A Second Language Acquisition Approach to Learning Programming Languages

Rachel Cunningham, Paula Sanjuan Espejo, Dr. Christina Frederick, Dr. Lulu Sun, Dr. Li Ding
Embry-Riddle Aeronautical University

Abstract

The instructional design for modules in the study was based on the evidence that learning a programming language is analogous to students acquiring a second language, and utilized tools from Second Language Acquisition (SLA) theory. A programming language has vocabulary, syntax, grammar and communicative outcomes that must be sufficiently developed for the learner to function successfully in the environment that utilizes the language. This proposed study utilized an SLA approach to programming language in a blended learning environment. Modifications to the course pedagogy included breaking the course topics into video lessons focusing on basic programming vocabulary, grammar, and syntax. These videos had opportunities for the students to practice new commands and dynamically apply the grammatical programming rules introduced in the lesson. Student performance in sections using the SLA approach will be compared with that of students in unaltered programming sections using student survey responses, class participation and course grades.

Keywords

Second language learning, engineering, programming.

Introduction

During the first year it is mandatory for engineering and computer science students to take a computer programming course. Within these courses students learn programming skills by utilizing common programming languages, such as MATLAB, C, C++, or Java. These courses are also important in teaching students basic applied problem solving within computer programming[1-4]. The amount of time practicing and applying the skills and techniques introduced in computer programming courses is often not enough for high-school and college students[5] to fully understand the conceptual complexity and logical reasoning processes that are required. Without adequate practice students find the course difficult which can, in turn, impact the success of students in STEM majors. The problem solving skills developed within a first year computer programming course are intended to be transferrable to other contextually relevant courses, such as in the math and science courses that these students must take. A better understanding of these skills, as well as how to apply them will better prepare students in succeeding, both in higher education and other contexts.

The instructional design for modules in the study was based on the premise that learning a programming language is analogous to students acquiring a second language, and utilized tools from Second Language Acquisition (SLA) theory. A programming language has vocabulary, syntax, grammar and communicative outcomes that must be sufficiently developed for the learner to function successfully in the environment that utilizes the language and effectively communicate with computers[5-9]. Although learning the vocabulary, grammatical structures, and punctuation builds knowledge about a language, this knowledge does not make one fluent in a spoken language. Similarly, it is necessary to have more understanding of a language beyond simply having rote-knowledge of programming rules and structures in order for a programmer to be successful. Typically, beginning courses for computer programming focus mainly on these rote elements of the programming language and do not emphasize or provide an adequate amount of problem solving practice[10]. This proposed work tests the hypothesis that the use of cognitive frameworks in second language acquisition for the development of a blended learning of programming languages can improve engagement and the learning experience of engineering students. Engagement and the student's learning experience is intended to be enhanced through the use of online components that continue connecting the students with each other and the material outside of classes[11], in addition to better preparing students for face-to-face classes and labs. Using this SLA approach to facilitate a Blended Learning experience (SLAaBLe) places greater emphasis on problem solving techniques that can be utilized in all courses and assists students in developing the skills needed to be successful for a higher learning.

Previous Work

At a private institution in the southeast, Introduction to Computing for Engineers (EGR115) is offered as an introductory course in computer programming, required for many students in various STEM majors. This introductory course is found to be one of the most difficult for students due to the lack of time to practice the material. The lack of adequate applied practice, along with the difficulty in understanding algorithms essential to programming, results in students with deficient comprehension of programming fundamentals. Based on feedback from both students and faculty, this course went through a number of iterations. In fall of 2009 a significant shift in the course structure involved changing from programming in C to programming in MATLAB, because many math and science disciplines utilize MATLAB as a major language for problem solving[12-14]. In addition to switching programming languages, the course structure changed from meeting three days a week, instead meeting two times for lecture and two times in a laboratory with hands-on work. Each lecture occurred for one hour in a large auditorium where topics were introduced and typically included 120 students. The lab sessions were more intimate, consisting of about 26 students, to allow for more one-on-one contact with the instructor, who assisted with problems and guided the practice for one hour. Following these course alterations feedback primarily indicated concern about attendance and effectiveness of the large lecture sessions, and the impact on the connection between students and the instructor[15].

To address the concerns regarding attendance and engagement, a blended learning approach was adopted the following year by almost half of the EGR115 course sections, with the hope of seeing improvement in student learning outcomes[16-18]. The two lab times remained unchanged and continued as one-hour sessions for practice. The difference between the traditional and blended course structure came in the form of altered lecture times for the blended courses. Instead of spending an hour in lecture, students within the blended courses were required to engage in approximately one hour of online self-guided study activities. These activities included PowerPoint lecture videos, an online discussion forum, and programming exercises and quizzes. The exercises and quizzes were required to be completed before the section met for a lab time, allowing students to discuss and ask questions about the activities, followed by additional hands-on practice. Even with this highly flexible access to learning materials, students were found to be intimidated by the complexity of understanding programming concepts and cognitive reasoning demands[19]. Students continued to be unprepared for lab sessions due to a lack of engagement and practice in online self-study activities and discussions. It was concluded that the anxiety and fear of students attempting to learn a programming language needed to be addressed with an online learning environment which is more compelling and engaging for students in math and science majors.

Method

The structure of the introductory computer programming course, EGR115 has been altered beyond the previous blended learning approach that was being utilized. In order to improve student learning outcomes and help students become more comfortable and engaged in the material a second language acquisition approach to learning computer programming has been adopted in three of the seven EGR115 sections for the fall 2015 and spring 2016 semesters. The sections utilizing the new approach are referred to as SLAaBLe sections while those which have continued to utilize a blended learning approach are referred to as non-SLAaBLe sections. SLA is composed of five stages. Stage 1) *preproduction*, in which individuals develop minimal comprehension, stage 2) *early production*, in which individuals develop limited comprehension, stage 3) *speech emergence*, in which individual develop increasing comprehension, stage 4) *intermediate fluency*, in which individuals develop very good comprehension, and stage 5) *advanced fluency*, in which individuals have a very high level of comprehension. In each of these stages different conceptual and cognitive skills are focused on and developed.

Each of these five stages have been adapted to the SLAaBLe sections so students can understand the similarities between SLA and computer programming language acquisition. The preproduction stage is created in these sections with visuals like images, objects, and animations to encourage basic or minimal levels of comprehension. To form early production skills multiple-choice and short answer questions are part of the online lessons and visuals. To support the limited comprehension developed by students an online discussion is available for students to post thoughts and questions about the lessons. Two student research assistants support the students in these discussions. Students are encouraged to engage in open conversation and contribute to the discussion as well as help each other and answer questions about the course material that other students cannot. The speech emergence stage occurs during lab practice on the day after students have worked through the first two stages in the online environment. Assignments during lab focus on increasing the student's comprehension by engaging in problem solving activities related to the topics. These assignments require the students to speak and do and are guided by a peer instructor who assists in rectifying incorrect thinking. Exercises at this stage include a group discussion along with a "think, pair, share" activity to help students advance to the next stage of acquisition. More complicated homework assignments are given following the guided exercises during lab sessions which the students complete individually, reinforcing the level of comprehension at the intermediate fluency stage. Advanced fluency, and a high level of competency and comprehension, is developed as the students work on and complete an open-ended final project. For this project the students are given guidelines and must choose a program to develop, utilizing knowledge acquired through working on the online and in-lab course assignments. Opportunities to present this project in the class allow students to express their level of comprehension of the course material as a whole.

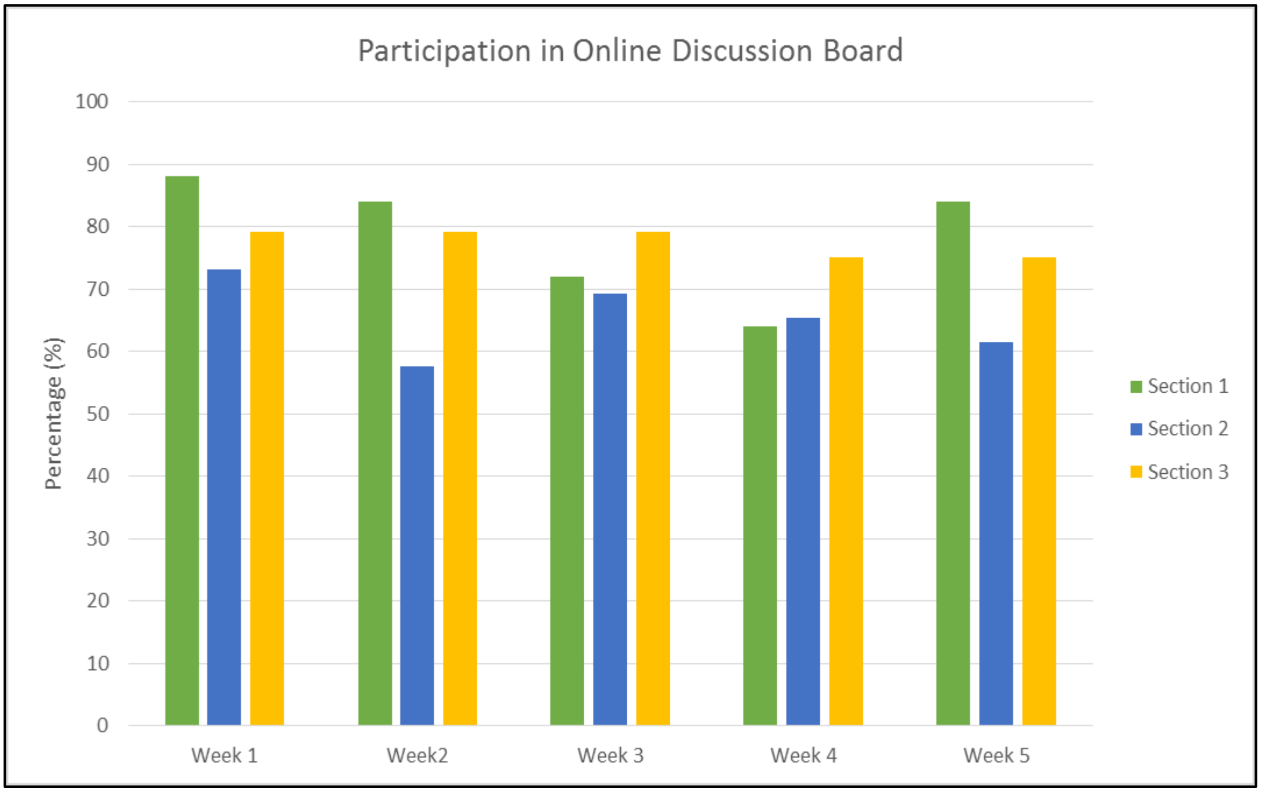
The five EGR115 course topics include introduction to MATLAB, data type, input and output, conditional statements, and loops. Each of these topics are designed to have students experience the first four stages of SLA while the fifth stage (the final project) comprehensively integrates all of the learned material so the students can illustrate fluency. The current SLAaBLe sections are testing the hypothesis that the use of cognitive frameworks in second language acquisition for the development of a blended learning experience of programming languages can improve engagement and how effectively engineering students learn the material. The current SLAaBLe course design places greater emphasis on developing problem solving skills. We expect that the qualitative and quantitative data collected during the study will help improve the implementation of the SLAaBLe in programming courses and the use of the cognitive frameworks in alternative settings.

Results

A preliminary analysis was conducted to determine the effectiveness of the online discussions in broadening the student participation. This analysis looked at how many students participated beyond the minimum participation required for a grade. Participation beyond the baseline was defined as any additional comments, questions, or answers to other students' posts. The participation from one of the SLAaBLe sections was collected and the total number of posts were recorded and compared to the number of posts that were beyond the minimum requirement. For the week 1 discussion, 52% of the 19 total students who posted at least the minimum requirement posted additional answers, comments, or questions. For the week 2 discussion there was the lowest total participation but of the 15 students who posted the minimum amount, 93% made additional posts. In the week 3 discussion, 33% of the 18 total students who made the minimum posts also made additional posts. Week 4 discussion had the lowest additional posts made (23%) by the 17 students who posted the minimum amount. In the final discussion, week 5, 25% of the 16 students who participated the minimum amount also made additional posts. Table 1 shows the percentage of students in each SLAaBLe section who participated in the five weeks of online discussion boards.

Table 1

Percentage of student participation in each of the three SLAaBLe EGR115 sections for Fall 2015



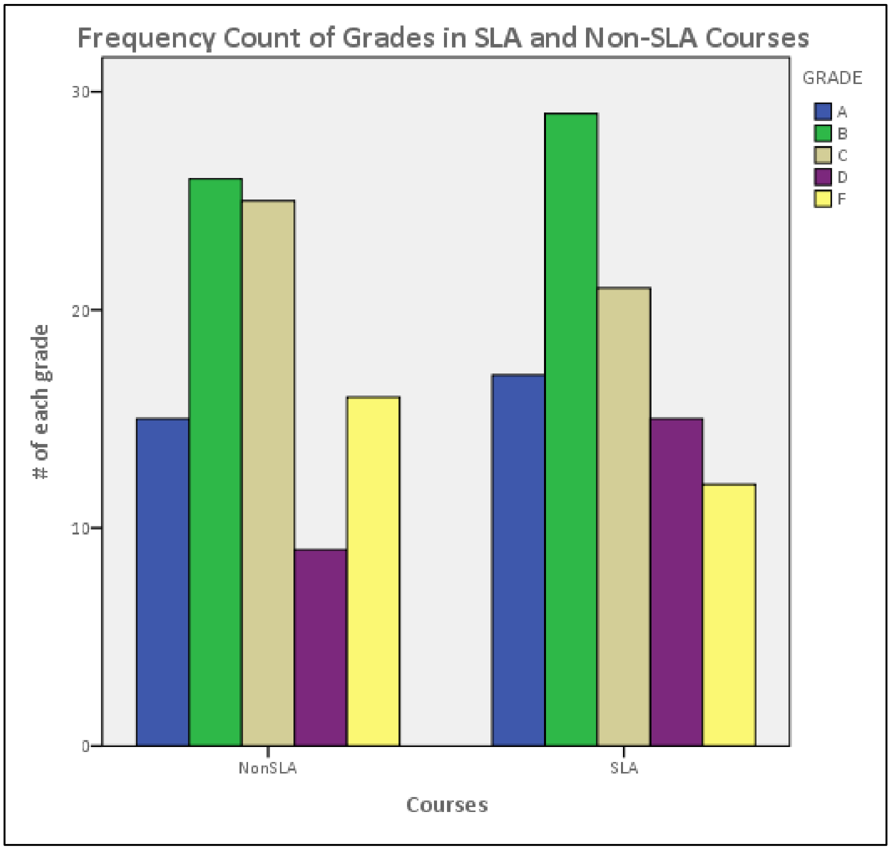
The online video and quiz lessons used for most topics were also analyzed for preliminary results. Of the 19 interactive videos utilized for this course, 17 included embedded quizzes related to the course topic for that week. Preliminary results indicate that almost all of the students who correctly answered all in-video questions had also watched 100% of the related video. Of the 17 video quizzes, only three students were able to correctly answer every question without watching 100% of the related video. It was also found that, on average, if a student answered any of the video questions correctly, then they had watched 100% of the video.

Students from the SLAaBLe course sections were interviewed for feedback regarding the course design. From these interviews it was suggested that the SLAaBLe course sections were effectively designed. Students indicated more engagement and comprehension of the online video and quiz lesson format, compared to the topics that were presented in a traditional non-video slide format. Students indicated a desire for the online video and quiz format be applied to all topics, stating that the video format was clearer. Students also commented on the laboratory sessions, indicating that the in class collaboration with a partner was helpful to learning and comprehension. Students also suggested that multiple short quizzes after each video were preferred to longer quizzes after the end of the entire module. This is believed to be desired because the students are able to test and check their comprehension of the topic throughout the module, rather than maintaining any incorrect or uncertain understanding until the end of the module.

A chi-square test of independence was calculated comparing the final grades of students in the SLAaBLe sections and non-SLAaBLe sections of the EGR115 course. There was no significant relationship associated between the course section and final grade, ($X^2(4) = 2.660$. $p = .616$). Students within the SLAaBLe sections did not score higher in the class than students in the non-SLAaBLe sections. Table 2 shows the bar chart of frequencies of final grades for the SLAaBLe and non-SLAaBLe sections.

Table 2

Comparison of students' final grades in the SLAaBLe and non-SLAaBLe sections for Fall 2015



Limitations

This study looked at the effectiveness of applying SLA strategies in an introductory computer programming course design. The qualitative results were positive; however, since students from only one university were investigated, these findings may not translate to introductory programming courses at other schools. Not every student in the SLAaBLe and non-SLAaBLe sections were able to be interviewed, limiting the amount and sources of feedback related to the courses. Also, because a portion of the results were based on self-reporting and feedback from students, they were subject to any bias the students' had regarding the course or professor. The results might still be widely applicable, as they will help guide effective computer programming course design at any school or university.

Next Steps

Feedback and data were collected after the Fall 2015 semester to assess effectiveness, and will be collected again after the Spring 2016 semester. After the results are collected and analyzed, those activities and modules which are found to be most effective in improving students' learning outcomes and programming language acquisition will be integrated into other course sections starting in the Fall 2016 semester. Following any changes made to the courses in this study will be recommendations for modifications to other programming courses at other institutions. The course modules will be disseminated to students and instructors who are either learning or teaching an introductory programming course to facilitate student learning outcomes. Likewise, the lessons learned from applying SLAaBLe frameworks in this context will be disseminated. This will provide a richer understanding of how SLAaBLe affects student's learning of the programming language.

Acknowledgements

The authors are grateful for the support provided by the National Science Foundation, Division of Engineering Education and Centers, grant number EEC 1441825. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would also like to thank and recognize the contributions from Professor Caroline Liron and Dr. Matthew Verleger on this project. Caroline Liron is an Assistant Professor in the Engineering Fundamentals Department, at Embry-Riddle Aeronautical University where she has been teaching since 2005. She currently teaches Introduction to Programming for Engineers and is involved in developing and maintaining the hybrid version of that class, and researching improvements methods to teach programming to incoming freshmen using new technologies. Dr. Matthew Verleger is an Assistant Professor of Engineering Fundamentals at Embry-Riddle Aeronautical University. His research interests are focused on using action research methodologies to develop immediate, measurable improvements in classroom instruction and the use of Model-Eliciting Activities (MEAs) in teaching students about engineering problem solving. He is an active member of ASEE and also serves as the developer and site manager for the Model-Eliciting Activities Learning System (MEALearning.com), a site designed for implementing, managing, and researching MEAs in large classes.

References

1 Bualuan, R., "Teaching Computer Programming Skills to First-year Engineering Students Using Fun Animation in MATLAB," Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL, 2006.

2 Devnes, P.E., "MATLAB and Freshman Engineering," Paper presented at the 1999 American Society for Engineering Education Annual Conference & Exposition, Charlotte, NC, 1999.

3 Morrell, D., "Design of an Introductory MATLAB Course for Freshman Engineering Students," Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI, 2007.

4 Naraghi, M.H.N. & Litkouhi, B., "An Effective Approach for Teaching Computer Programming to Freshman Engineering Students," Paper presented at the 2001 American Society for Engineering Education Annual Conference & Exposition, New York, 2001.
5 Solomon, J., "Programming as a Second Language," Learning & Leading with Technology, 32(4), 2004, 34-39.
6 Tran, L., "Computer Programming Could Soon Be Considered a Foreign Language in One State," Retrieved March 7, 2014, from http://www.policymic.com/articles/81067/computer-programming-could-soonbe-considered-a-foreign-language-in-one-state, 2014.
7 Tyre, P., "Is Coding the New Second Language?" Retrieved March 7, 2014, from http://www.smithsonianmag.com/innovation/is-coding-the-new-second-language-81708064/, 2013.
8 Van Roy, P., "The Role of Language Paradigms in Teaching Programming," Paper presented at the 34th SIGCSE Technical Symposium on Computer Science Education, New York, NY, 2003.
9 Wynn, M., "Ky. Ponders Teaching Computer Code as Foreign Language," Retrieved January 29, 2015, from http://www.usatoday.com/story/tech/2015/01/29/ky-computer-code-as-foreign-language/22529629/, 2014.
10 Victor, B., "Learnable Programming," Retrieved March, 7, 2014, from http://worrydream.com/LearnableProgramming, 2012.
11 Marsh, D., Blended Learning Creating Learning Opportunities for Language Learners, Cambridge University Press, 2012.
12 Azemi, A., Pauley, L.L., "Teaching the Introductory Computer-Programming Course for Engineering Using MATLAB and Some Exposure to C," Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL, 2006.
13 Bjedov, G. & Andersen, P., "Should Freshman Engineering Students be Taught a Programming Language," Paper presented at the Proceedings of the 26th Annual Frontiers in Education Conference, Salt Lake City, UT, 1996.
14 Herniter, M.E. & Scott, D.S., "Teaching Programming Skills with MATLAB," Paper presented at the 2001 American Society of Engineering Education Annual Conference & Exposition, New York, 2001.
15 Abbitt, J. & Carroll, B., "Using Technology to Enhance Undergraduate Learning in Large Engineering Classes," Paper presented at the 2013 American Society for Engineering Education Annual Conference & Exposition, Atlanta, GA, 2013.
16 Brown, C. & Meyers, D., "Experimental Hybrid Courses that Combine Online Content Delivery with Face-to-face Collaborative Problem Solving," Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI., 2007.
17 Brown, C., Lu, Y., Meyer, D., & Johnson, M., "Hybrid Content Delivery: Online lectures and Interactive Lab Assignments," Paper presented at the 2008 American Society for Engineering Education Annual Conference & Exposition, Pittsburgh, PA, 2008.
18 Yale, M., Bennett, D., Brown, C., Zhu, G., & Lu, Y., "Hybrid Content Delivery and Learning Styles in a Computer Programming Course," Paper presented at the 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, TX, 2009.
19 Sun, L., Kindy, M., & Liron, C., "Hybrid Course Design: Leading a New Direction in Learning Programming Languages" Paper presented at the 2013 American Society of Engineering Education Annual Conference & Exposition, San Antonio, TX, 2012.

Rachel Cunningham

Rachel is currently a doctoral student in the Human Factors Department at Embry-Riddle Aeronautical University Daytona Beach Campus. She received her B.S. in Landscape Architecture from the University of Kentucky. Rachel is a member of Psi Chi, an organizing committee member of Embry-Riddle’s Human Factors and Applied Psychology Student Conference, and Historian of Embry-Riddle’s Human Factors and Ergonomics Society student chapter. Recently, she presented a study on relatedness in gaming at the 2015 Southeastern Psychological Association Conference.

Paula Sanjuan Espejo

Paula is an undergraduate sophomore student in the Aerospace Engineering Program at Embry-Riddle Aeronautical University who was born in Spain. She is an Orientation Team Ambassador for the university, a member of the Honors Program, and a research assistant for the SLA-aBLE project in the Engineering Fundamentals Department. She is also Head of Mentors for the International Student Programming Council and is part of the Honor Student Association as well as of the structures and electrical team of Project Daedalus, the HAS research project, whose aim is to construct a 3D printing bending machine.

Christina M. Frederick, Ph.D.

Dr. Frederick is currently a Professor and Graduate Program Coordinator in the Human Factors and Systems Department at Embry-Riddle Aeronautical University in Daytona Beach, Florida. Dr. Frederick received her Ph.D. in 1991 from the University of Rochester with a major in Psychological Development. In 2000, Dr. Frederick joined the Human Factors and Systems Department at Embry-Riddle, where her work focused on applied motivation and human factors issues in aviation/aerospace. Dr. Frederick's current research interests examine how individual differences interact with technology to enhance educational engagement and performance.

Lulu Sun, Ph.D.

Dr. Sun is an Associate Professor of Department of Freshman Engineering at Embry-Riddle Aeronautical University, where she has taught since 2006. She received her B.S. degree in Civil Engineering from Harbin Engineering University (China) in 1999, and her Ph.D. degree in Mechanical Engineering from University of California, Riverside in 2006. Dr. Sun's Ph.D research, which was supported by UCR dean fellowship, USDA, UCR-LANL foundation, focused on pool fire experiments and numerical modeling of fire spread. The project of fire behavior in live fuels was ranked as best among 14 projects funded by the National Fire Plan of US in 2003.

Li Ding, Ph.D.

Li Ding has her Masters and Doctorate in Environmental Engineering from the University of Illinois at Urbana-Champaign, where she graduated 2004 with her MS and 2010 with her PhD. From 2011 to the present she has been an adjunct professor in the Department of Engineering Fundamentals & Department of Physical Sciences at Embry-Riddle Aeronautical University.