ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36695

Improving Student Learning Experience with MATLAB Grader and Live Scripts

Liya Ni

Dr. Liya (Grace) Ni is joining Biola University in the fall of 2022 as a Professor of Engineering and Director of Engineering Program in the School of Science, Technology and Health. She was previously a Professor of Electrical and Computer Engineering (ECE) in the Gordon and Jill Bourns College of Engineering at California Baptist University, where she worked as a faculty member from 2009 to 2022 and served as the ECE department chair from 2015 to 2021. Dr. Ni received her Ph.D. in Electrical and Computer Engineering from the University of Waterloo, Canada, in 2003, her M.Eng. in Electrical Engineering from the National University of Singapore, Singapore, in 1999, and her B. Eng. in Information and Control Engineering from the Xi'an Jiaotong University, China, in 1996. Her research interests include controls, robotics, haptics, and engineering education. Email: grace.ni@biola.edu

Keith Hekman (Professor)

Keith Hekman is a Professor at California Baptist University where he teaches courses on AutoCAD, Excel, SOLIDWORKS, LabVIEW, Machine Design, and Vibrations. His research has been on automated grading. Prior to teaching at CBU, he taught at Calvin College and the American University in Cairo. He received his PhD from the Georgia Institute of Technology

Improving Student Learning Experience with MATLAB Grader and Live Scripts

Abstract

Engineering educators use MATLAB in multiple disciplines. The new features of MATLAB, such as MATLAB Grader and live scripts, can help students gain not only skills of analysis and design with MATLAB as a tool but also a better understanding of engineering concepts. MATLAB programming exercises are required in homework in a Vibrations course and computer lab projects in a Signals and Systems course. Automated grading with MATLAB Grader provides immediate feedback, which allows students to learn from their own mistakes while they try to pass the multiple requirements for each exercise that the instructor configured in advance. MATLAB live scripts are used in the teaching of Signals and Systems. Students are provided with lab manuals written in live scripts, allowing the integration of formatted text, equations, images, and MATLAB code in a single environment. With the live scripts, the concepts related to the MATLAB programming exercises are reinforced without the need of referring to another document or PowerPoint slides, and the examples embedded in the document can be run section by section with results displayed below the code or in an output window next to the document. Interactive components, such as a slider bar to change the value of time t in the convolution integral procedure and animation of the approximation of a signal using Fourier Series, are included in the examples to further improve the student learning experience. The instructor also used MATLAB live scripts during their lectures. At the end of the semester, the students completed a survey about their experiences with MATLAB Grader and live scripts. Students found these tools helpful in learning Vibrations and Signals & Systems. Setting up the auto-grading system was more effort in one semester than grading the work but will result in a reduced professor workload in future semesters.

Introduction

MATLAB is a powerful software tool for computer-aided analysis and design in multiple engineering disciplines. Equipping engineering students with MATLAB proficiency has been incorporated into the curriculum of many engineering programs. MATLAB programming exercises are required in homework in a Mechanical Vibrations course and computer lab projects in a Signals and Systems course taught by the two authors respectively. Manual grading those exercises is a time-consuming process for the instructor or teaching assistant, so the feedback to the student comes when they are no longer thinking of the problem. Studies have shown that automated grading tools can help reduce the drop-out rate [1], improve student grades [1] [2], better engaging students in nontraditional teaching modalities such as condensed summer courses and hybrid (online/in-person) courses [3]. MATLAB Grader is a web-based automated grading tool that provides immediate feedback on MATLAB code submission, which allows students to learn from their own mistakes while they try to pass the multiple requirements for each exercise that the instructor configured in advance. Smith used MATLAB Grader in a hybrid Vibrations course [3]. Reck used ‘Cody Coursework’ a precursor to MATLAB Grader in a Control Systems course [4]. Gregg *et. al.* used MATLAB Grader in a system modeling course [5]. At our university, it was first adopted in the Mechanical Vibrations course, then based on the positive feedback from students it was implemented in Signals and Systems in the following semester.

Signals and Systems is a math-intensive course that contains complex mathematical operations such as convolution integral and sum, Fourier series and transform, differential and difference equations, Laplace transform and z-transform. The topic of how to use computer software such as MATLAB and MATHCAD as well as hands-on activities involving hardware to enhance student learning in Signals and Systems has been discussed by many engineering educators [6] [7] [8]. In as early as 2000, Aburdene *et. al.* proposed interactive laboratories using MATLAB programs linked to a Microsoft Word document (a "notebook") so that formatted text, executable MATLAB code, and plots of simulation results were all integrated in one interactive document [8]. This idea can now be realized using MATLAB live script (released by MathWorks in 2016), an interactive document and program that integrates formatted text, images, hyperlinks, equations, and MATLAB code.

Using live scripts, the code embedded in the text can be executed section by section and the generated results and graphics are displayed alongside the code. In Fall 2021, we converted our lab manuals for six MATLAB-based computer lab projects in Signals and Systems, initially created as Word documents, into MATLAB live scripts. In addition, MATLAB live script was used during lectures. Animations and simulation examples programmed in MATLAB can help students visualize the changes of signals in complex mathematical operations. Having the text explanations, equations, and interactive animation all in a single environment instead of switching between PowerPoint and MATLAB enhances the learning experience.

The purpose of this research is to assess the effectiveness of using MATLAB Grader to enhance student learning experiences and performance in the aforementioned two courses: Signals and Systems, and Vibrations. In addition, the usefulness of using MATLAB live scripts incorporated in Signals and Systems was also evaluated. The research was done at California Baptist University, a private college in Southern California. There are about 780 students in the College of Engineering, with about 260 studying Mechanical Engineering and 85 studying Electrical and Computer Engineering.

Contents and Assessments of the Two Courses

MATLAB Grader was first implemented in Mechanical Vibrations in Spring 2021. There were a combined 62 students in two sections of the class. The class is a required class for Mechanical Engineering (ME) students, taken by seniors in their last semester of studies. The class was delivered in a remote synchronous format since it was during the first year of the COVID-19 pandemic. The course is taken after students have taken Signals and Systems, so students have been working in MATLAB. MATLAB is used to predict responses using complex exponential calculations and also used for analyzing multiple degree-of-freedom systems. One or two MATLAB Grader homework problems were given for several homework assignments to develop the students' confidence by the time it was necessary for solving problems. Students were assigned ten textbook problems in addition to the two introductory problems on how to use MATLAB Grader. The MATLAB Grader homework problems represented 17% of the homework points, and homework was 10% of the students' grade in the course. At the end of the semester, students took a survey, and they found using MATLAB Grader beneficial and would recommend using it in other courses.

Based on the positive reception of the students, MATLAB Grader was then implemented in Signals and Systems in Fall 2021. There were 28 students in the sections that MATLAB Grader was adopted. The course is required for students in Electrical and Computer Engineering (ECE) and Mechanical Engineering (ME) majors. The ECE students normally take this course in the fall semester of their junior year, right after they completed Circuit Theory II as a prerequisite. However, most of the ME students take it in their senior year, with a one-year gap from Circuit Theory II and after they worked on their summer internships in industrial fields not quite related to this course. Therefore, the ME students were often less motivated and found the course more challenging. The course covers properties of signals and systems, time-domain analysis including convolution and differential/difference equations, Laplace transform and z-transform, Fourier Series and Fourier Transform, frequency response, Bode diagrams, and filter design. It is mainly focused on continuous-time signals and systems, while it gives a brief introduction to discrete-time signals and systems towards the end of the semester. Student mastery of concepts is assessed with weekly homework assignments, a midterm exam and a final exam, and several quizzes. Most of the lab projects are designed to teach students to use MATLAB and Simulink as tools to assist them in the study of signals and systems. The lab projects count towards 25% of the course grade, with a total of 450 points including 100 points on a two-week filter implementation project involving hardware and 50 points on each of the remaining seven MATLAB/Simulink based computer projects. Six of the computer projects, which contribute to 17% of the course grade, were graded using MATLAB Grader.

Design of Exercises in MATLAB Grader

Here we use the course Signals and Systems to explain how we designed the MATLAB exercises in the Grader. The MATLAB-based homework problems in the Mechanical Vibrations course were designed similarly.

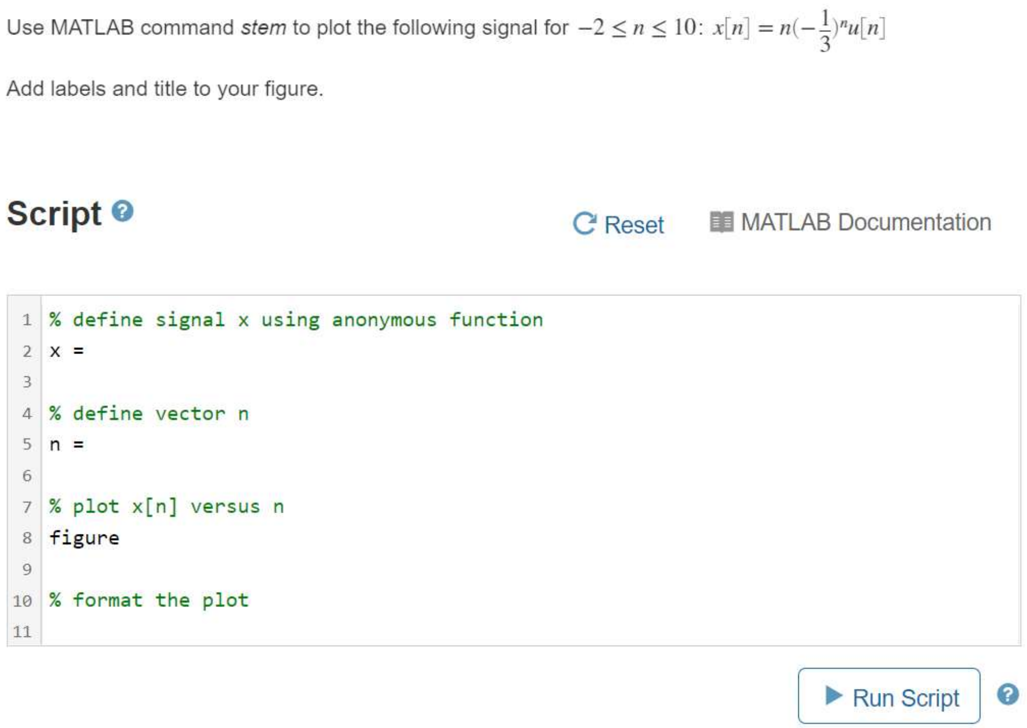
In this first version of integrating Signals and Systems lab projects into MATLAB Grader, the project on the analysis of system linearity and time-invariance using Simulink was excluded. The topics covered by the six projects built in MATLAB Grader are listed below in Table 1.

Table 1, Signals and Systems Lab Projects using MATLAB Grader

Lab #	Title	Topics
Lab 1	Using MATLAB to perform mathematic functions	complex numbers and variables; trigonometric functions; polynomials; LCM, GCD, and modulo
Lab 2	Preliminary study of continuous-time signals using MATLAB – definition, plotting, and operations	using anonymous functions to define signals; plotting signals, and operations on signals
Lab 4	Solving convolution integral and differential equations using MATLAB	introduction to Symbolic Math Toolbox; convolution integral; solving differential equations using the command dsolve

Lab 5	Laplace transform and transfer functions in MATLAB	use of Symbolic Math Toolbox to perform Laplace transform and inverse Laplace transform; Partial Fraction Expansion; transfer function, poles and zeros
Lab 6	Analysis of continuous-time signals and systems in frequency domain using MATLAB	approximating a periodic signal using its Fourier Series; Fourier Transform and Inverse Fourier Transform; plotting the magnitude and phase spectra of signals; using Fast Fourier Transform (FFT) for spectral analysis; plotting Bode diagrams
Lab 8	Analysis of discrete-time signals and systems using MATLAB	using anonymous functions to define discrete-time signals; numerical convolution sum; solving difference equations numerically; z-Transform; analyzing and simulating a discrete-time system with given transfer function

All six lab projects are listed under a MATLAB Grader course called "EGR331 Signals and Systems – MALTAB Exercises". A lab manual written in MATLAB live script is provided for each lab. Here we use Lab 8 Exercise 1 as an example to illustrate the development of exercises in MATLAB Grader. First the description and instructions were entered under the "Problem Description and Instructions" section. Equations, pictures, code, and hyperlinks are allowed to be inserted in this part. All the exercises for this course require learners to enter a MATLAB script although MATLAB Grader also allows the input to be a MATLAB function. The solutions were entered under the "Reference Solution" tab of the "Code" section. Then it was copied to the "Learner Template" tab next to the "Reference Solution" tab and the code to be filled in by students were removed. Comments to guide students in developing the correct solution and/or remind students on proper formatting were provided in the template. Figure 1 shows the learner view of Lab 8 Exercise 1.



Use MATLAB command *stem* to plot the following signal for $-2 \le n \le 10$: $x[n] = n(-\frac{1}{3})^n u[n]$

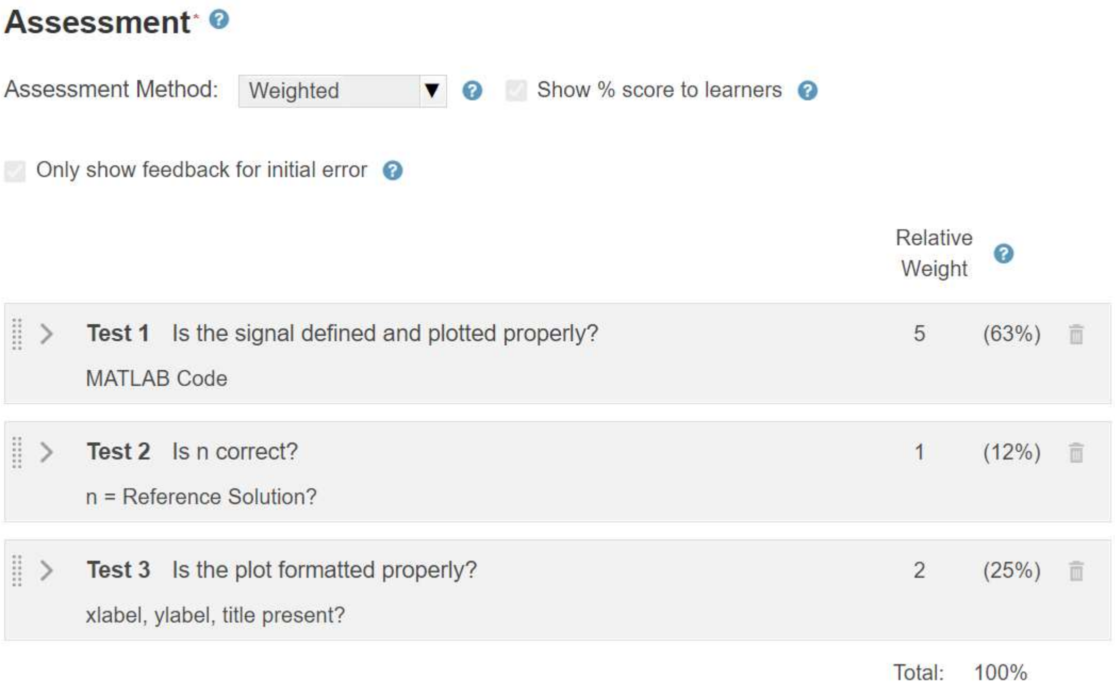
Add labels and title to your figure.

Script

```
% define signal x using anonymous function
x =

% define vector n
n =

% plot x[n] versus n
figure

% format the plot

```

Figure 1, Learner view of the problem description and code template

The tests were entered one by one under the “Assessment” section. There are four types of tests supported by MATLAB Grader: variable equals reference solutions; function or keyword is present; function or keyword is absent; and MATLAB code. As shown in Figure 2, the type of Test 1 is “MATLAB code” because the plot of the signal needs to be verified using MATLAB code. The function developed by Kosfelder [9] is modified and utilized for the evaluation of plots in MATLAB Grader. The type of Test 2 is “variable equals reference solutions” which simply compares the value of the student solution with that provided in the reference solution. The type of Test 3 is “function or keyword is present”. In this particular case the presence of functions *xlable*, *ylabel*, *title* is checked to ensure proper formatting of the plot. Points will be assigned to each test evenly if the instructor selects the “Correct/Incorrect” option under Assessment Method. Different weights can be assigned to different tests by choosing the “Weighted” option, as shown in Figure 2, if the instructor would like to emphasize certain aspect(s) of the solution.



Assessment*

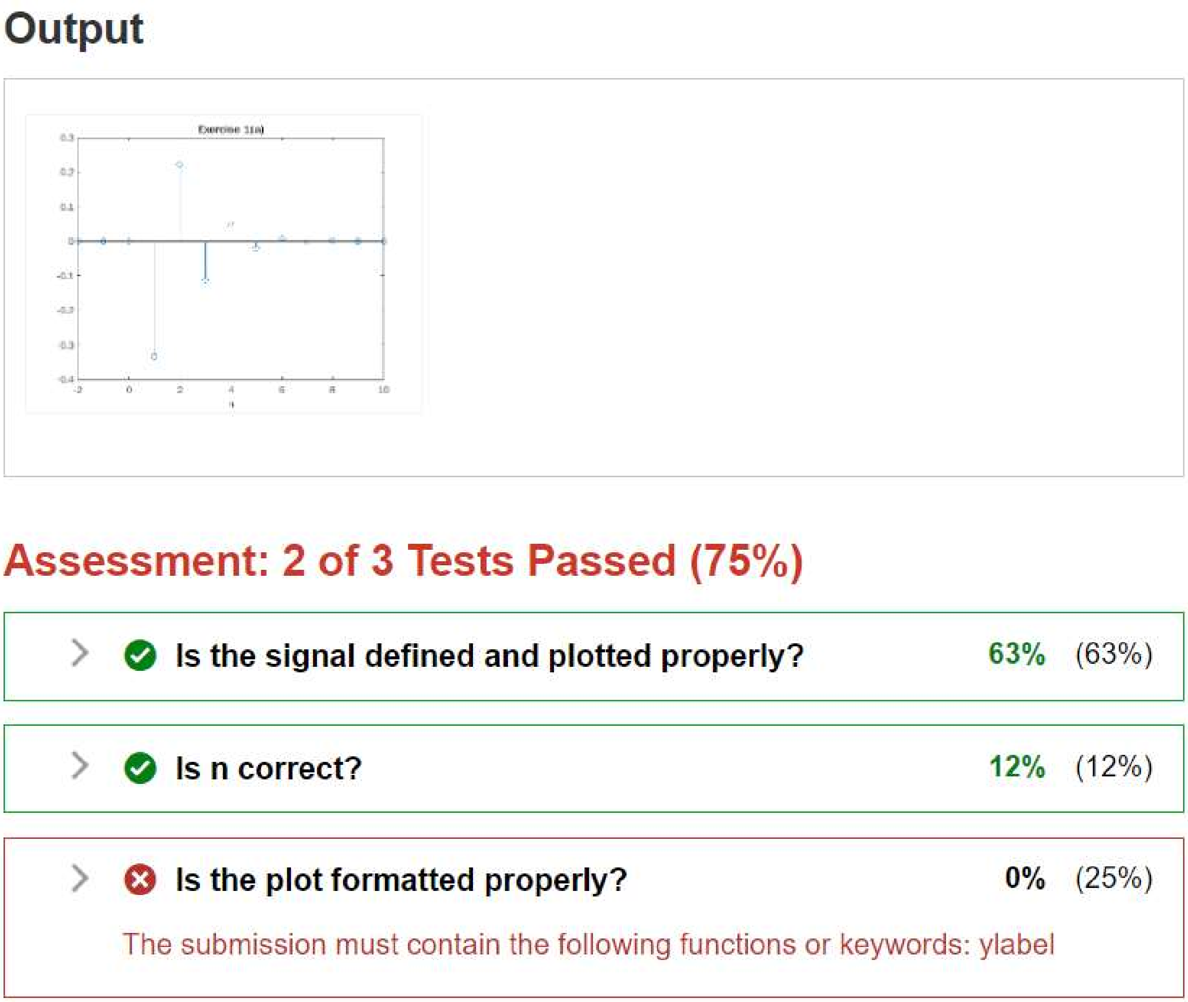
Assessment Method: Weighted ▼ ☑ Show % score to learners

☑ Only show feedback for initial error

		Relative Weight	
Test 1	Is the signal defined and plotted properly? MATLAB Code	5	(63%)
Test 2	Is n correct? n = Reference Solution?	1	(12%)
Test 3	Is the plot formatted properly? xlabel, ylabel, title present?	2	(25%)
		Total:	100%

Figure 2, Types of tests for assessment

Normally the default feedback shown under a failed test case is sufficient to help students correct the mistake. Additional feedback can be entered by the instructor if necessary. An example of a failed test with default feedback is shown in Figure 3. When students are attempting the exercises in MATLAB Grader, their progress can be monitored in real-time. The instructor can see how many students have completed each exercise. The learner analytics (shown in Figure 4) generated by MATLAB Grader provides more details on submissions required to solve the problems and average submissions required to pass each test. It can be used by the instructor to further improve students learning experience by adding more examples and exercises on topics that students struggle most, clarifying certain part of the problem description, breaking down a test into several tests, and providing additional feedback based on the common mistakes identified from the “Learner Solutions” under Learner Analytics.



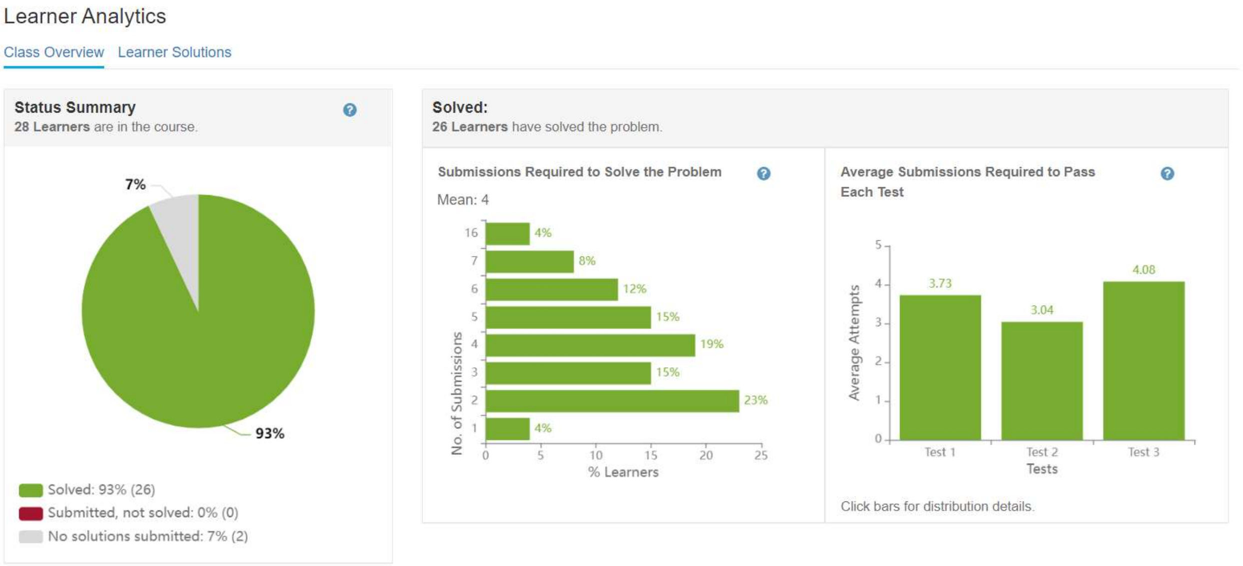
Output

Assessment: 2 of 3 Tests Passed (75%)

✓ Is the signal defined and plotted properly?	63%	(63%)
✓ Is n correct?	12%	(12%)
✗ Is the plot formatted properly?	0%	(25%)

The submission must contain the following functions or keywords: ylabel

Figure 3, A failed test with default feedback



Learner Analytics

Status Summary

28 Learners are in the course.

🟩 Solved: 93% (26)

🟥 Submitted, not solved: 0% (0)

⬜ No solutions submitted: 7% (2)

Solved:

26 Learners have solved the problem.

Submissions Required to Solve the Problem

Mean: 4

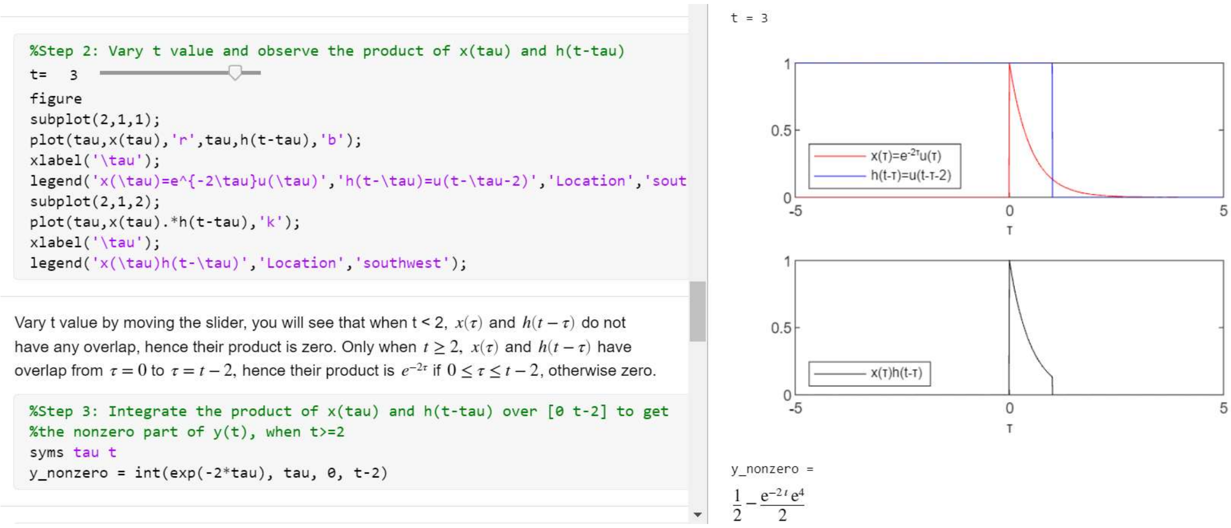
Average Submissions Required to Pass Each Test

Click bars for distribution details.

Figure 4, Learner analytics

Design of Lab Manuals and In-Class Examples in MATLAB Live Scripts

The Signals and Systems lab manuals from previous years were converted to MATLAB live script and interactive components were added. For example, in Lab 4 - Solving convolution integral and differential equations using MATLAB, there was an exercise asking students to determine the different cases in the graphical procedure of convolution integral and then use the Symbolic Math Toolbox to perform integral for each case. A slider bar was embedded in the live script so that students can vary the t value by moving the slider bar. The overlap between the input signal $x(\tau)$ and the folded and shifted impulse response $h(t-\tau)$ is displayed in the output window next to the live script and is updated in real-time corresponding to the change in the t value. A screenshot of this example is shown in Figure 5. In the live script lab manuals, the concepts and equations were presented with the proper format, followed by coding examples similar to the required lab exercises. Section breaks were added after each example or an individual step of a long example so that students can run the code segment one by one while reading through the lab manual. Instructions were provided in the following format: “Read Sections # below, run the live scripts by clicking ‘Run Section’ and observe the results in the output window, then go to MATLAB Grader to complete Exercise #”.



```
%Step 2: Vary t value and observe the product of x(tau) and h(t-tau)
t=   3
figure
subplot(2,1,1);
plot(tau,x(tau),'r',tau,h(t-tau),'b');
xlabel('\tau');
legend('x(\tau)=e^{-2\tau}u(\tau)','h(t-\tau)=u(t-\tau-2)','Location','sout
subplot(2,1,2);
plot(tau,x(tau).*h(t-tau),'k');
xlabel('\tau');
legend('x(\tau)h(t-\tau)','Location','southwest');
```

Vary t value by moving the slider, you will see that when t < 2, $x(\tau)$ and $h(t-\tau)$ do not have any overlap, hence their product is zero. Only when $t \geq 2$, $x(\tau)$ and $h(t-\tau)$ have overlap from $\tau = 0$ to $\tau = t-2$, hence their product is $e^{-2\tau}$ if $0 \leq \tau \leq t-2$, otherwise zero.

```
%Step 3: Integrate the product of x(tau) and h(t-tau) over [0 t-2] to get
%the nonzero part of y(t), when t>=2
syms tau t
y_nonzero = int(exp(-2*tau), tau, 0, t-2)
```

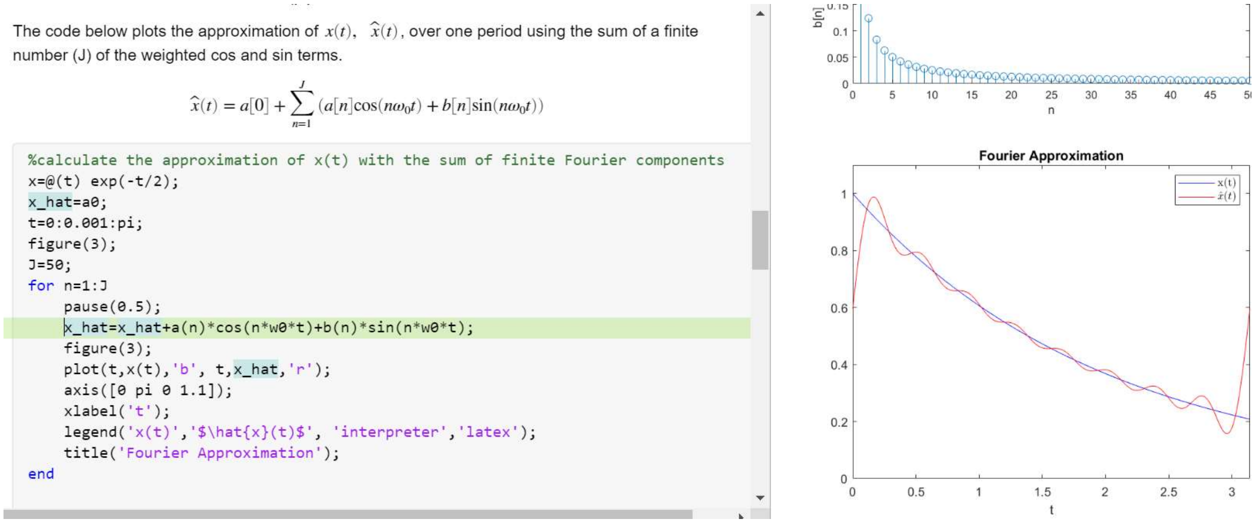
```
t = 3
```

```
y_nonzero =
```

$$\frac{1}{2} - \frac{e^{-2t}\,e^{4}}{2}$$

Figure 5, An interactive example written in MATLAB live script

MATLAB live script was also applied to some lectures in the Signals and Systems class. For example, a MATLAB script was created in the past to animate the partial-sum approximation of the Fourier Series representation to a periodic signal. After integrating it to a live script with the equations to compute Fourier Series coefficients and the partial-sum approximation, the instructor did not need to switch between PowerPoint and MALTAB in explaining this process. The learning experience for students also became more enjoyable. A screenshot of this example is shown in Figure 6.



The code below plots the approximation of $x(t)$, $\widehat{x}(t)$, over one period using the sum of a finite number (J) of the weighted cos and sin terms.

$$\widehat{x}(t) = a[0] + \sum_{n=1}^{J} (a[n]\cos(n\omega_0 t) + b[n]\sin(n\omega_0 t))$$

```
%calculate the approximation of x(t) with the sum of finite Fourier components
x=@(t) exp(-t/2);
x_hat=a0;
t=0:0.001:pi;
figure(3);
J=50;
for n=1:J
    pause(0.5);
    x_hat=x_hat+a(n)*cos(n*w0*t)+b(n)*sin(n*w0*t);
    figure(3);
    plot(t,x(t),'b', t,x_hat,'r');
    axis([0 pi 0 1.1]);
    xlabel('t');
    legend('x(t)','$\hat{x}(t)$', 'interpreter','latex');
    title('Fourier Approximation');
end
```

Figure 6, An animation example written in MATLAB live script

Analysis of Student Performance

For the Vibrations class, the MATLAB Grader questions were a small portion of the homework grade for the class. Figure 7 shows the students' grades on the MATLAB Grader problems vs their overall homework grades. A majority of the students did better on the MATLAB Grader problems than their overall homework in the course. Since 74% of the homework was done in a group, this might have improved some students' homework grades. There was one bonus MATLAB Grader problem, which a couple students attempted which is why some sores are above 100%.

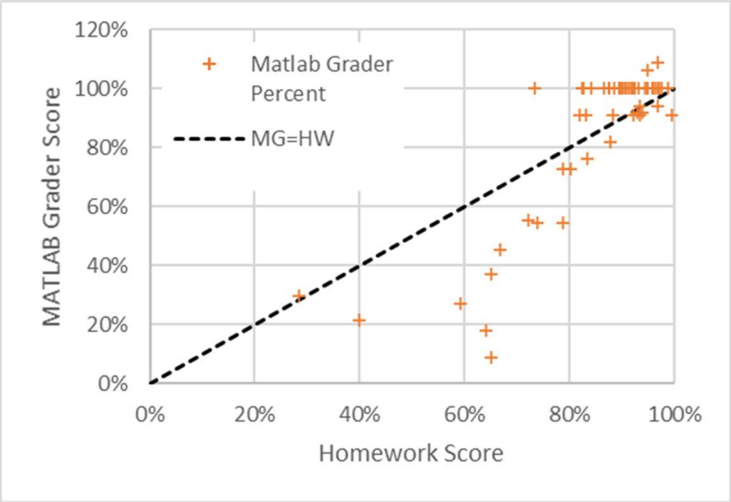


Figure 7, 62% of students did better on the MATLAB Grader than their overall homework.

Since most of the Signals and Systems labs were converted to MATLAB Grader, the students' lab performance on the MATLAB Grader graded labs were analyzed before and after the change. MATLAB Grader was introduced in the Fall of 2021. Figure 8 groups the grades into A (90%-100) to F (<60%) ranges, with the number of students in each range indicated. With MATLAB

Grader all of the students put in the effort to get the highest-grade range, while in other semesters, this was not the case. The figure also shows the class average, which was also higher using MATLAB Grader.

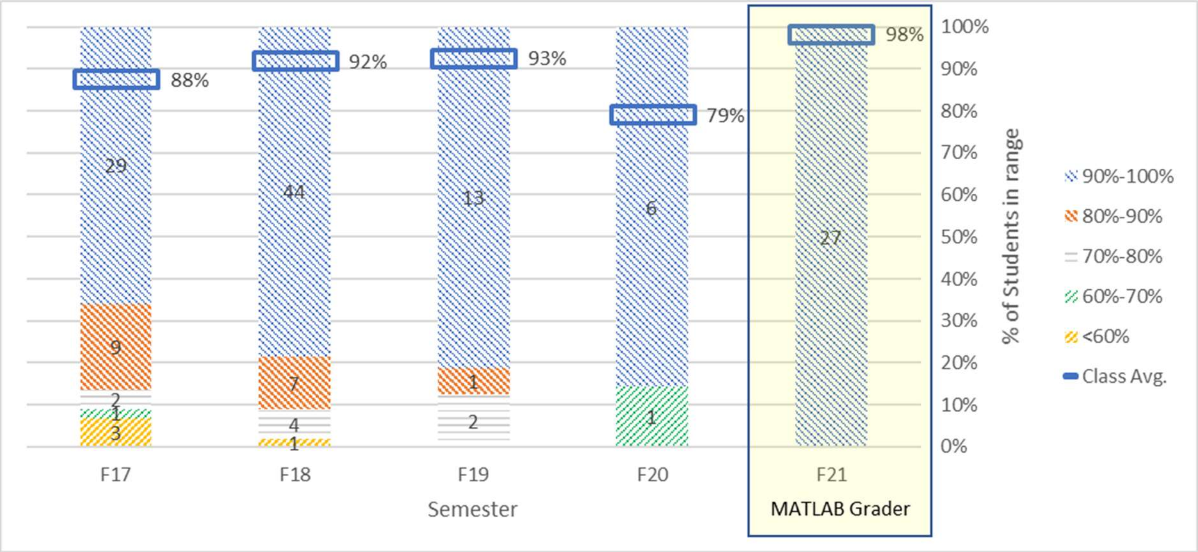


Figure 8, Signals and systems lab grades improved with MATLAB Grader (F21)

Student Feedback

After receiving approval from the university's Institutional Review Board, at the end of the semester, students were given extra credit to take an anonymous survey concerning their experiences with MATLAB Grader and live scripts or do an alternate task. Students were asked to rank statements on a Likert scale (Strongly Agree, Somewhat Agree, Neither Agree nor Disagree, Somewhat Disagree, and Strongly Disagree) the statements shown in Figure 9. In addition, they were given a space to comment. The top half of the figure shows the students' opinions of MATLAB Grader in both Vibrations and Signals and Systems. The bottom half shows their responses about MATLAB live scripts. The percentage of students with a positive response (Strongly Agree or Somewhat Agree) to each statement is indicated in parenthesis. Overall, both MATLAB Grader and live scripts were well received by the students.

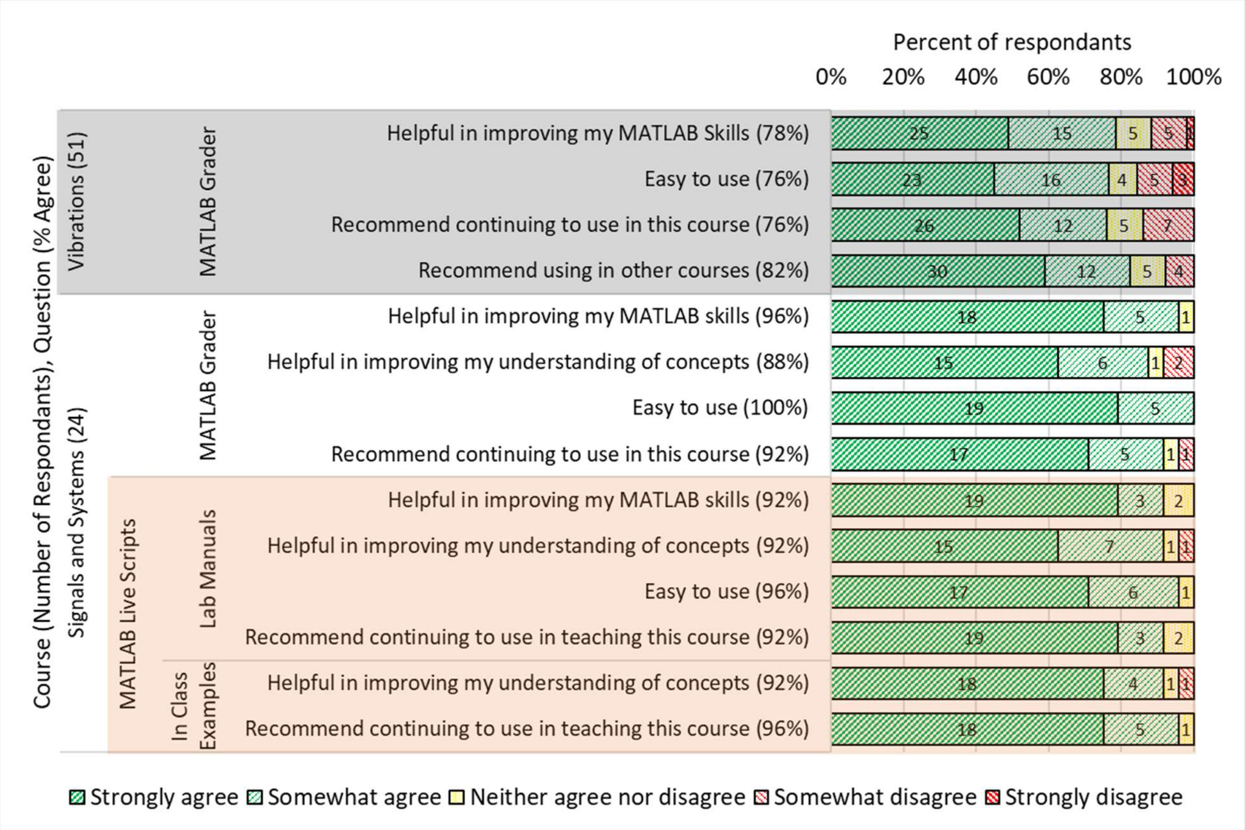


Figure 9, Survey responses concerning MATLAB Grader and MATLAB live scripts were positive

Below are some sample comments in Vibrations

- Matlab grader is a great concept, but I am not proficient enough in Matlab to where the grading tool is beneficial/helpful. Maybe spending part of a lecture reviewing Matlab commands/tips/syntax at the beginning of the semester would help clarify a lot of the initial issues students run into.
- Having the Matlab grader questions were the most helpful to me.
- As a person that struggles with MATLAB, MATLAB grader made it seem a bit easier since some of the code was provided.

Below are some sample student comments from the survey in Signals and Systems

- The grading system is very efficient. This structure for labs allowed greater student independence and helped boost student learning by not penalizing incorrect attempts.
- I really enjoyed using MATLAB Grader. The example exercises really helped me to understand the material better and work through the problems.
- Using Matlab grader was extremely helpful, and made the lab much more enjoyable, I also learned quite a bit about both signals and systems, as well as Matlab.
- I loved the use of MATLAB and Grader was extremely easy to use and get immediate feedback on my code. I would love it if more of the homework was in Grader, so that we can troubleshoot our answers and do more real-world application. I also think the live

> script examples for lab were extremely helpful but I would like to see a few more examples for each section because many times drawing conclusions about certain parts of code was really difficult with only one example and no explanation. Or instead, little explanations for each line of code in the example so that we know the why.

Discussions and Conclusions

MATLAB Grader has been introduced for homework problems in a Vibrations course and lab exercises in a Signals and Systems course. MATLAB Grader provides immediate feedback to the students, and the students had favorable opinions of MATLAB Grader. Although initially the time to develop MATLAB Grader questions is longer, they can be reused in future semesters, reducing the time of grading for the instructor.

In addition to MATLAB Grader, live scripts were used in teaching both the lecture and lab for Signals and Systems. Live scripts allow active MATLAB code to be embedded into a document to show the code within descriptive text. Ideally, the MATLAB exercises in Grader and the lab manuals in live script should be integrated in a MATLAB Onramp tutorial style, which embeds the exercises in a live script and allows Simulink exercises as well. Currently, this is not possible, but we hope to see this feature in the future release of MATLAB.

Overall, student responses concerning MATLAB Grader in both courses and MATLAB live scripts in Signals and Systems were positive. In the future offering of the Vibrations course, the MATLAB Grader problems can be improved to check the plots of the students more rigorously, rather than just checking to see if they have plotted the data. MATLAB Grader and live scripts are also introduced into the Control Systems course currently offered at the university in Spring 2022.

References

[1] R. S. Pettit, J. D. Homer, K. M. McMurry, N. Simone and S. A. Mengel, "Are automated assessment tools helpful in programming courses?," in *2015 ASEE Annual Conference & Exposition*, Seattle, WA, 2015.

[2] C. Cohenour and A. Anjum, "Automated Grading of Access® Databases Using the Matlab® Database Toolbox," in *2017 ASEE Annual Conference & Exposition*, Columbus, OH, 2017.

[3] N. Smith, "Integration of Instructional Technology Tools Including Matlab Grader to Enhance Learning in a Hybrid Vibrations Course," in *2020 ASEE Virtual Annual Conference Content Access*, virtual, 2020.

[4] R. M. Reck, "Student and Faculty Reflections After Using MathWorks' Cody Coursework in a Control Systems Course," in *2018 IEEE Frontiers in Education Conference (FIE)*, San Jose, CA, 2018.

[5] A. W. Gregg, O. Nicholas, T. Farnworth and C. Renton, "Automatic Assignment Marking using MATLAB Grader and Offline Unit Testing Code," in *Proceedings of the AAEE2020 Conference*, Sydney, Australia, 2020.

[6] C. Xu and S. H. Mousavinezhad, "Computer and Information Technology Tools in Signals and Systems," in *2016 ASEE Annual Conference & Exposition*, New Orleans, LA, 2016.

[7] J. Song, D. E. Dow and L. Ma, "In-Class Laboratory Exercises to Improve a Signals and Systems Course," in *2019 ASEE Annual Conference & Exposition*, Tampa, FL, 2019.

[8] M. F. Aburdene, R. J. Kozick and J. S. Baweja , "Interactive Signals and Systems Laboratories Using Notebooks," in *2000 ASEE Annual Conference & Exposition*, St. Louis, 2000.

[9] Kosfelder, "Function to evaluate plots (MATLAB® Grader)," 2022. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/78011-function-to-evaluate-plots-matlab-grader.