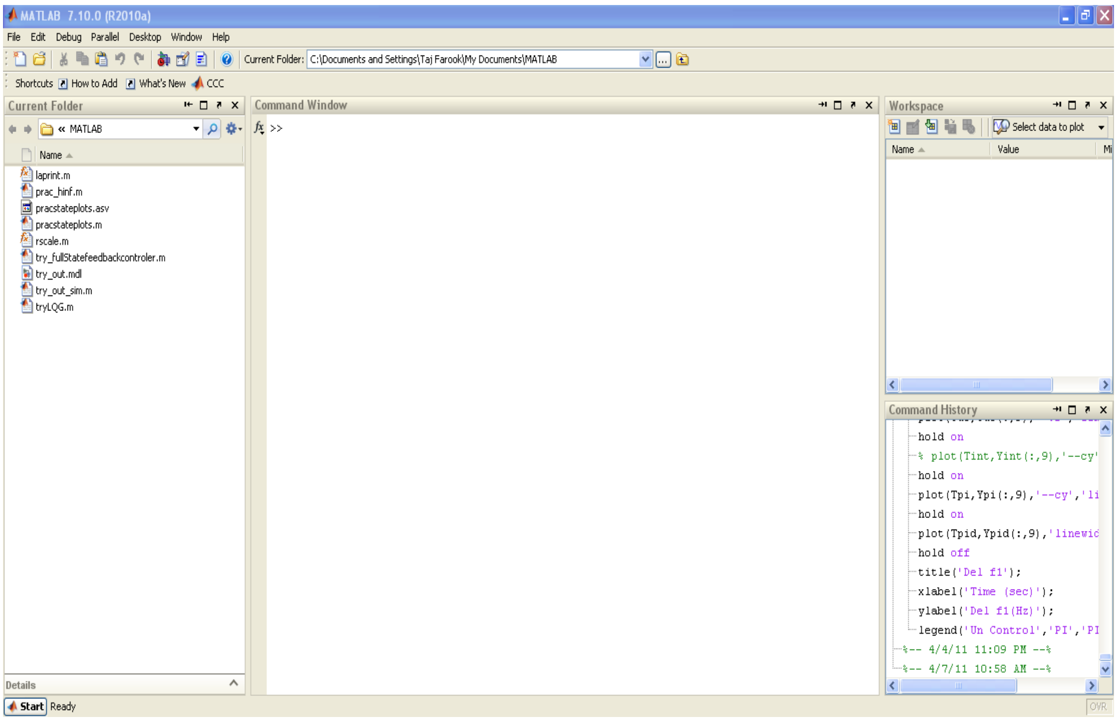


Fig. 1: MATLAB desktop layout

- **The Command Window:** the window in which the inputs and out puts can be observed.
- **The Command History:** the window consisting of the instruction given in the command during the previous sessions.
- **The Workspace:** the window consisting of the variables used in the programming.
- **The Current Directory:** the directory consisting of m-files and other files of use/work.

Variable:

A MATLAB variable is an object belonging to a specific data type. On MATLAB variable is basically a matrix. Matrices can be made up of real or complex numbers, as well as characters (ASCII symbols).

Defining MATLAB Variables

In general, the matrix is defined in the MATLAB command interface by input from the keyboard and assigned a freely chosen variable name in accordance with the following syntax:

```
>> x = 2.45
```

With this instruction, after a MATLAB prompt the number 2.45 (a number is a 1×1 matrix) will be assigned to the variable x and can subsequently be addressed under this variable name. All of the defined variables will be stored in the so-called workspace of MATLAB.

Rules for Variable Names

- MATLAB (beyond 7 Version) will support the variable names with 63 characters
- The names of variables can be of the alphabetical and numerical combinations
- The names of the variables should not start with numbers
- While naming a variable, make sure we are not using a name that is already used as a function name.
- MATLAB reserves certain keywords for its own use and does not allow overriding them. Hence the reserved key words can’t be used as variables.
- Special characters such as hyphen, % and other sign are not allowed to use as variable names.
- MATLAB is case sensitive; hence ALPHA and alpha are treated as separate variables.

Plotting

Plotting is one of the most useful applications of a math package to plot experimental or generated data

Basic 2 D plotting:

Plotting a function in MATLAB involves the following three steps:

1. Define the function
2. Specify the range of values over which to plot the function
3. Call the MATLAB plot(*x*, *y*) function

Example:

Define the function

For plotting the function y= cos(x) over a range of 0≤x≤pi

Specify the range of values over which to plot the function

To start, we have to define this interval and tell MATLAB what increment to use. The interval is defined using square brackets [ ] that are filled in the following manner:

Syntax: [start: interval: end]

Example: x=0: pi/10:2* pi

To assign this range to a variable name, we use the assignment operator. We also do this to tell MATLAB what the dependent variable is and what function we want to plot. Hence the command y = cos(x), returns the values of cos(x) to the variable y for the values of x:

Call plot(x, y) function

Now we can plot the function by entering the following command in the command window:

>> Plot(x, y)

After a moment MATLAB will open a new window on the screen with the caption *Figure 1*. The plot is found in this window.

```
>> x = [0:pi/10:2*pi];

>> y = cos(x);
```

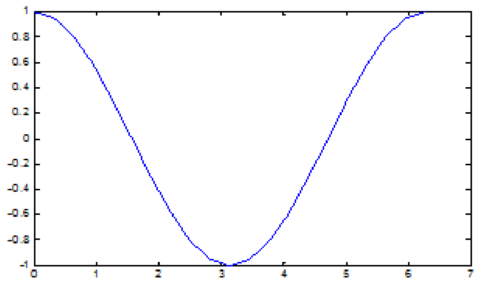


Fig. 2: Plotting the data

Options of 2D plotting:

- **Adding grid lines to the plot** We can add the axis lines(grid) on the plot by using a command *grid on* next to the plot command,
- **Adding labels to the plot:** for plotting the plot having labels can be done using the xlabel and ylabel functions. These functions can be used with a single argument, the label you want to use for each axis enclosed in quotes. Place the xlabel and ylabel functions separated by commas on the same line as your plot command. For example, the following text generates the plot shown in
  - `>> x = [0:pi/10:2*pi];`
  - `>> y = cos(x);`
  - `>> plot(x,y);`

    ```
    xlabel(‘x’);
    ylabel(‘cos(x)’);
    grid on
    ```

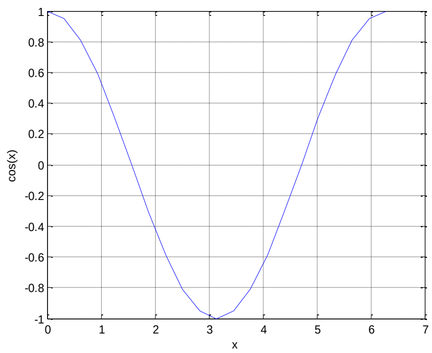


Fig. 3: Plotting the data

- **Adding title to the plot:**

  Titles can be added to the plot by using the function title (‘Title of the plot ‘)
- **Adding axis command:** MATLAB allows us to adjust the axes used in two-dimensional plots in the following way
  - Adding *axis square* to the line containing the plot command, this will cause MATLAB to generate a square plot.
  - Adding *axis equal*, then MATLAB will generate a plot that has the same scale factors and tick spacing on both axes.
  - To set the plot for user defined range call the function axis in the following way: *axis([xmin xmax ymin ymax])*
- **Multiple plots:** To plot multiple functions, we simply call the plot(x, y) command with multiple pairs x, y defining the independent and dependent variables used in the plot in pairs. This is followed by a character string enclosed in single quotes to tell us what kind of line to use to generate the second curve. In this case we have:

  >> plot (t_1, f, t_2, g);

  Where t_1 and f represents on set of data, similarly t_2 and g represents another set of data points.

- **Adding legend:** when more than two or more number of plots were plotted on the same figure window, know which curve is which it is required the discriminate the plots which can be done using the function *legend ('series1','series2','series3')*
- **Adding colors:** The color of each curve can be set automatically by MATLAB or we can manually select which color we want. This is done by enclosing the appropriate letter assigned to each color used by MATLAB in single quotes immediately after the function to be plotted is specified. Let's illustrate with an example.

  For plotting the first series in red color with dashed lines and second series with dot- dashed lines in blue color.

```
>> plot (t1, f, 'r--', t2, g, 'b.-');
```

Polar plots: MATLAB supports plotting the calculated data on to polar plots

For example let's generate a spiral. The so-called spiral of Archimedes is defined by the simple relationship:

$$r = a\boldsymbol{\theta}$$

Where a is some constant. Let's generate a polar plot of this function for the case where a = 2 and $0 \leq \boldsymbol{\theta} \leq 2\boldsymbol{\theta}$. The first statement is to assign a value to the variable a:

```
>> a = 2;
```

Now let's define the range of (θ).

```
>> theta = [0:pi/10:2*pi];
```

Now let's define the function r (θ).

```
>> r = a*theta;
```

The call to generate a polar plot is:

```
polar (theta, r)
```

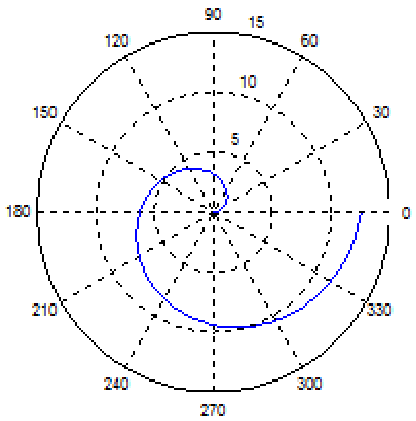


Fig. 4: Plotting the polar plot

Logarithmic and semi-logarithmic plots

Logarithmic and semi-logarithmic plots can be generated using the commands **loglog, semilogx, and semilogy**. The use of the above plot commands is similar to those of the plot command

The description of these commands is as follows:

loglog(x, y) - generates a plot of $\log_{10}(x)$ versus $\log_{10}(y)$

semilogx(x, y) - generates a plot of $\log_{10}(x)$ versus linear axis of y

semilogy(x, y) - generates a plot of linear axis of x versus $\log_{10}(y)$

Script and Functions:

Script files (m-File):

The commands in the Command Window cannot be saved and executed again. Also, the Command Window is not interactive. To overcome these difficulties, the procedure is first to create a file with a list of commands, save it and then run the file. In this way, the commands contained are executed in the order they are listed when the file is run. In addition, as the need arises, one can change or modify the commands in the file; the file can be saved and run again. The files that are used in this fashion are known as *script files*.

Thus, a script file is a text file that contains a sequence of MATLAB commands. Script file can be edited (corrected and/or changed) and executed many times.

Creating and Saving a Script File

Any text editor can be used to create script files. In MATLAB, script files are created and edited in the Editor/ Debugger Window. This window can be opened from the Command Window. From the Command Window, select *File*, *New* and then M-file. Once the window is open, the commands of the script file are typed line by line.

Example of a script file

```
x = 0:pi/100:2*pi;
y1 = 2*cos(x);
y2 = cos(x);
y3 = 0.5*cos(x);
Plot(x,y1,'--',x,y2,'-',x,y3,':')
xlabel('0 \leq x \leq 2\pi')
ylabel('Cosine functions')
legend ('2*cos(x)','cos(x)','0.5*cos(x)')
title ('Typical example of multiple plots')
axis ([0 2*pi -3 3])
```

Functions: functions are programs (or *routines*) that accept *input* arguments and return *output* arguments.

Anatomy of a M-File function

- Function m-files must start with the keyword function, followed by the output variable(s), an equals sign, the name of the function, and the input variable(s).
- If there is more than one input or output argument, they must be separated by commas. If there are one or more input arguments, they must be enclosed in brackets, and if there are two or more output arguments, they must be enclosed in square brackets. The following illustrate these points (they are all valid function definition lines):

  function [output]=function name(input arguments
- Function names must follow the same rules as variable names. The file name is the function name with ".m" appended. If the file name and the function name are different, matlab uses the file name and ignores the function name. You should use the same name for both the function and the file to avoid confusion.

Example

Function to find the sum of two numbers:

`function c= add(a, b)` // *function declaration*//

% the function takes a and b as input arguments and returns the sum into the variable c // *H1 line* //

`c=a+b;` // *evaluates the sum of two number*

Call syntax:

The function can be invoked by using the following command in the command prompt

```
>> c= add(3, 5)
>> c=8
```

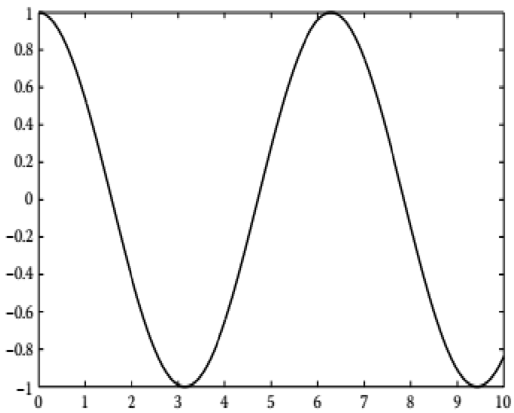
1. Generation of continuous time signals

Aim: To generate and plot different continuous time signals.

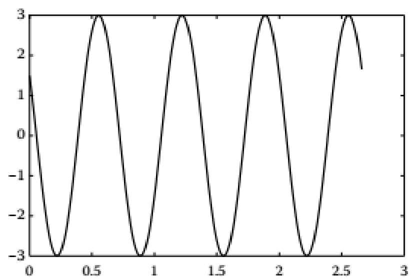
Pre lab: Before doing this experiment students need to study and write about different classification of signals and definitions of basic signals with their physical significance.

At lab: Open MATLAB command window and enter the commands for each problem and check the results.

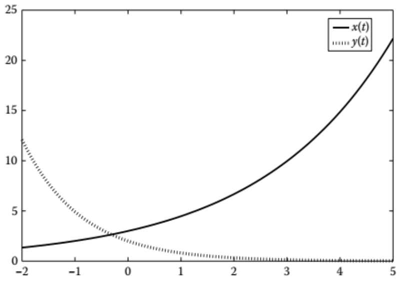
Sinusoidal signals y(t) = cos(t), 0 ≤ t ≤ 10

Code	Comments	Results
t=0:0.01:10;	Time (the independent variable t) is defined by using a very small step (time step = 0.01) in the continuous domain 0 ≤ t ≤ 10.	 Fig. 1.1: Sinusoidal signal
y=cos(t);	The dependent variable y(t) is defined in the continuous set of values -1≤ y(t) ≤ 1	
plot(t,y);	The analog signal is drawn by using the command plot	

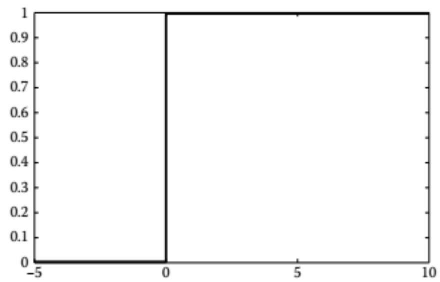
Plot the signal x(t) = 3cos(3πt+π/3) in four periods.

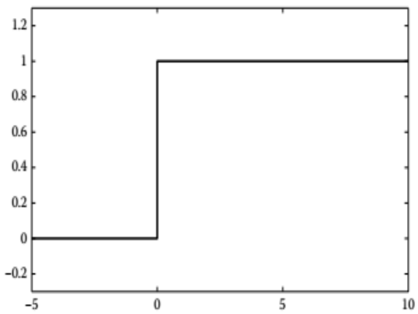
Code	Comments	Results
A=3;	The amplitude of the signal is 3	 Fig. 1.2: Sinusoidal signal
ω = 3 π;	The angular frequency is 3π	
α = π/3;	The phase is π /3.	
T=2*π/ ω;	The period is 2/3	
t=0:0.01:4*T;	The time is defined from 0 to 4T	
x = A*cos(ω*t+α)	Signal definition.	
plot(t,x)	The signal is plotted in time of four periods.	

Exponential signals Plot the signals x(t) = $3e^{0.4t}$ and y(t) = $2e^{-0.9t}$ in the time interval -2 ≤ t ≤ 5.

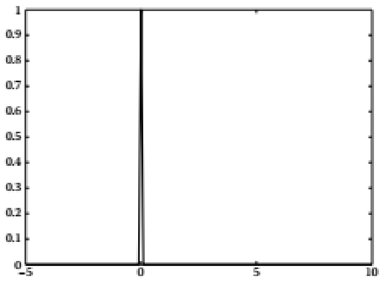
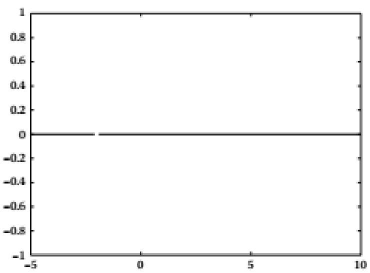
Code	Comments	Results
A=3;	The amplitude of the x signal is 3	
B=2;	The amplitude of the y signal is 2	Fig. 1.3: Exponential signals
t=-2:0.1:5;	Time is defined by using a very small step in the continuous domain -2 ≤ t ≤ 5.	
x=A*exp(0.4*t);	Representation of exponential signals	
y=B*exp(-0.9*t);		
plot(t,x,t,y,':')	The signal are plotted	
legend('x(t)','y(t)')	Legends are printed	

Step signal Plot unit step signal in the time interval -5 ≤ t ≤ 10.

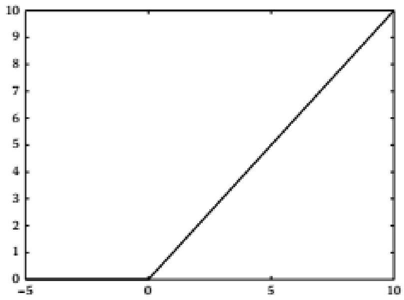
Method1 With use of the command *heaviside*.		
Code	**Comments**	**Results**
t=-5:0.1:10;	Definition of time intervel	
u= heaviside(t);	Definition of time u(t)	Fig. 1.4: Step signal
plot(t,u)	The signal is plotted	
Method 2	Use of piecewise functions.	
t1=-5:0.1:0;	Definition of the first time interval 5≤ t ≤ 0.	
t2=0:0.1:10;	Definition of the second time interval 0 ≤ t ≤10	
u1=zeros(size(t1));	Implementation of the part of u(t) that corresponds to time t1.	

u2=ones(size(t2));	Implementation of the part of u(t) that corresponds to time t2.	
t=[t1 t2];	Concatenation of the two time vectors.	
u=[u1 u2];	Concatenation of the two function vectors.	
plot(t,u)	Graph of the unit step function u(t) in the time interval $0 \leq t \leq 10$	 Fig. 1.5: Step signal
Method 2	Implementation with specific number of zeros and ones	
t=-5:0.1:10	Time definition	
u = [zeros(1,50) ones(1,101)];	The vector t consists of 151 elements. Thus, the first 50 elements of t are matched with zeros while the next 101 elements (including t = 0) of t are matched with ones. The two vectors are concatenated	
plot(t,u)	The signal is plotted	
ylim([0.3 1.3])	Y limts are changed	

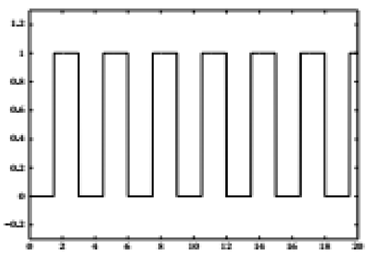
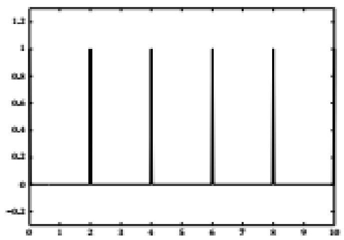
Unit impulse or Dirac Delta

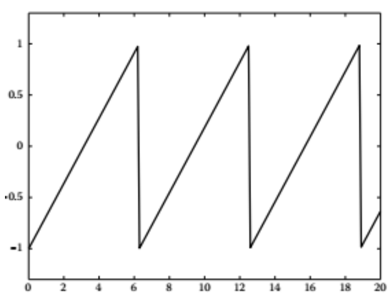
Code	Comments	Results
t1=-5:.1:-0.1;	Definition of the first time interval -5 ≤ t < 0	 Fig. 1.6: Unit impulse
t2= 0;	The second time interval is defined only for one time instance, namely for t = 0.	
t3= 0.1:.1:10;	Definition of the third time interval 0 < t ≤ 10.	
d1= zeros(size(t1));	Implementation of the part (vector of one element) of d(t) that corresponds to time t1	
d2= 1;	Implementation of the part (vector of one element) of d(t) that corresponds to time t2	
d3 =zeros(size(t3));	Implementation of the part of d(t) that corresponds to time t3.	
t = [t1 t2 t3];	Concatenation of the three time vectors.	
D = [d1 d2 d3];	Concatenation of the three function vectors.	
plot(t,d)	Graph of the Dirac function d(t).	
Method2		 Fig. 1.7: Unit impulse
t =-5:.1:10;	Definition of time.	
s =gauspuls(t)	Definition of the unit impulse	
plot(t,d)	Graph of the Dirac function d(t).	
d = dirac(t)	Graph of d(t). Notice that at the time instance t = 0 there is a gap in the graph that denotes infinity.	
plot(t,d)	Graph of the Dirac function d(t).	

Unit ramp signal

Code	Comments	Results
t1=-5:.1:-0.1;	Definition of the first time interval -5 ≤ t < 0	 Fig. 1.8: Unit ramp signal
t2= 0.1:.1:10;	Definition of the second time interval 0 < t ≤ 10.	
r1 = zeros(size(t1));b	The first part of r(t) that corresponds to time t1 is constructed	
r2 = t2;	The second part of r(t) that corresponds to time t2 is constructed	
t = [t1 t2];	Time concatenation	
r =[r1 r2];	Function concatenation	
plot(t,r)	Graph of the ramp function r(t)	
Method2		
t =-5:.1:10;	Definition of time.	
r =t.*heaviside(t);	Definition of the unit ramp	
plot(t,r)	Graph of the Dirac function r(t).	

Periodic signals

Code	Comments	Results
[s,t] =gensig('square',3,20,0.01) plot(t,s) ylim([.3 1.3])	Squares pulses that are repeated with period T =3, time duration t = 20, and sampling time ts = 0.01 s. Graph of the defined periodic signal. Indeed the square pulses are repeated every T = 3 s.	 Fig. 1.9: Square signal
[s,t] =gensig('pulse',2,10); plot(t,s) ylim([.3 1.3])	Unit pulses repeated with period T = 2 over the time interval 0 ≤ t≤ 10. Graph of repeated unit pulses.	 Fig. 1.10: Pulse signal

t = 0:0.1:20;
 	
s = sawtooth(t);
 	
plot(t,s);
 	
ylim([1.3 1.3])
	Definition of a triangle wave for 0 ≤ t≤ 20.
	
	Indeed the period is T = 2p, while the signal takes values in [1,1]
		
		
		Fig. 1.11: Sawtooth signal

Viva-voice:

1. Why we need to study about signals?
2. What are different classifications of signals?
3. Define basic signals and their physical significance.

Simulation results

Result:

2. Different operations on continuous time signals

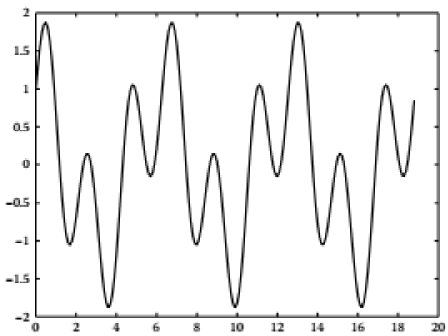
Aim: To do different operations on continuous time signals.

Pre lab: Before doing this experiment students need to study and write about different operations on signals. Solve and plot given problems theoretically.

At lab: Open MATLAB command window and enter the commands for each problem and compare the results with theoretical results.

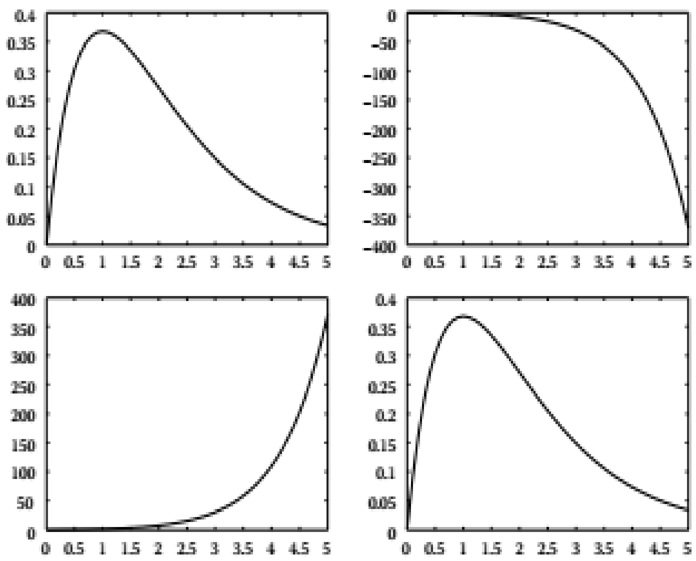
Addition of two signals

Code	Comments	Results

t = 0:.1:6*pi;
 	
x = cos(t)+sin(3*t);
 	
plot(t,x)
	Time definition (T =2 π to 3T = 6π).
	The signal x(t).
	From the graph of x(t) one can see that x(t) is indeed periodic with period T =2 π.
		
		Fig. 2.1: Addition of two signals

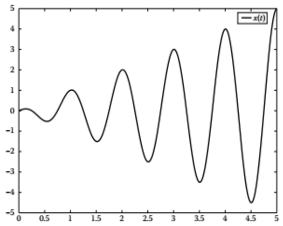
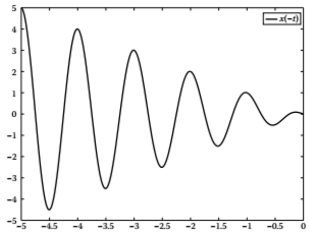
Even and odd signals

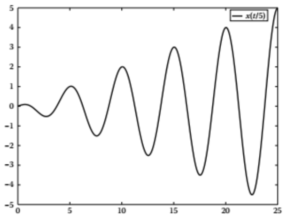
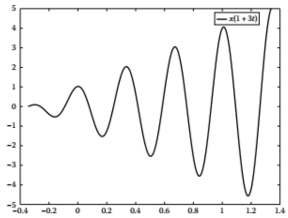
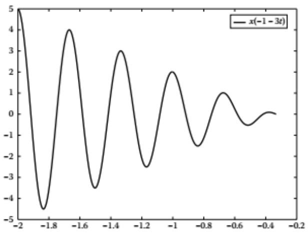
Code	Results

t = 0:.1:5;
x = t.*exp(t);
 	
xe = 0.5*t.*(exp(t)-exp(t));
 	
xo = 0.5*t.*(exp(t)+exp(t));
 	
subplot(221);
plot(t,x);
 	
subplot(222);
plot(t,xe);
 	
subplot(223);
plot(t,xo);
subplot(224);
plot(t,xe+xo);
	
	Fig. 2.2: Even and odd signals

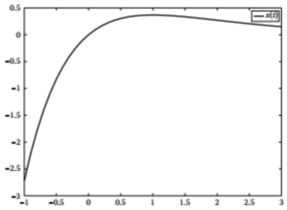
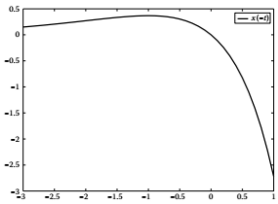
Suppose that $x(t) = t * \cos(2\pi t)$, $0 \le t \le 5$.

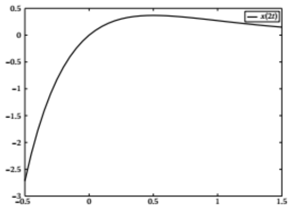
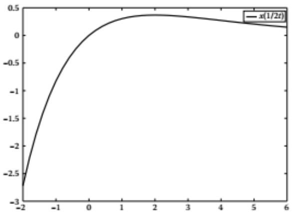
Plot the signals x(t), x(-t), x(t/5), x(1 +3t) , x(-1-3t)

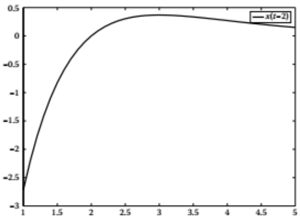
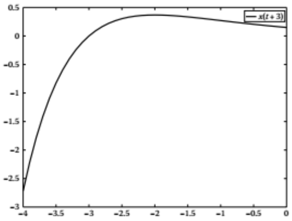
t = 0:0.01:5;
x =t.*cos(2*pi*t);
plot(t,x);
legend('x(t)');
	
	Fig. 2.3: Operations on signals
		plot(-t,x)
		legend('x(-t)');
			
			Fig. 2.4: Operations on signals

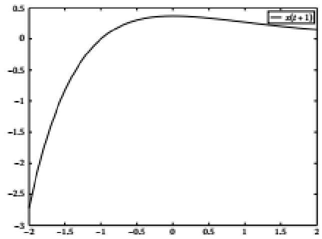
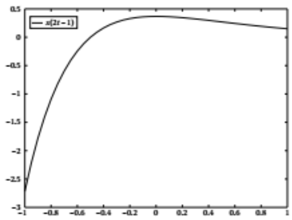
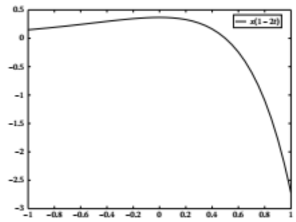
plot(5*t,x) legend('x(t/5)');	 Fig. 2.5: Operations on signals	plot((1/3)*(-1+t),x) legend('x(1+3t)');	 Fig. 2.6: Operations on signals
plot(-(1/3)*(1+t),x) legend('x(-1-3t)');	 Fig. 2.7: Operations on signals		

$x(t) = t*e^{-t}$, $-1 \le t \le 3$.

t = -1 ≤ t ≤ 3;
x = t.*exp(-t);
plot(t,x);
legend('x(t)');
	
	Fig. 2.8: Operations on signals
		plot(-t,x)
		legend('x(-t)');
			
			Fig. 2.9: Operations on signals

a = 2;
plot((1/a)*t,x)
legend('x(2t)');
	
	Fig. 2.10: Operations on signals
		a = 1/2;
		plot(
		(1/a)*t,x)
		legend('x(1/2t
		)');
			
			Fig. 2.11: Operations on signals

t0 = 2;
plot(t+t0,x)
legend('x(t-2)');
	
	Fig. 2.12: Operations on signals
		t0 = -3;
		plot(t+t0,x)
		legend('x(t+3)
		');
			
			Fig. 2.13: Operations on signals

plot(t-1,x) legend('x(t+1)')	 Fig. 2.14: Operations on signals	plot(0.5*(t- 1),x) legend('x(2t- 1)')	 Fig. 2.15: Operations on signals
plot(-0.5*(t- 1),x) legend('x(1-2t)'	 Fig. 2.16: Operations on signals		

Addition of signals

Multiplication of signals

Subtraction of signals

Amplitude and time scaling of signals

Odd and Even components of a signal

Result:

3. Properties of systems

Aim: To check different properties of given systems.

Pre lab: Before doing this experiment students need to study and write about different properties of systems and procedure for checking system properties.

At lab: Open MATLAB command window and enter the commands for each problem and check the results.

Causal and Non-causal Systems

Suppose that a system S1 is described by the i/o relationship y(t) = x(t + 1) while the i/o relationship of a system S2 is given by y(t) = x(t-1). Using the input signal x(t) = u(t)-u(t -1) find out if the two systems are causal.

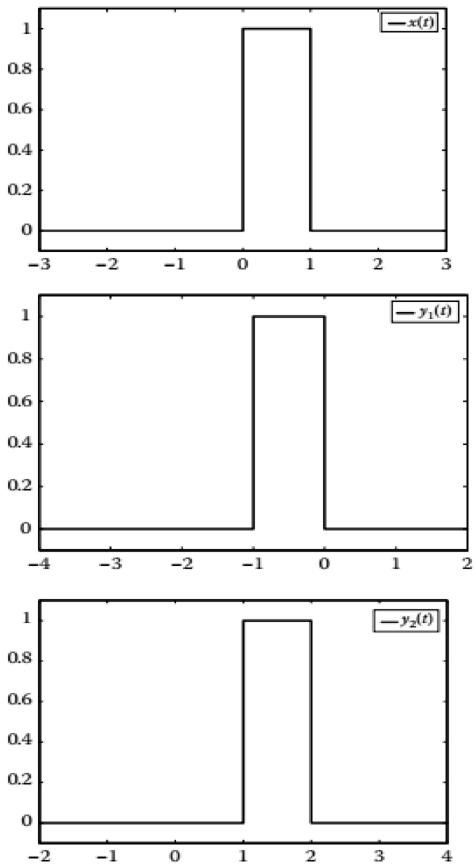
<table>
<tr><th>Code</th><th>Results</th><th>Comments</th></tr>
<tr><td>t1 = -3:.1:0;
x1
=zeros(size(t1));
t2 = 0:.1:1;
x2 =
ones(size(t2));
t3 = 1:.1:3;
x3 =
zeros(size(t3));
t = [t1 t2 t3];
x = [x1 x2 x3];
plot(t,x);
ylim([0.1 1.1]);</td><td rowspan="3">

Fig. 3.1: Causal and Non-causal Systems</td><td>Definition and graph in the time interval 3 ≤ t ≤ 3 of the input signal
x(t) = u(t)
u(t-1) = 1, 0 ≤ t ≤ 1
0, elsewhere</td></tr>
<tr><td>plot(t-1,x)
ylim([-0.1 1.1]);
legend('y_1(t)')</td><td>The output of S1 is given by y(t) = x(t + 1).
Inference: The input x(t) is zero for t < 0 but the output y(t) is nonzero for < 0, i.e., y(t) depends on future values of x(t); thus system S1 is not causal.</td></tr>
<tr><td>plot(t+1,x)
ylim([-0.1 1.1]);
legend('y_2(t)')</td><td>The output of S2 is given by y(t) = x(t - 1).
Inference: The output is zero for t < 1, i.e., y(t) depends only on past values of x(t); thus system S2 is causal.</td></tr>
</table>

Static (Memoryless) and Dynamic (with Memory) Systems

Using the input signal x(t) = u(t) - u(t - 1) find out if the systems described by the i/o relationships y(t) = 3x(t) is static or dynamic.

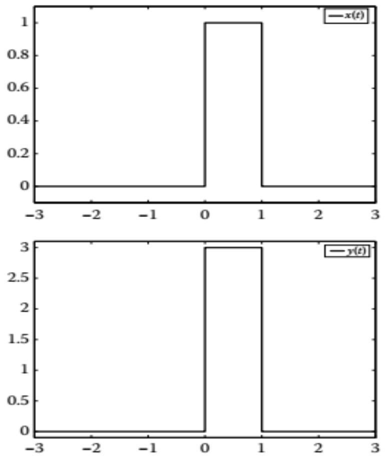
<table>
<tr><th>Code</th><th>Results</th><th>Comments</th></tr>
<tr><td>t1 = -3:.1:0;
x1 =zeros(size(t1));
t2 = 0:.1:1;
x2 = ones(size(t2));
t3 = 1:.1:3;
x3 = zeros(size(t3));
t = [t1 t2 t3];
x = [x1 x2 x3];
plot(t,x);
ylim([0.1 1.1]);</td><td rowspan="2">

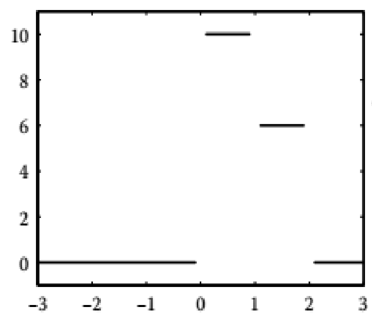
Fig. 3.2: Static (Memoryless) and Dynamic (with Memory) Systems</td><td>Definition and graph in the time interval $3 \le t \le 3$ of the input signal
x(t) = u(t) -u(t-1) = 1, $0 \le t \le 1$
0, elsewhere</td></tr>
<tr><td>plot(t,3*x);
ylim([-0.1 3.1]);
legend('y(t)')</td><td>The output of the system with i/o relationship
y(t) = 3x(t)
.</td></tr>
</table>

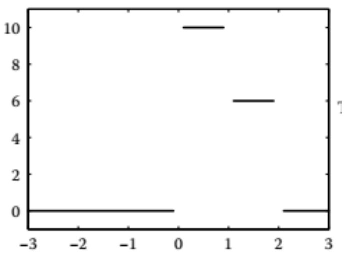
Inference:

Linear and Nonlinear Systems

Hint: For linear S{a1x1(t) + a2x2(t)} = a1S{x1(t)} + a2S{x2(t)}

Let x1(t) = u(t) - u(t - 1) and x2(t) = u(t) - u(t - 2) be input signals to the systems described by the i/o relationships y(t) = 2*x(t) and y(t) = x2(t). Determine if the linearity property holds for these two systems. To examine if the systems are linear, we use the scalars a1 = 2 and a2 = 3. The time interval considered is $3 \le t \le 3$. For the system described by the i/o relationship y(t) = 2x(t) the procedure followed is

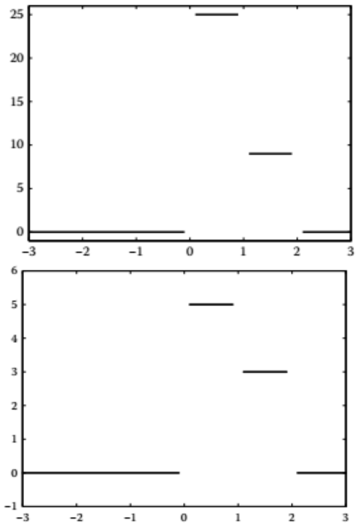
<table>
<tr><th>Code</th><th>Results</th><th>Comments</th></tr>
<tr><td>t =-3:.1:3;
x1 = heaviside(t)-heaviside(t-1);
x2 = heaviside(t)-heaviside(t-2);</td><td rowspan="2"></td><td>Definition of the input signals x1(t) and x2(t).</td></tr>
<tr><td>a1 = 2; a2 = 3;
z = a1*x1+a2*x2;
y = 2*z;
plot(t,y);
ylim([-1 11]);</td><td>The expression a1x1(t)+a2x2(t) is defined. S{a1x1(t)+a2x2(t)} is computed and the result is plotted</td></tr>
</table>

z1 =2*x1;
z2 = 2*x2;
y =a1*z1þa2*z2;
plot(t,y);
ylim([1 11]);
 	
.
	
	Fig. 3.3: Linear and Nonlinear Systems
		Definition of S{x1(t)} and S{x2(t)}.
		
		a1S{x1(t)} + a2S{x2(t)}, is computed and the result is plotted

Inference:

.

Examine if the linearity property holds for the system with i/o relationship y(t) = x^2(t).

<table>
<tr><th>Code</th><th>Results</th><th>Comments</th></tr>
<tr><td>t =-3:.1:3;
x1 = heaviside(t)-heaviside(t-1);
x2 = heaviside(t)-heaviside(t-2);</td><td rowspan="3">
Fig. 3.4: Linear and Nonlinear Systems</td><td>Definition of the input signals x1(t) and x2(t).</td></tr>
<tr><td>a1 = 2;
a2 = 3;
z = a1*x1+a2*x2;

y = z^2;
plot(t,y);
ylim([-1 26]);</td><td>The expression a1x1(t)+a2x2(t) is defined.

S{a1x1(t)+a2x2(t)} is computed and the result is plotted</td></tr>
<tr><td>z1 = x1.^2;
z2 = x2.^2;

y = a1*z1+a2*z2;
plot(t,y);
ylim([1 6]);.</td><td>Definition of S{x1(t)} and S{x2(t)}.

a1S{x1(t)} + a2S{x2(t)}, is computed and the result is plotted</td></tr>
</table>

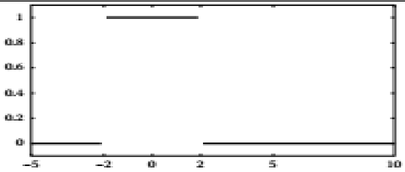
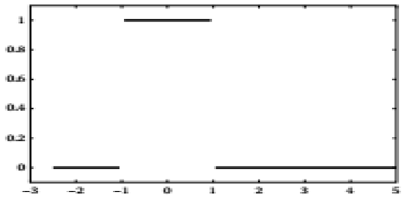
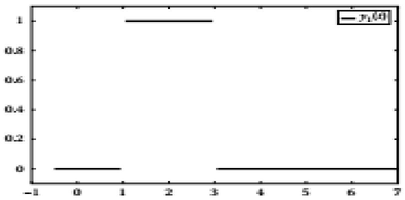
Inference:

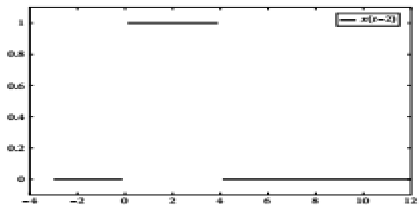
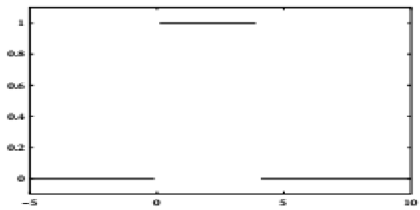
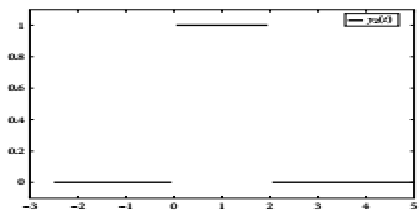
.

Time-Invariant and Time-Variant Systems

Consider a system described by the i=o relationship y(t) = x(2t). Find out if this is a time invariant system by using the input signal x(t) = u(t + 2) u(t - 2).

Code	Results	Comments

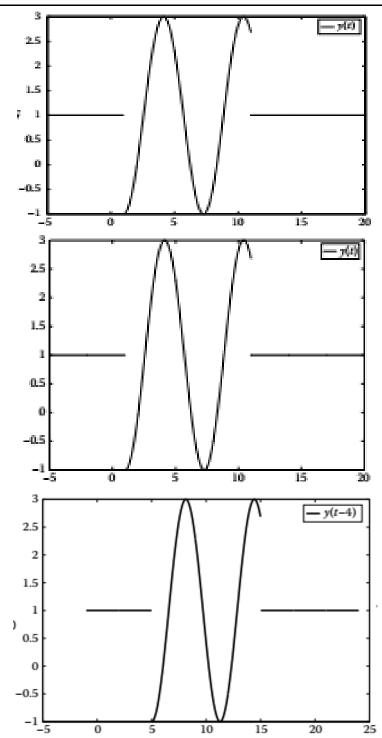
t = -5:.1:10;
 	
x = heaviside(t+2) -
heaviside(t-2);
plot(t,x);
ylim([.1 1.1]);
 	
plot((1/2)*t,x);
 	
ylim([.1 1.1]);
 	
plot((1=2)*tþ2,x);
 	
ylim([.1 1.1]);
 	
Legend('y_1(t)')
	
	
	
	Fig. 3.5: Time-Invariant and Time-Variant Systems
		The input signal x(t) =u(t + 2) - u(t - 2).
		
		The system response y(t) = x(2t) to the input signal x(t) = u(t + 2)-u(t - 2).
		
		The shifted by 2 units signal y(t - 2). The mathematical expression for
		y1(t) = y(t - 2) is
		y1(t) = u(t + 1) - u(t + 3).

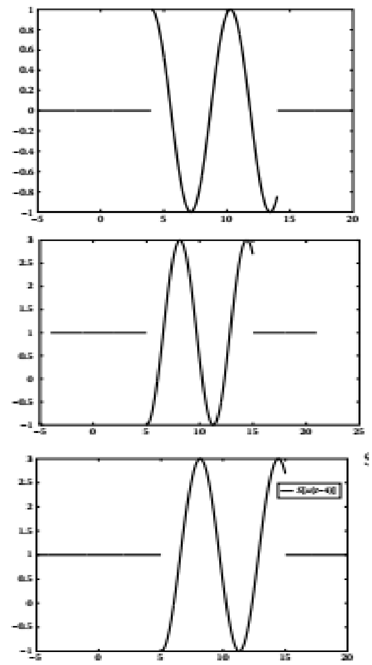
plot(t+2,x);
ylim([.1 1.1]);
legend('x(t-2)')
 	
t= -5:.1:10;
x2= heaviside(t) -
heaviside(t4);
plot(t,x2);
ylim([-.1 1.1]);
 	
plot((1=2)*t,x2);
ylim([-.1 1.1]);
legend('y_2(t)')
	
	
	
	Fig. 3.6: Time-Invariant and Time-Variant Systems
		The input signal x(t) is shifted by t0 = 2 units to the right, i.e., we plot the signal
		x(t - 2).
		
		The shifted input signal x2(t) = x(t - 2) is actually given by x2(t) = u(t) - u(t - 4). The (shifted) input signal is defined and plotted for confirmation.
		
		The system response y2(t) = x2(2t) is plotted. The mathematical expression for y2(t) is y2(t) = u(t) - u(t - 2).

Inference:

Consider a system described by the i/o relationship y(t) = 1 - 2x(t - 1). Determine if this is a time-invariant system by using the input signal x(t) = cos (t)[u(t) - u(t - 10)].

Code	Results	Comments

t = -5:.01:20;
p = heaviside(t1) - heaviside(t11);
y = 1-2*cos(t1).*p;
plot(t,y)
legend('y(t)')
 	
p = heaviside(t) - heaviside(t - 10);
x = cos(t).*p;
plot(t+1,1-2*x)
legend('y(t)')
 	
plot(t+4,y)
legend('y(t-4)')
	
	Fig. 3.7: Time-Invariant and Time-Variant Systems
		The system response y(t) to the input x(t) is defined and plotted.
		
		Alternative way of plotting the signal y(t) in terms of x(t).
		
		The shifted by 4 units to the right output signal y(t - 4).

p = heaviside(t-4) heaviside(t-14);
x = cos(t-4).*p;
plot(t,x)
 	
p = heaviside(t-4) - heaviside(t - 14);
x = cos(t-4).*p;
plot(t+1,1-2*x)
 	
t = -5:.01:20;
p = heaviside(t-5) - heaviside(t-15);
y2 = 12*cos(t-5).*p;
plot(t,y2);
legend('S[x(t4)]')
	
	Fig. 3.8: Time-Invariant and Time-Variant Systems
		Graph of the shifted input signal x(t - 4).
		
		First way to plot the system response y2(t). The response y2(t) = S{x(t - 4)} to the input signal x2(t) = x(t - 4) is plotted in terms of x2(t).
		
		Second way to plot the system response y2(t).The response y2(t) = S{x(t - 4)} of the system to the shifted signal x2(t) is computed as y2(t) = 1 - 2cos(t-5)[u(t5) - u(t15)].

Inference:

Simulation results

Causal and Non-causal Systems

Static (Memoryless) and Dynamic (with Memory) Systems

Linear and Nonlinear Systems

Result:

4. Convolution of continuous time signals

Aim: To perform convolution of continuous time signals.

Pre lab: Before doing this experiment students need to study and write about convolution of of continuous time signals and they need to solve given problems by graphical method.

At lab: Open MATLAB command window and enter the commands for each problem and check the results.

A linear time-invariant system is described by the impulse response

$$h(t) = \begin{Bmatrix} 1 - \mathrm{t}, 0 \ \le \ \mathrm{t} \le \ 1 \\ 0, \text{elsewhere:} \end{Bmatrix}$$

Calculate the response of the system to the input signal

$$x(t) = \begin{Bmatrix} 1, 0 \ \le \ \mathrm{t} \le \ 1 \\ 0, \text{elsewhere:} \end{Bmatrix}$$

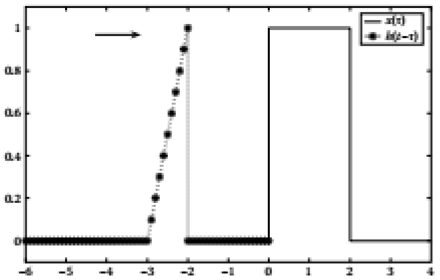
<table>
<tr><td colspan="4">Step1: For t < 0, the two signals do not overlap (zero overlap stage). Thus, the response of the system is y(t) = 0</td></tr>
<tr><td></td><td>

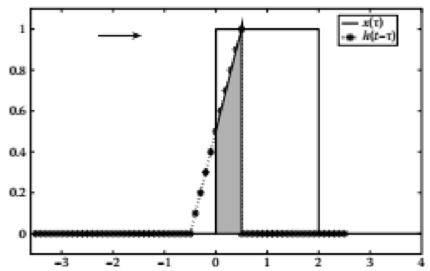
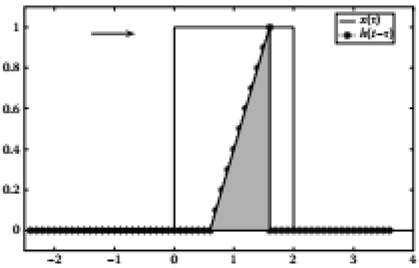
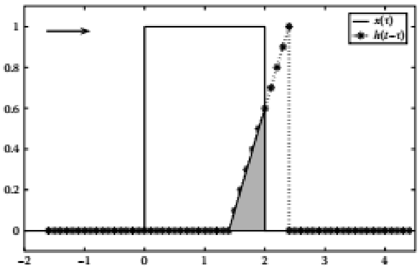
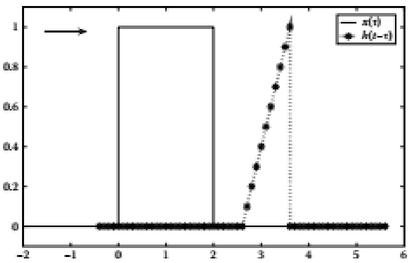
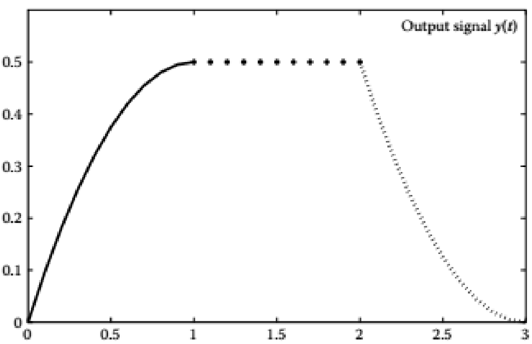
Fig. 4.1: Convolution of continuous time signals</td><td>y(t) = 0</td><td></td></tr>
<tr><td colspan="4">Step2: For 0 < t < 1, the two signals start to overlap (entry stage—partial overlap). The limits of the integral are the limits that specify the shadowed area.</td></tr>
<tr><td></td><td>

Fig. 4.2: Convolution of continuous time signals</td><td>syms t r
f = 1-t+r;
y = int(f,r,0,t)</td><td>y = t1-2/*t^2</td></tr>
<tr><td colspan="4">Step3: For 1 < t < 2, the two signals overlap completely (complete overlap stage). The only difference to the previous calculation is the integral limits.</td></tr>
</table>

<table>
<tr><td>.</td><td>
Fig. 4.3: Convolution of continuous time signals</td><td>y = int(f,r,t-1,t)
simplify(y)</td><td>y =1-t+1/2*t^2-1/2*(t-1)^2
ans = 1/2</td></tr>
<tr><td colspan="4">Step4: For 2 < t < 3, the two signals overlap partially (exit stage).</td></tr>
<tr><td></td><td>
Fig. 4.4: Convolution of continuous time signals</td><td>y = int(f,r,t1,2)</td><td>y = 5-t-t*(3-t)-1/2*(t-1)^2</td></tr>
<tr><td colspan="4">Step5: For t > 3, there is no overlap; thus, the output is y(t) = 0, t > 3.</td></tr>
<tr><td></td><td>
Fig. 4.5: Convolution of continuous time signals</td><td>y(t) = 0</td><td></td></tr>
<tr><td colspan="4">Step6: Combining all the derived results</td></tr>
<tr><td></td><td>t1 = 0:.1:1;
t2 = 1:.1:2;
t3 = 2:.1:3;
y1 = t1(t1.^2)=2;
y2 = 0.5*ones(size(t2));
y3 = 0.5*((3t3).^2);
plot(t1,y1,t2,y2,‘.’,t3,y3,‘:’)
ylim([0 0.6])
title(‘Output signal y(t)’);</td><td colspan="2">
Fig. 4.6: Convolution of continuous time signals</td></tr>
</table>

By using ***conv*** command

<table>
<tr><td colspan="3">Step:1 Defining of signals</td></tr>
<tr><td>step = 0.01;

t = 0:step:2;
x = ones(size(t));

t1 = 0:step:1;
t2 = 1+step:step:2;

h1 = 1-t1;
h2 =zeros(size(t2));
h = [h1 h2];</td><td colspan="2">The time step has to be quite small in order to approximate accurately the continuous-time signals.

The input signal x(t) = 1, $0 \le t \le 2$ is defined.

The time intervals for the two parts of h(t) are defined

The impulse response h(t) is defined in the wider time interval, namely, in the time interval where the input x(t) is defined.</td></tr>
<tr><td colspan="3">Step:2 Convolution of signals using conv command</td></tr>
<tr><td>y = conv(x,h)*step;</td><td colspan="2">The response y(t) of the system is computed by convoluting the input signal x(t) with the impulse response signal h(t) and multiplying the result with the time step .</td></tr>
<tr><td colspan="3">Step:3 Plotting results</td></tr>
<tr><td>ty = 0:step:4;

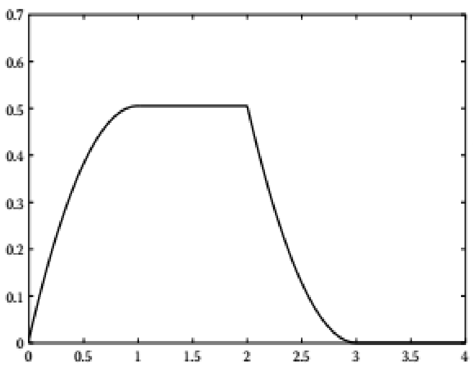
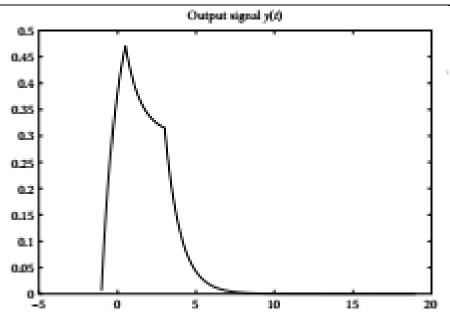
plot(ty,y);</td><td>The time interval in which the output signal y will be plotted is $0 \le t \le 4$, namely, it is double from the time interval where the vectors x and h are defined.
The response of the system y(t) computed from the convolution between the input x(t) and the impulse response h(t) is plotted.</td><td>

Fig. 4.7: Convolution of continuous time signals</td></tr>
</table>

<table>
<tr><td colspan="3">Step:1 Defining of signals</td></tr>
<tr><td>t1 = [0:0.01:1.5];
t2 = [1.5+0.01:0.01:4]
t3 = [4.01:0.01:10];
x1 = 0.6*ones(size(t1));
x2 = 0.3*ones(size(t2));
x3 = zeros(size(t3));
x =[x1 x2 x3];
h = exp([t1 t2 t3]);</td><td colspan="2">The first rule is applied; that is, the input and impulse response signals are defined over the same time interval (0 t 10), while the partial time intervals are constructed in a way that they do not overlap (second rule).</td></tr>
<tr><td colspan="3">Step:2 Convolution of signals using conv command</td></tr>
<tr><td>y = conv(x,h)*0.01;</td><td colspan="2">The output of the system is computed by multiplying the result of the convolution with the time step (third rule).</td></tr>
<tr><td colspan="3">Step:3 Plotting results</td></tr>
<tr><td>plot(1:0.01:19, y)
title('Output signal y(t)')</td><td>The output is plotted in the double time interval(fourth rule). Notice that the output signal is shifted by one unit to the left since the input signal was shifted one unit to the right</td><td>
Fig. 4.8: Convolution of continuous time signals</td></tr>
</table>

<table>
<tr><td colspan="2">Step:1 Defining of signals</td></tr>
<tr><td>t1 = [1:0.01:0.5];
x1 = .6*ones(size(t1));
t2 = [.5+0.01:0.01:3];
x2 = .3*ones(size(t2));
t3 = [3.01:0.01:10];
x3 = zeros(size(t3));
x =[x1 x2 x3];
t1 = -1:.01:-.01;
t2 = 0:.01:10;
h1 = zeros(size(t1));
h2 = exp(t2);
h ¼=[h1 h2];</td><td>The output y(t) is plotted over the time interval 2 ≤ t ≤ 20 as the input and impulse response signals are defined in the time interval
1 ≤t ≤ 10. Thus, when the input or the impulse response signals are nonzero for t < 0 the time interval where the output signal is plotted must be doubled for positive and negative values of t.</td></tr>
<tr><td colspan="2">Step:2 Convolution of signals using conv command</td></tr>
<tr><td>y = conv(x,h)*0.01;</td><td>The output of the system is computed by multiplying the result of the convolution with the time step (third rule).</td></tr>
</table>

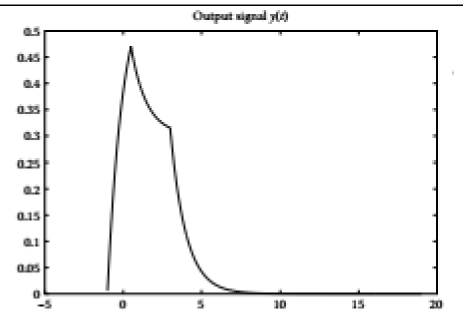
<table>
<tr><td colspan="3">Step:3 Plotting results</td></tr>
<tr><td>plot(1:0.01:19, y)
title(‘Output signal y(t)’)</td><td>The output is plotted in the double time interval(fourth rule). Notice that the output signal is shifted by one unit to the left since the input signal was shifted one unit to the right</td><td>

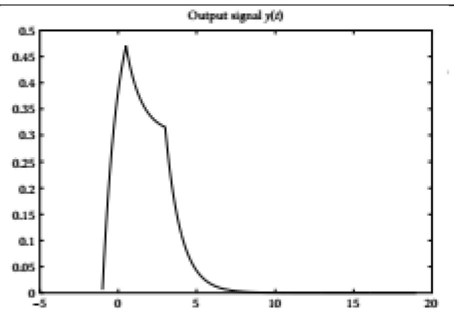
Fig. 4.9: Convolution of continuous time signals</td></tr>
<tr><td colspan="3">Step:1 Defining of signals</td></tr>
<tr><td>t1 = [1:0.01:0.5];
x1 = .6*ones(size(t1));
t2 = [.5+0.01:0.01:3];
x2 = .3*ones(size(t2));
t3 = [3.01:0.01:10];
x3 = zeros(size(t3));
x =[x1 x2 x3];
t1 = -1:.01:-.01;
t2 = 0:.01:10;
h1 = zeros(size(t1));
h2 = exp(t2);
h ¼=[h1 h2];</td><td colspan="2">The output y(t) is plotted over the time interval 2 ≤ t ≤ 20 as the input and impulse response signals are defined in the time interval 1 to 10.
Thus, when the input or the impulse response signals are nonzero for t < 0 the time interval where the output signal is plotted must be doubled for positive and negative values of t.</td></tr>
<tr><td colspan="3">Step:2 Convolution of signals using conv command</td></tr>
<tr><td>y = conv(x,h)*0.01;</td><td colspan="2">The output of the system is computed by multiplying the result of the convolution with the time step (third rule).</td></tr>
<tr><td colspan="3">Step:3 Plotting results</td></tr>
<tr><td>plot(1:0.01:19, y)
title(‘Output signal y(t)’)</td><td>The output is plotted in the double time interval(fourth rule). Notice that the output signal is shifted by one unit to the left since the input signal was shifted one unit to the right</td><td>

Fig. 4.10: Convolution of continuous time signals</td></tr>
</table>

Simulation results

Result:

5. Transformations of signals into time and frequency domains

Aim: To do transformations of different signals.

Pre lab: Before doing this experiment students need to study and write about Fourier and Laplace transformations and they need solve each problem.

At lab: Open MATLAB command window and enter the commands for each problem and check the results.

Compute the Fourier transform of the function x(t) = e^{t^2}

Code	Results	Comments
syms t w		Declaration of symbolic variables
x = exp(t^2);		Representation of x(t)
fourier(x)	ans = pi^(1/2)*exp(1/4*w^2)	The Fourier transform of X(w) = pi^(1/2)*exp(1/4*w^2)
int(x*exp(j*w*t),t,inf,inf)	ans = exp(1/4*w^2)*pi^(1/2)	The result is verified

Compute the inverse Fourier transform of the function X(w) = 1/(1 + jw)

Code	Results	Comments
X = 1/(1+j*w);		Input function
ifourier(X)	ans = exp(-x)*heaviside(x)	The inverse Fourier transform of a function is expressed with x as the independent variable. The result is e^{-x}u(x)
ifourier(X,t)	ans = exp(-t)*heaviside(t)	The inverse Fourier transform is expressed with t as the independent variable. The result is e^{-t}u(t).

Command	Result	Time domain	Frequency domain
syms t w w0 t0	Symbolic variables declared	x(t)	X(w)
x = dirac(t); fourier(x,w)	ans = 1	δ(t)	1
fourier(1,w)	ans = 2*pi*dirac(w)	1	2*pi*dirac(w)
X = 1/(j*w)+pi*dirac(w); ifourier(X,t)	ans = heaviside(t)	u(t)	1/jw + π*δ(w)
x = dirac(t-t0); fourier(x,w)	ans = exp(-j*t0*w)	δ(t-t0)	e^{-jwt0}
X = 2*pi*dirac(w-w0); ifourier(X,t)	ans = exp(j*w0*t)	exp(j*t0*w)	2*pi*dirac(w-w0)
X=pi*(dirac(w-w0)+dirac(w+w0)); ifourier(X,t)	ans = cos(w0*t)	cos(w0t)	π*δ(w-w0)+ π*δ(w+w0)
X = (pi/j)* (dirac(w-w0)-dirac(w+w0)); x = ifourier(X,t)	x = sin(w0*t)	sin(w0t)	(π/j)*δ(w-w0)+ (π/j)*δ(w+w0)
a = 8; x = exp(-a*t)*heaviside(t); fourier(x,w)	X = 1/(8 + j*w)	e^{-at} u(t), Re(a) > 0	1/(jw + a)
x = t*exp(-a*t)*heaviside(t); fourier(x,w)	ans = 1/(8 + j*w)^2	te^{-at} u(t), Re(a) > 0	$1/(jw + a)^2$
n = 4; X = 1/(j*w+a)^n; ifourier(X,t)	ans = 1/6*t^3*exp(-8*t) *heaviside(t)	$(t^{n-1} /(n-1)!)e^{-at}$ u(t), Re(a) > 0	$1/(jw + a)^n$

Result:.

6. Transient analysis of RL circuit

Aim: To study and plot transient behaviour of RL circuit.

Pre lab: Before doing this experiment students need to study about transient behaviour of RL circuit, derive expression for response of RL circuit and plot model graph for the response.

Circuit connections

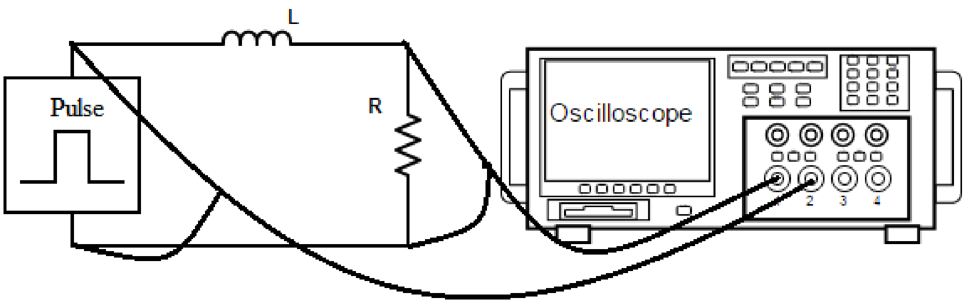


Fig. 6.1: Circuit arrangement of RL Circuit

At lab:

1. Connect the circuit as per circuit diagram shown below.
2. Apply pulse wave as input.
3. Observe input and output wave forms, note down few points for drawing graph and compare with theoretical one.
4. Observe the wave forms for different combinations of R and L values.

Inference:

Viva-voice:

Post lab: Automobile Ignition Circuit

The ability of inductors to oppose rapid change in current makes them useful for arc or spark generation. An automobile ignition system takes advantage of this feature. The gasoline engine of an automobile requires that the fuel-air mixture in each cylinder be ignited at proper times. This is achieved by means of a spark plug (Fig.), which essentially consists of a pair of electrodes separated by an air gap. By creating a large voltage (thousands of volts) between the electrodes, a spark is formed across the air gap, thereby igniting the fuel. But how can such a large voltage be obtained from the car battery, which supplies only 12 V? This is achieved by means of an inductor (the spark coil) *L*. Since the voltage across the inductor is

$$v = L\, di/dt$$

we can make it large by creating a large change in current in a very short time. When the ignition switch in Fig is closed,

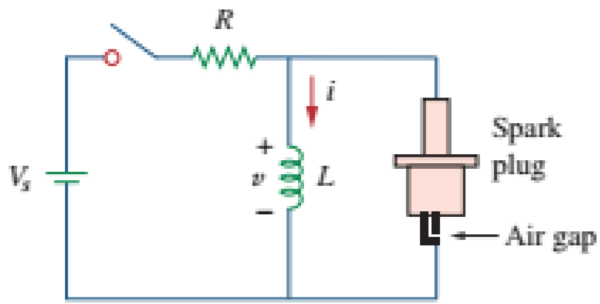


Fig. 6.2: Practical application of RL Circuit

the current through the inductor increases gradually and reaches the final value of $i = V_s / R$ where $V_s = 12V$ Again, the time taken for the inductor to charge is five times the *time constant* of the circuit $\tau = L/R$.

$$t_{charge} = 5\,\tau$$

Since at steady state, *i* is constant, di/dt=0 and the inductor voltage *v*=0. When the switch suddenly opens, a large voltage is developed across the inductor (due to the rapidly collapsing field) causing a spark or arc in the air gap. The spark continues until the energy stored in the inductor is dissipated in the spark discharge. In laboratories, when one is working with inductive circuits, this same effect causes a very nasty shock, and one must exercise caution.

Output:

Result:

7. Transient analysis of RC circuit

Aim: To study and plot transient behaviour of RC circuit.

Pre lab: Before doing this experiment students need to study about transient behaviour of RC circuit, derive expression for response of RC circuit and plot model graph for the response.

Circuit connections

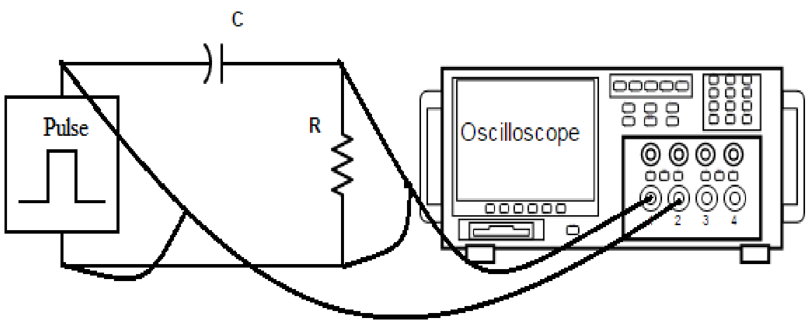


Fig. 7.1: Circuit arrangement of RC circuit

At lab:

1. Connect the circuit as per circuit diagram shown below.
2. Apply pulse wave as input.
3. Observe input and output wave forms, note down few points for drawing graph and compare with theoretical one.
4. Observe the wave forms for different combinations of R and C values.

Inference:

Post lab: : Delay Circuits

An *RC* circuit can be used to provide various time delays. Figure shows such a circuit. It basically consists of an *RC* circuit with the capacitor connected in parallel with a neon lamp. The voltage source can provide enough voltage to fire the lamp. When the switch is closed, the capacitor voltage increases gradually toward 110 V at a rate determined by the circuit's time constant, $(R1 + R2)C$. The lamp will act as an open circuit and not emit light until the voltage across it exceeds a particular level, say 70 V. When the voltage level is reached, the lamp fires (goes on), and the capacitor discharges through it. Due to the low resistance of the lamp when on, the capacitor voltage drops fast and the lamp turns off. The lamp acts again as an open circuit and the capacitor recharges.

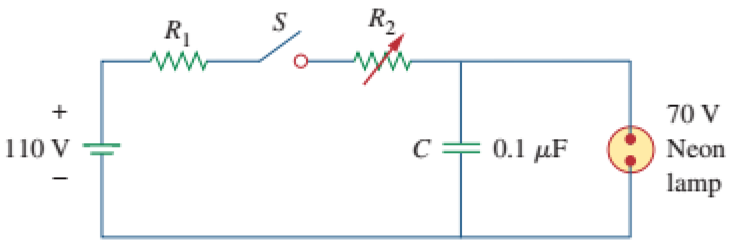


Fig. 7.2: Practical application of RC Circuit

By adjusting we can introduce either short or long time delays in to the circuit and make the lamp fire, recharge, and fire repeatedly every time constant because it takes a time period to get the capacitor voltage high enough to fire or low enough to turn off. The warning blinkers commonly found on road construction sites are one example of the usefulness of such an *RC* delay circuit.

Output:

Result:

8. Transient analysis of RLC circuit

Aim: To study and plot transient behaviour of RLC circuit.

Pre lab: Before doing this experiment students need to study about transient behaviour of RLC circuit, derive expression for response of RLC circuit and plot model graph for the response.

Circuit connections

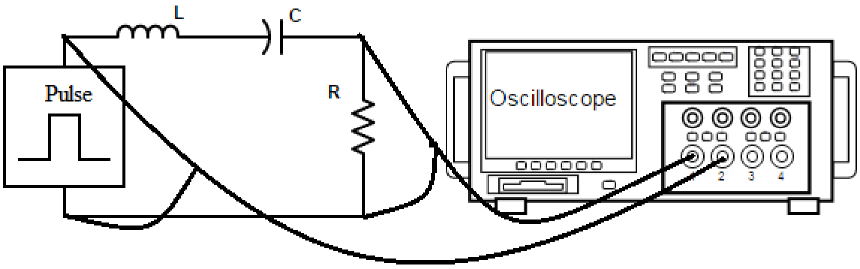


Fig. 8.1: Circuit arrangement of RLC circuit

At lab:

1. Connect the circuit as per circuit diagram shown below.
2. Apply pulse wave as input.
3. Observe input and output wave forms, note down few points for drawing graph and compare with theoretical one.
4. Observe the wave forms for different combinations of R, L and C values.(Fix L&C values and vary R value or vice versa)

Post lab:

Design example: Auto Airbag Igniter

Airbags are widely used for driver and passenger protection in automobiles. A pendulum is used to switch a charged capacitor to the inflation ignition device, as shown in Figure. The automobile airbag is inflated by an explosive device that is ignited by the energy absorbed by the resistive device represented by R. To inflate, it is required that the energy dissipated in R be at least 1 J. It is required that the ignition device trigger within 0.1 s. Select the L and C that meet the specifications.

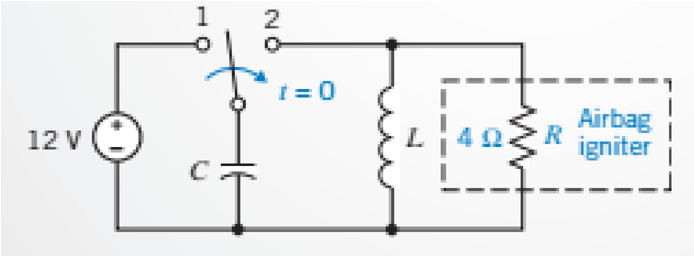


Fig. 8.2: Practical application of RLC Circuit

Describe the Situation and the Assumptions

1. The switch is changed from position 1 to position 2 at t = 0.
2. The switch was connected to position 1 for a long time.
3. A parallel RLC circuit occurs for $t \geq 0$.

We assume that the initial capacitor voltage is v(0) = 12 V and $i_L(0)= 0$ because the switch is in position 1 for a long time prior to t = 0. The response of the parallel RLC circuit for an under damped response is of the form

$$V(t)= e^{-\alpha t}(B_1\cos\omega_d t+B_2\sin\omega_d t)$$

We choose an under damped response for our design but recognize that an over damped or critically damped response may satisfy the circuit's design objectives. Furthermore, we recognize that the parameter values selected below represent only one acceptable solution.

Because we want a rapid response, we will select $\alpha = 2$ (a time constant of 1/2 s).

Where $\alpha = 1/(2RC)$.
Therefore, we have

$$C=1/(2R\alpha) =1/16 \text{ F}$$

$$\omega_0=2\pi/T=5\ \pi \text{ rad/sec}$$

$$L=1/\ \omega_0^2 C =0.065\text{H}$$

Verification of Proposed design:

$$V(t)= e^{-2t}(B_1\cos\omega_d t+B_2\sin\omega_d t)$$

Applying initial conditions

$$B_1 = v(0) = 12 \text{ V}$$

and

$$\omega_d B_2 = \alpha B_1 - B_1/RC = -24$$

$$B_2 = -1.54$$

Because B2 << B1, we can approximate

Voltage as $V(t) = 12e^{-2t}\cos\omega_d t$

Power as $P(t) = v^2/R = 36e^{-4t}\cos^2\omega_d t$

The actual voltage and current for the resistor R are shown in Figure for the first 100 ms. If we sketch the product of v and i for the first 100 ms, we obtain a linear approximation declining from 36 W at t = 0 to 0 W at t = 95 ms. The energy absorbed by the resistor over the first 100 ms is then

$$W = (1/2)*36*0.1 = 1.8J$$

Therefore, the airbag will trigger in less than 0.1 s, and our objective is achieved.

Output:

Result:

9. Two port network parameters

Aim: To study and find different network parameters for two port T-network and π-network.

Pre lab: Before doing this experiment students need to study about two port network parameters, procedure for deriving them and need to solve problems theoretically.

At lab:

1. Connect the circuit as per circuit diagram
2. Apply sufficient input voltage
3. Note down the readings meters and input voltage in table.
4. Repeat above steps for all circuits.
5. Calculate different network parameters by using given formulas.

Impedance parameters (from 1&2 circuits)

$$Z = \begin{bmatrix} z11 & z12 \\ z21 & z22 \end{bmatrix} = \begin{bmatrix} \frac{V1}{I1} & \frac{Voc1}{I2} \\ \frac{Voc2}{I1} & \frac{V2}{I2} \end{bmatrix}$$

Admittance parameters (from 3&4 circuits)

$$Y = \begin{bmatrix} y11 & y12 \\ y21 & y22 \end{bmatrix} = \begin{bmatrix} \frac{I1}{V1} & \frac{Isc1}{V2} \\ \frac{Isc2}{V1} & \frac{I2}{V2} \end{bmatrix}$$

Hybrid parameters (from 3&2 circuits)

$$h = \begin{bmatrix} h11 & h12 \\ h21 & h22 \end{bmatrix} = \begin{bmatrix} \frac{V1}{I1} & \frac{Voc1}{V2} \\ \frac{Isc2}{I1} & \frac{I2}{V2} \end{bmatrix}$$

Transmission line parameters (from 1&3 circuits)

$$T = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \frac{V1}{Voc2} & -\frac{V1}{Isc2} \\ \frac{I1}{Voc2} & -\frac{I1}{Isc2} \end{bmatrix}$$

Repeat the same procedure for π-network also and compare with theoretical calculations.

When I2=0

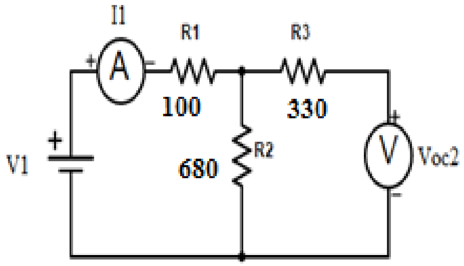


Fig. 9.1: Open circuiting Port II

Table 9.1: Open circuit parameters

V1	I1	Voc2

When I1=0

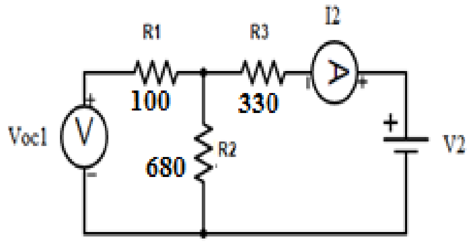


Fig. 9.2: Open circuiting Port I

Table 9.2: Open circuit parameters

V2	I2	Voc1

When V2=0

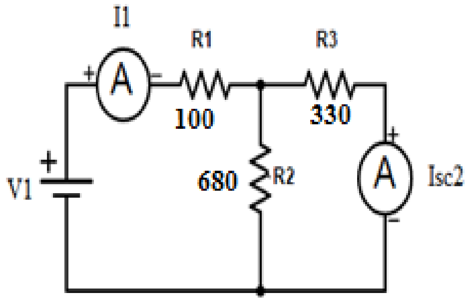


Fig. 9.3: Short-circuiting Port II

Table 9.3: Short-circuit parameters

V1	I1	Isc2

When V1=0

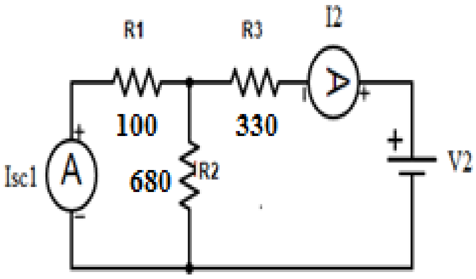


Fig. 9.4: Short-circuiting Port I

Table 9.4: Short-circuit parameters

V2	I2	Isc1

Impedance parameters

$$Z = \begin{bmatrix} z11 & z12 \\ z21 & z22 \end{bmatrix} =$$

Admittance parameters

$$Y = \begin{bmatrix} y11 & y12 \\ y21 & y22 \end{bmatrix} =$$

Hybrid parameters (from 2&3 circuits)

$$h = \begin{bmatrix} h11 & h12 \\ h21 & h22 \end{bmatrix} =$$

Transmission line parameters (from 1&3 circuits)

$$T = \begin{bmatrix} A & B \\ C & D \end{bmatrix} =$$

When I2=0

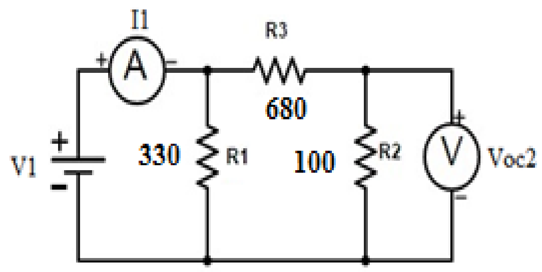


Fig. 9.5: Open circuiting Port II

Table 9.5: Open circuit parameters

V1	I1	Voc2

When I1=0

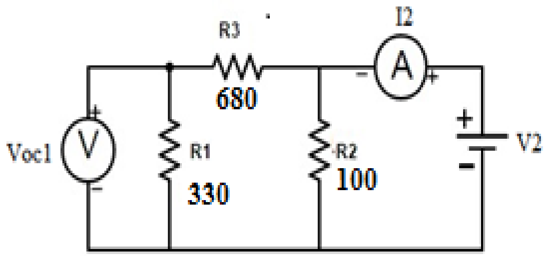


Fig. 9.6: Open circuiting Port I

Table 9.6: Open circuit parameters

V2	I2	Voc1

When V2=0

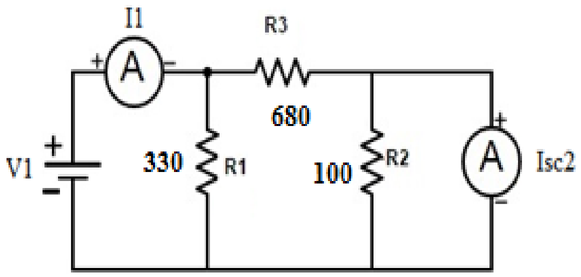


Fig. 9.7: Short-circuiting Port II

Table 9.7: Short-circuit parameters

V1	I1	Isc2

When V1=0

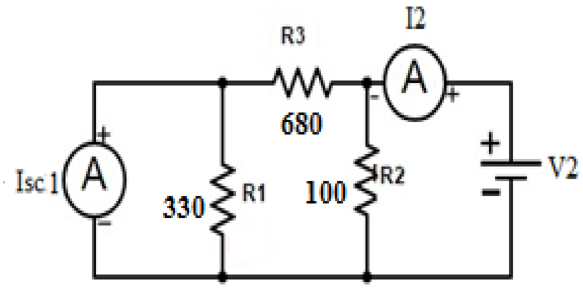


Fig. 9.8: Short-circuiting Port I

Table 9.8: Short-circuit parameters

V2	I2	Isc1

Impedance parameters (from 1&2 circuits)

$$Z = \begin{bmatrix} z11 & z12 \\ z21 & z22 \end{bmatrix}$$

Admittance parameters (from 3&4 circuits)

$$Y = \begin{bmatrix} y11 & y12 \\ y21 & y22 \end{bmatrix}$$

Hybrid parameters (from 3&2 circuits)

$$h = \begin{bmatrix} h11 & h12 \\ h21 & h22 \end{bmatrix}$$

Transmission line parameters (from 1&3 circuits)

$$T = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Result:

10. Two port network parameters for inter connected networks

Aim: To study and find network parameters for inter connected networks.

Pre lab: Before doing this experiment students need to study about inter connected two port network parameters, procedure for deriving them and need to solve problems theoretically.

At lab:

Series-Series connection:

1. Connect the circuit as per circuit diagram (take previous experiment T and π networks)
2. Apply sufficient input voltage
3. Note down the readings meters and input voltage in table.
4. Repeat above steps for the other circuit.
5. Calculate Z-parameters by using given formula and verify with theoretical values.

When I2=0

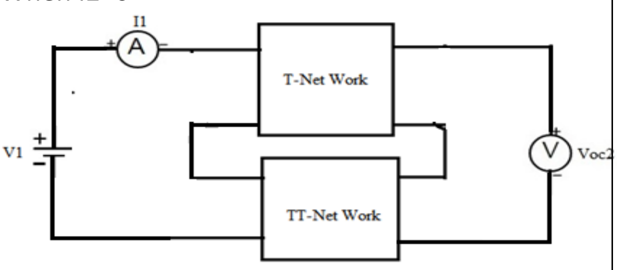


Fig. 10.1 Open circuiting Port II

Table 10.1: Open circuit parameters

V1	I1	Voc2

When I1=0

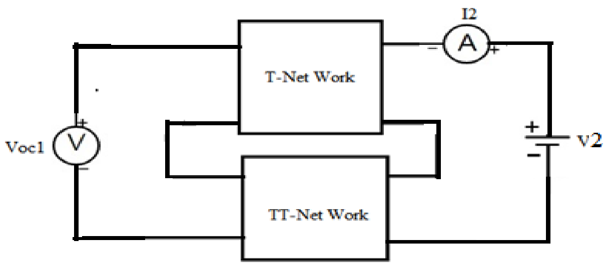


Fig. 10.2 Open circuiting Port I

Table 10.2: Open circuit parameters

V2	I2	Voc1

Impedance parameters

$$Z = \begin{bmatrix} z11 & z12 \\ z21 & z22 \end{bmatrix} = \begin{bmatrix} \frac{V1}{I1} & \frac{Voc1}{I2} \\ \frac{Voc2}{I1} & \frac{V2}{I2} \end{bmatrix} = Z_T + Z_\pi =$$

Parallel - Parallel connection:

1. Connect the circuit as per circuit diagram(take previous experiment T and π networks)
2. Apply sufficient input voltage
3. Note down the readings meters and input voltage in table.
4. Repeat above steps for the other circuit.
5. Calculate y-parameters by using given formula and verify with theoretical values.

When V2=0

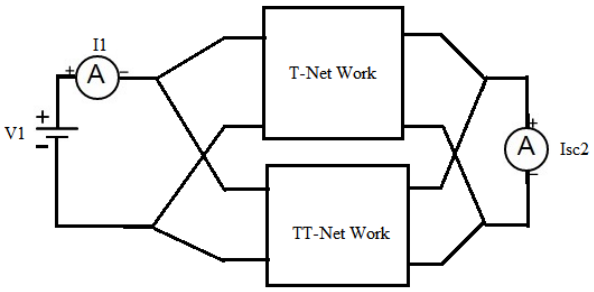


Fig. 10.3: Short-circuiting Port II

Table 10.3: Short-circuit parameters

V1	I1	Isc2

When V1=0

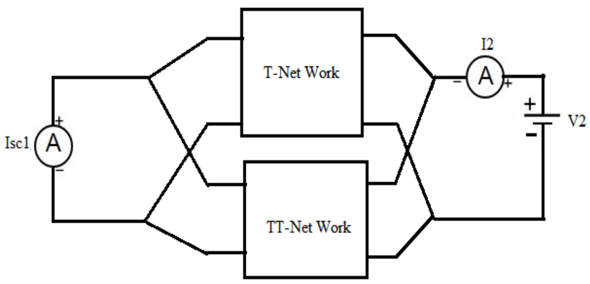


Fig. 10.4: Short-circuiting Port I

Table 10.4: Short-circuit parameters

V2	I2	Isc1

Admittance parameters

$$Y = \begin{bmatrix} y11 & y12 \\ y21 & y22 \end{bmatrix} = \begin{bmatrix} \frac{I1}{V1} & \frac{Isc1}{V2} \\ \frac{Isc2}{V1} & \frac{I2}{V2} \end{bmatrix}$$

Series - Parallel connection:

1. Connect the circuit as per circuit diagram (take previous experiment T and π networks)
2. Apply sufficient input voltage
3. Note down the readings meters and input voltage in table.
4. Repeat above steps for the other circuit.
5. Calculate h-parameters by using given formula and verify with theoretical values.

When V2=0

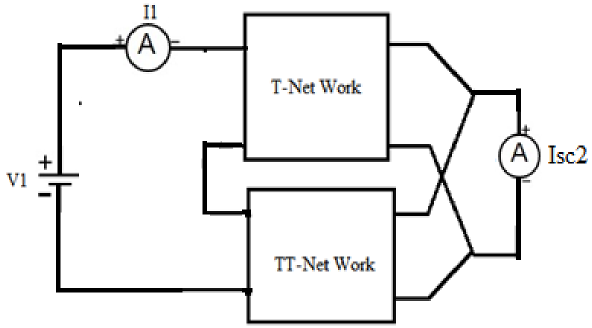


Fig. 10.5: Short-circuiting Port II

Table 10.5: Short-circuit parameters

V1	I1	Isc2

When I1=0

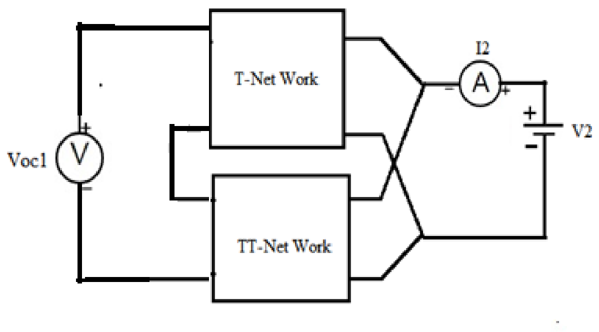


Fig. 10.6 Open circuiting Port I

Table 10.6: Open circuit parameters

V2	I2	Voc1

Hybrid parameters

$$h = \begin{bmatrix} h11 & h12 \\ h21 & h22 \end{bmatrix} = \begin{bmatrix} \frac{V1}{I1} & \frac{Voc1}{V2} \\ \frac{Isc2}{I1} & \frac{I2}{V2} \end{bmatrix}$$

Cascaded connection:

1. Connect the circuit as per circuit diagram (take previous experiment T and π networks)
2. Apply sufficient input voltage
3. Note down the readings meters and input voltage in table.
4. Repeat above steps for the other circuit.
5. Calculate h-parameters by using given formula and verify with theoretical values.

When I2=0

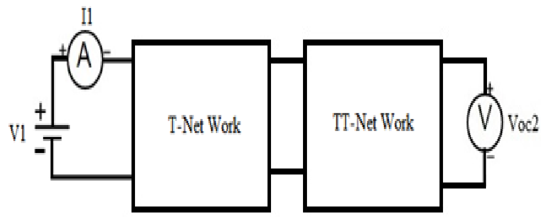


Fig. 10.7 Open circuiting Port II

Table 10.7: Open circuit parameters

V1	I1	Voc2

When V2=0

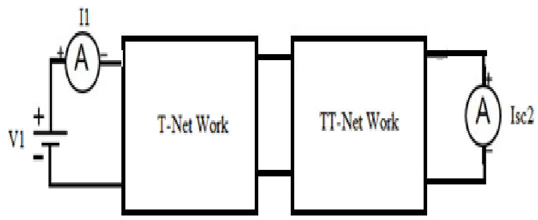


Fig. 10.8: Short-circuiting Port II

Table 10.8: Short-circuit parameters

V1	I1	Isc2

Transmission line parameters $$T = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \frac{V1}{Voc2} & -\frac{V1}{Isc2} \\ \frac{I1}{Voc2} & -\frac{I1}{Isc2} \end{bmatrix}$$

Result:

11. Design, analysis and application of Low pass and high pass filters.

A *filter* is a device that changes the amplitude (height) of an AC voltage (a voltage in the form of a sine wave) as the frequency of the input voltage changes. Filters have two terminals. The input terminals take in the input voltage, which passes through the filter and onto the output terminals, where the resulting output waveform can be observed. Figure is a basic representation of a filter.

Fig. 11.1 Filter block diagram

There are several types of filters, but in this experiment, we will be looking at three types.

A *low-pass* filter is a filter that allows a signal of a low frequency (i.e. a low amount of oscillations per second) to pass through it. Consequently, it attenuates (reduces) the amplitude of an input signal whose frequency is higher than the *cutoff* frequency. A *high-pass* filter is a filter that passes high frequencies well, but attenuates (or reduces) frequencies lower than the cutoff frequency. A *band-pass* filter is a device that passes frequencies within a certain range and rejects (attenuates) frequencies outside that range.
These three filters will be investigated in this experiment.

Low-pass filter

Figure shows a simple low-pass filter consisting of a resistor and a capacitor, which should be constructed on your breadboard. Notice that the input is connected in series with the resistor, and the output is the voltage across the capacitor. The input and output have one common terminal, which is the low (ground, or reference) side of each.

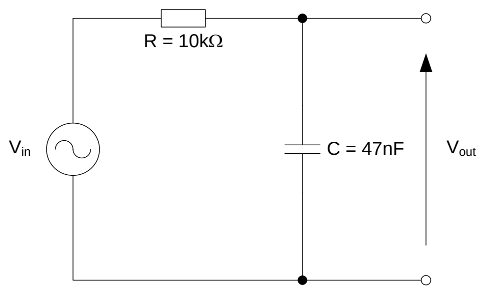


Fig. 11.2 Low pass filter

Procedure

Set up the function generator so that it produces a sinusoidal waveform, with a peak to peak voltage of 10V. Use the oscilloscope to verify this. Use Channel 1 of the scope to display V_{in}, and Channel 2 to display V_{out}. You may need to set up the triggering function of the scope, especially for the lower frequencies. Starting at 50Hz, vary the frequency of the input signal up to 2500Hz (2.5kHz) in a sufficient number of steps. For each increment, note down the peak to peak voltage of the output for each frequency, and tabulate your results in your lab book. Plot a graph of the amplitude of V_{out} against the frequency, which should resemble Figure.

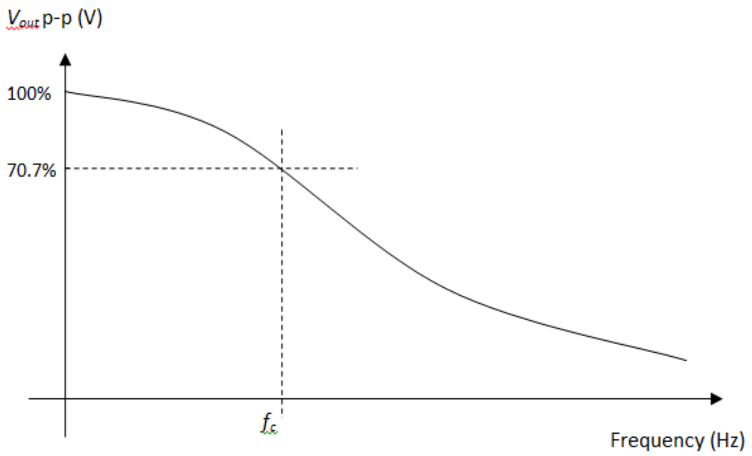


Fig. 11.3 Frequency response of low pass filter

From your graph, mark off 70.7% of your peak voltage, and note down the corresponding cutoff frequency, f_c. The cutoff frequency can also be calculated from values of the components in the circuit, i.e. the resistor R and the capacitor C, using the following formula.

$$f_c = \frac{1}{2\pi RC} \text{ Hz}$$

Calculate the cutoff frequency using the above formula, and account for any discrepancies between the calculated value and the measured value.

Experimental results

Table 11.1 Frequency response of low pass filter

S. No.	Frequency	Vin	Vout	20log(Vout/Vin)
1	50			
2	100			
3	500			
4	1000			
5	1500			
6	1750			
7	2000			
8	2500			
9	2750			
10	3000			

High-pass filter

Circuit

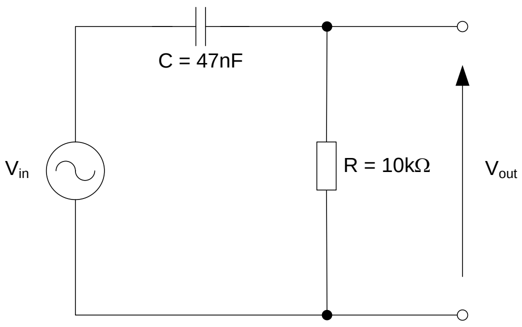


Fig. 11.5: High pass filter

Figure shows an *RC* network that behaves as a high-pass filter. Notice that the high-pass filter is the same as the low-pass filter, but with the positions of the resistor and capacitor interchanged. Here the input is in series with the capacitor and the output voltage is taken across the resistor.

Procedure

Repeat the tests as outlined in but this time start your frequency readings at 100Hz and work your way up to 10kHz. Record all your results in your lab book. The cutoff frequency can be calculated in the same way as for the low-pass filter. Note that your corresponding graph will not be the same as that shown in Figure.

Repeat the spectrum analysis test, but this time set the frequency scale to 500Hz. Compare the two tests and explain your observations.

Table 11.2 Frequency response of high pass filter

S. No.	Frequency	Vin	Vout	20 log(Vout/Vin)
1	50			
2	100			
3	500			
4	1000			
5	1500			
6	1750			
7	2000			
8	2500			
9	2750			
10	3000			

Result:

12. Design, analysis and application of Band Pass and Band stop filters.

Band-Pass Filter and Band Stop Filter

Figure shows a series *RLC* (resistor-inductor-capacitor) circuit used as a filter. Here the output voltage is taken across the resistor. The input is in series with the inductor and the capacitor.

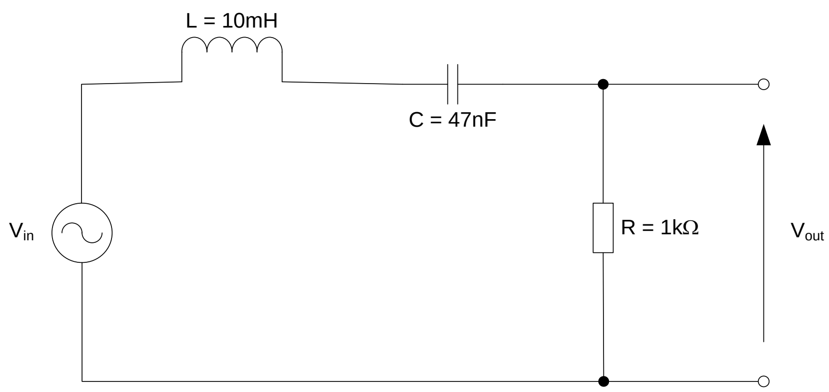


Fig. 12.1: Band pass filter

Procedure

Repeat the tests as outlined in Section (frequencies from 2kHz up to 30kHz). However, the characteristics for a band-pass filter differ greatly from that of the previous two filters investigated. When plotting your graph of the amplitude of V_{out} against the frequency, make sure it resembles that of Figure

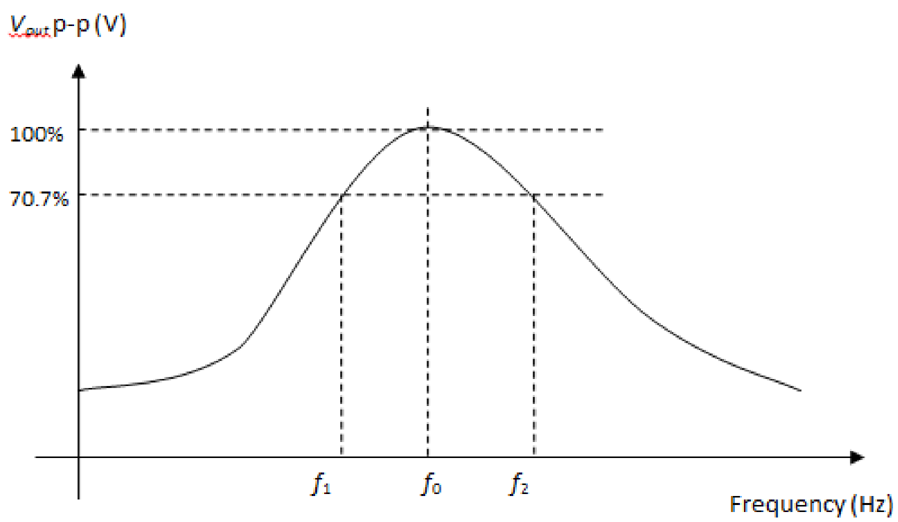


Fig. 12.2: Characteristics of a Band pass filter

As mentioned earlier, a band-pass filter allows signals to pass whose frequencies lie within a certain *band*, i.e. those frequencies that are between a lower cutoff frequency f_1 and an upper cutoff frequency f_2, as shown in Figure . The *bandwidth* (BW) of a band-pass filter is the difference between the upper and lower cutoff frequencies.

$$BW = f_2 - f_1 \text{ Hz}$$

From your graph, mark off the peak value of the output voltage, and consequently, draw a horizontal line marking off 70.7% of this value so that it intersects the graph at the points shown in Figure. Note in your lab books f_0, f_1, and f_2 as measured from the graph, and thus calculate the bandwidth for your circuit.

Formulae can also be used to calculate the cutoff frequencies and central frequency (also called the *resonant* frequency).

$$f_1 = \frac{1}{4\pi}\left(\sqrt{\left(\frac{R}{L}\right)^2 + \frac{4}{LC}} - \frac{R}{L}\right) \text{Hz}$$

$$f_2 = \frac{1}{4\pi}\left(\sqrt{\left(\frac{R}{L}\right)^2 + \frac{4}{LC}} + \frac{R}{L}\right) \text{Hz}$$

$$f_0 = \frac{1}{2\pi\left(\sqrt{\mathrm{LC}}\right)}$$

Use the above formulae to compute the desired frequencies, based on the nominal values of the components used in your band-pass filter circuit. Next compute the bandwidth using

$$BW = \frac{R}{2\pi L} \text{ Hz}$$

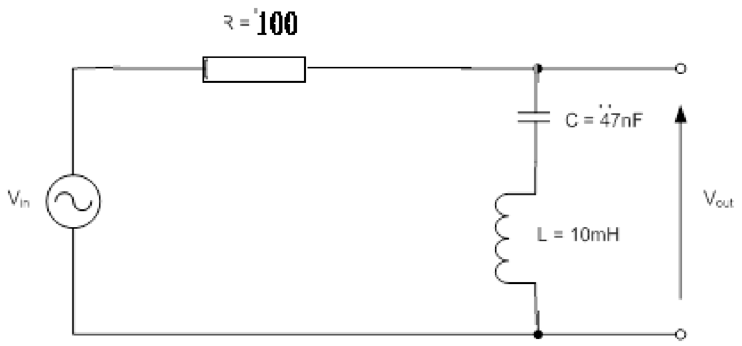


Fig. 12.3: Band Stop filter

Band Pass filter

Table 12.1 Frequency response of band pass filter

S. No.	Frequency	Vin	Vout	20 log (Vout/Vin)
1	50			
2	100			
3	500			
4	1000			
5	1500			
6	1750			
7	2000			
8	2500			
9	2750			
10	3000			
11	5000			
12	30000			
13	35000			
14	45000			
15	50000			
16	75000			
17	100000			
18	350000			

Band stop filter

Table 11.2 Frequency response of band stop filter

S. No.	Frequency	Vin	Vout	20 log (Vout/Vin)
1	50			
2	100			
3	500			
4	1000			
5	1500			
6	1750			
7	2000			
8	2500			
9	2750			
10	3000			
11	5000			
12	15000			
13	20000			
14	30000			
15	35000			
16	45000			
17	50000			
18	75000			
19	100000			
20	350000			

Result: