

Emona-based Interactive Amplitude Modulation/Demodulation iLab

by

Edison M. Achelengwa

S.B. Electrical Engineering and Computer Science, M.I.T., 2010

Submitted to the Department of Electrical Engineering and Computer Science

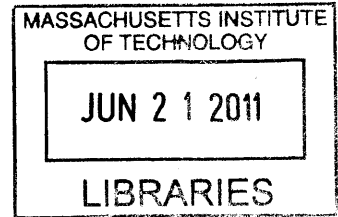
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

June 2011

Copyright 2011 Edison M. Achelengwa. All rights reserved.



ARCHIVES

The author hereby grants to M.I.T. permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.

Author _____

Department of Electrical Engineering and Computer Science

May 20, 2011

Certified by _____

Jesús A. del Alamo
Professor of Electrical Engineering
Thesis Supervisor

Accepted by _____


Dr. Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Emona-based Amplitude Modulation/Demodulation iLab

by

Edison M. Achelengwa

Submitted to the

Department of Electrical Engineering and Computer Science

May 23, 2011

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The MIT iLab Project has developed online laboratories (iLabs) which are lab stations that can be accessed and controlled remotely over the Internet. With iLabs, students can conduct real experiments on real equipment over the Internet. With the introduction of the National Instrument's Educational Laboratory Virtual Instrument Suite, NI ELVIS, in the development of iLabs, students to gain a better understanding of engineering concepts by obtaining real data from electronic labs. One of such crucial engineering concepts is telecommunications which plays a key role in transmitting information between people, systems and computers. There are many telecommunication schemes which exist today. The iLab developed in this thesis implements an experiment for studying one of such schemes, Amplitude Modulation. The NI ELVIS is used together with a device called the Emona Digital and Analog Telecommunications Experimenter (DATEX) to achieve the Amplitude Modulation lab setup. This iLab is an Interactive iLab, which gives one student at a time complete, real-time control over the lab set up. The Amplitude Modulation iLab will permit students to tune various controls and observe the behavior and changes of relevant signals, both in time domain and frequency domain. It will also permit students to compare different signals and retrieve data locally for post processing.

Thesis Supervisor: Jesús del Alamo

Title: Professor of Electrical Engineering

DEDICATION

I dedicate this thesis

To my late father,
Capt. Chuh Henry Achelengwa,
who taught me the values of discipline,
discipline, hardwork and being God-fearing
at a tender age

And my late uncle,
Charles Tangie,
who guided me through seven years of
secondary school and high school

Acknowledgements

I would like to acknowledge several people who have made it possible for me to see this project through. I am grateful to my supervisor, Professor Jesús del Alamo and Jud Harward for the opportunity to undertake this project and make a significant impact in education in Africa as well. I am also thankful to Jim Hardison for helping me to define my project goals, project steps, setting up weekly progress report meetings, giving valuable feedback, assisting me with proposals, outlines and submissions.

I would like to thank other colleagues at MIT's Center for Educational Computing Initiative for the valuable contributions. I thank Kirky DeLong and Philip Bailey for helping me solve many technical and administrative problems which were beyond my scope. I am grateful to Meg Westlund for organizing so many aspects of my iLabs experience and to Maria Karatzas for making sure I had the funds to work and travel for this project.

My gratitude also goes to Olayemi Oyebode, who first introduced me to the opportunities the iLab Project had to offer, who was a valuable mentor in helping me interact with the iLab partners in East Africa.

I will like to extend my appreciation to the African iLab partners. I thank Dr. Alfred Mwambela, Josiah Nombo, Baraka Maiseli and the entire iLabs team at the University of Dar es Salaam; Cosmas Mwikirize, Arthur Asiimwe, Nicholas Mpanga and the entire iLabs team at Makerere University for their hospitality and sharing with me their iLab project ideas.

On a more personal note, I will also like to extend my appreciation to my church, Pentecostal Tabernacle which has been like a home-away-from-home for me. I am so appreciative of the Senior Pastor, Bishop Brian C. Greene and Elder Roy Ray who have been so welcoming to me. I would also like to thank Pastor Herman Greene and Mother Marcia Greene who provided me both moral and spiritual support whenever I needed it.

I will also like to thank my fraternity, Tau Chapter of Delta Psi, Number Six Club. I am so grateful for the wonderful experiences, the company and the friends.

Finally, I will like to extend my appreciation to my family and family friends. I would like to thank my mother, Chuh Susana Tangie Awafor, who has been with me every step of the way from the time my father passed away through secondary school, applying to MIT and applying to the M.Eng program till the completion of this M.Eng thesis. I am so appreciative of her words of encouragement and support. I would like to also thank the Akwo family, especially Auntie Margaret Akwo, who first introduced me to MIT. I thank Uncle Francis Ngwandi and family as well Uncle Francis Ndifang and family for their kindness and hospitality.

Contents

1	Introduction	17
1.1	The iLab Project	18
1.2	The iLab-Africa Initiative	19
1.3	iLabs on the ELVIS	20
1.4	Thesis Overview	21
2	Overview of ISA Architecture	23
2.1	ISA Architecture	23
2.1.1	Lab Server	25
2.1.2	Lab Client	25
2.1.3	Service Broker	26
2.1.4	Scheduling Services	27
2.1.5	Experiment Storage Service	28
2.2	Summary	29
3	Implementation of AM System	31
3.1	Overview of Emona DATEx	31
3.2	Amplitude Modulation	33
3.3	Amplitude Demodulation	35

3.4	LabVIEW Control	37
3.5	Oscilloscope and Dynamic Signal Analyzer	38
3.6	Two-channel Oscilloscope	41
3.7	NI Switch	42
3.8	Summary	44
4	User Interface: Front Panel	45
4.1	Front Panel Tabbing	46
4.2	AM Experiment Tab	47
4.2.1	Block Diagram	48
4.2.2	Graph Time Domain	48
4.2.3	Frequency Domain (Power Spectrum)	49
4.2.4	Probing Multiple Signals	51
4.3	Publishing VI on ISA as iLab	53
4.3.1	LabVIEW Set-up and Configuration	53
4.3.2	Interactive Lab Server and Bootstrapping	54
4.4	Running the iLab	55
4.5	Test Results	57
4.5.1	Testing Probing of Signals	57
4.5.2	Testing Digital Input Options	61
4.6	Summary	63
5	Future Work and Conclusion	65
5.1	Future Work	65
5.2	Conclusion	67
A	Amplitude Modulation and Demodulation Review	69

A.1 Amplitude Modulation Review	70
A.2 Amplitude Demodulation Review	72
Bibliography	76

List of Figures

1.1	NI ELVIS II+	20
2.1	Batched iLab Architecture	24
2.2	Interactive iLab Architecture	24
2.3	ELVIS 5.0 Lab Client	26
3.1	Block Diagram Example - Amplitude Modulation System	32
3.2	Emona Front Panel (Top Face)	33
3.3	Emona DATEx workstation with NI ELVIS	33
3.4	DATEx block diagram of Amplitude Modulation	34
3.5	”Patch Cord” Implementation of Amplitude Modulation	34
3.6	DATEx block diagram of Amplitude Demodulation	36
3.7	DATEx block diagram of Amplitude Demodulation	36
3.8	AM iLab Core Functionality	38
3.9	Cases for Time Domain Plot Rendering	40
3.10	Cases for Frequency Domain Plot Rendering	41
3.11	NI Switch Components	43
3.12	NI Switch	44
4.1	Tabs on User Interface (Front Panel)	47

4.2	Main Experiment Tab	47
4.3	User Interface Block Diagram	48
4.4	Time Domain Graph Indicator	49
4.5	Power Spectrum Graph Indicator	51
4.6	Selector For Probing Signals	52
4.7	Service Broker Login	55
4.8	AM iLab Embedded Front Panel on Lab Server Page	56
4.9	For Running AM iLab	56
4.10	User probes $s(t)$ on Channel 0 and $m(t)$ on Channel 1	58
4.11	User probes $c(t)$ on Channel 0 and $r(t)$ on Channel 1	59
4.12	Still Probing $c(t)$ and $r(t)$ - Adjusted x-axis Scales	59
4.13	User probes $r'(t)$ using Ch0 and $b(t)$ using Ch1 of Oscilloscope	60
4.14	Still probing $r'(t)$ and $b(t)$ - Viewing Peaks in vicinity of DC	60
4.15	User probes $m'(t)$ using Ch0 and $s'(t)$ using Ch1 of Oscilloscope	61
4.16	User probes $s'(t)$ using Ch0 and $-Ga*s'(t)$ using Ch1 of Oscilloscope	61
4.17	Testing 2-kHz Digital Input	62
4.18	Testing 8-kHz Digital Input	63
A.1	Amplitude Modulation Blockdiagram	70
A.2	Time and Frequency Domain plots of Amplitude Modulation Signals	71
A.3	Envelop Detector	73
A.4	Time and Frequency Domain plots of Amplitude Demodulation Signals	74

List of Tables

3.1	Relevant Signals For Amplitude Modulation	35
3.2	Relevant Signals For Amplitude Demodulation	36
4.1	Test Settings of Components	57

Chapter 1

Introduction

Any form of education in an engineering or science discipline is incomplete without a means of testing and appreciating theories learned in class. The ability to carry out experimentation demonstrating theories through laboratory work is an integral part of an engineering, science and technology education. In laboratories, students can learn how to process real data, understand and appreciate discrepancies between their observations and the predictions according to theories. Not only do students appreciate those discrepancies, they learn how to make compromises to minimize the imperfections of their observations. This is a valuable skill for an engineer to have as engineers are problem solvers.

Unfortunately, sometimes giving a traditional lab experience to students is costly, logistically hard and in some cases, impractical. For example, most of the introductory classes at MIT do not provide lab components for all its topics due to the large size of the class population. Also certain schools around the world may not be able to afford the necessary equipments for the traditional lab experience.

Even traditional laboratories have their own shortcomings. Setting up the lab is time consuming and costly. Once the lab is setup, it is only open during stipulated times to protect

the equipments from theft or damage by students. The equipment needs to be maintained. Furthermore, the equipment may be under-used due to limited lab hours.

All of the above issues with traditional laboratories and the risks in damaging the expensive equipment were all motivations for the iLab Project.

1.1 The iLab Project

iLabs are remote laboratory stations developed at MIT in order to address the shortcomings of conventional laboratories. These experimental setups can be accessed and controlled remotely through the Internet, allowing students and educators to carry out experiments from anywhere at any time [1, 2]. Users are able to access the laboratory experiments 24/7. Also students get to bypass the hurdle of setting up the equipments and hence that difficulty does not take their attention away from the actual experience and from the main point of a given experiment.

The iLab Project was started at MIT in 1998 by Prof. Jesús del Alamo, a professor at the Microsystems and Technology Labs (MTL) at MIT. Prof. Jesús del Alamo, acquired the Agilent 4155B semiconductor parameter analyzer, which he wanted to make available to his students. Unfortunately the equipment was very expensive and there was the risk of it being damaged if multiple students used it. On the other hand, the equipment was under-used and there was a need to give the class a lab component. This gave birth to the Microelectronics Device Characterization iLab (also called Microelectronics Weblab) which consisted of a web-based client which could acquire data from a server connected to the Agilent 4155B. Since then several other remote experiments have been developed in various disciplines. These include the Heat Exchanger and Polymer Crystallization in chemical engineering, a Shake Table in civil engineering, a Dynamic Signal Analyzer in electrical engineering and Force on

Dipole Experiment in Physics [3].

Unlike conventional laboratories, iLabs greatly reduces the risk of misusing a sensitive and expensive piece of equipment while increasing the availability of that same equipment. It reduces maintenance costs financially and in terms of time. iLabs can be shared and accessed widely by students around the world. While anyone can access the laboratories, only a few institutions need to maintain the equipment.

As a scalable and easily accessible type of laboratory experience, MIT iLabs have been used widely in courses at over 18 universities in Europe, Asia, North America, Australia and Africa[4]. The iLab Project have sparked interest in certain African universities, specifically in sub-Saharan Africa.

1.2 The iLab-Africa Initiative

iLabs is most beneficial to students of developing countries where reliable equipments are not easy to get and classes tend to be large. This is the case for students in African universities. Mutual interest between MIT iLabs and certain professors at these African universities have led to formal partnerships with Obafemi Awolowo University (OAU) in Ile-Ife, Nigeria, the University of Dar Es Salaam (UDSM) in Tanzania and Makerere University (MAK) Kampala, Uganda [5]. The purpose of the iLab-Africa initiative is to explore the benefits and challenges of establishing iLabs in sub-Saharan Africa. Such efforts could be extended to other developing countries. The iLab-Africa program has helped OAU, UDSM and MAK to include laboratory experiences in their technical course curricula. These universities often lack the state-of-the-art equipment and infrastructure resources available to institutions like MIT. With iLabs, they can overcome these limitations by saving on purchasing costs - taking advantage of economies of scale that MIT iLabs provides or by accessing iLabs at other uni-

versities. In the early days of the initiative, students from these universities accessed iLabs developed by MIT. Now, the universities have their own development teams creating iLab experiments that are better suited for their individual curricula.

1.3 iLabs on the ELVIS

MIT together with the three sub-Saharan Universities - OAU, UDSM and MAK have developed a wide variety of electronic laboratories. These laboratories are usually developed using National Instrument's (NI) Educational Laboratory Virtual Instrument Suite (Figure 1.1).

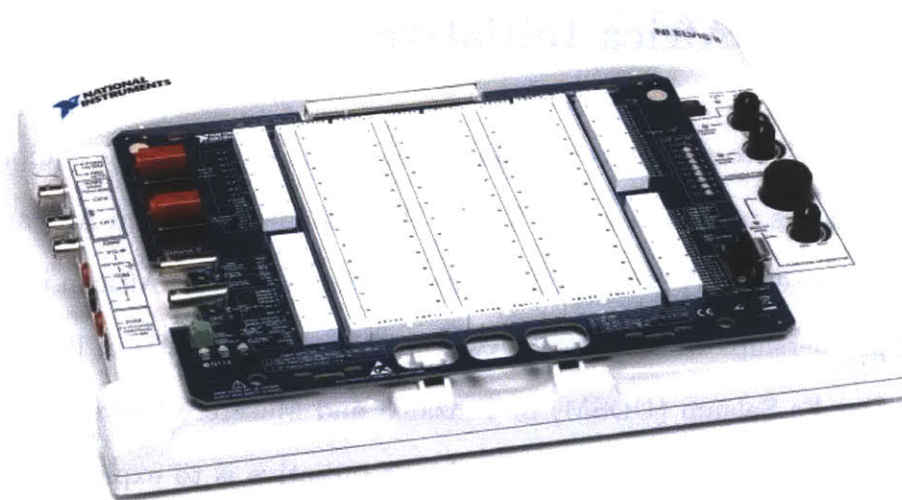


Figure 1.1: NI ELVIS II+

The NI ELVIS is a small-footprint, all-in-one electronics workstation that performs the functions of instruments which are typically found in an electrical engineering laboratory. It contains 12 instruments which include an Oscilloscope, Digital Multimeter, Variable Power Supply, Bode Analyzer, Dynamic Signal Analyzer and Arbitrary Waveform Generator[6, 7].

Its variety of instruments and versatility has made it very suitable for the development of a wide range of electronics experiments both at MIT and in Africa.

The NI ELVIS II⁺ connects to a computer using a USB 2.0 high speed cable. The NI ELVIS can be programmatically accessed and controlled within NI's LabVIEW (Laboratory Virtual Instrumentation Workbench), a graphical development environment containing and integrating a multitude of hardware devices and libraries. Previous versions of ELVIS-based iLabs improved the student learning by providing access to more of the ELVIS's instruments varied experiments.

The NI ELVIS comes with a removable prototyping board on which developers can construct electrical circuits which connect the different ELVIS's instruments to achieve a particular experimental goal. In some cases, third-party add-on modules can be attached to the NI ELVIS to create new types of experiments. Such modules include the Emona HELEx for Hydrogen Fuel cell experiments, Emona Fiber Optics Telecommunications Experimenter (FOTEx) and the Emona's Digital and Analog Telecommunications Experimenter (DATEx)[8]. For each of these add-ons, Emona provides drivers which are compatible with the LabVIEW development environment. This allows such third-party add-ons, such as the Emona add-ons, to be programmatically controlled in the same development environment as the NI ELVIS. The Emona DATEx is of interest in this thesis because of the relative ease with which it can be used to set up a variety of telecommunication experiments which will otherwise be difficult and expensive to set up in a traditional lab.

1.4 Thesis Overview

This thesis will detail the development of the Interactive Amplitude Modulation Experiment using the Emona DATEx and NI ELVIS in the LabVIEW environment. To establish a bit of

a background, Chapter 2 will examine the iLab Shared Architecture (ISA) and its different components.

Chapter 3 will talk about the implementation of the Amplitude Modulation and Demodulation system on the Emona DATEx and all backend functionalities. First, it will examine the Emona DATEx in detail, giving a description of all the implemented sub-modules on the Emona DATEx. Then it will mention all the components necessary for a desktop implementation of the AM Experiment. Then the LabVIEW implementation of the controls will be shown as well. This chapter will also raise the different issues encountered in implementing this experiment, detail the proposed solutions for addressing these issues and expose the functionalities that will enhance the student's experience.

Chapter 4 will talk about the user interface implementations of the backend functionality. Different features such as the tabbing of the user interface and the buttons and the block diagram will be examined in detail. The different components for enhancing the user's experience such as cursors, buttons, indicators and other design choices will also be explained. After discussing the implementation of the user interface, the chapter will then discuss how to publish the Amplitude Modulation LabVIEW application on the ISA as an iLab. Chapter 4 will then discuss and display some test results to ascertain the correct functioning of the iLab.

Finally, Chapter 5 will conclude by detailing what is next for the Amplitude Modulation iLab and what useful features can be added. It will also discuss how the efforts of Amplitude Modulation iLab can be extended to other telecommunication experiments.

Chapter 2

Overview of ISA Architecture

Initial versions of Microelectronics WebLab and other online laboratories only consisted of the Lab Client and the Lab Server. Due to their success and the desire to develop more such labs, there was a need for a more standard approach to developing online laboratories in a cost-effective way and also easy to maintain. The iLab Project was created as a result of these needs. This led to the infrastructure called the iLab Shared Architecture (ISA)[9].

2.1 ISA Architecture

The iLab Shared Architecture (ISA) is a toolkit of reusable modules and a set of standardized protocols to facilitate the rapid development and effective management of iLabs. The ISA is designed to be highly scalable giving the lab provider minimum work possible when it comes to new iLab development. The ISA is also decentralized, secure, open to wide variety of web service specifications and compatible to commercial software such as NI LabVIEW [9, 10].

The ISA is capable of supporting two types of iLabs, Batched iLabs and Interactive iLabs.

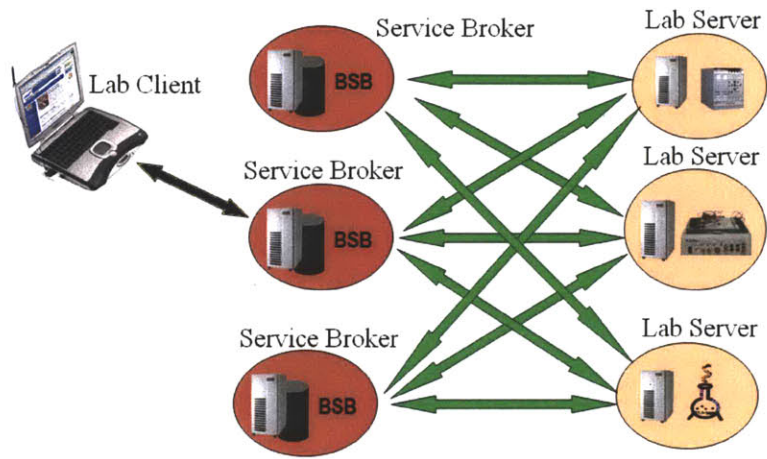


Figure 2.1: Batched iLab Architecture

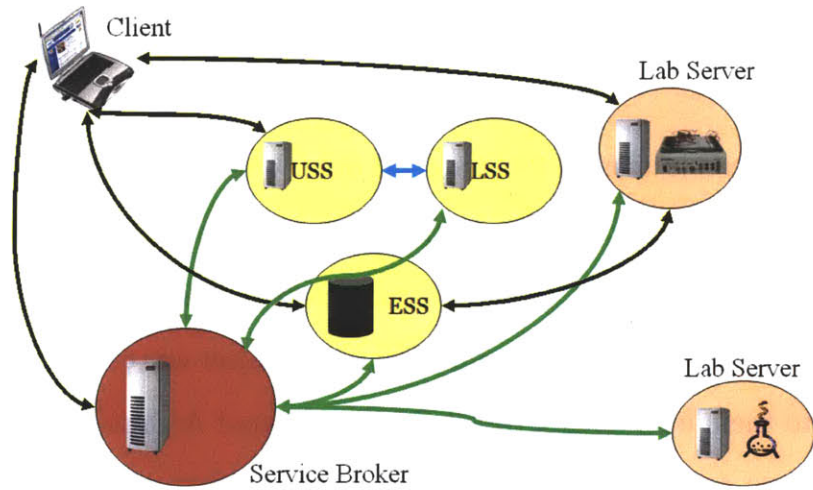


Figure 2.2: Interactive iLab Architecture

Figures 2.1 and 2.2 shows the architectures for Batched iLabs and Interactive iLabs. The first version of the ISA supported only Batched iLabs. In Batched iLabs, students completely configure the experiment before running it and get the results after the experiment has finished execution. Batched iLab experiments typically take on the order of 10-100 seconds. This allows multiple users to virtually access a Batched iLab simultaneously as each user's configuration is queued and ran in a First-In-First-Out order.

Interactive iLabs, on the other hand, permit the user to configure the experiment while it is running and observing the results. Consequently the user needs more time to "interact"

with the iLab and hence only one user at a time can perform experiments in an Interactive iLab. Interactive iLabs are more relevant in this thesis since the Amplitude Modulation iLab will be an Interactive iLab.

Batched iLabs and Interactive iLabs differ from each other in many aspects. One of these aspects is what components each iLab needs and how these components interact with each other in a student's iLab experience. In order to better appreciate how these categories of iLabs differ, it is only logical to examine in detail what each of these ISA components are and what their functions are. This is done in the next subsections.

2.1.1 Lab Server

The Lab Server hosts the laboratory hardware and controls the execution of the experiment and acquisition of data from the lab set-up. It communicates with the laboratory hardware to both perform experiments and retrieve results. The Lab Server also includes a website that is the interface through which lab administrators create, modify and delete experiments as well as configure the connection settings to Service Brokers. Finally, the Lab Server has databases that store experimental results for retrieval at later times for post processing.

2.1.2 Lab Client

The Lab Client is a web based user interface through which a student configures an experiment and runs it. Through the user interface, the user sends the commands to the server for running the experiment, retrieves data and sends the data back to the user. Lab Clients are developed using different technologies. The Microelectronics Device Characterization iLabs uses Java-based clients. In general, Batched iLabs such as the ELVIS iLab 1.0 - 5.0 [11, 12, 13, 14, 15] series of iLabs are developed using Java-based clients. Figure (2.3) shows

what an ELVIS iLab 5.0 client looks like. Java-based clients provide flexibility but are more difficult to implement. Interactive iLabs, such as the Amplitude Modulation iLab, are developed using LabVIEW-based Lab Clients. The design of this interface will be examined shortly.

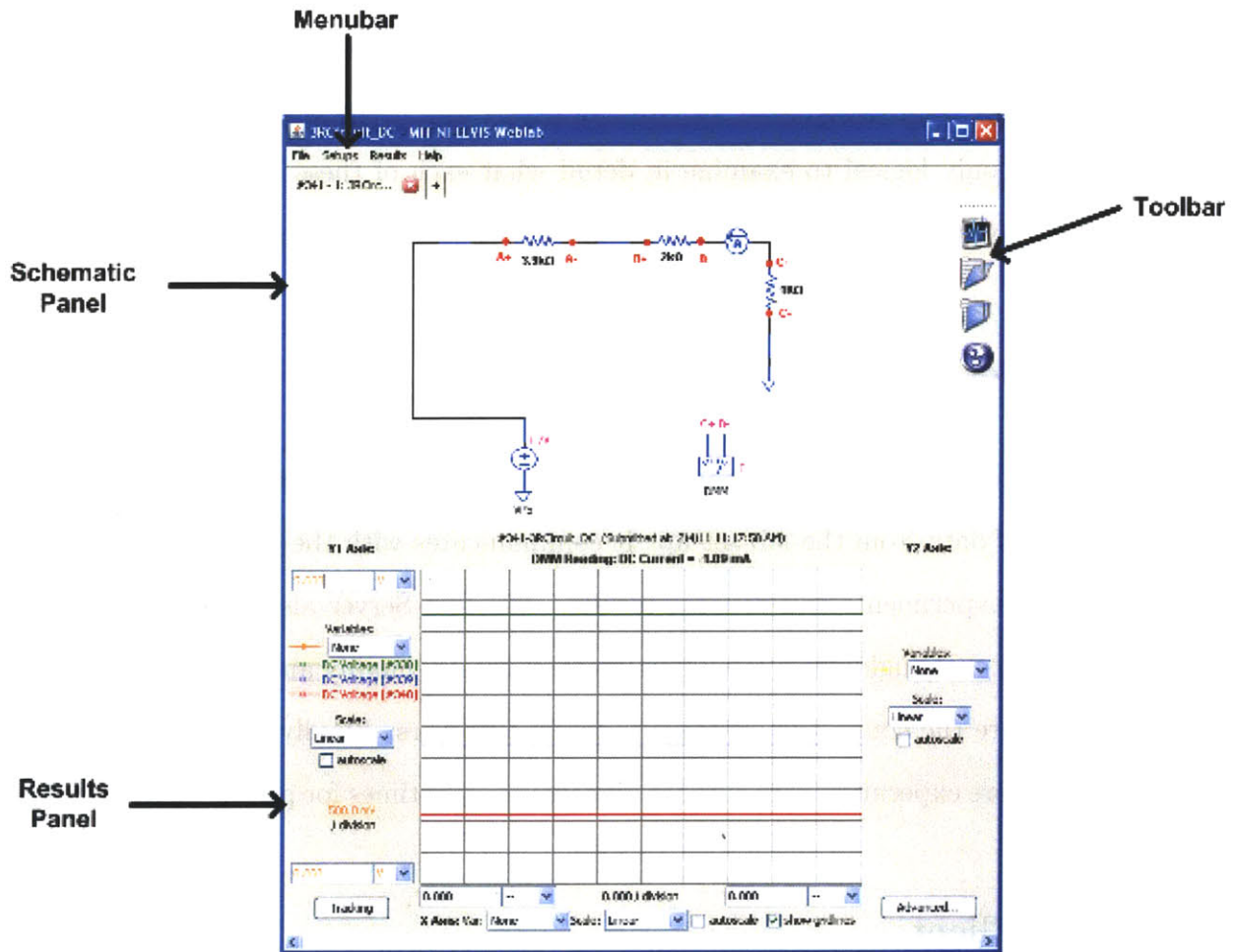


Figure 2.3: ELVIS 5.0 Lab Client

2.1.3 Service Broker

Unlike the Lab Client and Lab Server, the Service Broker only came about with the creation of the iLabs Shared Architecture. The Service Broker provides functionality that is generic

for all iLabs such as authentication, authorization and data storage. Deploying an iLab application involves configuring the application by sharing credentials between the Service Broker and the Lab Server. The Service Broker is also responsible for user and administrator group management. A Service Broker communicates with multiple servers and multiple Lab Clients. As observed in Figure 2.1, in a Batched iLab Architecture, the Service Broker acts as a mediator between the Lab Client and the Lab Server. Consequently one Service Broker communicates with multiple Lab Servers and multiple Lab Clients.

So far we have examined three components - Lab Server, Lab Client and Service Broker. The next ISA components which will be examined are those which are more specific to Interactive iLabs.

2.1.4 Scheduling Services

Interactive iLabs permit a user to have complete control over the lab set-up and only one user should have access to an iLab at any given time in order for the user to fully appreciate the experience. Hence users need to schedule time to reserve the lab set-up for their use. This leads to the need for Scheduling Services. The ISA makes use of two types of scheduling services, Lab Side Scheduling and User Side Scheduling. As the names of these services may imply, User Side Scheduling is what the user uses while Lab Side Scheduling sets times when the actual iLab is available for usage. These two services work in conjunction with each other.

User Side Scheduling (USS)

The Service Broker is a critical agent in the scheduling process as it authenticates a user and determines which labs a user is permitted to launch. Once a user is authorized, the

User Side Scheduling (USS) service is invoked. It is through the USS that the user actually schedules the time slot to have direct access with the Interactive Lab. The user can select from a set of available time blocks available to the group he/she belongs to. Once a user selects a time, the USS stores that reservation and the user can come to the Service Broker at his/her scheduled time-slot and redeem that reservation using the USS. At this point, the user can launch the Interactive iLab. The USS is also responsible for notifying users about how much time they have left on their reservation and preventing the user from overstaying their reservation.

Lab Side Scheduling

The USS is used in conjunction with Lab Side Scheduling (LSS). The LSS is responsible for defining scheduling policies for a particular iLab. These policies define maximum possible time and least possible time a user can stay on an iLab experiment. While the USS permits users to reserve time slots from available time blocks, the LSS defines these available time blocks. The Service Broker can only permit a user to launch an iLab if its policies say that it is available at the time the user wants to schedule the iLab. The LSS can be used to schedule multiple Lab Servers. The Lab Administrator is generally the person concerned with the LSS because he/she can estimate how much time a user needs to spend on an iLab. The Lab Administrator can use the LSS to schedule non-overlapping time blocks or time spans for these multiple classes.

2.1.5 Experiment Storage Service

In Batched iLabs, the Service Broker does all the data storage. This eliminates the need for direct interaction between Lab Server and Lab Client. Yet the very nature of Interactive

iLabs makes direct interaction between the Lab Client and Lab Server inevitable. Consequently the Service Broker can no longer act as a mediator between the Lab Client and Lab Server in an Interactive iLab experience. Instead, the Service Broker sets up a relationship between the Lab Client and Lab Server. The Service Broker is no longer holding experiment parameters, commands or results. This brings up the need to diverge the data storage for the interactive experiments to another service. This service is the Experiment Storage Service (ESS).

The ESS is a stand-alone web service that allows Service Brokers, Lab Servers and Lab Clients to store experiment data. As an independent system, a single ESS can potentially be used by many Service Brokers and their associated labs.

Records on the ESS consist of experiment data. Such data include binary data (images, video or audio) and XML based text/numeric data. Administrative data describing the student who owns the data and when it was collected are stored on the Service Broker. Students can use the Service Broker in conjunction with the ESS to retrieve past experimental data for post processing.

2.2 Summary

The ISA toolkit designed to facilitate the development of iLabs has six components: Lab Client, Lab Server, Service Broker, User Side Scheduling, Lab Side Scheduling and Experiment Storage Services. While Batched iLabs only make use of the Lab Client, Service Broker and Lab Server, Interactive iLabs make use of all these components. Interactive iLabs are of interests in this thesis as the Amplitude Modulation iLab will be interactive in nature. Because of the ease of development which the ISA toolkit provides, the only components which need redevelopment for any new Interactive iLab are the Lab Server and the Lab Client.

The next chapters will examine in detail the implementation of the Amplitude Modulation application on the Lab Server side as well as the configuration of the Lab Server and the user interface design.

Chapter 3

Implementation of AM System

This chapter starts of the description of the implementation of the Amplitude Modulation iLab on the Lab Server side. The NI ELVIS was already described in Section 1.3. This chapter will kick off with a background on the Emona DATEx. It then describes the system block diagrams of Amplitude Modulation and Demodulation. Then it will examine the Amplitude Modulation application on LabVIEW, the different challenges which came with implementing the application and the proposed solutions.

3.1 Overview of Emona DATEx

In Section 1.3 it was mentioned that NI's ELVIS has a removable prototyping board which can be replaced by third party add-on modules. The Emona DATEx is one of those add-on modules.

The Emona DATEx, Digital Analog Telecommunications Experimenter, is an add-on which could serve in offering basic training in various concepts of telecommunications and communications [16]. This add-on is fully integrated with the NI ELVIS platform and is controllable

through the NI LabVIEW development environment. All DATEx knobs and switches are manually controllable or programmable from LabVIEW.

By using the DATEx add-on, students are able to construct block diagrams of various communication systems. A "block diagram" is a simple high-level representation of a more complex circuit. A good example of an Amplitude Modulation block diagram is given in Figure 3.1.

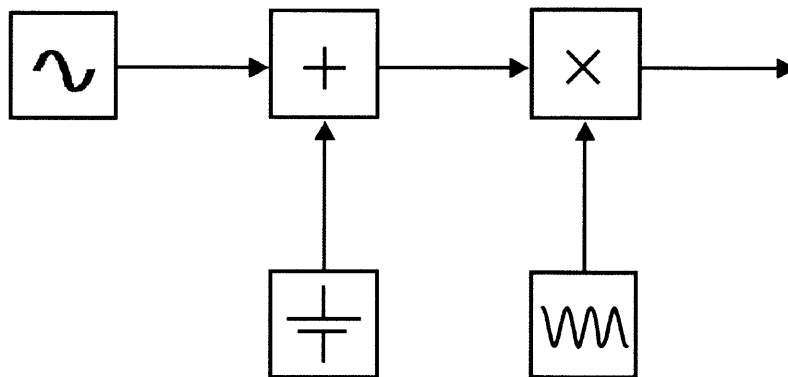


Figure 3.1: Block Diagram Example - Amplitude Modulation System

The Emona implements common blocks such as Adders, Amplifiers, Multipliers, Low Pass Filters, Band Pass Filters, Phase Shifters, etc. Figure 3.2 shows the top face of the Emona DATEx and the different blocks which it implements. As observed, due to the integration with the NI ELVIS, some of the I/O jacks are derived from NI ELVIS components such as the Function Generator and Variable Power Supply. Hence the NI ELVIS is used in conjunction with the Emona DATEx, which is crucial for the Amplitude Modulation iLab. Figure 3.3 shows the Emona DATEx as connected to the NI ELVIS.

The Emona DATEx was originally developed in the 1970s by Tim Hooper and further developed by Emona Instruments. It is used by thousands of students around the world to implement pretty much any form of modulation or coding found in a telecommunication protocol.

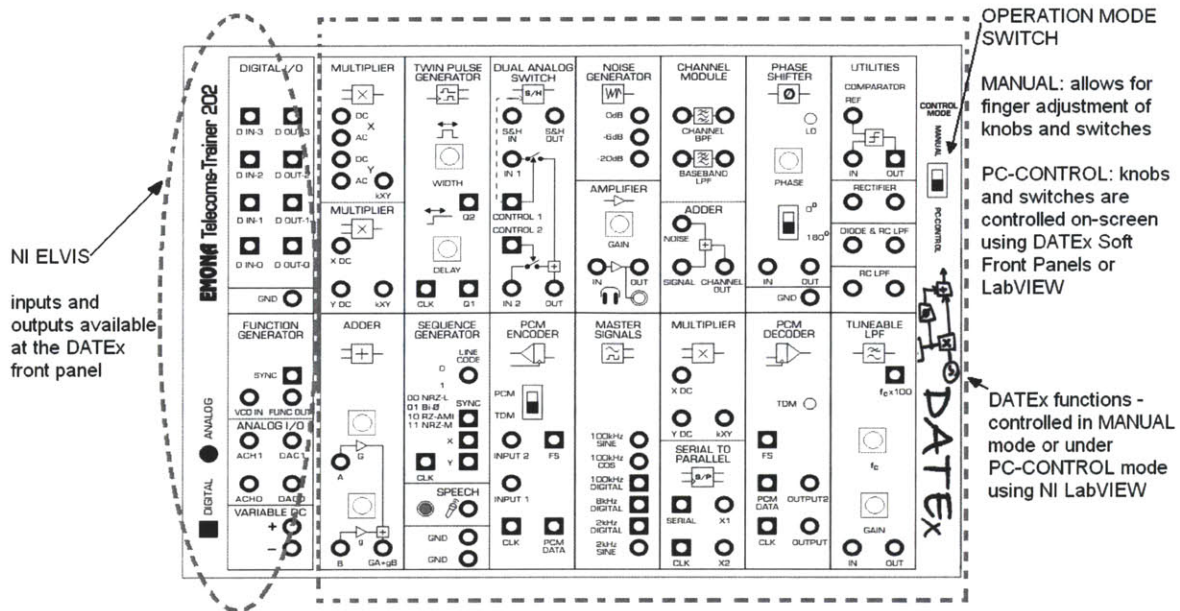


Figure 3.2: Emona Front Panel (Top Face)

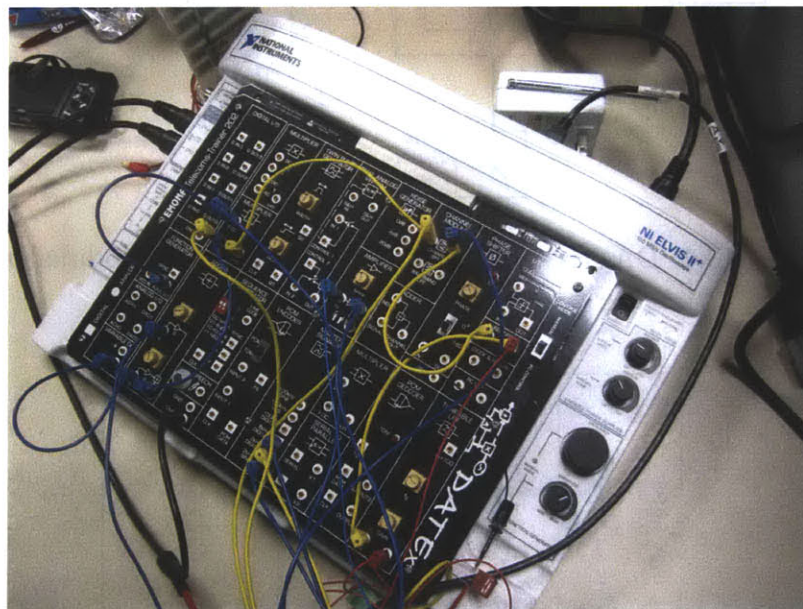


Figure 3.3: Emona DATEx workstation with NI ELVIS

3.2 Amplitude Modulation

To implement the Amplitude Modulation iLab on the DATEx, one of the considerations was to give the students as much flexibility as possible. Hence, as many tunable compo-

nents as possible were used for the modulation block diagram. Figures 3.4 and 3.5 shows the block diagram and the patch cord implementation of the modulation part of the experiment respectively [17]. Patch cords are the mini-banana-to-mini-banana connectors used to connect relevant I/O ports on the DATEx to implement a particular block diagram (for more on modulation system as implemented on the Emona DATEx, see appendix section A.1). Tunable components in the used in the modulation system are the Adder, the Variable Power Supply and the Function Generator. Table 3.1 gives a summary of all signals in the modulation part of the experiment. These are the signal names that will be seen by the user all through the experiment.

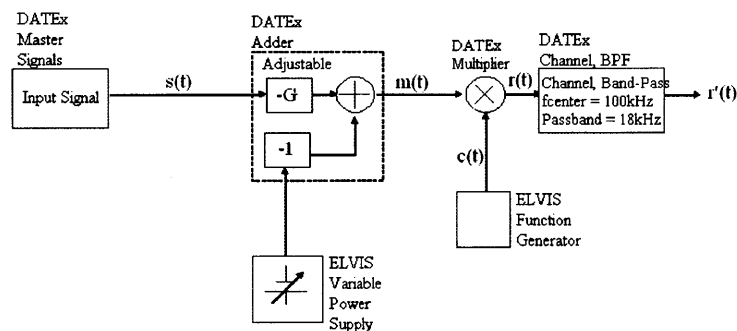


Figure 3.4: DATEx block diagram of Amplitude Modulation

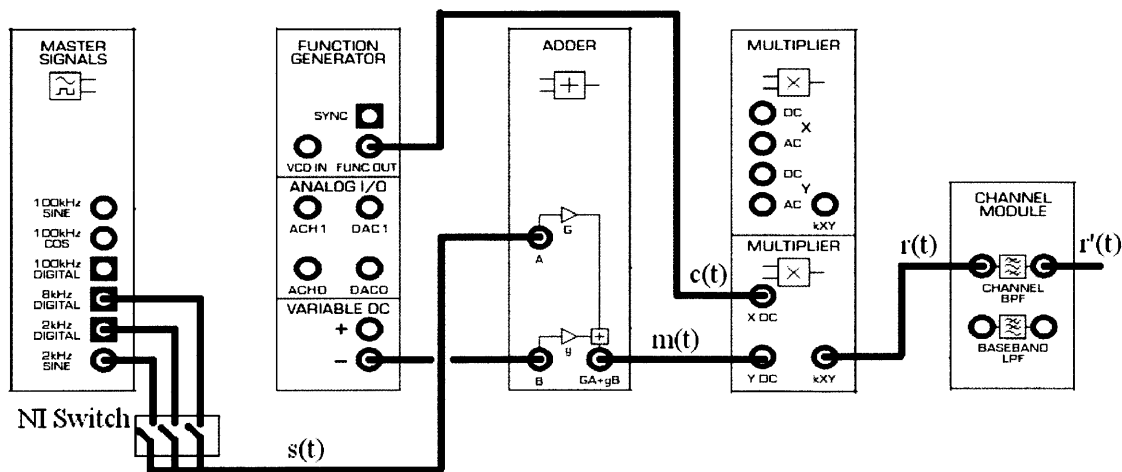


Figure 3.5: "Patch Cord" Implementation of Amplitude Modulation

First the input signal $s(t)$ is given a DC offset by passing it through the tunable DATEx Adder. The DC offset comes from the NI ELVIS Variable Power Supply. The resulting signal is the message signal, $m(t)$. Then the message signal $m(t)$ is used to modulate another sinusoid called the carrier signal, $c(t)$. The modulation process is a multiplication of the message $m(t)$ and carrier signal $c(t)$ using the DATEx Multiplier. The carrier, $c(t)$ is derived from the NI ELVIS Function Generator. The resulting modulated signal, $r(t)$, is transmitted through a medium which is modeled on the DATEx hardware using a Band Pass Filter on the Channel Module.

Signal Name	Description
$s(t)$	Input signal (baseband)
$m(t)$	Input signal with DC offset (message signal)
$c(t)$	Carrier signal
$r(t)$	Modulated signal
$r'(t)$	Recieved (Filtered) modulated signal

Table 3.1: Relevant Signals For Amplitude Modulation

3.3 Amplitude Demodulation

Just like in the case of modulation, in order to enhance the user's experience, as many tunable components are used to create the demodulation portion of the experiment. Figures 3.6 and 3.7 shows the block diagram and the patch cord implementation of the demodulation part of the experiment respectively[17]. For demodulation, an Envelope Detector is used which consists of a rectifier and a Low Pass Filter (for more information on demodulation system as implemented on the Emona DATEx, see appendix section A.2). For demodulation, tunable components are the Low Pass Filter and the Amplifier. Table 3.2 gives a summary of all relevant signals in the demodulation part of the experiment. These are the signal names that will be seen by the user all through the experiment.

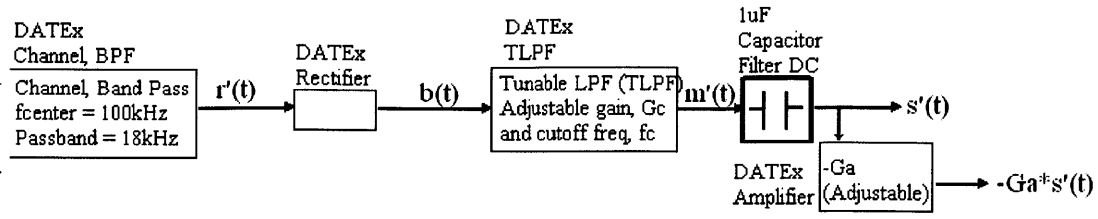


Figure 3.6: DATEx block diagram of Amplitude Demodulation

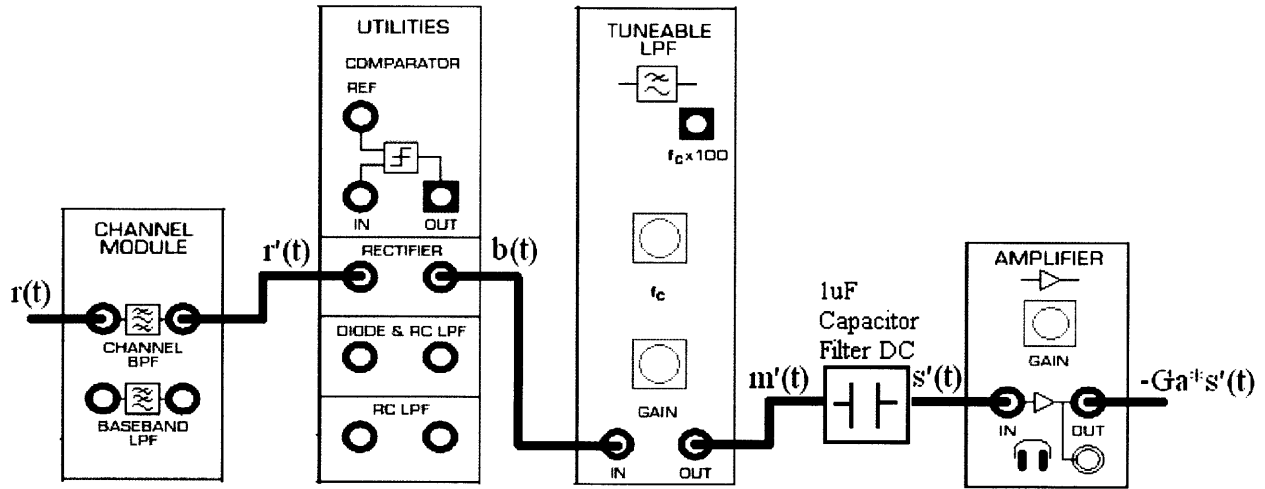


Figure 3.7: DATEx block diagram of Amplitude Demodulation

The modulated signal, after transmission through Band Pass Filter becomes $r'(t)$. Then it is rectified using one of the DATEx rectifiers, producing $b(t)$. Then $b(t)$ is sent through a DATEx Tunable Low Pass Filter producing $m'(t)$ which is the output message with a DC offset. Then the DC component of $m'(t)$ is filtered using a capacitor producing the output signal $s'(t)$. In order to better observe that output, it is sent to the DATEx Tunable Amplifier producing $-Ga*s'(t)$.

Signal Name	Description
$r'(t)$	recieved (filtered) modulated signal
$b(t)$	rectified recieved signal
$m'(t)$	output message with DC offset
$s'(t)$	output signal (AC component of output message)
$-Ga*s'(t)$	Amplified output signal

Table 3.2: Relevant Signals For Amplitude Demodulation

3.4 LabVIEW Control

Now that we have examined the hardware implementation of the Amplitude Modulation/De-modulation system, we need to examine the software that will control the system. The Emona DATEx is fully integrated with the NI ELVIS and can be accessed and controlled through LabVIEW.

LabVIEW is a development environment created by National Instruments as a means of building instrument-control programs using a graphical, data-flow centric language. Programs created in LabVIEW are referred to as Virtual Instruments or VIs. Sometimes, VIs can be used to control real lab hardware such as the Emona DATEx and the NI ELVIS. Every LabVIEW application has two main parts - a Front Panel and a block diagram. It is important not to be confused between the block diagram of a communication system and a LabVIEW block diagram. Since LabVIEW gives a graphical, data-flow representation of variables and parameters and VIs then the backend of a LabVIEW application is called a "block diagram". In order to avoid confusion, "block diagram" will be used for a telecommunication system block diagram and "LabVIEW block diagram" will be used for the backend block diagram of a LabVIEW application.

We need to be able to control each of the tunable blocks in the AM system. For the NI ELVIS components, NI provides Express VIs (high level control VIs). Emona provides a polymorphic DATEx VI which can be configured appropriately for all the DATEx tunable blocks.

Figure 3.8 shows the core functionality of the LabVIEW application (VI) which controls the iLab. It shows the NI ELVIS Express VIs for the Function Generator, Variable Power Supply and the Oscilloscope. It also shows the three instances of the Emona Polymorphic VIs configured as Adder, Tunable LPF and Amplifier.

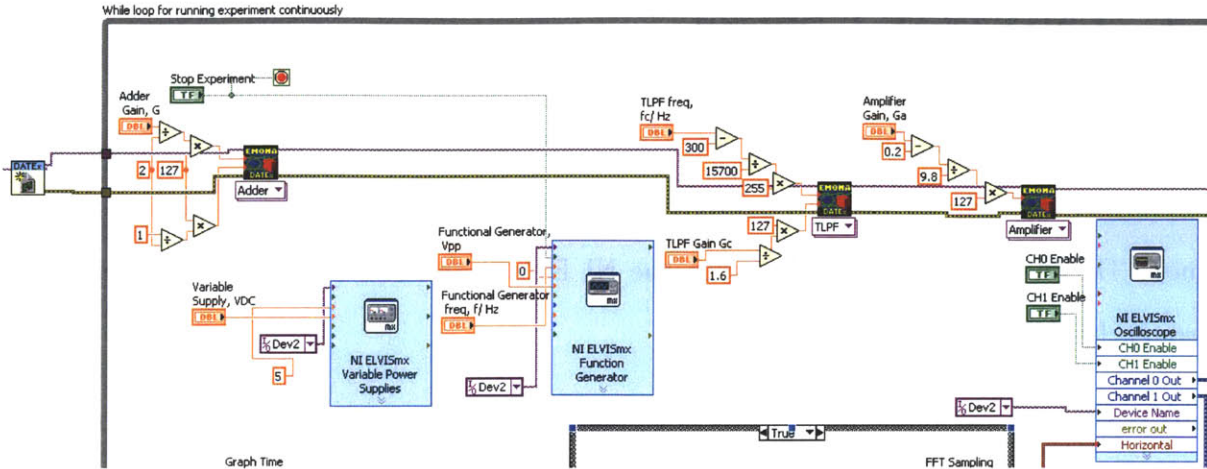


Figure 3.8: AM iLab Core Functionality

The Amplitude Modulation VI has a main "while" loop which runs continuously until the user presses the "Stop" button on the user interface. Almost all of the functionality of the backend LabVIEW block diagram is found within this "while" loop. From now on the "while" loop will be referred to as the AM main loop.

3.5 Oscilloscope and Dynamic Signal Analyzer

In communication or telecommunication experiments, it is quite valuable to see how a signal varies in time and also to see the frequency content of a signal. Frequency domain plots of signals are valuable to explain why certain protocols are used for modulation and demodulation and why some times a Low Pass Filter or a High Pass Filter is used. The NI ELVIS has both an Oscilloscope and Dynamic Signal Analyzer (DSA). Unlike the Oscilloscope, which displays the time domain waveform data, the DSA is capable of acquiring the Data and representing frequency domain characteristics of signals.

Unfortunately, the Oscilloscope and the DSA cannot run at the same time because they use the same resources in the NI ELVIS hardware. Furthermore, the NI ELVIS driver does not

provide an Express VI for the DSA in the LabVIEW environment the way it does for the Oscilloscope. Consequently we are left with the Oscilloscope which is capable of collecting raw data in Time Domain. Yet it will be of great benefit to the student's experience to be able to see both the time domain and frequency domain plots.

One way around this problem is to use the Oscilloscope as the data acquiring component and processing these data within LabVIEW to produce the frequency domain plots. This can be done using one of the numerous FFT (Fast Fourier Transform) Power Spectrum VIs provided by LabVIEW.

Despite finding the above solution there remained another problem. The way graphs work in LabVIEW, if the graph displayed time domain data, the time span of the graph will automatically adjust itself to the time span to which the data corresponds i.e. the time span = Sampling Period \times Number of Samples. For frequency domain plots, the graphs automatically adjusts its frequency range from negative of half the Sampling Rate to half the Sampling Rate with the resolution of the points (frequency step length) being the reciprocal of the time span in the time domain. This complicates the graph displays in that certain signals as seen in the time domain will be well captured at a particular sampling frequency and number of samples but when the FFT is done on that same data the sampling rate could be too small and the signal could be miss-represented with peaks showing up at aliases. Consequently the VI was designed such that, for every iteration of the AM main loop either the time domain plot is being rendered or the frequency domain plot is being rendered. A control variable is being toggled to make sure that the alternation happens between time domain and frequency domain plot rendering. It is worth noting that this alternation happens fast enough for the user not to realize that it is taking place. In this way, a user is free to control settings to see time domain plots and control other settings for the frequency domain plot and the Oscilloscope is alternatively configured with those

settings.

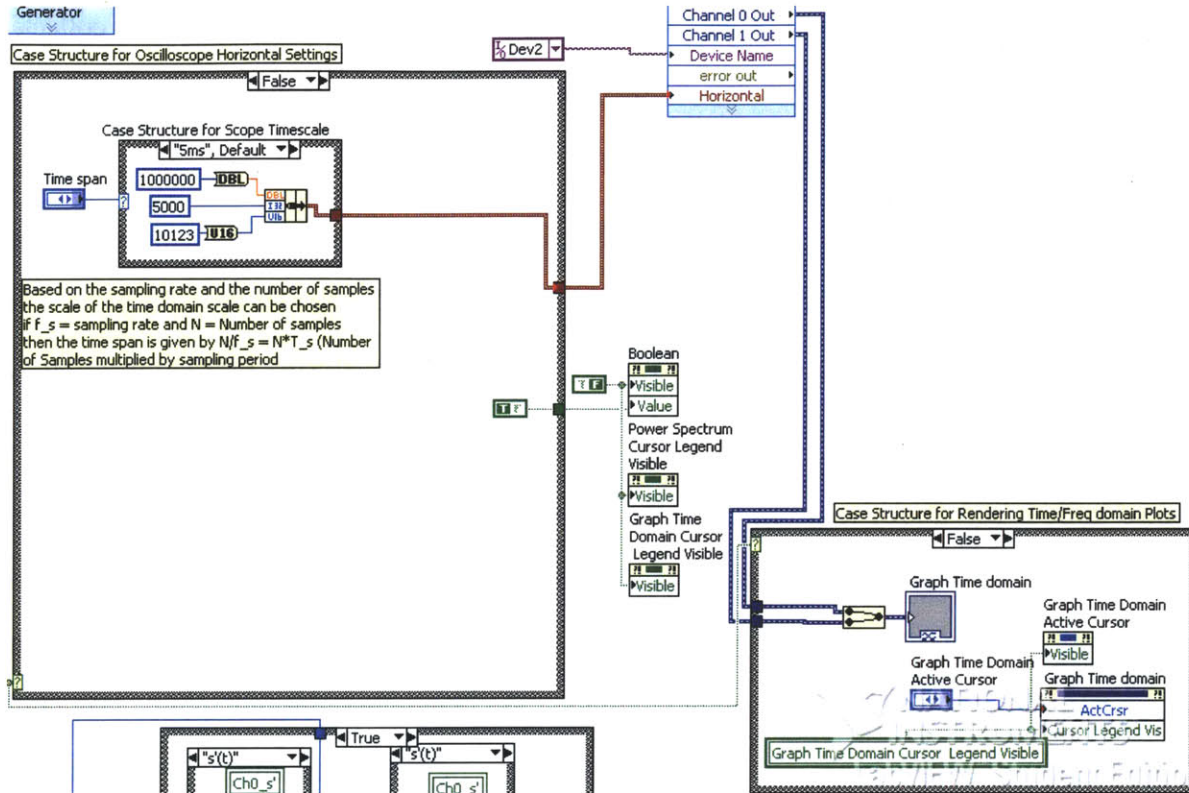


Figure 3.9: Cases for Time Domain Plot Rendering

Figure 3.9 shows the cases (within case structures) for setting Oscilloscope for time domain rendering and actual Graph Time Domain rendering. Figure 3.10 shows the cases for setting the Oscilloscope for the frequency domain rendering and actual Power Spectrum rendering. It is worth noting that before rendering the power spectrum the data is passed through FFT Power Spectrum VIs which calculates the FFT and then sends the data to the Power Spectrum plot.

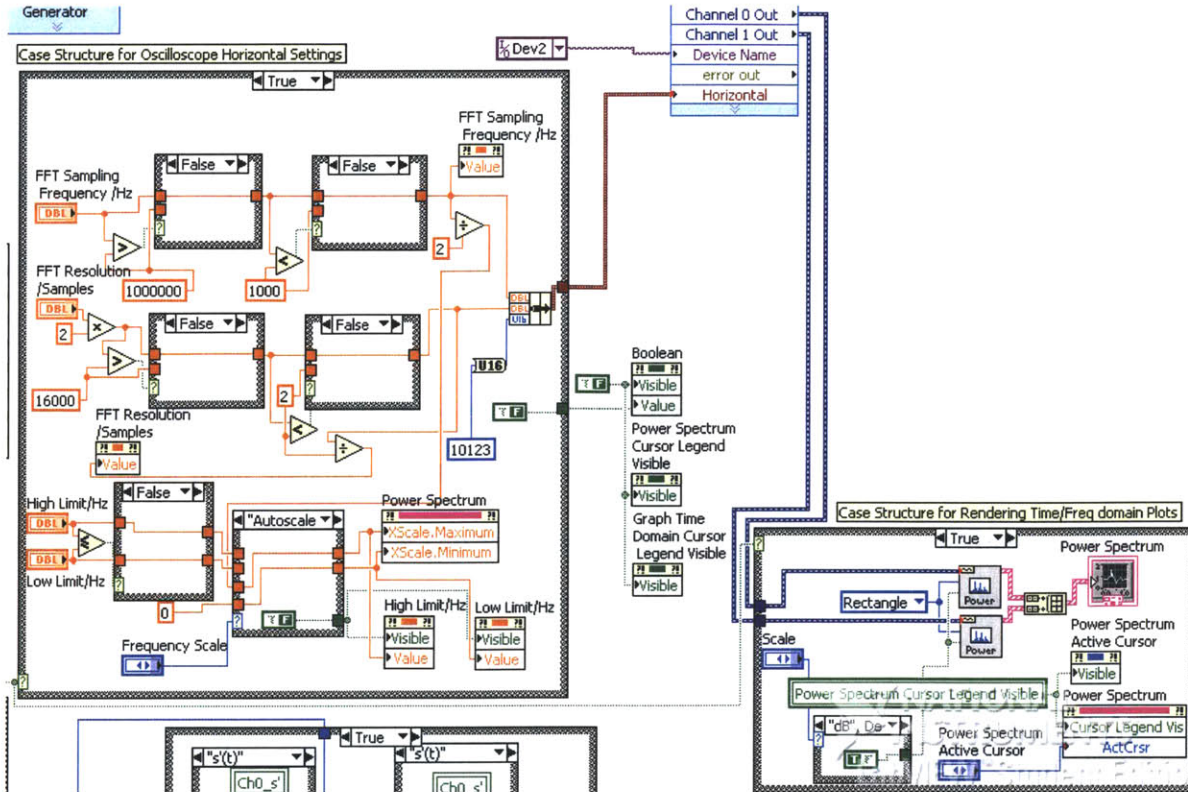


Figure 3.10: Cases for Frequency Domain Plot Rendering

3.6 Two-channel Oscilloscope

Looking at tables 3.1 and 3.2, there are nine relevant signals in the whole system. In order to enhance the user's experience it is important that he/she be able to probe all these signals in a flexible manner. Unfortunately, the very nature of iLabs makes physical probing of different signals impossible from the user's side. Also the Oscilloscope has only two channels that can only collect and display data. Therefore, only two signals can be studied. Hence there is a need to for another device that will assist the Oscilloscope in viewing all the nine signals. One possible device which can assist the Oscilloscope is an NI Switch.

3.7 NI Switch

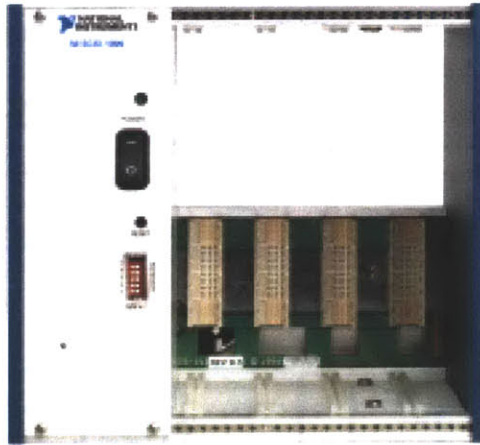
A National Instrument's Switch consists of two types of components - the chassis and the switch modules. The chassis contains the circuitry for powering the switch modules, interfacing with the computer, fanning the NI Switch system and it contains slots which can take up to 4 switch modules.

Switch modules are add-on boards which have hundreds of mechanical relays and can be inserted into the chassis. [18, 19]. For this particular iLab, the chassis used is an NI SCXI-1000 (Signal Conditioning eXtensions for Instrumentation) while one switch module is used, the NI SCXI-1169. Figure 3.11 shows the SCXI-1000 chassis and an SCXI-1169 switch module.

The SCXI-1169 is equipped with up to 100 SPST (Single-Pole-Single-Throw) mechanical relay switches. Like the NI ELVIS, the NI Switch comes packaged with its driver called NI-SWITCH. The NI-SWITCH driver is compatible with LabVIEW, allowing individual relays on the switch module to be controlled programmatically. Consequently, using the NI Switch, we can create a switch implementation which, based on the user's choice, can tell the each channel of the Oscilloscope to chose between all 9 relevant signals of the system.

Now that we have introduced the NI Switch hardware, we should examine how it is controlled in LabVIEW. Figure 3.12 shows switch control mechanism on the LabVIEW block diagram.

There are two controls called "Choose Ch0 signal" and "Choose Ch1 signal", which tell the Amplitude Modulation VI to set the switches appropriately so that the Oscilloscope acquires the signals chosen by the user. When the user first starts the Amplitude Modulation experiment, the switches are initialized accordingly before the AM main loop in Figure 3.12(b). Local variables whose values are the same as the "Choose Ch0 signal" and "Choose Ch1



(a) NI Switch Chassis SCXI-1000

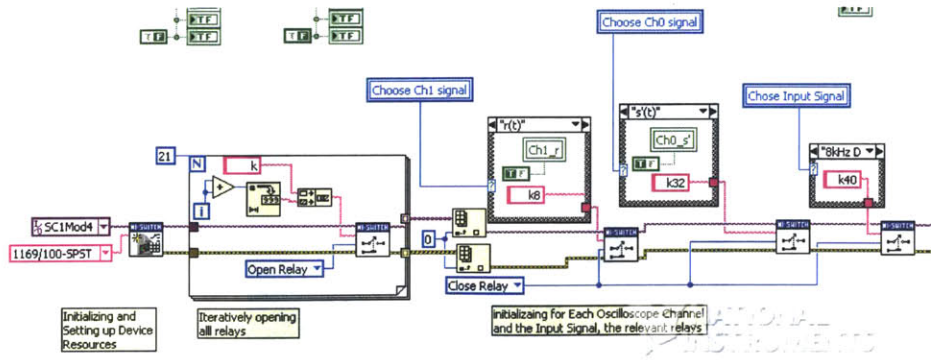


(b) NI Switch Module SCXI-1169

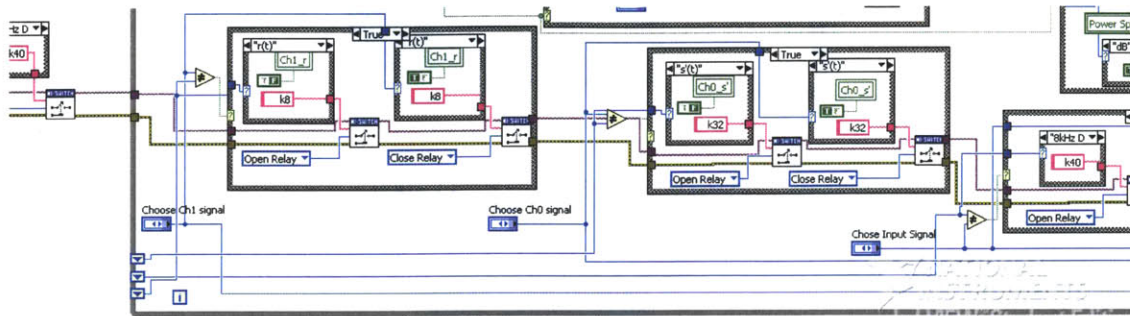
Figure 3.11: NI Switch Components

signal” controls are used for this initialization. Then after the initialization, the AM main loop runs until the user stops the experiment. Within this loop, the ”Choose Ch0 signal” and ”Choose Ch1 signal” are used accordingly to change the state of the switches. Shift registers are used in conjunction with the channel-choosing controls to make sure switches are only affected when the user changes the value of the channel-choosing controls.

Another careful look at Figure 3.12 reveals a third control which interacts with the NI Switch hardware called ”Choose Input Signal”. With this additional functionality, the user is free



(a) NI Switch Initialization



(b) NI Switch control within "While" loop

Figure 3.12: NI Switch

to supply the Amplitude Modulation system with a 2-kHz Sinusoid, a 2-kHz Square Wave or an 8-kHz Square Wave. These signals are all produced by the DATEx Master Signals block. This added functionality explains the "NI Switch" in 3.5.

3.8 Summary

We have examined the desktop implementation of the Amplitude Modulation system using the Emona DATEx add-on to the NI ELVIS-II. We have also examined the different issues of functionality such as probing multiple signals with the Oscilloscope and displaying both time domain and frequency domain data. We examined the core functionality as seen in the LabVIEW block diagram. The next chapter examines the user interface.

Chapter 4

User Interface: Front Panel

The previous chapter was focused on explaining the functionality of the Amplitude Modulation iLab application as implemented within the LabVIEW block diagram. As mentioned in section 3.4, each LabVIEW application has a backend LabVIEW block diagram and a Front Panel. The Front Panel has a lot of virtual control elements that represent common components such as knobs, levers, switches and buttons. It also has indicators which display results such as graphs or LED lights. In Interactive iLabs based on LabVIEW, the Front Panel is the user interface. LabVIEW has a feature called a Remote Panel which permits a Front Panel to be embedded on a web page and the whole application can be accessed and controlled remotely. This chapter will explore how the different AM iLab functions are represented in the Front Panel along with additional features created on the Front Panel to enhance the user's experience with the iLab.

4.1 Front Panel Tabbing

The functionality of this iLab is represented within a *tab container*. LabVIEW provides all sorts of components necessary for building a user interface. There is a class of these components called containers. As the name implies, a container can hold other components and carry those components around. One of such containers is a tab container. Tab containers can have multiple tabs, each with its content. Below is the list of tabs found in this iLab and their description:-

1. **Introduction and Objectives:** As the name of this tab implies, it gives a short introduction to the Amplitude Modulation iLab and lists the different contents. This page also explains to the students the aim of the iLab and what the student stands to gain from his/her experience with the iLab.
2. **AM Experiment:** This is the most important page of the AM iLab interface as the page contains the AM iLab client. Every control on the backend LabVIEW block diagram and indicators and graphs have UI component representations on this page. The bulk of the rest of this chapter will be a detailed examination of this tab.
3. **Math Theory:** There are currently three of these tabs which cover a short review of what Amplitude Modulation and Demodulation is. It features graphs in both time domain and frequency domain which should refresh the user's memory.
4. **How To Use AM iLab:** There are currently four tabs which cover this topic and serve as a user guide on how to use the AM iLab.

Figure 4.1 shows a snapshot of the tab container and headings of the different tabs.

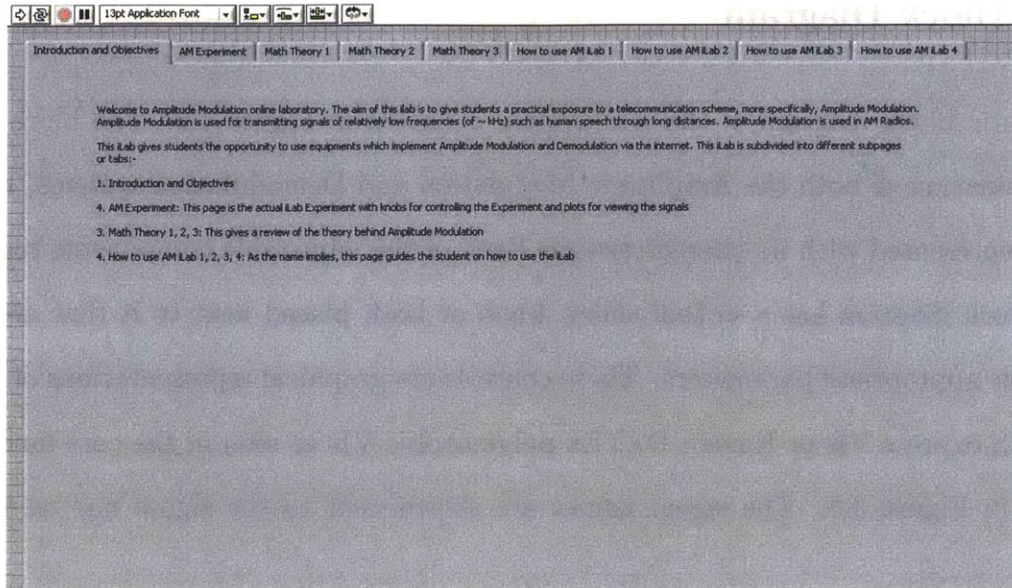


Figure 4.1: Tabs on User Interface (Front Panel)

4.2 AM Experiment Tab

The AM Experiment Tab is the most important tab on the Front Panel. Figure 4.2 shows a snapshot of the AM Experiment Tab.

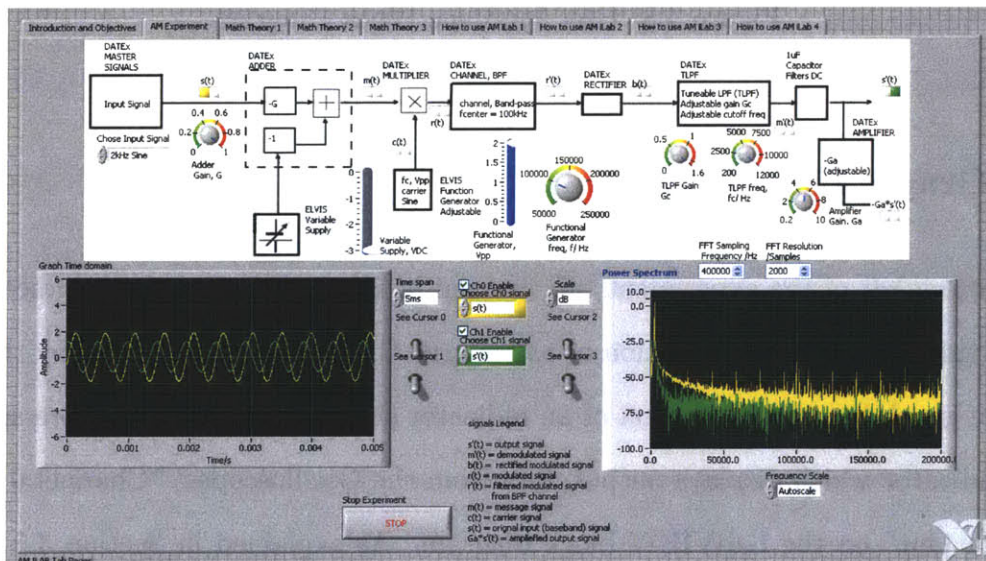


Figure 4.2: Main Experiment Tab

4.2.1 Block Diagram

A schematic block diagram of the AM system is shown in Figure 4.2. The block diagram is a combination of both the Amplitude Modulation and Demodulation systems, with each module represented with its characteristics. Each of the adjustable components represented on the block diagram has a virtual slider, knob or both placed next to it that are used to control the appropriate parameters. These controls are graphical representations of inputs to the ELVIS express VIs or Emona DATEx polymorphic VIs as seen in the core functionality diagram in Figure 3.8. The signal names are shown next to the signal line on the block diagram.

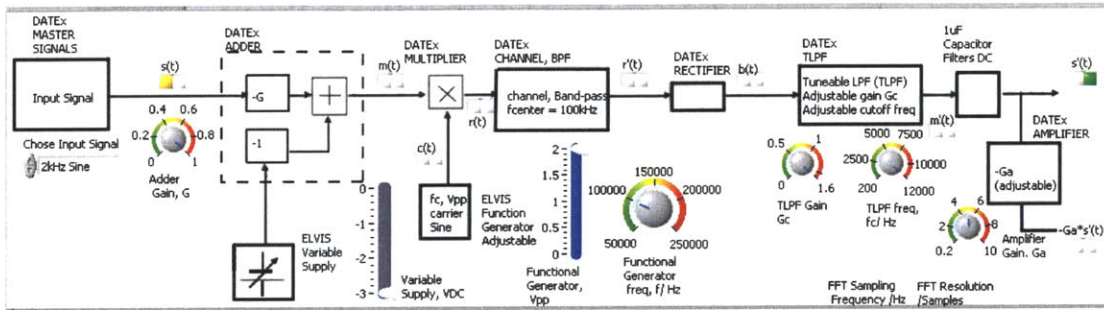


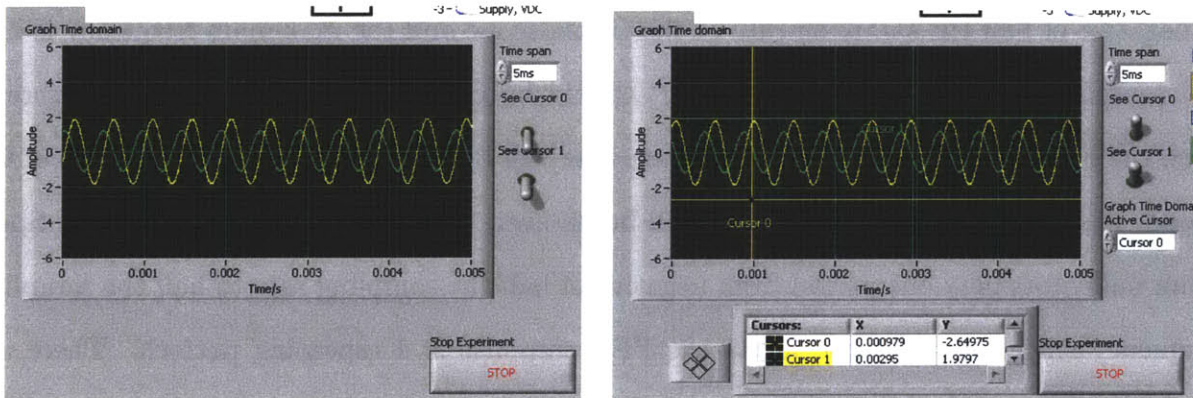
Figure 4.3: User Interface Block Diagram

4.2.2 Graph Time Domain

On the bottom left of the experiment tab, as seen in Figure 4.2, is a graph indicator which shows the time domain representation of two signals that are being observed by the Oscilloscope. The graph is a representation of an indicator shown in Figure 3.9. More specifically, it is the indicator which displays output data from the Oscilloscopes' "Channel 0 Out" and "Channel 1 Out" on the LabVIEW block diagram. As explained in Section 3.5 there was an issue of representing Time Domain and Frequency Domain plots at the same time. Consequently, the Graph Time Domain indicator is refreshed once per two iterations of the AM

main loop in the LabVIEW block diagram. This indicator is set such that the vertical axis scale is fixed from -6V to 6V. The horizontal axis on the other hand is automatic and its span can be selected using the "Time span" selector on the top right of the graph indicator. This selector is used to control a case structure in the block diagram shown in Figure 3.9. The Time span is given by $\text{Sampling Period} \times \text{Number of Samples}$.

Finally, there are two virtual levers on the right of the graph, labeled "See Cursor0" and "See Cursor1" which, as the names imply, can be used to switch on/off two cursors and if at least one cursor is visible, the cursor legend will become visible. Figure 4.4 shows the time domain graph when the cursor is not visible and when the cursors are visible. The cursors can be used to read numerical values from the output waveforms. In addition to the visible cursor legend, there is another cursor selector which the user can use to select which cursor should be active.



(a) Cursors off

(b) Cursors on

Figure 4.4: Time Domain Graph Indicator

4.2.3 Frequency Domain (Power Spectrum)

Due to the necessity of observing both time domain and frequency domain, there is a need for a second graph to display the Power Spectrum of the relevant signals. As in the case

of the time domain graph, the Power Spectrum graph is a graphical representation of an indicator shown in Figure 3.10 of the LabVIEW block diagram. It receives outputs from two Power Spectrum VIs which process data from the Oscilloscope. Just like the time domain graph, this graph is also refreshed every two iterations of the main program loop (this loop alternates between refreshing time domain and frequency domain data).

The vertical scale of the Power Spectrum graph, shown on the bottom right of Figure 4.2 represents the magnitude of the Power Spectra. The y-axis scale can either be linear or in dB. There is a selector on the right which serves to switch between these two settings. The scale range is always automatic based on minimum and maximum waveform magnitude.

For the horizontal (frequency) scale, there are a few controls to examine. There is a selector on the bottom of the indicator called Frequency Scale which can be used to make the frequency scale automatic or manually settable. When it is automatic, the minimum frequency is negative of half the sampling rate of the Oscilloscope and the maximum frequency is half the sampling rate of the Oscilloscope. It should be noted that since the data is discrete, then the FFT (Fast Fourier Transform) response will be periodic at the sampling rate of the scope. Also since the signal is real, it will be an even function hence the FFT Power Spectrum indicators only one period from negative of half the sampling rate to half the sampling frequency since beyond these limits the FFT response is a repeating pattern. There are two controls on the top of the Graph indicator called "FFT Sampling Rate/Hz" and "FFT Resolution/Samples" respectively. The former sets the sampling rate and, consequently, the maximum displayable frequency while the latter sets how dense the graph will be in terms of number of points for the corresponding frequency span.

When the scale is set to manual, two additional controls appear called "Low Limit/Hz" and "High Limit/Hz" on the left and right of the "Frequency Scale" selector respectively. These can be used to manually zoom in or out of the graph plot. Since the ELVIS is a physical

piece of hardware, it has its limits such as maximum possible samples and maximum possible sampling rate. There is logic within the case structure of Figure 3.10 which addresses these limits and prevents the Oscilloscope VI from giving an error.

Finally, just like the Time Domain Indicator, the Power Spectrum has its own cursors and the corresponding levers "See Cursor2" and "See Cursor3" are on the left of the Power Spectrum indicator. These have the same functionality as the levers for the cursors of the time domain indicator. Figure 4.5 shows the Power Spectrum on automatic frequency scale with cursors off and on manual frequency scale with cursors all on.

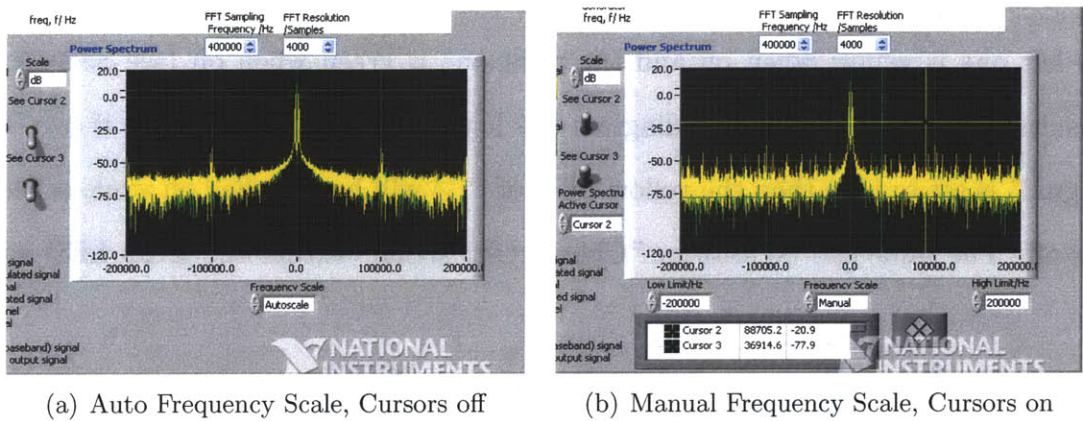


Figure 4.5: Power Spectrum Graph Indicator

4.2.4 Probing Multiple Signals

As explained in Section 3.7, the Oscilloscope is only capable of acquiring and displaying data from at most two channels but the AM system has nine relevant signals which can be observed. Hence the NI Switch was introduced to assist the Oscilloscope in viewing these signals. The selectors for these features can be seen between the Graph Time Domain and Power Spectrum graph indicators. These selectors were already introduced in Section 3.7. They are called "Choose Ch0 signal" and "Choose Ch1 signal". They are represented on

the Front Panel as drop down selectors with options being the signal names as seen on the schematic block diagram on the Front Panel. Figure 4.6 shows the selectors. As observed from Figure 4.6, Channel 0 selector is placed on a yellow LED while the Channel 1 selector is placed on a green LED. This is intuitive because in the two graphs, Ch0 signals are yellow in color while Ch1 signals are green in color. By selecting a signal using the "Choose Ch0 signal" or "Choose Ch1 signal" we open and close switches so that the Oscilloscope Ch0 or Ch1 can be connected accordingly.

For a more intuitive touch, pairs of LED indicators are placed next to the signal names on the block diagram from Figure 4.3. For each signal, both LEDs are white when off but when on the left one is yellow when on and the right one is green when on (see Section 4.5 on test results). The LabVIEW block diagram has logic which makes sure that if say "Choose Ch0 signal" is set to $r'(t)$ then the yellow LED of $r'(t)$ will be switched on while other LEDs remain off. The same can be said for "Choose Ch1 signal" except the green LED will turn on.

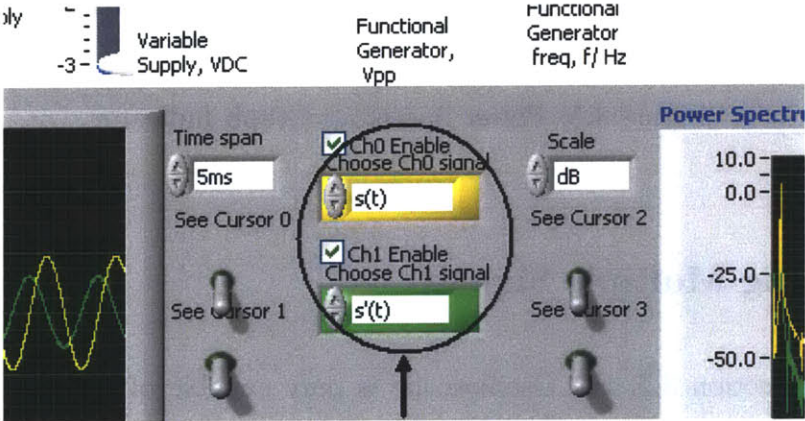


Figure 4.6: Selector For Probing Signals

In addition to the Oscilloscope signal selectors, there is a "Choose Input Signal" selector on the block diagram of Figure 4.3 under the input representation. This drop-down selector permits the input to the AM system to be one of three choices "2kHz Sine", "2kHz Digital"

or "8kHz Digital." As the names of the options imply, the user can either set the input of the AM system to be a 2-kHz sinusoid, 2-kHz Square Wave or an 8-kHz Square Wave. An explanation of this is given in Section 3.7. Once again this choice is made possible by the NI Switch.

Finally, there is a signal legend between the two graphs, as seen in Figure 4.2, which gives the user a brief description of each signal represented on the schematic block diagram.

4.3 Publishing VI on ISA as iLab

The ISA toolkit makes it quite easy to upload VIs on the ISA as iLabs. Now that the Amplitude Modulation VI has been created with all its functionality, the next step is to configure the LabVIEW application and publish it on the ISA as an iLab. To get to this process we are assuming that we have a working Service Broker, working USS, LSS and working ESS. The process of publishing the VI is not unique to the AM iLab VI but is generic for all Interactive iLab applications.

4.3.1 LabVIEW Set-up and Configuration

Before the VI application is published on the ISA, LabVIEW needs to be configured appropriately. Two ports for HTTP are required and by default, port 80 is for the .Net web services and port 81 is for the LabVIEW Web Server. The LabVIEW Web Server is configured by going to LabVIEW >> Tools >> Options enabling the Web Server, setting the port to 81 and enabling Remote Panel access, web services and other necessary accesses. Also under "Options", the VI Server is configured appropriately by enabling TCP/IP, ActiveX and allowing server resources. The default port, 3363, for the VI Server should be

maintained.

Furthermore the LabVIEWDCOM needs to be configured. The location of the LabVIEW execution file and the security settings have to be set accordingly. Other things which need to be added are the DataSocket server storing data, Remote Panel Manager for having a working Remote Panel. For more detailed step by step information see documentation at <https://wikis.mit.edu/confluence/display/ILAB2/iLab+Downloads> online, more specifically, the "iLabs InteractiveLabServer with LabVIEW Setup and Configuration" under the "Documentation" sub-topic [20].

4.3.2 Interactive Lab Server and Bootstrapping

Now that LabVIEW has been configured accordingly, it is important to set up the Interactive Lab Server. The ISA toolkit has both source files and a pre-built distribution of the Interactive Lab Server. This process involves setting up certain key credentials on the Lab Server such as the name of the Lab Server and TCP/IP port used. These credentials will be given to the Service Broker to help it locate the Lab Server when the user is trying to run the Interactive iLab.

After the Lab Server has been configured, its configuration credentials are shared with Service Broker. At this point, the Lab Side Scheduling policies are configured using the LSS and the sharing credentials between the Lab Server and the ESS, USS and LSS and setting administrators. This process by which the individual Interactive ISA components are configured and the credentials are shared with each other is called Bootstrapping and it is generic to the development of all Interactive iLabs. For more information on this see documentation on <https://wikis.mit.edu/confluence/display/ILAB2/iLab+Downloads> online, more specifically, the "Merged iLabs Bootstrapping Guide" under the "Documentation"

sub-topic[21].

With this, the AM iLab is published on the ISA and can be utilized.

4.4 Running the iLab

To launch the iLab, the user needs to go the Service Broker and log in. Figure 4.7 shows Service Broker Login

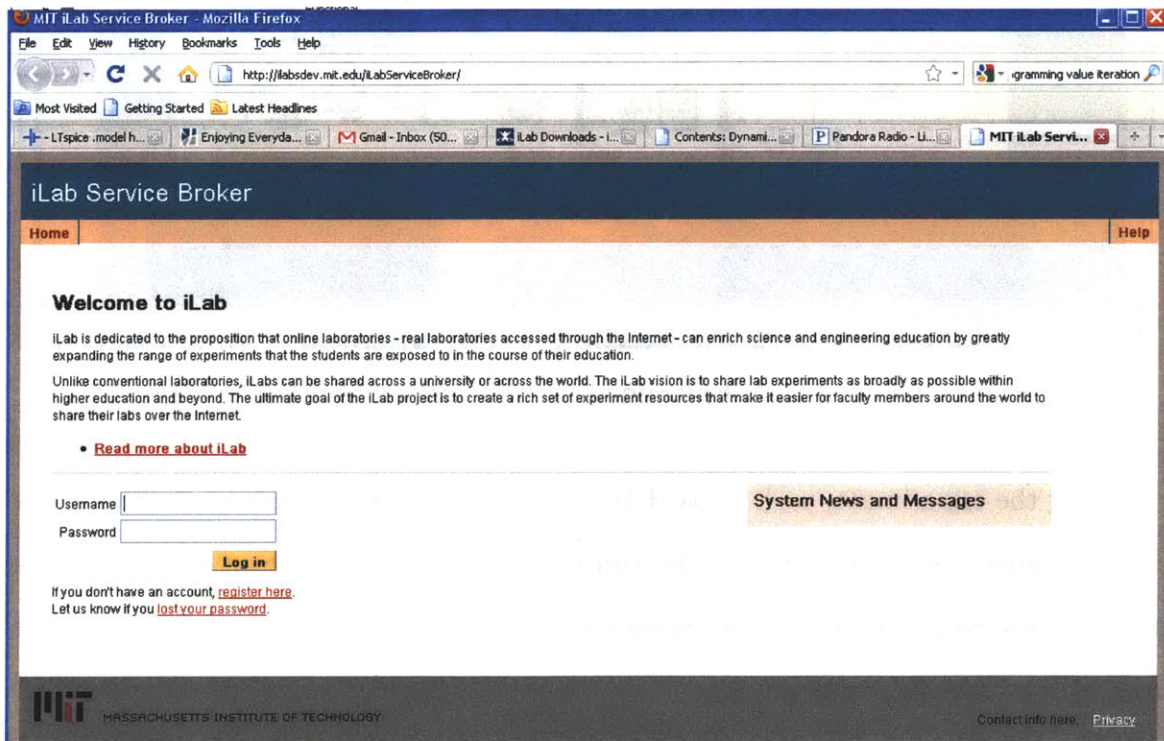


Figure 4.7: Service Broker Login

In summary, the user logs in and is directed to the experiment of he/she is authorized to use it. Then the user can schedule time and come at that scheduled time to launch the AM iLab experiment. This process of login in, scheduling time and launching the iLab is generic to all Interactive iLabs. The "Merged iLabs Bootstrapping Guide" [21] documentation gives a detailed run down of what the user should see as it examines another Interactive iLab called

”Time Of Day” iLab.

Once the user launches the AM iLab, he/she is redirected to the Lab Server and should see the user interface as illustrated in Figure 4.8.

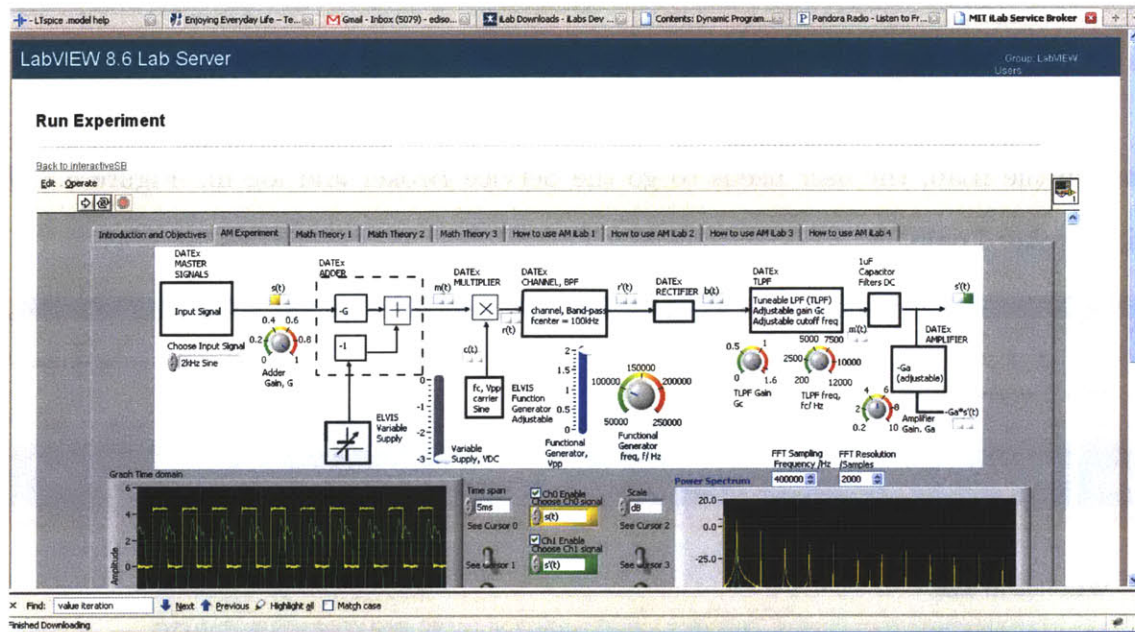
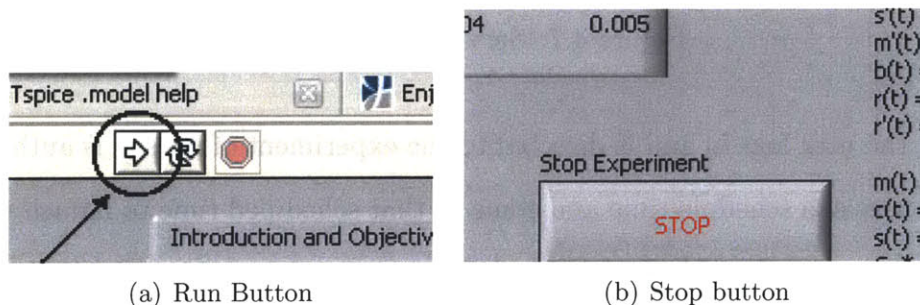


Figure 4.8: AM iLab Embedded Front Panel on Lab Server Page

For running the experiment, each LabVIEW application has a run button, ”white arrow” at top left of application Front Panel. The run button is shown in Figure 4.9(a). In order to stop the experiment, the user should press the ”Stop” button of Figure 4.9(b) found at the bottom of the ”AM Experiment” tab.



(a) Run Button

(b) Stop button

Figure 4.9: For Running AM iLab

4.5 Test Results

For testing, all the tunable devices were set such that a 2-kHz sinusoid input will be well transmitted. Table 4.1 gives a summary of all test settings of each component. It should be noted that the DATEx Adder gains are all negative and the DC offset provided by the NI ELVIS Variable Power Supply is negative. Consequently input signal $s(t)$ is given an effective positive DC offset to produce message, $m(t)$.

Component	Parameter Name/Description	Value
DATEx Adder	Gain on input $s(t)$ (G)	-1
	Gain on DC offset	-1
NI ELVIS Variable Power Supply	DC offset	-3
NI ELVIS Function Generator	Carrier frequency	100 kHz
	Waveform type	Sinusoid
	Peak-to-Peak Amplitude	2 V
DATEx Channel Block, Band Pass Filter	Center frequency	100 kHz
	Passband	18 kHz
	Stopband	12 kHz
	Gain	1
	Filter type	6th order Chebychev
DATEx Tunable Low Pass Filter	Gain	1
	Cut-off frequency	12 kHz
DATEx Amplifier	Gain	-5

Table 4.1: Test Settings of Components

4.5.1 Testing Probing of Signals

To test probing of signals, the 2-kHz sinusoid was used as input to the AM system. This is set at the "Choose Input Signal" control on the system block diagram. Figure 4.10 is a snapshot of the user interface when the user has chosen $s(t)$ (input signal) to be viewed from Ch0 of the scope and the user has chosen $m(t)$ (message signal) to be viewed from Ch1 of the scope. The time span of the time domain plots is set to 5ms while frequency range of the

power spectra is set to span -5 kHz to 5 kHz. As expected, the yellow LED next to $s(t)$ on the block diagram is on while the green LED next to $m(t)$ is on as well. For power spectra, both $s(t)$ and $m(t)$ have 2-kHz peaks as expected. Also, $m(t)$ has another significant peak at DC which is consistent with the DC component that has been added.

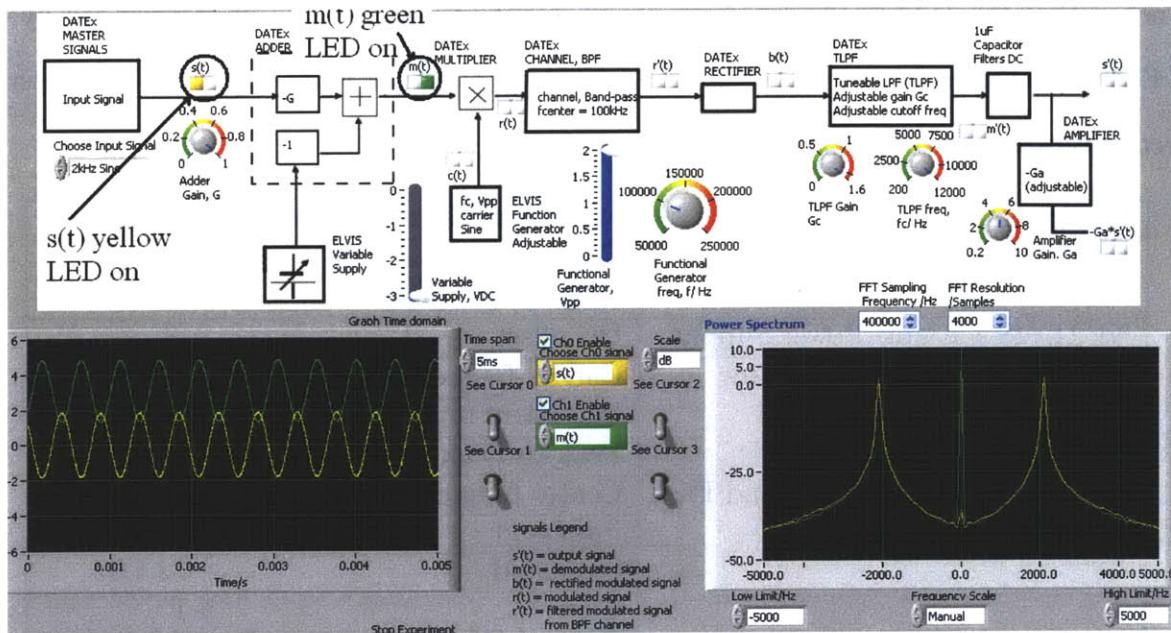


Figure 4.10: User probes $s(t)$ on Channel 0 and $m(t)$ on Channel 1

Now the user probes, carrier, $c(t)$ and modulated signal, $r(t)$ using Ch0 and Ch1 of the Oscilloscope respectively. Figure 4.11 shows the time and frequency domain plots. The time domain time span 5ms while the power spectra frequency range is -200 kHz to 200 kHz. As expected, the signals look very dense because the frequency is much higher at 100 kHz. There is also a distinct envelop which the modulated signal $r(t)$ has by virtue of $m(t)$. We can also notice that the power spectra of both signals have significant peaks only in the vicinity of 100 kHz.

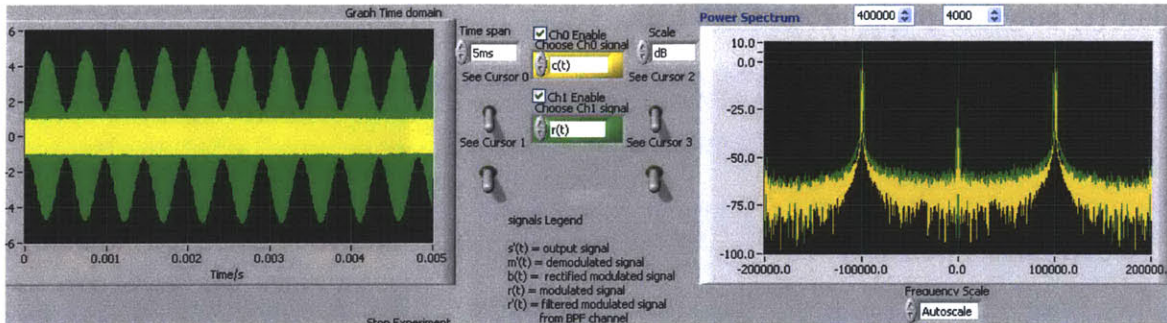


Figure 4.11: User probes $c(t)$ on Channel 0 and $r(t)$ on Channel 1

To better appreciate both signals, the time span was set to $500 \mu\text{s}$ while the frequency range was set to 90 kHz - 110 kHz. Figure 4.12 shows the results of these changes. Now the time domain waveforms are more visible. For the power spectra, the carrier, $c(t)$, has a 100-kHz peak while the modulated signal has 3 distinct peaks at 98 kHz, 100 kHz and 102 kHz approximately. These peaks are consistent with Amplitude Modulation theory.

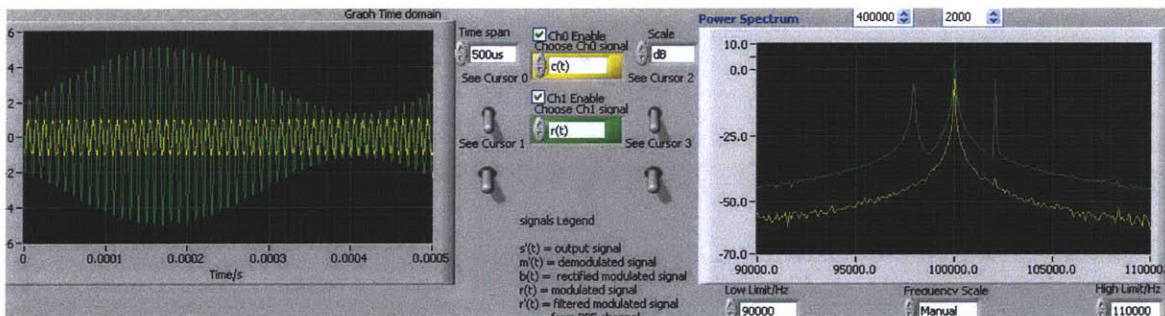


Figure 4.12: Still Probing $c(t)$ and $r(t)$ - Adjusted x-axis Scales

Now the user probes $r'(t)$, received signal after transmission, and rectified signal, $b(t)$ using Ch0 and Ch1 respectively of Oscilloscope. Figure 4.13 shows snapshot of results. Time domain time span is set at $500 \mu\text{s}$. For the frequency domain scale, the sampling rate of the scope was increased to 1 MHz so the automatic frequency scale will adjust itself to a range of -500 kHz to 500 kHz. In time domain, we see that $r'(t)$ is very similar to $r(t)$ from figure 4.12 while $b(t)$ only has the positive portions of $r'(t)$. For the power spectra, we can notice that $b(t)$ has peaks around DC and around multiples of 100 kHz which is the carrier frequency

while $r'(t)$ only has peaks around 100 kHz.

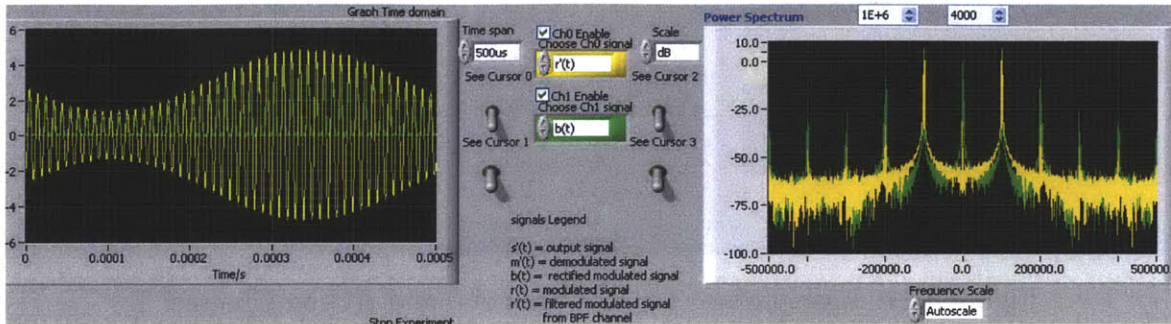


Figure 4.13: User probes $r'(t)$ using Ch0 and $b(t)$ using Ch1 of Oscilloscope

The user can take a more careful look at the vicinity of DC by changing the frequency range to span -10 kHz to 10 kHz. Figure 4.14 shows the results. We notice that $b(t)$ has a 2-kHz peak and a peak at DC. This suggests using a Low Pass Filter to isolate those components.

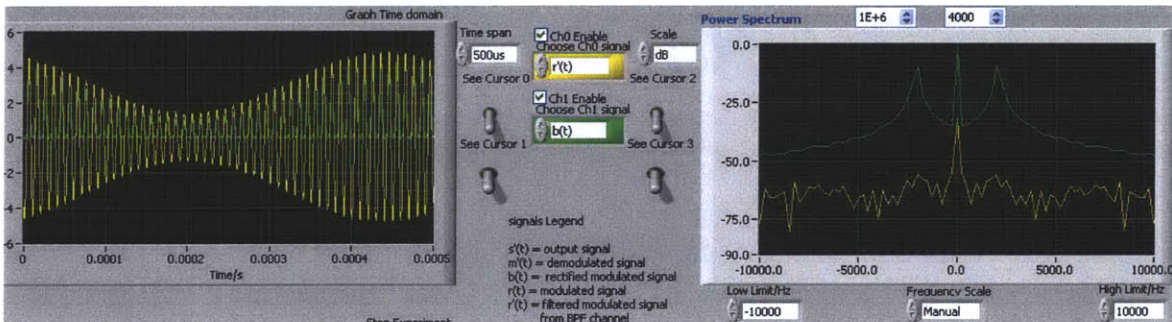


Figure 4.14: Still probing $r'(t)$ and $b(t)$ - Viewing Peaks in vicinity of DC

The user then probes $m'(t)$ and $s'(t)$ which are output message signal with DC and without DC. Figure 4.15 shows the results. The time domain span is set to 5ms while the frequency range spans -10 kHz to 10 kHz. As expected, both power spectra have 2-kHz peaks but $m'(t)$ has an additional significant DC peak.

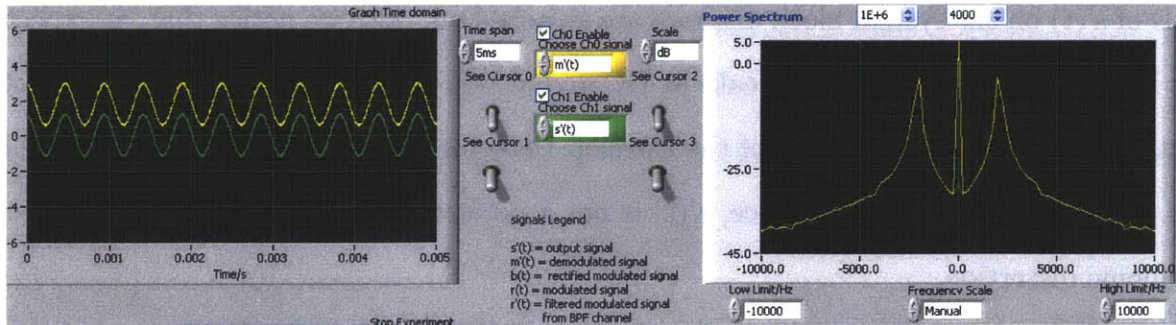


Figure 4.15: User probes $m'(t)$ using Ch0 and $s'(t)$ using Ch1 of Oscilloscope

Finally user probes $s'(t)$ and its $-Ga*s'(t)$. Figure 4.16 shows the results. The time domain plots have 5-ms time span while power spectra have 20-kHz frequency span. We notice that $-Ga*s'(t)$ is 180° out of phase with $s'(t)$ as expected due to negative amplification and the amplified output is slightly clipped. Both signals have significant peaks at 2-kHz on their power spectra.

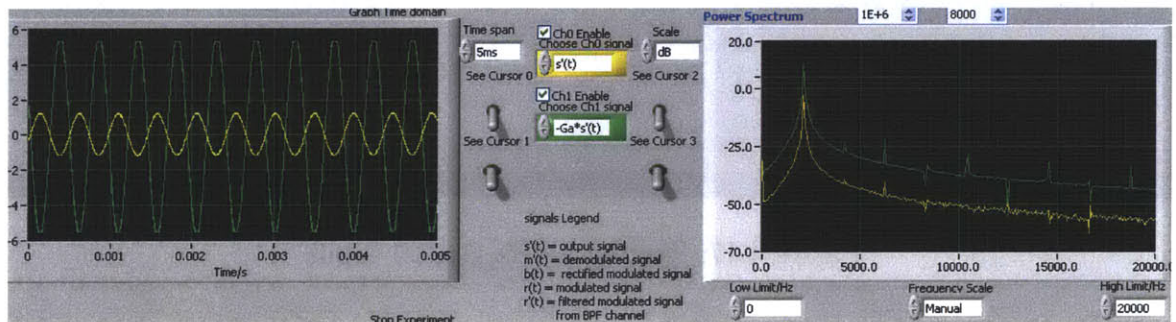


Figure 4.16: User probes $s'(t)$ using Ch0 and $-Ga*s'(t)$ using Ch1 of Oscilloscope

4.5.2 Testing Digital Input Options

So far the tests on probing made use of the 2-kHz sinusoid as input but as mentioned in Section 3.7, there are 2 other input options, 2-kHz Digital (Square Wave) signal and 8-kHz Digital (Square Wave) signal. Figure 4.17 shows the user interface with 2-kHz Digital input into the system. Input, $s(t)$, is probed using Ch0 of the Oscilloscope and output, $s'(t)$, is probed using Ch1 of the Oscilloscope. Consequently, the yellow LED of $s(t)$ and the green

LED of $s'(t)$ on the system block diagram are switched on. Also we notice that, as expected, the Power Spectrum of $s(t)$ has peaks at DC and at odd multiples of 2-kHz. We also notice that, due to the Low Pass Filter of the Envelop Detector, $s'(t)$ cannot be a square wave but has up to the 5th harmonic. Hence, $s'(t)$ is not a perfect sinusoid but exhibits a significant harmonic distortion.

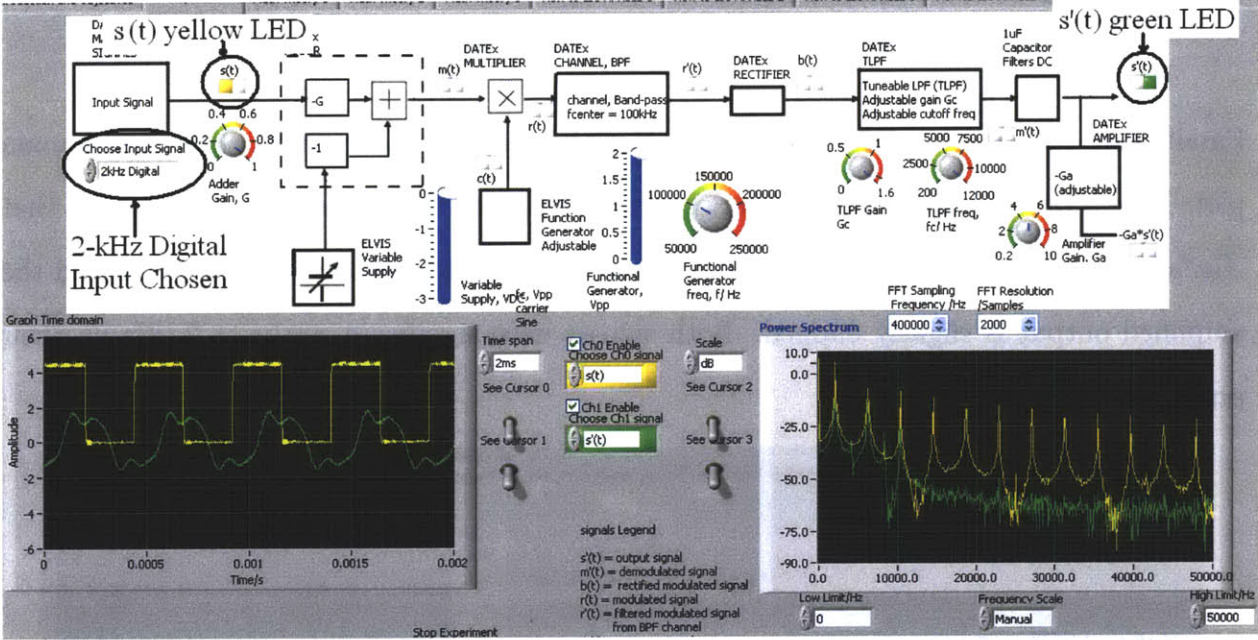


Figure 4.17: Testing 2-kHz Digital Input

Figure 4.18 shows a test result using the 8-kHz Digital (Square Wave) input. We can also notice the peaks at DC and odd multiples of 8-kHz. Now $s'(t)$ is a sinusoid because the limit of the Low Pass Filter limit is less than the 3rd harmonic frequency of the 8-kHz signal. Consequently the output $s'(t)$ only has the 8-kHz sinusoid component of the original 8-kHz Digital signal.

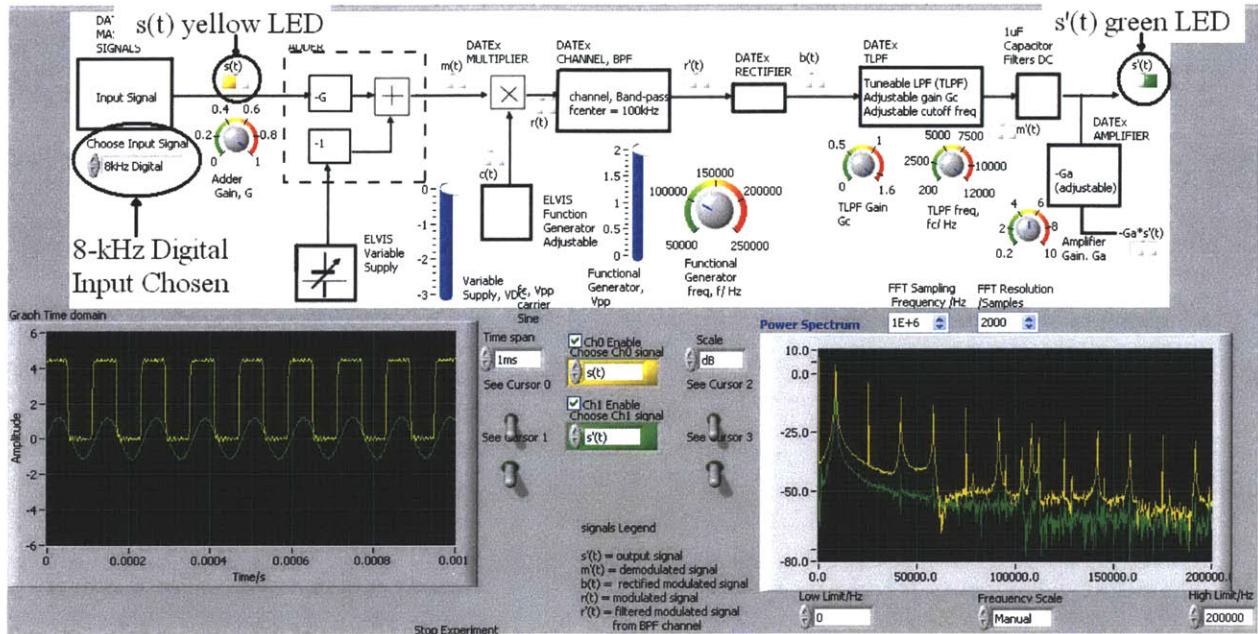


Figure 4.18: Testing 8-kHz Digital Input

4.6 Summary

The user interface of the Amplitude Modulation iLab is the Front Panel. Each control (input) has a graphical representation on the Front Panel. There are multiple ways that numerical inputs can be represented such as knobs, sliders, drop-downs, etc. Boolean inputs can also be represented through LED lights, buttons, levers or switches. Also enumerated inputs can be represented as drop-downs. The AM iLab user interface permits the user to see explore and understand the underlying block diagram of the system, tune the tunable devices and see the results in real time. Also the results are displayed both as time domain representations and power spectra (frequency domain). In addition to this the user can view all the relevant signals of the system and the user can choose between three different inputs.

Chapter 5

Future Work and Conclusion

Previous chapters detailed the background of iLabs, iLabs in Africa, the hardware used for the development of the Amplitude Modulation iLab (NI ELVIS, Emona DATEx). Detailed descriptions of the LabVIEW block diagram implementation of the Amplitude Modulation VI application and the user interface (Front Panel) were also given. This chapter will discuss what comes next in terms of future work and recommendations.

5.1 Future Work

With a working Amplitude Modulation iLab which is capable of acquiring data in time and frequency domain, the next step is to permit students to download that data for more detailed analysis. The very nature of an Interactive iLab does not permit a student to analyze data while he/she is running the iLab. Students need to be able to store the data which they accumulate during their session. This is because the session with the iLab will generally not be long enough to permit students to analyze their data in real time. This step of analyzing data is very necessary as a professor can use this to evaluate the students'

level of comfort, understanding of the iLab and understanding of the concepts the iLab was designed to teach. The professor can give an assignment with questions which make use of the AM iLab to evaluate the student. Hence storage of data is of paramount importance. This is where the Experiment Storage Service comes (ESS). The Lab Server uses DataSockets as the medium for storing data onto the ESS. Data from previous sessions of a user can be retrieved from the ESS. Consequently DataSocket VIs will be added accordingly in the Amplitude Modulation VI described in Chapters 3 and 4.

There are also a number of recommendations that could be explored in order to enhance the user's experience.

Firstly, it will be valuable if the user could listen to the input baseband signal, $s(t)$ and the output amplified output signal, $-Ga*s'(t)$. This will add richness to the interactive experience of the user and will give the user more insight on the necessity of having the right settings for the proper transmission of sound. The user will be able to hear the output being changed as he/she is changing settings. This could be achieved using a camera whose sound input is connected to the outputs of the Emona DATEx. A network camera which would have its own server and run concurrently and independently with the iLab would be ideal to implement this feature.

Secondly, for the current AM iLab, the inputs have been monotone signals such as 2kHz or 8kHz sine or square waves. It will be interesting for a user to be able to play around with an actual sound file. A simple first step to achieve this could be to have one sound file on the server which will be used and adding the option to the "Choose Input Signal" selector on the block diagram. The sound data can be written on one of the NI ELVIS physical input channels. It might be even more interesting if the user could upload his/her own audio files. This interesting addition will really be especially valuable when coupled with the live audio feedback from the network camera.

Thirdly, one shortcoming of this AM iLab is that the system block diagram is frozen and cannot be changed by the user. Hence the user does not take a more active role in the design. An interesting addition will be to use switching and some kind of interactive block diagram which, based on the relative positions of the switches, can be used to implement multiple telecommunication schemes. A good start could be to set up a block diagram with all the components necessary for four closely related telecommunication schemes and setting up a switching logic on the LabVIEW block diagram. When the user closes/opens the switches, one of the four telecommunication schemes is implemented. This will enable students to take on more design-oriented roles as opposed to just observation and analysis.

5.2 Conclusion

At this point, we have a working Interactive Amplitude Modulation iLab which can be accessed online on the Service Broker website. This iLab permits students to observe data while they are controlling settings on the iLab both in the time domain and in the frequency domain. Furthermore it also permits students to probe all relevant signals represented on the system block diagram. All the above features could be use as a blueprint for future interactive telecommunication online experiments.

This Amplitude Modulation iLab is only the beginning of efforts to give students the opportunity to learn telecommunication in an easy and accessible way. This iLab could be best used by classes of relatively large sizes which provide lectures on telecommunications but cannot really offer a traditional lab experiences. Such classes at MIT include 6.002 (Circuits and Electronics) and 6.003 (Signals and Systems) which are core requirements in the Electrical Engineering and Computer Science Department. Also iLab is the beginning of efforts that could be extended to the African partners in OAU, UDSM and MAK.

Amplitude Modulation is not the only telecommunication scheme which can be explored using the Emona DATEx. Up to 17 telecommunication experiments can be implemented using the DATEx hardware [17]. This lab will motivate the implementation of other telecommunication experiments including Sampling, Bandwidth Limiting and Reconstruction, FM, DSBSC, SSB, ASK, PSK, etc.

Appendix A

Amplitude Modulation and Demodulation Review

Communication systems play a key role in our modern world in transmitting information between people, systems and computers. Electronically, the signal to be transmitted is processed by a transmitter or a modulator to a form which can go through the transmission medium relatively unaffected. At the receiver, the signal is appropriately post-processed to retrieve the original signal. This post-processing is absolutely necessary because most transmission media will allow signals of a certain frequency range e.g. the earth's atmosphere will rapidly attenuate signals in the audible frequency range (10Hz to 20kHz), whereas it will propagate signals at a much higher frequency range over long distances [22].

The general process of embedding an information-bearing signal into a second signal is typically referred to as modulation while extracting the information-bearing signal from the second signal is called demodulation. There are different modulation techniques and each technique has its corresponding demodulation technique. In Amplitude Modulation (AM), the signal which needs to be transmitted is used to modulate the amplitude of the carrier,

which is another sinusoidal signal.

A.1 Amplitude Modulation Review

Multiple telecommunication schemes make use of sinusoidal AM in which a sinusoid $c(t)$ (called the carrier) has its amplitude multiplied by the signal which needs to be transmitted $m(t)$. The message signal is usually obtained by giving a DC offset to the original signal to be transmitted, $s(t)$. Generally the input signal $s(t)$, (and consequently $m(t)$) is within the audible range (10 Hz to 20 kHz) while the carrier signal $c(t)$ could typically be of the range 300 MHz to 300 GHz. In the AM experiment of this thesis, a 100 kHz signal carrier was used because the band pass filter best transmits 100 kHz. Equations A.1 below summarize the operations for modulation and Figure A.1 details it as a system block diagram. A summary of the signals and their description can be found in Table 3.1. Figures A.2 shows time domain and frequency domain plots of all relevant signals of the modulation. In these plots, the assumption is that the input signal $s(t)$ is sinusoidal.

$$\begin{aligned} m(t) &= DC + s(t) \\ r(t) &= m(t) \cdot c(t) \end{aligned} \tag{A.1}$$

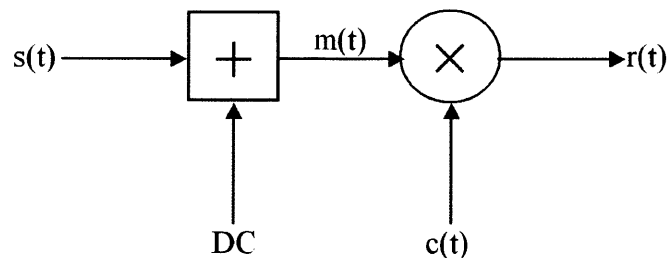
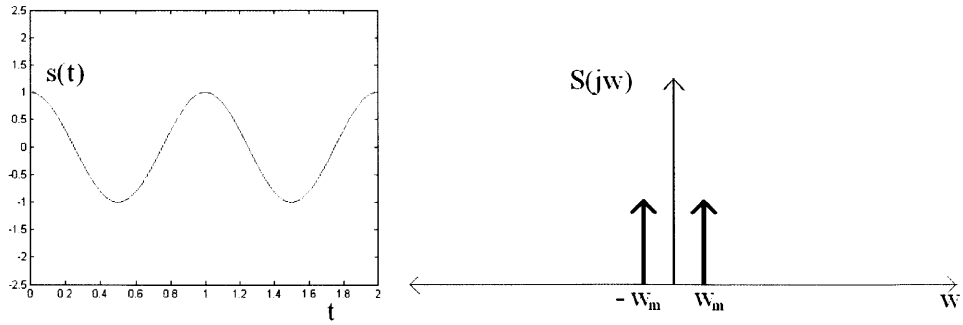
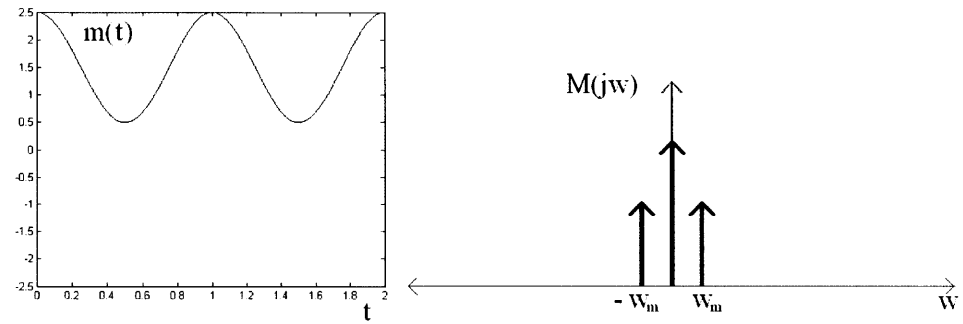


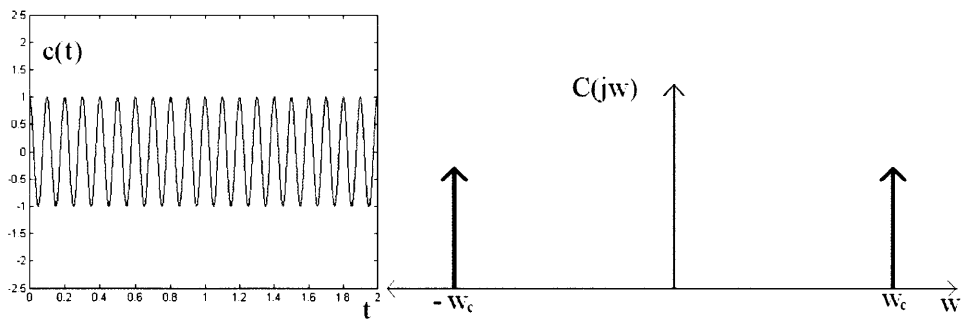
Figure A.1: Amplitude Modulation Blockdiagram



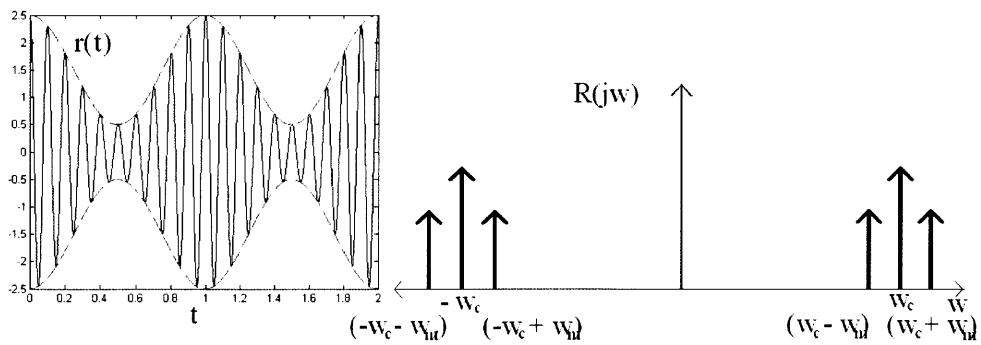
(a) $s(t)$ (Input Baseband Signal)



(b) $m(t)$ (Message Signal/Input Baseband Signal with DC offset)



(c) $c(t)$ (Carrier Signal)



(d) $r(t)$ (Modulated Signal)

Figure A.2: Time and Frequency Domain plots of Amplitude Modulation Signals

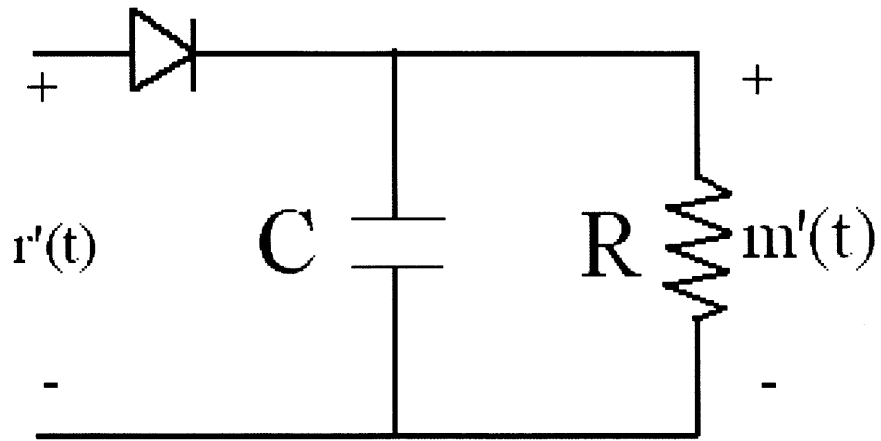
A.2 Amplitude Demodulation Review

After the sound has been transmitted, Demodulation of the received signal, $r'(t)$, is achieved by a method known as asynchronous demodulation. This method is called this way because it eliminates the need for synchronizing the carrier signal between receiver (demodulator) and transmitter (modulator). In synchronous demodulation the carrier signal is used to demodulate the received signal.

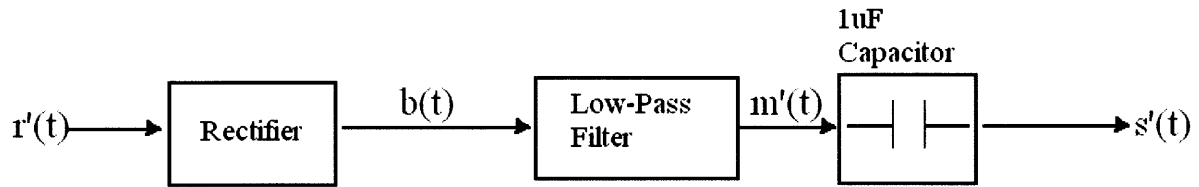
In order for asynchronous modulation to work correctly, certain assumptions are made. First the message signal $m(t)$ is positive (hence $s(t)$ is DC-offset to obtain $m(t)$) and the frequency range of $m(t)$ is very small compared with the carrier frequency. Under the assumption that $r'(t) \cong r(t)$ (good transmission), Figure A.4(a) shows $r'(t)$. The Envelope formed by $m(t)$ can easily be observed from that plot. For a signal to be demodulated using asynchronous demodulation, the modulated signal should have that typical structure. $m(t)$ (or a scaled version $m'(t)$) can be recovered through a demodulator called Envelope Detector.

The Envelope Detector consists first of a rectifier then a low pass filter. First the received signal from the transmission medium, $r'(t)$ (Figure A.4(a)) is rectified producing $b'(t)$ (Figure A.4(b)). The output of the rectifier, $b'(t)$ is sent through a low pass filter which smoothens the rectified $b'(t)$ producing a $m'(t)$ which is a scaled version of the original message $m(t)$. Then the DC component of $m'(t)$ can be removed to obtain $s'(t)$, scaled version of input signal $s(t)$.

Table 3.2 gives a summary description of all signals involved in the demodulation. Figure A.4 shows time domain and frequency domain plots of all relevant signals of the demodulation process.

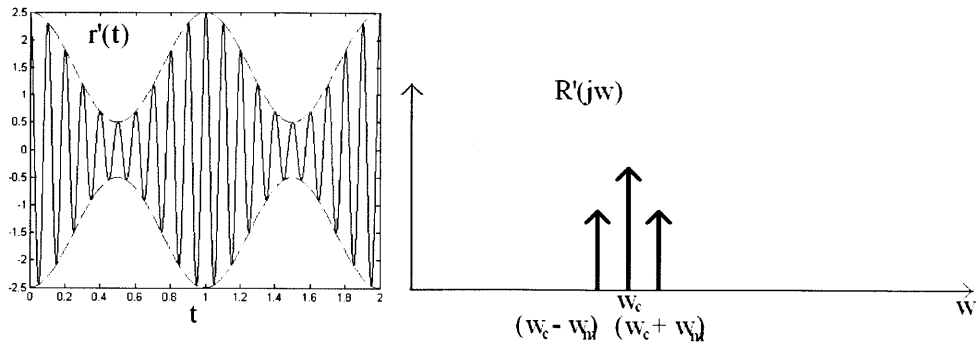


(a) Example of an Envelope Detector Circuit

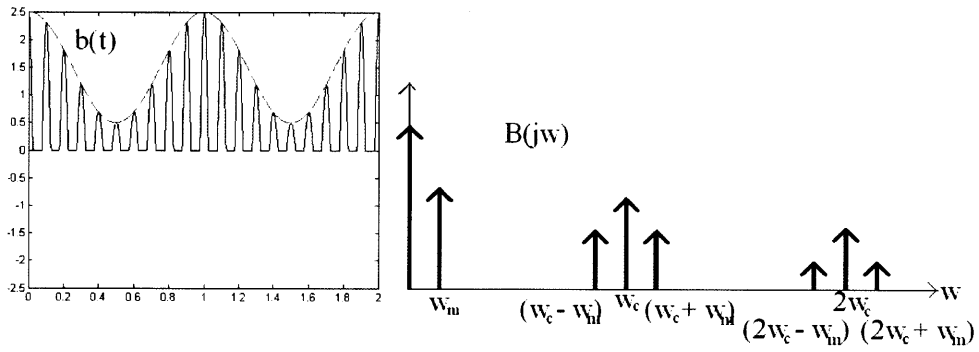


(b) Envelop Detector in Demodulation Block diagram

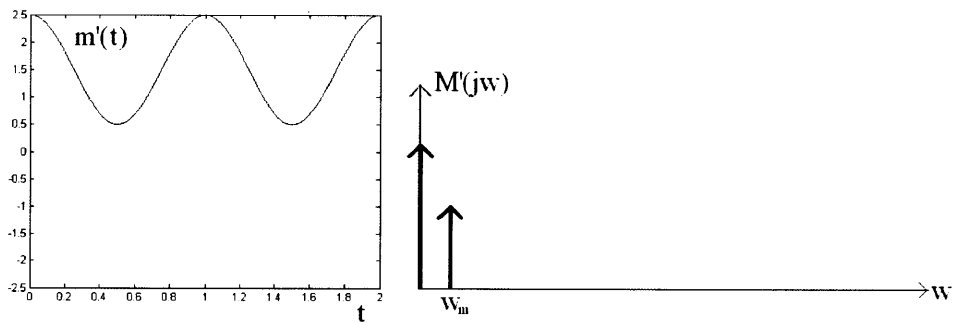
Figure A.3: Envelop Detector



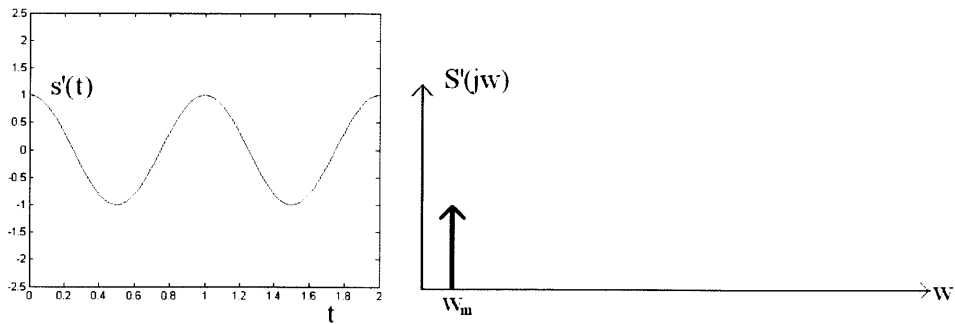
(a) $r'(t)$ (received signal from transmission medium)



(b) $b(t)$ (rectified received signal)



(c) $m'(t)$ output message with DC offset



(d) $s'(t)$ scaled version of baseband input

Figure A.4: Time and Frequency Domain plots of Amplitude Demodulation Signals

Bibliography

- [1] iCampus, “The MIT-Microsoft Alliance. iLab: Remote Online Laboratories,” April 2008. [Online]. Available: <http://icampus.mit.edu/projects/ilabs.shtml>
- [2] K. DeLong, “About iLabs,” August 2009. [Online]. Available: <http://wikis.mit.edu/confluence/display/ILAB2/about+iLabs>
- [3] —, “ilabs,” October 2009. [Online]. Available: <http://wikis.mit.edu/confluence/display/ILAB2/iLabs>
- [4] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, T. Mao, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, R. Zbib, and D. Zych, “The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories,” *Proceedings of the IEEE*, vol. 96, no. 6, pp. 931–950, June 2008.
- [5] K. Theroux, “Linking African Universities with MIT iLabs,” *Carnegie Reporter*, vol. 3, no. 4, Spring 2006. [Online]. Available: <http://carnegie.org/publications/carnegie-reporter/single/view/article/item/153/>
- [6] N. Instruments, “What is NI ELVIS?” April 2009. [Online]. Available: <http://zone.ni.com/devzone/cda/tut/p/id/8599>

- [7] —, “NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS),” September 2006. [Online]. Available: <http://zone.ni.com/devzone/cda/tut/p/id/3711>
- [8] E. I. P. Ltd., “TIMS Telecommunications — University, Modulation, Transmission, Wireless.” [Online]. Available: http://www.tims.com.au/tims-main.taf?do=page&page=about_tims&_UserReference=8F0813749139B5B34D023E87
- [9] J. L. Hardison, K. DeLong, P. H. Bailey, and V. J. Harward, “Deploying Interactive Remote Labs Using the iLab Shared Architecture,” *38th ASEE/IEEE Frontiers in Education Conference*, October 2008.
- [10] P. Mitros and K. DeLong, “The iLab Shared Architecture,” November 2010. [Online]. Available: <http://wikis.mit.edu/confluence/display/ILAB2/Developers>
- [11] S. Gikandi, “ELVIS iLab: A Flexible Platform for Online Laboratory Experiments in Electrical Engineering,” Master’s thesis, Massachusetts Institute of Technology, 2006.
- [12] R. Shroff, “A Versatile Internet-Accessible Electronics Workbench with DC Domain Experimentation and Troubleshooting Capabilities,” Master’s thesis, Massachusetts Institute of Technology, 2009.
- [13] A. Jiwaji, “Modular Development of an Educational Remote Laboratory,” Master’s thesis, Massachusetts Institute of Technology, 2008.
- [14] H. Soumare, “Introduction of Digital Experimentation Capabilities on the ELVIS iLab Platform,” Master’s thesis, Massachusetts Institute of Technology, 2009.
- [15] O. A. F. Oyebode, “A New Client Interface for the ELVIS iLab with Enhanced Capabilities,” Master’s thesis, Massachusetts Institute of Technology, 2011.

- [16] A. Breznik and C. Manfredini, *Emona DATEx Telecommunications Trainer for NI ELVIS - User Manual*, 1st ed., Emona Instruments Pty Ltd, 86 Parramatta Road, Camperdown NSW 2050, Australia, 2007-2010.
- [17] B. Duncan, *Emona DATEx Lab Manual, Volume 1, Experiments in Modern Analog and Digital Telecommunications For NI ELVIS II*, 3rd ed., Emona Instruments Pty Ltd, 86 Parramatta Road, Camperdown NSW 2050, Australia, 2007-2009.
- [18] N. Instruments, *Computer-Based Instruments: NI-SWITCH Software User Manual*, February 2001. [Online]. Available: www.ni.com/pdf/manuals/370154b.pdf
- [19] —, *NI Switches: Getting Started Guide*, January 2008. [Online]. Available: www.ni.com/switches
- [20] K. DeLong and P. Bailey, “iLabs Interactive Lab Server with LabVIEW Setup and Configuration,” January 2011. [Online]. Available: <https://wikis.mit.edu/confluence/download/attachments/21140627/iLabs+InteractiveLabServer+with+LabVIEW+Setup+and+Configuration.doc?version=1&modificationDate=1219237930000>
- [21] —, “Merged iLabs Bootstrapping Guide,” January 2011. [Online]. Available: https://wikis.mit.edu/confluence/download/attachments/21140627/Merged_iLabs_Bootstrapping_v5.doc?version=1&modificationDate=1219237945000
- [22] A. V. Oppenheim and S. H. Willsky, A. S. with Nawab, *Communication Systems*, 2nd ed. Asoke K. Ghosh, Prentice-Hall of India Private Ltd, 1997, ch. Chapter 8: Communication Systems, pp. 583–594.