# APPENDIX
# (A)

# Appendix (A*)*

# MATLAB

Getting Started ***MATLAB®*** is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. This collection includes the following topics.

## What Is MATLAB?

MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include Math and computation Algorithm development Data acquisition Modeling, simulation, and prototyping Data analysis, exploration, and visualization Scientific and engineering graphics Application development, including graphical user interface building .

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or Fortran. The name **MATLAB stands for matrix laboratory**. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of add-on application-specific solutions called toolboxes.

## Toolboxes.

Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

## MATLAB system

MATLAB system consists of five main parts:

***Development Environment.*** This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the ***MATLAB desktop*** and ***Command Window***, a ***command history***, an ***editor and debugger***, and browsers for viewing help, the workspace, files, and the search path.

**The MATLAB Mathematical Function Library**.

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

**The MATLAB Language**.

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs. Graphics.   MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

The MATLAB Application Program Interface (API).   This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

## Documentation  MATLAB

Provides extensive documentation, in both printed and online format, to help you learn about and use all of its features. If you are a new user, start with this book, Getting Started with MATLAB, which introduces you to MATLAB. It covers all the primary MATLAB features at a high level, including many examples to help you to learn the material quickly: Development Environment--Introduces the MATLAB development environment, including information about tools and the MATLAB desktop. Manipulating Matrices--Introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices. Graphics--Introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images. Programming with MATLAB--Describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays. This section also provides an overview of the demo programs included with MATLAB. To find more detailed information about any of these topics, use the MATLAB online help. The online help provides task-oriented and reference information about MATLAB features. The MATLAB documentation is also available in printed form and in PDF format.

## Development Environment

The Development Environment covers starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files, including the MATLAB desktop. For more information about the topics covered here, see the corresponding topics in Development Environment", which is available in the online as well as in the printed manual.

The graphical user interface to MATLAB .Desktop Tools Use the Command Window for running functions and entering variables, Start button for launching tools, demos, and documentation, Help browser for accessing documentation, Current Directory browser for accessing files, Workspace browser for viewing variables, Editor/Debugger for modifying MATLAB program files (M-files), and Profiler for optimizing M-file performance .Other Development Environment Features Import and export data, improve M-file performance, interface with source control systems, and access MATLAB from Microsoft Word using the MATLAB Notebook feature.

## MATLAB Start and quit MATLAB

Starting and Quitting MATLAB Starting MATLAB On Windows platforms, to start MATLAB, **double-click** the MATLAB shortcut icon on your Windows desktop. On UNIX platforms, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens--see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup, and reduce startup time in some situations. For more information, see the documentation for starting MATLAB. Quitting MATLAB To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish. m script. MATLAB Online Help

## MATLAB Desktop

MATLAB Online Help To view the online documentation, select MATLAB Help from the Help menu in MATLAB. For more information about using the online documentation, see Help Browser. For MATLAB, the documentation is organized into these main topics: Development Environment--Provides complete information on the MATLAB desktop. Mathematics--Describes how to use MATLAB mathematical and statistical capabilities.

### Programming and Data Types

Describes how to create scripts and functions using the MATLAB language.

### Graphics

Describes how to plot your data using MATLAB graphics capabilities.

### 3-D Visualization

Introduces how to use views, lighting, and transparency to achieve more complex graphic effects than can be achieved using the basic plotting functions.

### Creating Graphical User Interfaces

Describes how to use MATLAB graphical user interface layout tools.

### External Interfaces/API

Describes MATLAB interfaces to C and Fortran programs, Java classes and objects, COM objects, data files, serial port I/O, and DDE. In addition to the above documentation, MATLAB documentation includes the following reference material: Functions - By Category--Lists all the core MATLAB functions. Each function has a reference page that provides the syntax, description, mathematical algorithm (where appropriate), and related functions. You can also access any function reference page using the "Functions - Alphabetical List". Handle Graphics Property Browser--Enables you to easily access descriptions of graphics object properties. For more information about MATLAB graphics, see Handle Graphics External Interfaces/API Reference--Covers those functions used by the MATLAB external interfaces, providing information on syntax in the calling language, description, arguments, return values, and examples. MATLAB online documentation also includes Examples--An index of major examples included in the documentation. Release Notes--Introduces new features and identifies known problems in the current release.

### Printable Documentation

Provides access to the PDF versions of the documentation, which are suitable for printing.

## Demos

Getting Started with Demos Use the Demos feature in the Help browser to access demonstrations of MATLAB and related products. The style of the demos varies, but for many of them, you can Run the demo. Learn more about the topic. View the source code for the demo. Copy source code from the demo to your own M-files. To Run Demos In the left pane, expand the listing for a product area (for example, MATLAB). Within that product area, expand the listing for a product or product category (for example, MATLAB Graphics). Select a specific demo from the list (for example, Visualizing Sound). In the right pane, view instructions for using the demo. Function Alternative You can also type demo in the Command Window to open the Help browser to the Demos tab, or go directly to the demos for a specific product or category. For example demo matlab graphics lists the demos for MATLAB Graphics. Notes The Search tab will not find terms contained in the demos. The Product Filter does not apply to demos. Demos are shown for all your installed products, even if you did not install the online documentation for those products. See Also In the Contents pane, see the Examples listing for a product. It contains links to the major examples included in the documentation, providing more code samples you can view, run, or copy.

### Demos Toolboxes

Are specialized collections of M-files (MATLAB language programs) built specifically for solving particular classes of problems. Our toolboxes represent the efforts of some of the world's top researchers in fields such as controls, signal processing, system identification, and others. The following toolboxes have demos to browse through. Try these demos to see which toolboxes might be appropriate for the work you do. Note that this is a comprehensive list of toolboxes. Your particular installation of Math Works products will likely include only some of these products.

**Toolbox Description :**

**Communications Design** and analyze communications systems

**Control System** Design and analyze feedback control systems

**Curve Fitting** Perform model fitting and

**Analysis Data Acquisition** Acquire and send out data from plug-in data acquisition board

**Database** Exchange data with relational database

**Filter Design** Design and analyze advanced floating-point and fixed-point filters **Financia**l Model financial data and develop financial analysis algorithms

**Fuzzy Logic** Design and simulate fuzzy logic systems

**Image Processing Perform** image processing, analysis, and algorithm development **Instrument Control** Control and communicate with test and measurement **instruments LMI Control Design** robust controllers using convex optimization techniques MATLAB Link for Code Composer Studio Use MATLAB with RTDX-enabled Texas Instruments digital signal processors

**Mapping** Analyze and visualize geographically based information

**Model Predictive Control** Control large, multivariable processes in the presence of constraints Mu-Analysis and Synthesis Design multivariable feedback controllers for systems with model uncertainty

**Neural Network** Design and simulate neural networks

**Optimization** Solve standard and large-scale optimization problems

**Partial Differential** Equation Solve and analyze partial differential equations **Robust Control** Design robust multivariable feedback control systems

**Signal Processing** Perform signal processing, analysis, and algorithm development

**Spline** Create and manipulate spline approximation models of data

**Statistics** Apply statistical algorithms and probability models

**Symbolic Math** Perform computations using symbolic mathematics and variable-precision arithmetic

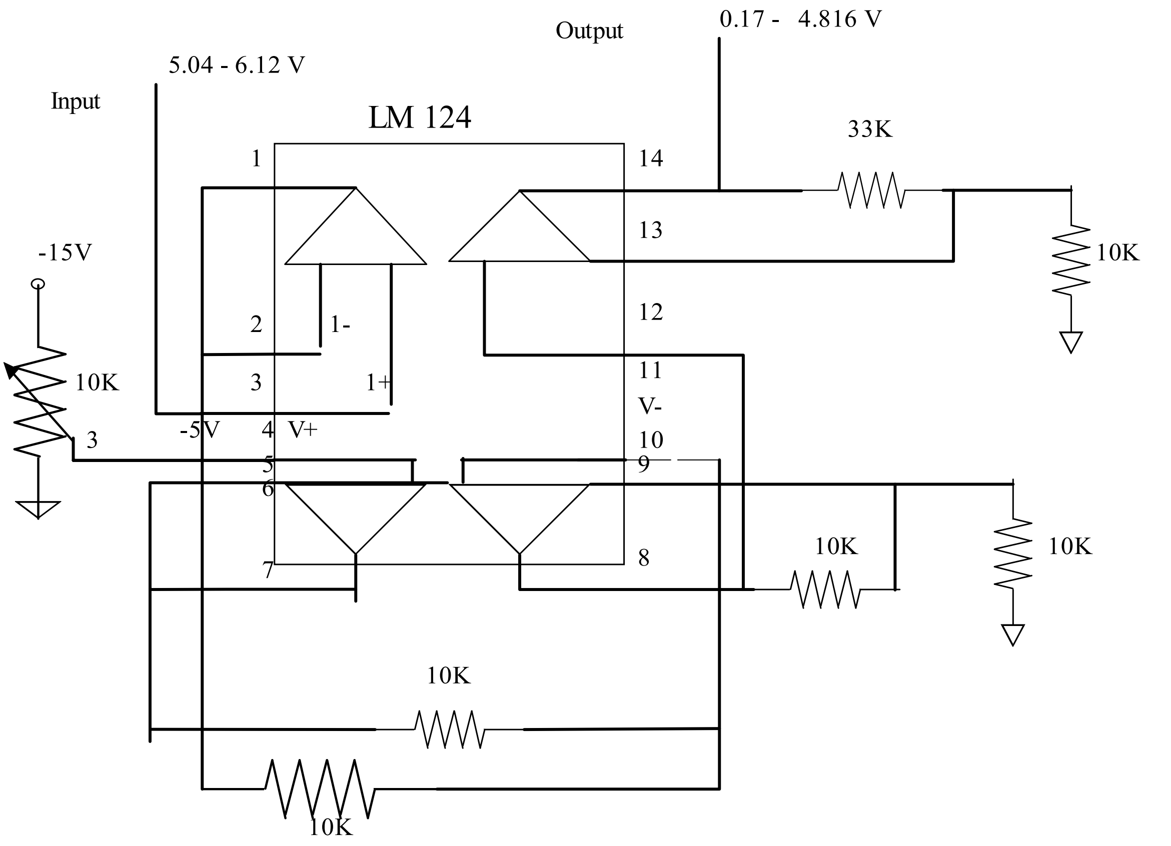
**System Identification** Create linear dynamic models from measured input-output data .

**Virtual Reality** Create and manipulate virtual reality worlds from within MATLAB and Simulink.

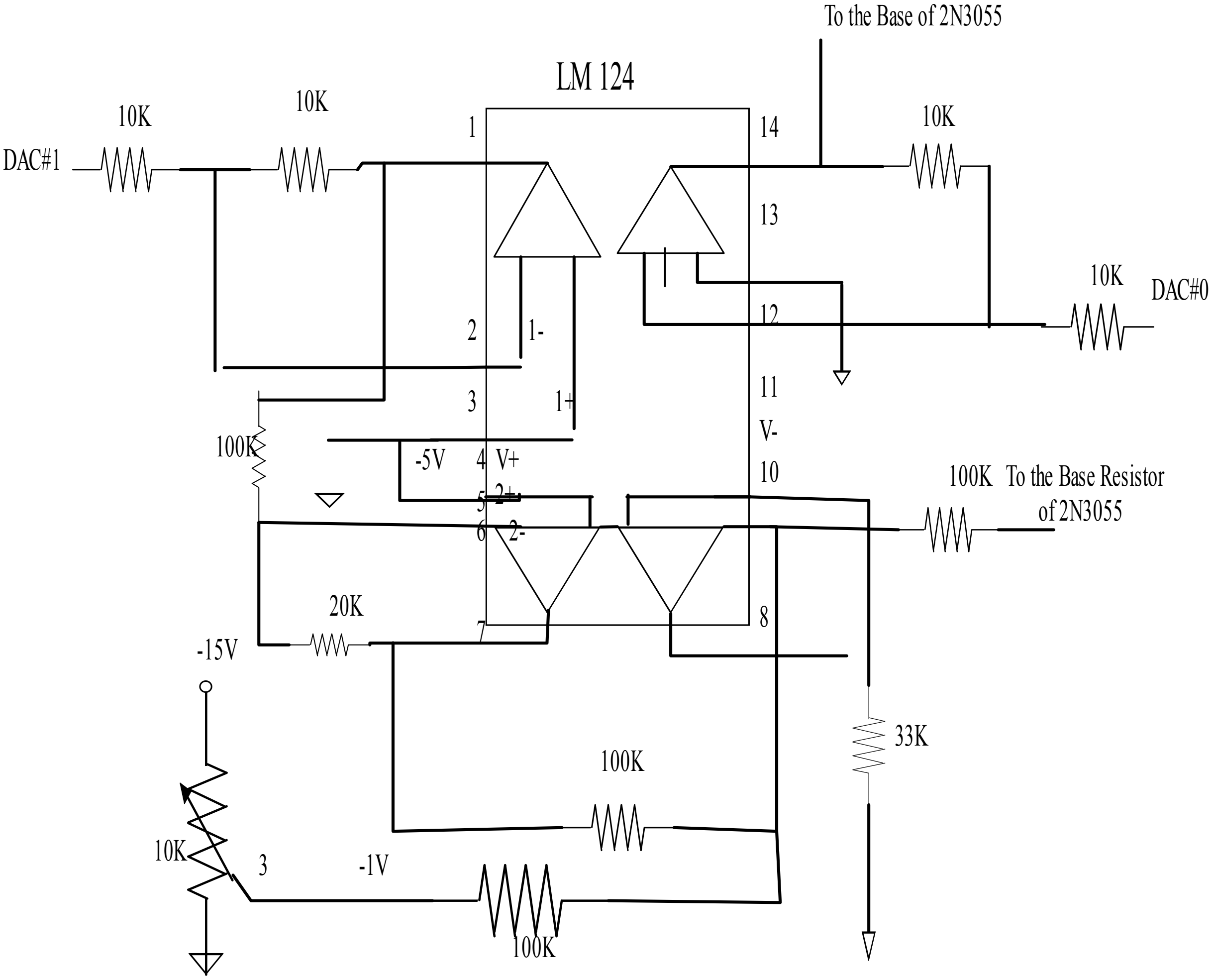
**Wavelet** Analyze, compress, and de noise signals and images using wavelet techniques

# APPENDIX

# (B)



A-    A/D OP-amp Circuit



B- Shunt/Field OP-amp Circuit

# APPENDIX
# (C)