Utilization of MATLAB Simulink Exercises for an Undergraduate Communications Course

Abstract

A set of six MATLAB Simulink laboratory exercises was previously designed in 2011 for an undergraduate analog/digital communication course. This paper presents our experience of a pilot test on these exercises, followed by their modification and enhancement, and concluding with an application of the modified exercises in the spring and fall 2012 offerings of the course. The pilot test by a student from the spring 2011 class identified several areas of improvement for the previous design of the lab exercises, including lack of details of lab instructions, high level of difficulty in the first two labs, and partial completeness of the last two labs. Significant effort was then put into the modification and re-design of these labs in fall 2011, and the enhanced labs were applied in spring and fall 2012 with enrollment of nine and seven students, respectively. Feedback from students was solicited after each lab exercise. The results show that the Simulink labs were well received by students, in comparison with a traditional lecture-only approach or a MATLAB script programming approach. Some slight changes were also made to address the minor flaws in these labs. Currently, the Simulink lab manuals and solutions are available to be disseminated. The solutions are available to instructors by emailing Dr. Y. Rosa Zheng zhengyr@mst.edu. The lab manuals are available to the public for free download at http://web.mst.edu/~zhengyr/EE243/EE243LabManuals2012.zip.

Background

The first communications course at our undergraduate level covers three main topics:

1) Review of linear systems topics such as the Fourier series, the Fourier transform, power spectral analysis, and impulse/frequency responses.
2) Basic analog modulation and demodulation techniques such as Amplitude Modulation (AM), Single-Side Band modulation (SSB), and Frequency/Phase Modulation (FM/PM).
3) Digital baseband transmission concepts such as line coding, pulse shaping, Inter-Symbol Interference (ISI), and Zero Forcing (ZF) Equalization.

This course is open to students who have taken continuous-time linear systems but not necessarily discrete-time linear systems. It is meant to provide students with a theoretical foundation for advanced courses such as Communication Systems II, Communication Circuits, and Wireless Communications. The course has been traditionally offered by the lecture-only approach; however, in a couple of semesters, lab projects using MATLAB scripts offered in the textbook were incorporated in addition to the lectures. Although the MATLAB projects provide a good aid for learning the heavy communications theory and are welcome by many of the students, more than 50% of the enrolled students were not ready for the extensive MATLAB programming. This is mainly due to their limited training in MATLAB scripting and lack of the foundation of discrete-time linear systems. Therefore, a set of Simulink projects was designed in spring 2011 for this course[1] because Simulink offers easy-to-use block diagram models that inherently take care of the sampling issues.

MATLAB Simulink has been used as an educational tool for teaching block-diagram based simulations in other institutes and for many courses[2,3,4]. It is also widely used in industry and research institutes for real-life applications in areas such as control system design and signal processing.[5,6] Advantage for the beginner is that Simulink has an intuitive point-and-click interface, where circuit blocks can be drag-and-dropped from a library, then connected with wires. Figure 1 shows a basic illustration of the Simulink design layout, in this case used to create a graphical representation of a third-order FIR (finite impulse response) filter.

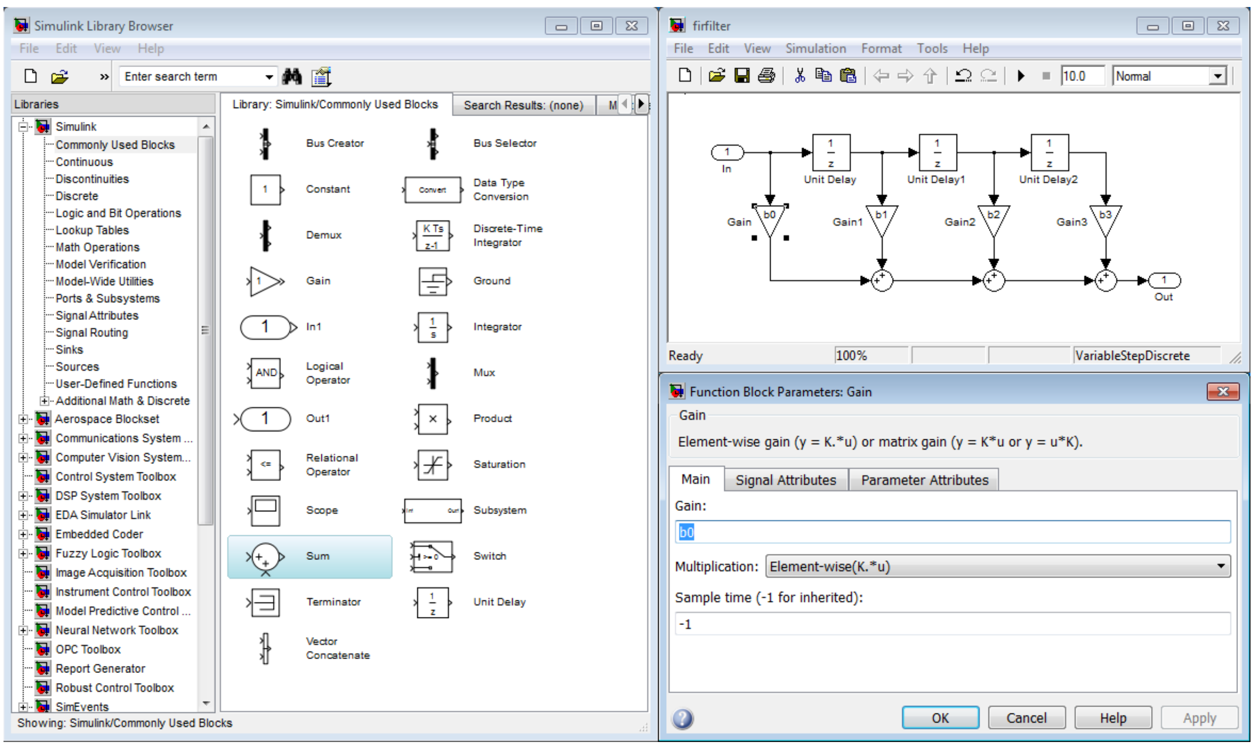

Figure 1 **–** Example of Simulink program interface.

Despite its ease-of-use in creating the most basic of block diagrams for simulation, the Simulink software harbors many unique nuances that must be understood in order to make good use of the software. The original design of the Simulink lab exercises aimed at teaching the most important skills relevant to Simulink in combination with teaching concepts from communications theory. The layout of the six lab projects was well thought out, as shown in Table 1. The original lab manuals were created by following Gagne's pedagogical model of nine levels of instruction.[7]

Lab	Topics covered	Simulink skill
I	Frequency Domain Analysis	Building a Model
II	Linear Systems	Subsystems & Masks
III	Amplitude Modulation	Library Building
IV	Frequency & Phase Modulation	Model Referencing
V	Pulse Code Modulation & Line Codes	Using Stateflow
VI	Zero Forcing Equalizer	Interacting with MATLAB

Table 1 - Simulink Laboratory Projects covering theory in combination with Simulink skills

Pilot Test of the Original Labs

A pilot test of these labs was performed by the first author of this paper, who took the Communications course in spring 2011 that used the MATLAB scripting approach. That author then became a master's student in fall 2011 and performed the pilot test. Despite being an outstanding student in the spring 2011 class and having just learned the theory thoroughly, the first author had to make a great effort to follow the original lab instructions and had to spend a large amount of time to perform the tasks required in the lab manuals. The pilot test identified several areas of improvement for the original design of the lab exercises. These include:

1) Unrealistic assumptions:
   The original design required that the students work through a Simulink demo before performing the lab exercises so that they gain the basic skills of making a model, using the Simulink help files, and navigating through libraries. The lab exercises were then designed with expectations and requirements that were far too high for the level of the course. In practice, it is rare that an undergraduate student in our university would spend much time on a demo if it is not an assignment that would be graded. It is also difficult for an average undergraduate to grasp all of the required skills just by working through a single demo.

2) Lack of details in lab instructions:
   Based on the unrealistic assumption that a student would know where to find help instruction in Simulink, the original lab manuals contain limited instructions on how to perform the lab exercises. The first author attempted to perform these labs as an undergraduate, having no prior Simulink experience, and was repeatedly confused by what was being asked and how to accomplish it. It was simply too difficult to search through the help manuals trying to figure out how to perform the tasks required by the lab manual. The frustration was worse during the first two lab exercises because the student

just started to learn the basic of Simulink skills and was overwhelmed by the large size of the Simulink help system, most of which is designed as a reference rather than a tutorial.

3) Inappropriate level of difficulty and unrealistic requirements:
   The level of difficulty for all six of the original labs was considered too high for average undergraduate students at our university, especially for this first communications course. Some models required advanced knowledge in Digital Signal Processing (DSP) or Digital Communications that are not prerequisites of this course. The amount of work required by the original lab manuals was also enormous and unrealistic for the given length of lab sessions. For example, multiple types of modulators and demodulators were required in Lab 3 and Lab 4 that could not be done in a two-hour session.

4) Incomplete lab designs:
   The original designs of Lab 5 and Lab 6 were partially functional except for a few example modules. Lab manuals for these two labs were completely missing.

5) Selection of Simulink solvers:
   The original design of the labs used a large amount of library blocks that utilize continuous-time (CT) solvers such as ode45. The primary reason for selecting the CT solvers was that the theory in the course is based mainly on CT systems and some students may not have a discrete-time linear system background yet. However, the available CT blocks in Simulink are very limited, thus making the tasks difficult to accomplish. On the other hand, Simulink has a much larger selection of DT blocks that not only run much faster than CT blocks, but also provide more convenient solutions to the tasks required in the labs.

In addition, the pilot test found out that the Simulink help manual fails to explicitly teach/emphasize the difference between the DT and CT solvers. This caused a significant amount of confusion and grief to both the original and new designers of the labs. After balancing the pros and cons of the DT and CT solvers, the decision was made to switch all labs but one to the DT solver. The ode45 CT solver was kept for Lab 2 to explicitly teach the difference between the two types of solvers in Simulink. All of the other five labs now use DT solvers, with the lab manuals instructing the students to set specified sampling frequencies for signal sources and having all subsequent blocks inherent this sampling frequency. When variable sampling time is required in one model, the lab instructions provide the sampling time entries without explaining the principle of discrete-time linear systems. This approach allows the students to use the DT models in Simulink without requiring them to understand the Nyquist sampling theorem.

Redesign of the Laboratory Exercises

After the pilot test, significant effort was put into the modification and re-design of these labs to address the issues that were found. The main considerations in modifying the labs were reorganization of objectives, addition of informational content, and adjustment of difficulty. The coverage and layout of the six lab projects were kept the same as shown in Table 1. Since the labs are meant to be supplemental to a lecture course, the number of required tasks in each lab was trimmed down to fit into a 2-hour lab session per project; while the excluded tasks were listed as bonus part for more capable students. Those extra tasks may be used for a full laboratory course if needed. The manual instructions were completely rewritten to provide much more details for each lab. In particular, the instructions for the first two labs provide the student step-by-step procedures to perform each task and to learn the basics of Simulink without searching the help manual. The instructions to the later labs gradually reduce the amount of explicit details and instead provide general guidance. All labs except Lab 2 were redesigned to use Simulink DT blocks rather than CT blocks.

Although the difficulty and scope of the labs were reduced overall to create an experience that is not overly demanding yet still educationally significant, the amount of information in the lab manuals pertaining to the remaining material was increased. Also, explanations for the functioning of certain Simulink features (such as continuous vs. discrete solvers and normal vs. accelerated simulation modes) were added; it is always a good idea for students to understand the "why" for the use of a software feature rather than to be told to "just do it this way because the lab manual says so."

Overview of the Newly Designed Lab Exercises

The complete set of exercises presented in this paper consists of six labs designed to build understanding in the fundamental operations of Simulink as well as provide experience in applying concepts learned in the communications systems lecture. It is important to note that, since these exercises were taught as an additional requirement of a 3-credit hour lecture course (class meets 3 hours per week), less material is included than would be expected for a full laboratory course. However, some material listed as bonus part that was cut from the original set of exercises could most certainly be added back in to create a new laboratory course.

The six lab exercises under test were as follows:

Lab 1 - Intro to Simulink and Frequency Analysis *(shown in Figure 2, left side)*
The first exercise covers the most basic of operations in Simulink, which includes placing blocks from libraries, connecting them together, and using basic signal sources and scope outputs. Basic FFT operations and Fourier series signal representations are also covered here. Since the students are assumed to have no prior experience with Simulink at this

point, the lab manual consists of very explicit instructions right down to every mouse click and keypress. This way, the students are eased into the Simulink program and can come away from the first lab feeling good about it.

<u>Lab 2 - Hierarchical Design and Linear Systems</u> *(shown in Figure 2, right side)*
The second exercise steps up the complexity by introducing subsystems and masks. Students are required to use these tools to implement a switchable source similar to the built-in Signal Generator block, which is then used in another model. A demonstration of system linearity is also performed here using transfer function blocks. This is the only lab out of the six that uses a continuous-time Ordinary Differential Equation (ODE) solver in Simulink; all other exercises use discrete-time solvers, which are much more practical and flexible in most cases.

<u>Lab 3 - Simulink Libraries and Amplitude Modulation</u> *(shown in Figure 3, left side)*
The third exercise requires students to create their own Simulink library with custom-made amplitude modulator and demodulator blocks. Unlike the previous two labs, the students are not explicitly told what blocks to use in the AM modulator; they are required to call upon previously learned knowledge to construct a model that simulates the basic AM equation. Also, the demodulator was designed as a product detector, which must be implemented by finding and using a low-pass digital filter. Both of these tasks serve to reinforce previously learned material while expanding upon the knowledge of how discrete sampling works in Simulink.

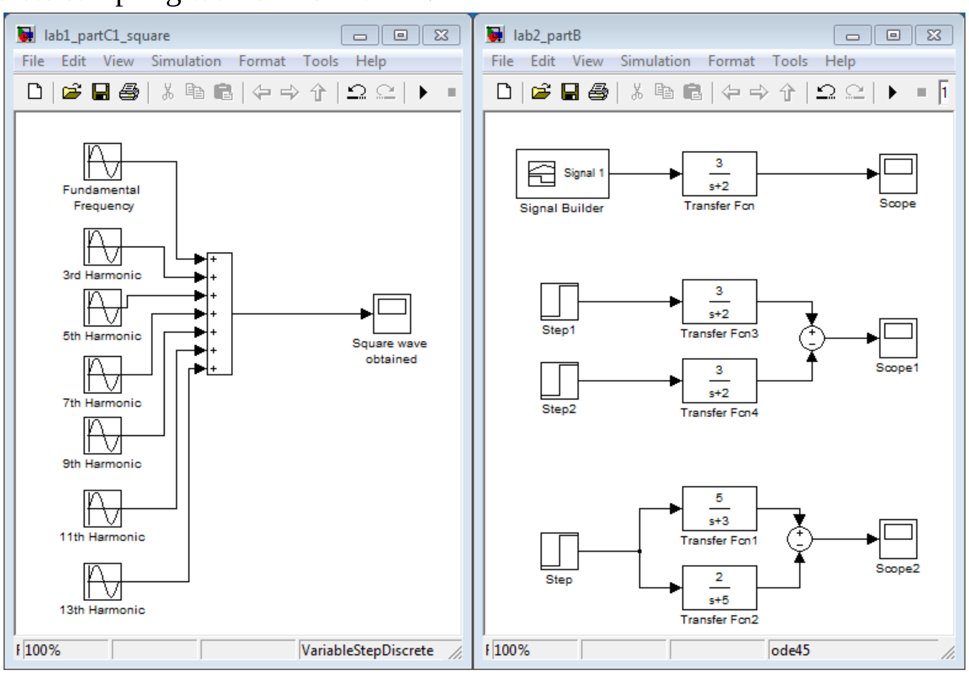

Figure 2 – Example screenshots from Simulink projects.
Left: In Lab 1, students construct a square wave from the sum of its Fourier Series harmonics.
Right: In Lab 2, students visually observe the property of system linearity.

Lab 4 - Model Referencing and Angle Modulation *(shown in Figure 3, right side)*

The fourth exercise centers around the creation of PM and FM modulators, this time using model referencing rather than simple subsystems or libraries. When model files are referenced in this manner, they can be accelerated by pre-compiling code, which is one of the important features of Simulink that is worth knowing about. Basic information about PM and FM is given, but students are still required to seek out and learn about the VCO block as well as external triggering for a sinusoidal source. This lab is not quite as long as the previous one, so it gives students a slight reprieve before the fifth one. Also, students are allowed extra credit by building an FM demodulator from scratch, which is quite difficult for a beginner at Simulink but allows students who might have missed a lab to catch up.

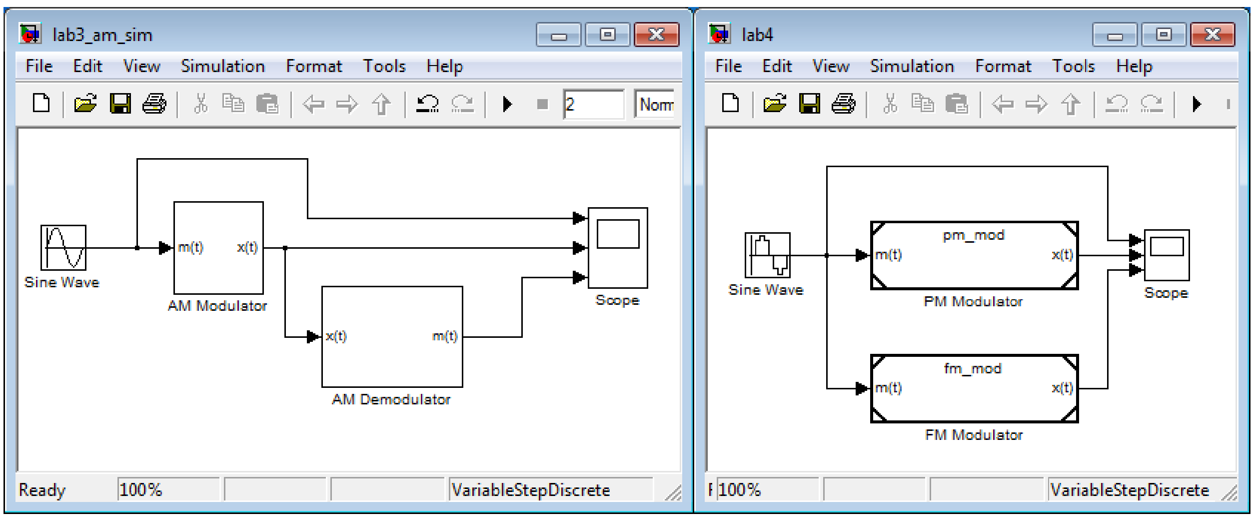

Figure 3 – Example screenshots from Simulink projects.
Left: In Lab 3, students construct AM library blocks which are used to modulate and demodulate an input signal.
Right: In Lab 4, students use external model file referencing to create PM and FM modulators.

Lab 5 – Line Coding and Decoding *(shown in Figure 4, left side)*

Unlike the previous four lab exercises, no new Simulink features are presented in the fifth one. Instead, students are asked to research various line coding schemes before coming into the lab, then implement them in Simulink from scratch using blocks of their choice. This lab is the turning point where students are freed from detailed instructions and expected to experiment and look up blocks that will allow them to accomplish the task at hand. By requiring students to invent their own solutions, they learn how to research and complete tasks on their own rather than being led by the hand as was done in the previous labs.

Lab 6 – Zero-Forcing Equalization *(shown in Figure 4, right side)*

The final exercise is mainly an observation task where students connect pre-constructed blocks to create a zero-forcing equalizer that removes ISI from a signal sent through a static channel with additive white noise. Since most of the model construction is already done, the emphasis of this lab is on noting the effects of the equalizer as various parameters (such as input signal type, static channel response, noise power, and equalizer length) are adjusted. In research, careful observation and interpretation of simulation results is just as important as constructing the simulations in the first place, so this final exercise rounds out the educational experience quite well.

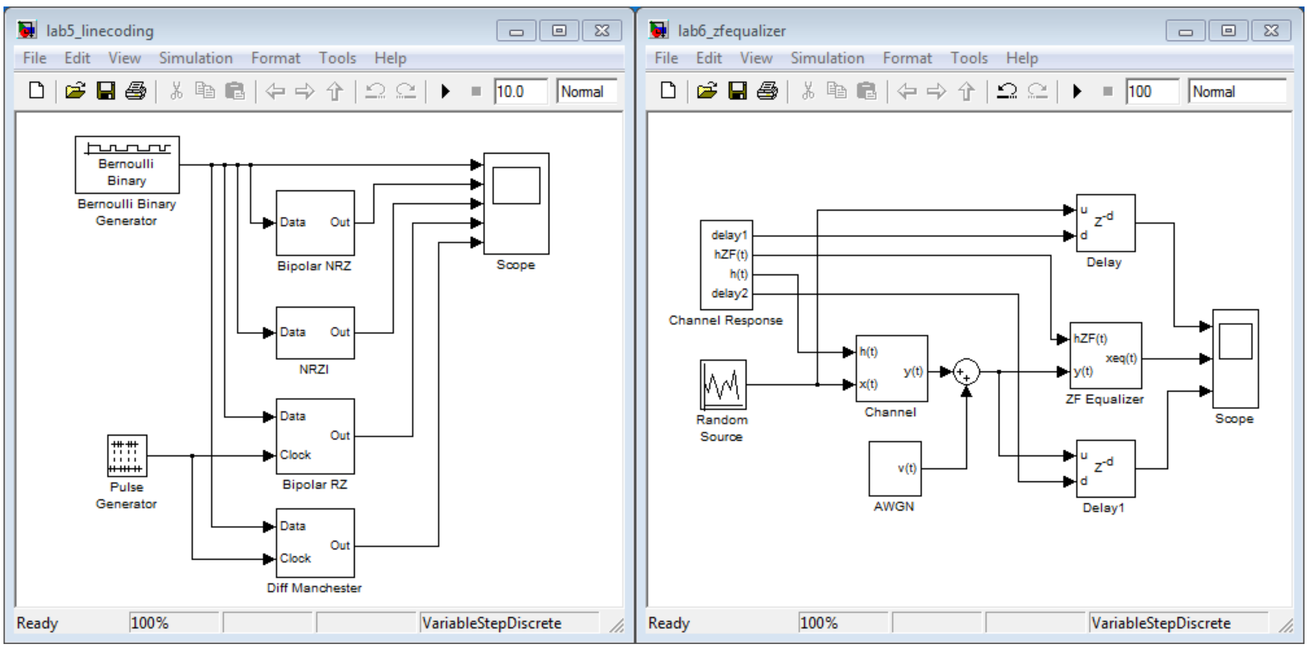

Figure 4 – Example screenshots from Simulink projects.
Left: In Lab 5, students construct subsystems that implement common line coding techniques.
Right: In Lab 6, students observe the effects of a zero-forcing equalizer on a static channel with additive noise.

Each of the six lab exercises consists of three major tasks. These are:

1. Preliminary – There is a short period of time between the posting of each lab manual for download and the start of the corresponding in-class lab session, and there is often some preliminary work that is to be done during this time so that the student can come to the lab session with the knowledge necessary to perform it. The preliminary work usually consists of some simple calculations or research; examples include calculating the Fourier series for some simple waveforms before the first exercise, and researching line coding schemes before the fifth one. The preliminary questions are checked before each lab session and are given 5-10 points out of the total of 100 points per lab grade.

2. Procedure – The actual construction and simulation of models in Simulink are performed during in-class laboratory sessions where teacher's assistants are available in person for the students to ask for help. The simplest lab exercises may only require a single 1-hour session to complete, whereas more complicated ones (such as the line coding exercise) may require two or more.

3. Report – After completing a lab exercise, a detailed report on what was done and what results occurred is due before the next lab session. The content requirements for the lab report are documented in the syllabus, so there should be little ambiguity over what material is expected. In each report, a series of post-lab questions pertaining to theory and/or lab results must also be answered.

This three-step approach is designed to obtain retention of material, as well as reinforcement of previously learned concepts from linear systems and communications. Since this set of labs is designed to be taught alongside a course in electrical engineering communications, the material in the labs should be sequenced so that the corresponding lectures have covered it a week or so in advance. For example, not much will have been learned during the first few weeks of lecture, so the first two labs only present material from linear systems, which is a standard prerequisite for communications. The remaining four labs present material from communications in the order they are usually taught, starting with AM, then progressing to PM, FM, line coding, and finally equalization. By lining up the material in this manner, concepts from communications will be subject to a greater degree of retention.

Utilization of the Newly Designed Labs in Classroom Teaching

The enhanced labs were applied in a communications course in Spring 2012 with an enrollment of nine students. The lecture portion of the course was conducted in a normal classroom with three one-hour sessions per week. The lab projects were added in weekly one-hour sessions as the lecture progressed. The first author served as Teaching Assistant (TA) for the lab sessions and graded the students' lab report work. A preliminary exercise was required before most labs, and the students were required to conduct the lab exercises in the designated computer learning lab during the allocated lab sessions. Each student worked individually through the lab exercises. Only five students conducted the sixth lab because only the best 5 out of 6 lab reports were counted in the final grade. The full set of labs was worth 20% of the final grade in the course.

The final grading scheme for the laboratory reports is as follows. These descriptions for each required section of the report are taken directly from the syllabus presented to the students:

Objectives (5%) – Basic objectives of the lab, in your own words.

Procedure (10%) – Brief description of how you did the lab exercise, step-by-step. This section does not have to be long; it is not necessary to copy the steps from the lab manual.

Results (30%) – The final results of the lab, including any screenshots and figures requested in the lab manual. Include any important observations or difficulties you had in this section.

Conclusions (25%) – Describe any important conclusions you have taken away from the lab here. Important things to consider for this section are: *What have you learned from this lab? How is the material here important in the real world? Overall, how valuable was this lab to your learning of the EE243 material in your opinion?*

Answers to Post-Lab Questions (10%) – Simply provide the answers to the post-lab questions here. Nothing fancy needed here; short-answers will be fine.

Model Files (20%) – Submit all model files, M-files, and other related files resulting from your completion of the lab as email attachments with your report.

This format is fairly standard in undergraduate lab reports at our university; since students were likely already familiar with this arrangement, it was a sensible choice.

Results and Student Feedback from Laboratory Sessions

Feedback from students was solicited after each lab exercise in order to gauge teaching effectiveness. The results show that the Simulink labs were well received by students, in comparison with the traditional lecture-only approach and the MATLAB script programming approach. Ordinarily, the communications course includes MATLAB-based exercises instead; the students seemed pleased that this was being replaced with a chance to learn the more visually-oriented Simulink.

Though the labs themselves were received fairly well, there was still a bit of displeasure among the students in the spring 2012 offering because they felt this was too much extra work for a 3 credit-hour lecture course. Nonetheless, when it became well-known that this set of exercises was the alternative to doing the MATLAB exercises and the total amount of work would remain the same for the whole semester, the students reverted to a somewhat neutral stance. In the fall 2012 offering, most of the students clearly showed their favor of doing Simulink labs over MATLAB scripting.

A chronological breakdown of the student experiences for each lab exercise is as follows:

Lab 1 - Intro to Simulink and Frequency Analysis

Since the tasks necessary to complete the first lab were outlined very clearly step-by-step in the lab manual, students had little trouble with it. The main purpose of this lab was to familiarize students with the most basic operations in Simulink, such as connecting blocks and running simulations, and in that capacity it accomplished its purpose. The only difficulty experienced by students was in the preliminary for the lab, where students were expected to calculate Fourier series coefficients, which would then be used to construct basic waveforms from a sum of sinusoids. However, many of them had trouble with this part, even when given the coefficients directly, because of some oddities in the way Simulink handles sine wave sources. Overall, the smooth conduct of the first lab raised the students' interest in Simulink.

Lab 2 - Hierarchical Design and Linear Systems

The second lab was more in-depth than the first, and some students were confused by the instructions for masking subsystems in the spring 2012 offering. Since the author of the lab manual was present to clarify what needed to be done, the lab was completed by the students successfully. Several clarifications were then made in the lab manual for use in fall 2012 class and no problem was encountered.

Lab 3 - Simulink Libraries and Amplitude Modulation

The third lab included the first instance where students were told to create a model to accomplish a task without explicitly being told which blocks to use. Most students were able to create the AM modulator successfully without hints, but a few did need to be led in the right direction first (i.e. they were told how many gain and sum blocks they needed, but not the order in which they needed to be arranged). The demodulator was trickier because it involved a digital filter that is generally learned in a higher level DSP course, but it was nevertheless essential to create a working demodulator. Therefore, the lab manual went more into detail on how to make the model without explaining how the filter works; as a result, most of the students performed this part very well. In the fall 2012 offering, some students were concurrently learning a DSP course. Therefore, they had even explored the filter characteristics on their own.

Lab 4 - Model Referencing and Angle Modulation

The fourth lab was initially designed to showcase the performance difference between normal and accelerated models. However, the lab computers did not have the necessary MEX compilers to use the accelerator mode, so this part had to be excluded. The angle modulation part of the lab was completed without much difficulty. The extra credit demodulation part was attempted by some of the students in the fall 2012 class since the build-in FM demodulator in Simulink was allowed to be used for completing this part.

Lab 5 - Line Coding and Decoding

Despite being the first lab where almost no explicit instruction was given, students performed reasonably well on the fifth lab. This was taken as evidence that the students were indeed learning the intricacies of Simulink well. A couple students even found an easier approach to the problem than the instructor's solution using special blocks. Hints were eventually given to stragglers, but these hints were more along the lines of a "push" to get them going in the right direction rather than an outright spoiler to the solution.

Lab 6 - Zero-Forcing Equalization

In the spring 2012 class, many students who had already gotten sufficient scores on the previous labs did not attempt the sixth because only the best 5 out of 6 labs counted for points. Roughly half of the class participated in this lab, but a problem was revealed as the lab session started: the pre-built models made use of new blocks that were not available on the old version of Simulink loaded on the computers. To fix this, students were forced to wait 5 minutes while the instructor quickly rebuilt the model using older blocks and re-uploaded it. To avoid problems like this in the future, all blocks used in example models should be version-checked before being posted for download by the students. Once the correct model was uploaded, students had an easy time performing this lab. In the fall 2012 class, the syllabus was modified to state that the best 13 out of the 14 (six labs plus eight quizzes) were counted towards the final grade, most students selected to work on the lab so that they can drop a low grade in quizzes.

The educational impact of these labs was best seen in the performance of the students on the line coding lab. In that lab, they were required to research four line coding schemes such as bipolar NRZ and differential Manchester, then implement each one as a masked subsystem in Simulink using any method of their choice. This served as both a learning experience and an evaluation of their accumulated Simulink skills. The results of this evaluation are as follows: around half of the students completed the full exercise with only a slight amount of guidance (mostly on the transition-based line codes, where a memory element was required), whereas the other half needed a bit more help to get the lab done. Since student abilities vary, an outcome like this is satisfactory. On average, the students learned most of the basics of Simulink quite well and the interest on learning the theory was also increased. Although the labs can be done in teams consisting of two students per team, almost all students chose to do the labs individually so that they can really learn the material. Several students also asked challenging questions beyond the required lab tasks during the lab sessions. Two students in fall 2012 class also applied the Simulink tool to another course they were taking and got excellent results. Students' comments and rating in teaching evaluation also demonstrated that they truly liked the Simulink approach.

As this was the first teaching experience for the first author, he learned a great deal about teaching and working with students. Experience gained through this teaching may also be helpful to other instructors using this set of lab exercises.

1) It is important to test the lab exercises on the computers that students use. This is true for any laboratory-based courses. Compatibility of software versions, missing software or hardware components, access rights to drivers, etc. would cause many problems and shall be tested prior to the lab sessions.

2) It is important to clearly state deadlines and consequences of late submission. A lack of hard deadlines and late-submission consequences was also assumed by many students. Despite repeated reminders, a lot of students forgot to submit the model files they used in the lab. The solution was to grade late submissions much more harshly; it is fine if a student needs more time to complete a report, but the quality of the submission must reflect this extra time spent.

3) It is important to clearly specify expectations in a grading rubric. Since the syllabus did not clearly specify a grading scheme for writing quality in particular, the quality of the lab reports was below standard. As a trade-off, the decision was made to remove a great deal of points from unacceptable work, with an offer to restore them if corrections were made by the student. In this manner, the student is not penalized for initially expecting low standards, but they must still improve the quality of their work to acceptable levels if they want to obtain full credit.

The authors also realized the importance of the assessment scheme on the quality of student learning, and it was noticed that the students' training on technical writing was not emphasized enough in the practice of previous lab courses. Many problems existed in the reports submitted by the students, including deficiencies in basic structure and flow of English, lack of proper grammar and spelling, etc. This issue was also reported to the undergraduate curriculum committee of the department.

Conclusion

A set of six MATLAB Simulink laboratory exercises, previously designed for an undergraduate communications course, has been thoroughly tested by a pilot test. These lab designs have then been modified and utilized in classroom teaching which were well received by students. The laboratory exercises are ready to be disseminated to the public. The lab manuals are available to for free download at http://web.mst.edu/~zhengyr/EE243/EE243LabManuals2012.zip. The lab solutions are available to instructors by emailing Dr. Y. Rosa Zheng zhengyr@mst.edu.

Acknowledgements

This work was supported by Dr. Zheng's NSF Career award #ECCS-0846486.

References

[1] C. Aroskar and Y. R. Zheng, "Design of Simulink Projects for an Undergraduate Communications Course," *Annual Conference of American Society of Engineering Education (ASEE)*, pp. 1-9, June 26-29, 2011.

[2] K. Modi, E. Hong and B. Bhattacharya, "Interactive models for teaching digital signal processing," *DSP/SPE 2009*, pp. 238-243, January 4-7, 2009.

[3] J. Turner and J.P. Hoffbeck, "Putting Theory into Practice with Simulink," *Proceedings 2005 ASEE Annual Conference & Exposition*, June 2005.

[4] M. Rice, "*Teaching Digital Communication Theory with Simulink at Brigham Young University*," MATLAB Digest: Academic Edition, vol. 3, no. 2, April 2009.

[5] J. Eker and A. Cervin, "A MATLAB toolbox for real-time and control systems co-design," *Real-Time Computing Systems and Applications*, pp. 320-327, 1999.

[6] M. Karimi-Ghartemani, H. Mokhtari, M.R. Iravani and M. Sedighy, "A signal Processing system for extraction of harmonics and reactive current of single-phase systems," *IEEE Transactions on Power Delivery*, vol. 19, pp. 979-986, July 2004.

[7] R.M. Gagne, *The Conditions of Learning and Theory of Instruction*, Holt, Rinehart and Winston Inc., New York, 1985.