# Student and Faculty Reflections After Using MathWorks' Cody Coursework in a Control Systems Course

Rebecca M. Reck, *Senior Member, IEEE*
Mechanical Engineering Department
Kettering University
Flint, Michigan, USA
rreck@kettering.edu

***Abstract*—This innovative practice work in progress paper examines the student and faculty reactions after a pilot implementation of Cody Coursework for submission of MATLAB scripts in a senior level dynamic systems and control systems course. The goal in implementation was to increase students' use of scripts in MATLAB and reduce the amount of time instructors spend grading open-ended design problems. The pilot was evaluated with reflections from the faculty team shared in a post-course discussion. Students in the course were also asked to complete a survey after the course. Additionally, scores on two assignments from before the pilot study were compared to scores during the pilot. Based on the pilot implementation, it was determined that Cody Coursework reduced the grading load of the instructors and the students were generally satisfied with the tool.**

***Keywords—assessment, education technology, feedback, automatic grading***

## I. Introduction

New technologies, such as MathWorks' Cody Coursework, can add value to a course if they help students achieve the course learning objectives and/or reduce the instructor workload. However, instructors and students must first learn the new technology and overcome resistance to change before achieving the desired outcomes.

The senior-level course in this study already incorporated a significant amount of work in MATLAB and Simulink into the laboratory and lecture. Feedback from previous courses indicated the students wanted to have more practice using MATLAB and Simulink. The instructors desired a faster way to grade MATLAB assignments, especially open-ended design problems. Based on this feedback, the researcher looked for technologies that could address both the student and instructor feedback. MathWorks' Cody Coursework seemed to be a plausible solution because it provided a means to give automated feedback on student MATLAB scripts, which could provide the opportunity for more practice and reduce the grading workload at the same time.

In this study, the we sought to understand the benefits of adding automatic grading into the course. Before incorporating Cody Coursework into the course, we consulted MathWorks experts and training materials. We also found examples of problems from similar courses where other faculty had successfully used Cody Coursework [1-4]. Then, we collected field notes from the instructors in order to understand the instructor learning curve and distill best practices for assignment development moving forward. After the course ended, we asked students to complete surveys to gather their impressions of three different assignment types: paper submission, online submission to Blackboard, and online submission to Cody Coursework.

Based on initial analysis Cody Coursework did reduce the instructor workload for grading. However, there was some additional workload while learning the tool and overcoming bugs in the software. The students had a learning curve to overcome to submit assignments in Cody Coursework as well. Some of this was improved by changing the assignments' instructions and code templates based on feedback along the way. Overall, the first time using the tool went as well as expected with no major complaints from students or instructors.

## II. Background

Both [5] and [6] emphasize creating assessments that align with the course learning goals and objectives. Svinicki and McKeachie [5] suggest that assessments should be learning experiences and evaluate understanding. They also mention that providing feedback is more important than the grade itself [5]. Feedback should be understandable, selective, specific, timely, contextualized, nonjudgmental, balanced, and transferable [5]. Suskie [6] suggests using a variety of assessments to construct a clear understanding of student learning.

Felder and Brent [7] suggest that instructors should consider technology that aligns with the course objectives and provides feedback to the student and instructor about student attainment of the objectives. Svinicki and McKeachie [5] warn faculty to use careful thought and planning when adopting instructional technologies. They suggest that planning includes consideration of course content, capabilities of the tool, student access and comfort with technology, and the instructor's view of the role of technology in the teaching and learning process [5]. One example of an instructional technology is a software-checker for automatically grading and providing feedback on programming assignments [3, 4, 8, 9]. There are several commercial and instructor-developed software checkers [3, 10, 11]. These tools can decrease grading time and provide timely feedback to

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

students [3, 4, 8, 9]. New commercially-developed tools can save additional development time of student interfaces and grade calculations [3, 4, 12]. Many such examples in the literature focus on the implementation of automatic graders rather than the student response and relative performance.

## III. Course Integration

I piloted Cody Coursework in four assignments for MECH430: Dynamic Systems II in the summer term of 2017. The detailed course context, motivation, and assignment details before and after the pilot are described in this section.

### A. Course Context

MECH430 is a senior-level, required course in dynamic systems modeling and control system design. The course is 11 weeks long, including finals. Each week there are two 90-minute lectures and one 2-hour laboratory session. One of the course objectives states that by the end of the course students should be able to simulate the engineering system performance using accepted computer software, such as MATLAB and Simulink.

The course assignments include quizzes (take-home and in-class), pre-lab assignments, laboratory experiments, and a group project. To complete the pre-lab assignments, laboratory experiments, and the project, students are required to use MATLAB and/or Simulink. Most of the take-home quizzes also require the students to use MATLAB and/or Simulink. Prior to enrolling in MECH430, most students have only used MATLAB and Simulink in one other mechanical engineering course. On the first day of MECH430, most students rated their experience with MATLAB and Simulink as three or lower on a scale of 1 (no experience) to 5 (very experienced).

### B. Motivation

After teaching MECH430 for several terms, I identified several areas for improvement; two of them were related to MATLAB. First, the design problems in assigned for students to complete took a significant amount of time to grade. Second, several students asked to learn more about MATLAB and Simulink after completing the course. Therefore, I looked for ways to incorporate more MATLAB and/or Simulink in the course without increasing grading (or preferably, actually decreasing) the grading for the additional MATLAB content.

I looked for options to incorporate technologies that could support automatic grading and additional student exposure to MATLAB and/or Simulink. The best solution I identified was Cody Coursework, a website from MathWorks that supports automatic grading, immediate feedback for students, and self-directed learning [3, 8, 9]. From a student's perspective, the advantages include its integration with their existing MathWorks online account and it accessibility on any device. From the instructor's perspective, it provides an intuitive user interface and is supported by MathWorks; as an experienced MATLAB user I found it to have a low learning curve.

### C. Assignment Creation

Cody Coursework has two types of problems: functions and scripts [13]. For all of my pilot assignments, I selected the script submission because we do not cover functions in the course. The automatic assessment is split into individual tests that can be made available as pretests or not (hidden) [13]. If a test is made available as a pretest, students may check their script with the pretest as many times as they would like and they will see the code of the test as well as the feedback. If the test is not available as a pretest (hidden), students will see if they passed or failed the test and any feedback that is included for failed tests. The hidden tests are run when the student submits their script for assessment and will count toward their total number of submissions. There are four types of tests to select from: variable equals reference solution, function or keyword is present, function or keyword is absent, and MATLAB code.

After reviewing the information provided by the MathWorks, I identified four assignments as candidates for piloting Cody Coursework: laboratory experiment 6, take-home quiz 6, pre-lab assignment 7, and take-home quiz 7. Each of these assignments already contained design problems in MATLAB and/or Simulink and were among some of the most time consuming to grade. I decided to make the first assignment part of a laboratory experiment so that students would have an opportunity to use the tool with a laboratory instructor there to help if needed. The other three assignments were completed outside of class; however, students were able to come see the lecture and laboratory instructor during office hours.

#### 1) Laboratory experiment 6 and take-home quiz 6

In the second part of the sixth experiment, students are asked to design two simple filters based on what they have learned about Bode plots. In previous terms, students completed their filter design and provided the final transfer function in the laboratory worksheet. In many cases, it was difficult to grade these transfer functions without generating the Bode plot myself. Take-home quiz 6 also had two similar filter design problems.

For each of these filter design problems, an input that consisted of the sum of three or four sinusoids at different frequencies was given along with the desired range for the output amplitude at steady-state. Each filter design was a separate problem in Cody Coursework. There was a pre-test that plotted the Bode plot of their filter so that they could visually inspect the filter shape before submitting their script. The hidden tests checked the magnitude at each frequency to determine if the output amplitude would meet the specification.

#### 2) Pre-lab assignment 7

In pre-lab 7, students were asked to design a proportional + derivative controller with a specific range for settling time and maximum percent overshoot (MPO). In previous terms, I asked students to provide a plot of the closed-loop response, so that I could visually inspect the response to determine if the design met specifications. For cases that were hard to verify visually, I had a script which I could plug in the student's value for each gain and then determine if the MPO and settling time met specifications. This script made the tests easy to develop in Cody Coursework.

#### 3) Take-home quiz 7

In take-home quiz 7, students design a lead-lag controller (root locus design) to meet specifications. While this problem was easier to grade, students made several mistakes on this problem and I thought there might be a benefit to the quick

feedback provided by Cody Coursework that might help eliminate their errors. In this problem, the pre-test showed the student the root locus of their new system and the desired pole location so they could visually inspect that the root locus of their system passed through the desired pole locations of the final system. The hidden tests verified the values of the intermediate calculations that were graded in previous terms. The problem also required a plot of the final closed loop system, so there was also a test that checked if specific functions like step, title, plot, xlabel, and ylabel were present in the script.

*4) Assignment logistics*

For each assignment, I provided a written description on paper, the same description no the problem page in Cody Coursework, and a template script with all of the variable names that were required for the tests in the assessment. The template script also included comments with each variable and hints on the overall structure for the required parts of the script.

Both laboratory-related assignments were set to allow unlimited submissions, however, each quiz had a limit of three submissions per design problem. After the due date, I downloaded the "last, best solution" report for each problem. The report included the number of total tests passed. To convert the number of passed tests into a grade, I created a linear equation that related the number of tests passed to the total number of points for the assignment.

## IV. Preliminary Results

After the pilot, I sent a survey to the students and spoke with the other laboratory instructor for the summer term. She also has experience teaching the lecture portion of the course, so we compared each of the assignments before and after from the instructors' perspective.

### *A. Student Survey and Reflections*

Seven of the 33 students in the summer term responded to the survey. One set of Likert statements started with the prompt, "What are your impressions after using Cody Coursework?" The Likert scale ranged from 1 (strongly disagree) to 4 (strongly agree). The response to each statement in that category is shown in Fig. 1. Overall, the response from students was positive. One student stated, "I think Cody is a good addition." Another student that responded indicated that he was very frustrated with Cody Coursework. However, he also indicated a general frustration with computers as well, so it is likely he would have also been frustrated with the previous assignments.

Two students indicated that having multiple attempts on problems with the ability to fix mistakes was helpful. One survey participant specifically said, "The feedback on why it wasn't working / what it needed was great." Although one student indicated that on the problems with limited submissions it was stressful if you did not have the correct answer and were down to the last submission.

### *B. Quiz Grades Before and After Introducing Cody Coursework*

After the pilot, I was curious to see if the grades differed between terms. I only compared the two quizzes, because I graded all of the quizzes in both terms, which eliminated the variable of multiple instructors grading laboratory-related assignments. Each quiz had 10 points and the points allocated for the MATLAB design were the same in both terms. The total scores were used for statistical analysis because that was the only data available for the spring term.

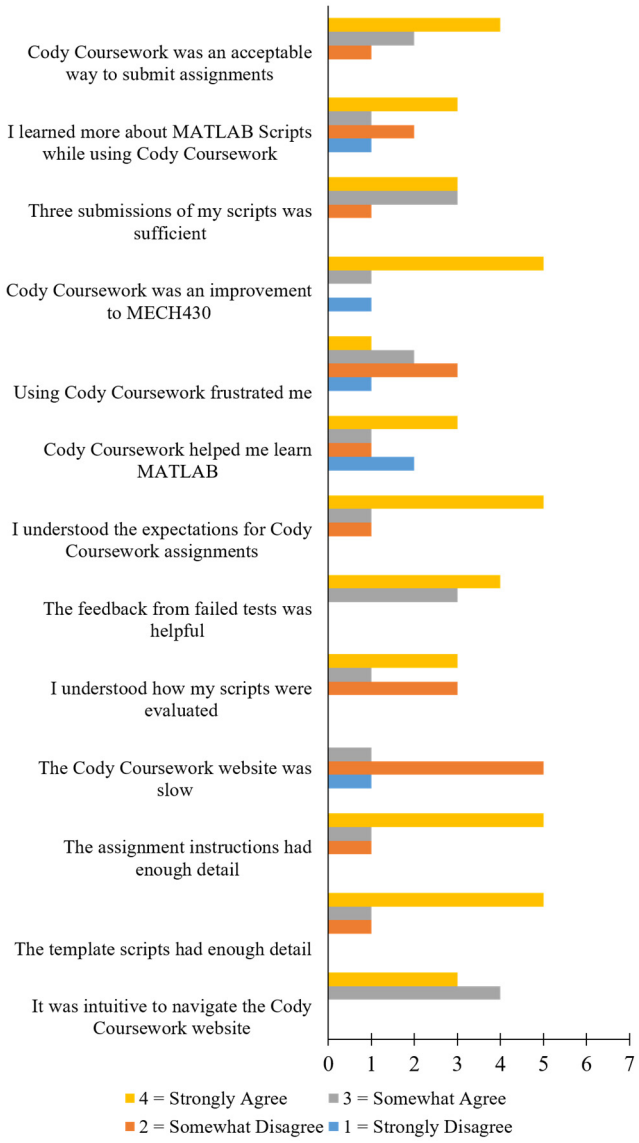


Fig. 1. Student reactions to using Cody Coursework

I started with basic descriptive statistics for both quizzes in the spring 2017 term (without Cody Coursework) and the summer 2017 term (with Cody Coursework). Since the quiz scores were not normal, I used a Wilcoxon-Rank Sum test to compare the means of each quiz. The statistics and p-value are shown in Table 1. With an $\alpha = 0.05$, the null hypothesis of equal means cannot be rejected for quiz 7. However, the null hypothesis of equal means can be rejected for quiz 6. On quiz 6, the five points that were not allocated to MATLAB designs were the same for both terms. However, the allocation of the 5 points for the MATLAB design was different between the terms, so it

is hard to say that the difference in scores between terms on quiz 6 represents a difference in understanding of the material.

Table 1 - Quiz statistics

| | Spring 2017[a] | | | Summer 2017[b] | | | Wilcoxon Rank Sum |
|---|---|---|---|---|---|---|---|
| | *n* | *M* | *SD* | *n* | *M* | *SD* | *p-value* |
| Quiz 6 | 50 | 8.2 | 1.7 | 33 | 8.8 | 1.9 | 0.02 |
| Quiz 7 | 49 | 7.2 | 2.5 | 32 | 7.6 | 2.3 | 0.50 |

[a] without Cody Coursework
[b] with Cody Coursework

### C. Faculty Reflections

After the end of the pilot, the instructors discussed lessons learned and observations during the term. We discussed the learning curve, student behavior, cheating, problems that happened, and thinking ahead for reuse in subsequent terms.

#### 1) Learning curve

There was a learning curve for both students and instructors in the first term. The students were learning a new interface for submitting assignments. I think the first assignment in lab helped them build confidence with the tool prior to individual assignments. The learning curve for the laboratory instructor responsible for helping students was minimal as she was an experienced MATLAB and Simulink user.

As the instructor who created all of the problems, I had the biggest learning curve, since I had not generated automated tests before. One major problem was the use of the break command in a while loop. In the script I ran to check the control design for pre-lab 7, I used a while loop to determine the settling time from simulated data. However, using a break command in Cody Coursework caused the entire site to stop responding. In general, finding the right parameters to test and determining how to test them required thought and creativity.

#### 2) Student behavior

There were two notable student behaviors that should be considered: using tests for debug and not reading comments. When I limited the number of tests, I had hoped that students would develop their design offline in MATLAB and then copy their solution into the Cody Coursework website to submit and run the tests. In reality, it seems that most students developed their scripts on the Cody Coursework website and used the tests (both pre-test and hidden tests) as a debug tool. I think this behavior created some of the stress and frustration with the limited submissions.

In my first assignment, I put most of the detail about each required variable in comments in the template script provided on the website. Based on discussions with students during the term, it seems that there was some confusion about my expectations for each variable when the information was only provided in the assignments. In the subsequent assignments, I put descriptions of each variable in the comments of the template script as well as the paper and electronic assignment descriptions. The extra description seemed to provide more clarity about expectations.

In the future, I would like to include a suggested workflow and more clear expectations on grading when I introduce Cody Coursework as a tool in the course. I may also consider raising the limit of submissions from three to five.

#### 3) Cheating

Based on some simple inspections of the code, I have found scripts submitted on the Cody Coursework website that were too similar to be developed independently. Currently, there is not a built-in plagiarism detector for Cody Coursework. Therefore, I have been searching for other tools to help identify plagiarism and collaboration in subsequent terms. Other instructors have also explored several options to discourage cheating [4].

#### 4) Problems that occurred during the pilot

Two major problems occurred during the pilot. The first problem was with the report generation in Cody Coursework. I was advised by MathWorks staff to use the pre-release version of Cody Coursework V2 because it would be easier to reuse problems going forward. The reports I downloaded did not associate the submitted scripts with the correct student. I contacted the MathWorks and they were able to generate the correct reports and send them to me. That glitch has been fixed in the final release of V2.

The second problem was my own attention to detail when copying and pasting between tests. On one of the filter problems, I was checking the wrong frequency for one of the amplitude requirements. That meant that all of the students improperly failed that test by no fault of their own. Some of the students were frustrated or confused by their results. Unfortunately, none of them asked about the failed case until after the assignment was due. Cody Coursework does allow the instructor to update a test and reevaluate the submitted solutions. Also, when downloading a report for grading, Cody Coursework has the option of downloading the last, best solution for each student so that if an earlier submission had passed the updated version of the incorrect test, the student would still get credit. Fortunately, there is a way to recover from these typos for grading and now I am more careful in double-checking each test before the assignment goes live.

#### 5) Thinking ahead

Assignments can be copied from one course to another, so it can be helpful to think ahead to what might change in an assignment if the problems are reused in a subsequent term. I tried to design template scripts to be generic for each type of problem. I also used extra variables in the reference solution to make it easier to change numbers in the problem statement each term. These small additions during the pilot made the questions much easier to reuse in a subsequent term.

## V. Conclusions and Future Work

In general, I considered the pilot test of using Cody Coursework a success. I have learned from the problems and gathered feedback from students to incorporate when I use Cody Coursework again. It did reduce the time I spent grading. The initial setup was time consuming, but I suspect that will decrease with continued use. Based on the student feedback it seems that using Cody Coursework is at least as good as the previous assignment design from the student perspective. I also plan to gather feedback from students in subsequent terms to continue to improve the feedback provided on the pilot assignments and to introduce additional assignments.

## ACKNOWLEDGMENT

The author would like to thank Dr. D. Peters for her support of this research.

## REFERENCES

[1] J. P. Hoffbeck, H. E. Dillon, R. J. Albright, L. Wayne, and T. A. Doughty, "Teaching programming in the context of solving engineering problems," in *2016 IEEE Frontiers in Edu. Conf. (FIE)*, 2016.

[2] M. A. Verleger, "Just Five More Minutes: The Relationship Between Timed and Untimed Performance on an Introductory Programming Exam," presented at the 2016 ASEE Annu. Conf. & Exposition, New Orleans, LA, 2016.

[3] R. C. Hill and Y. Parvini, "Automated Grading with a Software-Checking Program in the System Dynamics and Control Curriculum," presented at the 2018 Amer. Control Conf. (ACC), Milwaukee, WI, 2018.

[4] B. W. Char, I. Daulagala, N. Kandasamy, and T. T. Hewett, "Using Automatic MATLAB Program Testing for a First-Year Engineering Computation Course," presented at the 2016 ASEE Annu. Conf. & Exposition, New Orleans, LA, 2016.

[5] M. D. Svinicki and W. J. McKeachie, *McKeachie's Teaching Tips: Strategies, Research, and Theory for College and University Teachers* vol. 14: Wadsworth Cengage Learning, 2014.

[6] L. Suskie, *Assessing Student Learning: A Common Sense Guide*: Jossey-Bass, 2009.

[7] R. M. Felder and R. Brent, *Teaching and Learning STEM: A Practical Guide*, First ed. San Francisco, CA: Jossey-Boss, 2016.

[8] D. G. T. Antunes. (2016). *Automating Grading of Assignments in a MATLAB Programming Course*. [Online]. Available: https://www.mathworks.com/company/newsletters/articles/automating-grading-of-assignments-in-a-matlab-programming-course.html

[9] R. A. Canfield. (2015). *Teaching Computational Methods to 140+ Second-Year Engineering Students at Virginia Tech (v2 ed.)*. [Online]. Available: https://www.mathworks.com/company/newsletters/articles/teaching-computational-methods-to-140-second-year-engineering-students-at-virginia-tech.html

[10] T. Staubitz, H. Klement, J. Renz, R. Teusner, and C. Meinel, "Towards practical programming exercises and automated assessment in Massive Open Online Courses," in *2015 IEEE Int. Conf. on Teaching, Assessment, and Learning for Eng. (TALE)*, 2015, pp. 23-30.

[11] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppälä, "Review of recent systems for automatic assessment of programming assignments," in *Proceedings of the 10th Koli Calling Int. Conf. on computing Edu. Res.*, 2010, pp. 86-93.

[12] R. Green, "Mastery Learning With The Matlab Webserver," presented at the 2000 ASEE Ann. Conf., St. Louis, MO, 2000.

[13] MathWorks. (2018). *Documentation: Cody Coursework*. [Online]. Available: https://www.mathworks.com/help/coursework/