

## Using LEGO Mindstorms and MATLAB in Curriculum Design of Active Learning Activities for a First-year Engineering Computing Course

### Dr. Shelley Lorimer, Grant MacEwan University

Shelley Lorimer is an Associate Professor in Engineering (BSEN) Transfer Program at MacEwan University. She is an instructor in the introductory engineering courses as well. The BSEN program at MacEwan has grown from forty students since it started almost fifteen years ago, to the current 216 students. The majority of the students in the program transfer to second year engineering at the University of Alberta.

Shelley is a graduate of the University of Alberta in engineering and is a registered professional engineer with APEGA (Association of Professional Engineers, Geologists and Geophysicists of Alberta). Prior to her career at MacEwan, Shelley worked in industry as a research engineer and a consulting engineer for several years.

### Dr. Jeffrey A. Davis, Grant MacEwan University

Dr. Davis obtained his PhD at ETH Zurich specializing in multiphase flows and thermal hydraulics in nuclear reactors. With a passion for teaching, Dr. Davis' research focuses on pedagogical topics such as student engagement, active learning, and cognitive development. Projects he is currently working on include "Development of a risk assessment model for the retention of students", "Development of Student Assessment Software", and "Improving Student Engagement through Active Learning".

### Miss Olivia Tronchin

## **Using LEGO® Mindstorms and MATLAB in curriculum design of active learning activities for a first-year engineering computing course**

This paper is an **evidence-based practice** research study to improve course delivery in computer programming. Courses and materials in computer programming tend to be abstract, which can lead to many students having difficulties learning and being engaged with the material. With a more hands-on practical approach, students may find themselves immersed in the material and motivated to understand and apply concepts learned in class to real-world applications. Previous studies in the literature have shown that LEGO® Mindstorms can be used to enhance active learning for students, particularly when used to demonstrate computer programming concepts. However, many of the studies have typically been limited to design courses and not first-year specific. There is not substantial information showing a developed curriculum for a first-year engineering programming course. The current paper examines the feasibility of using robotics (specifically LEGO® Mindstorms) combined with computer programming (MATLAB) as it relates to the curriculum of a first-year engineering computing course, and how it can be implemented.

Specifically, the goals of this scholar activity were threefold: i) to investigate the literature to explore the use of active learning tools in first-year engineering education, ii) to determine the capabilities of the LEGO® Mindstorms platform as an “active learning” tool for first-year engineering and computer science students at MacEwan University and iii) to use the information gained to propose and test active learning lab activities for first-year programming courses. This research uses appreciative inquiry to examine the feasibility of using LEGO® Mindstorms EV3 robot with the MATLAB programming environment in our first-year engineering course. The functionality, specifically sensor and motor capabilities, of the robot was then compared with intended learning outcomes. A checklist of desirable curriculum learning outcomes was used as the rubric in the feasibility analysis. Motivation for the research was derived from the literature which showed potential for the use of robotics for active learning.

Results of this investigation have shown that LEGO® Mindstorms is a viable teaching tool for a first-year engineering computing course to develop fundamental programming skills and hands-on problem-solving skills. The feasibility study focused on the five sensors and motors that are available for the Mindstorms EV3 robot to design active learning activities using the MATLAB toolbox. The capabilities of the motors and sensors were found to be more than adequate to cover the first-year computing curriculum. Through the assessment of the various sensors, learning activities were designed to reinforce learning outcomes and provide students with interactive practical opportunities to apply their knowledge. A set of assignments was created and tested by a first-year engineering student and reviewed to by the course coordinator to ensure that they satisfied the learning outcomes of the course and that the programming level was consistent with the status quo. Examples of the tested learning activities demonstrating curriculum relevant material in the context of our first-year computing course are presented. A discussion of the sensor capabilities which provide the framework for the curriculum is also provided. The sensors are discussed qualitatively in a broader context that would allow development of additional learning activities as the need arises. A detailed curriculum map using Bloom’s taxonomy for the cognitive domain is presented for both the classroom and lab environment. This curriculum map is linked to learning outcomes for the course.

## **Introduction**

The impetus for this research came from papers by Behrens et al. [1], [2] which suggested that LEGO® Mindstorms robots coupled with the MATLAB programming language could be used in undergraduate courses in electrical engineering to motivate students using “practical situations” driven by “active learning” activities. The activities proposed by those authors were discipline specific to electrical engineering and design, but their results motivated us to consider a similar idea for a more general first-year engineering computing course that was not discipline specific. Many of the ideas recommended by Behrens et al. [1], [2] were examined and tested in this study using a more general approach to programming with MATLAB to demonstrate basic programming structures using LEGO® Mindstorms robots. The curriculum for our first-year engineering computing course is common to all engineering disciplines, and therefore the learning outcomes are broader and sometimes more difficult to satisfy. It should be noted that students in our first-year program do not choose engineering disciplines until they enter second year. This research focused on establishing the feasibility of these recommendations within the context of our own curriculum.

The feasibility of using LEGO® Mindstorms with MATLAB to introduce active learning to a first-year computing course was conducted through an internal summer project grant at our institution in the context of our first-year engineering programming course in MATLAB. Ease of use for our first-year engineering students was paramount in our deliberations. This evidence-based practice research (feasibility) study was conducted by a first-year engineering student (co-author), henceforth referred to as research assistant, that had just successfully completed first-year and had taken our first-year engineering computing course, and by two of our engineering instructors. The research assistant was familiar with the course requirements and syllabus. The research assistant developed and coded all the activities discussed later in this paper and provided feedback regarding ease of use. This qualitative feedback, along with the curriculum mapping to Bloom’s taxonomy provided evidence for the positive conclusions reached regarding future implementation of LEGO® Mindstorms in our first-year MATLAB programming course.

## **Literature Review**

A part of this research study was to conduct a literature review of research work in active learning applied to teaching first-year computing courses in engineering. The scope of the review was expanded to a more general consideration of robotics as an active learning tool in computing courses (not necessarily first-year), encompassing general science programs as well. A review of the literature, in general, supported the use of LEGO® Mindstorms (robotics) with some chosen programming language for teaching programming and/or discipline specific design courses [1] – [50]. The review did not reveal any detailed studies of the LEGO® Mindstorms EV3 robot coupled with the MATLAB programming environment to teach an introductory first-year engineering functional programming course. In many of the studies, the results were qualitative. There were few quantitative studies to confirm researchers’ anecdotal observations that robotics are effective active learning tools. In the following paragraphs, we present a more detailed discussion of the papers relevant to our consideration of the feasibility of using LEGO® Mindstorms EV3 and MATLAB in our first-year engineering computing course.

Behrens et al. [2] found that in the term following their robotics/MATLAB electrical engineering course, “the number of students who passed were highest for the subjects with contents related to the project”; in this case Mindstorms project. The authors concluded that although the data was not indicative of absolute quantitative proof of the concept, the project was successful. Alvarez et al. [3], however, were unable to show any statistical significance between groups that used and did not use robots to enhance basic programming skills. Anecdotally, teachers that participated in the study by Alvarez et al. noted that the experimental group using robots could “properly use conditional and iterative conditions much earlier” than the group that did not use robots. A follow-up study by Alvarez et al. [4] suggested that student attrition in the course used for the study was significantly reduced. It was further concluded in this study that the robots should be constructed prior to the course to shift the focus of the course to programming instead of robotics design.

Much of the literature surveyed was not specific to MATLAB but did offer significant qualitative evidence to support the use of active learning. The literature further supported “hands-on” active learning in improving the students’ educational experiences. Aznar et al. [5] have gone so far as to suggest competency in programming can only be achieved by changing teaching methodologies; moving away from traditional methods that are focused on knowledge transfer only. They did not recommend a specific platform for achieving this result, although they did recommend the use of interdisciplinary robotics projects for first-year engineering students. Chen et al. [6] successfully coupled LEGO® Mindstorms with Microsoft Visual programming to motivate 400 freshman students, improve course evaluations in a first-year computing science course and encourage better student participation. Although much of the literature was very positive in support of using robots in computer programming, there were several studies with cautionary advice.

Cliburn [7] noted that the programming interface and robotics project should be chosen carefully so as not to increase the workload of the students. Minimizing the additive workload concept was an important consideration in our feasibility study as we had also recognized what Cliburn had observed. Cliburn recommended that “instructors should carefully consider whether the Mindstorms project will add to the course’s content before choosing to adopt them”. Cliburn also noted that the application of the robotics with C++ programming language was successful in an introductory object-oriented computer science programming course but fell short for the sequel to the course. He noted a lack of individual participation in the team environment that was used to deliver the course.

Fagin & Merkle [8] used data from over 800 students in computer science to quantitatively assess the success of using robots in an introductory computing course. This study was more quantitative examining 9/48 sections using “robotics” compared to the remaining sections without. The results from their study were negative in terms of scores i.e. scores were lower in the robotics section when comparing two groups (robotic and non-robotics based). Interestingly, the results did not discourage the authors. The authors attributed this unexpected anomaly in the results to the lack of ease of use, instructor inexperience and reduced access due to the limited number of robot kits, again suggesting that the choice of robot and programming language is quite critical. Hence the need for this in-depth study.

Cruz-Martin et al. [9] had a positive experience with LEGO® Mindstorms NXT and a C-like programming language for sophomore students in a systems engineering course. One of the conclusions reached was that the application of robotics significantly reduced the effort of teaching faculty in the preparation of exercises while at the same time improving both professor and student experiences. The authors noted a slight improvement in grades and concluded that the use of LEGO® robots had “more benefits than drawbacks”.

Fan [10] conducted a study with 300 students using a robot simulator (instead of robotics kits like Mindstorms EV3) with MATLAB. This study was more design focused. The authors noted improved interest in learning skills from assignments and a strong correlation between overall exam scores and simulator usage.

Danahy et al. [11] provided a summary on the role LEGO® robotics has played in college engineering education over a 15-year period (1998 – 2013) noting a substantial improvement in ease of use with the introduction of the EV3 robot in 2013. He envisions LEGO® products to be transformational through the added programming capabilities with the newer version, opening the door to broader applications in undergraduate engineering computing courses. His conclusions generously supported LEGO® robotics in college level engineering education, although the case studies that he considered leaned more towards the LEGO® modular programming language rather than programming languages like MATLAB.

Hirst et al. [12] reviewed various programming environments for teaching robotics using LEGO® Mindstorms, concluding that none fully met their requirements but did not consider MATLAB. Klassner et al. [13] used older and newer versions of LEGO® Mindstorms robots with MATLAB to demonstrate genetic algorithms. Their conclusions were positive about student engagement but were uncertain regarding the students’ comfort level with the MATLAB programming language. Their experience was also positive on the ease of use of the new Mindstorms robots coupled with the MATLAB environment supporting some of the results that we obtained during our feasibility analysis. Lawhead et al. [14] provided sample exercises for first-year computer science curricula using LEGO® Mindstorms RCX bricks with Java as the programming language, arguing the benefits of using physical robots to demonstrate abstract programming concepts.

Katz [15] conducted an extensive review of the literature for “active learning” in academic environments. He noted that there seem to be some opposition to active learning suggesting that it can be detrimental to student progress. However, he suggested that the process necessary to incorporate active learning in the classroom itself might increase “forethought, creativity and willingness to experiment” (presumably for instructors, but also maybe students) which in turn might alter dogmatic paradigms of instructors and, with the engagement of students, improve the knowledge and skill sets of “today’s” students. Our own anecdotal experiences in the classroom would tend to support this contention.

There were several articles on active learning, in general, which added to the support for considering this type of learning environment [51] – [55] in a computing course. Christie et al. [51] suggested that it is necessary to adapt engineering education in favor of “more engaging” and “active” learning environments to address the needs of educating diverse groups of learners.

The authors also encouraged engineering educators to conduct research into the “best ways of activating learning” within the context of their own teaching environment. Lima et al. [52] provided a summary of active learning, also for engineering education and concluded that active learning promotes “analysis”, “synthesis” and “evaluation”, which coincidentally belong to the upper hierarchy of Bloom’s learning taxonomy. Oddie et al. [53] put active learning in the context of computing science, suggesting that problem solving is one of the key skills required to learning programming fundamentals. Conclusions based on their research suggested that robotics encourages engagement and motivates students to learn. Parjono [54] provided a discussion on the nature of active learning concluding that this type of learning is necessary for the development of the students’ “own knowledge”, which is the foundation of Bloom’s cognitive domain. Prince [55] studied the effectiveness of active learning in a more quantitative exploration using the literature to provide evidence. He concluded that faculty should use more non-traditional approaches like active learning to “promote academic achievement”. He found support for all forms of active learning examined, which included student activities in the classroom, promoting student engagement, collaborative and cooperative learning, and problem-based learning (PBL), all of which could be achieved using robotics to teach computer programming.

Based on the literature review, we concluded that the LEGO® Mindstorms robots with MATLAB could provide the framework for the hands-on active learning activities and act in a complementary way to reinforce the delivery of content in fundamental programming course. The literature reviewed in this study confirmed the need to conduct a feasibility study of this platform with respect to our introductory computing course. It also provided a template for various considerations, such as: ease of use, cognitive value, ability to improve understanding of abstract concepts, meeting learning outcomes and compatibility between robot and programming platforms.

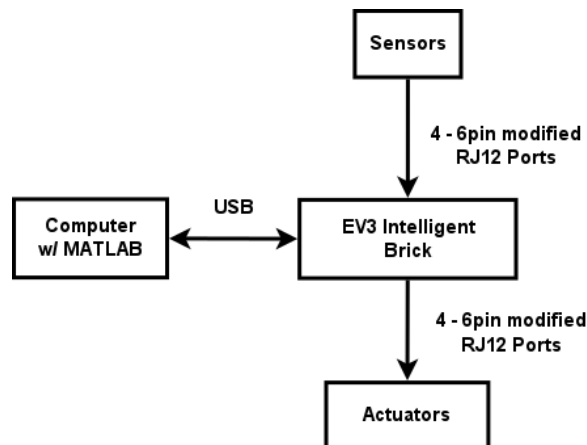
### **LEGO® Mindstorms**

The EV3 Intelligent Brick is the programmable prototyping board for the LEGO® Mindstorms robot kit which costs approximately \$400 CAD. It is powered by a Siltara ARM9 CPU running a Linux based operating system. There is 64MB RAM, 16 MB flash memory with 32 GB additional memory from a micro SDHC card [59]. Figure 1 shows a diagram of how components are connected using the EV3 brick. First, to send code to the EV3, it is connected to a computer through a USB connection. There is also support to connect wirelessly to a computer using Bluetooth or Wi-Fi. The Brick has 4 ports for connecting external sensors through a 6-pin modified RJ12 connection. Similarly, there are 4 output ports to connect actuators. The Brick also includes a 178x128 pixel display screen, speakers, and 3 LEDs. The libraries needed to program the EV3 with MATLAB are freely available and are supported on multiple platforms. Additional libraries are available for Octave but is out of the scope of the current paper. The current setup shows that it is simple to connect the EV3 to a personal computer and connect various sensors. One downfall to the current setup is the cost of the LEGO® Mindstorms kit. To mediate the cost students could work in groups and/or the kits could be used in multiple courses.

The sensors available for EV3 kits depend on which version of the kit is purchased. Five sensors were considered here: touch, color, gyro, ultrasonic and infrared sensor with beacon. A possible

alternative to the LEGO® sensors is to connect an Arduino with sensors to the brick. However, this requires more setup for the instructor and knowledge of circuits. In addition, there are numerous other third-party sensors available. Three servo motors were used in this study: two large and one medium. Additional motors are also available. Information regarding the kits can be found on the LEGO® Mindstorms website.

**Figure 1: Diagram of the setup of the EV3 Brick**



### Sensor/Motor Evaluation for LEGO® Mindstorms EV3

One aspect in the scope of the project was to examine the five sensors and motors: to see if they afforded ample flexibility to cover the learning outcomes in the course. It should be noted that the sensor and motor assessments were conducted by the research assistant (first-year engineering student) to qualitatively answer ease of use questions as well as suitability to course content. The summary that follows was based on the testimonials of the student and provided the evidence needed to assess the overall MATLAB/Mindstorms platform. The qualitative summaries for the sensors and motors are based on the final conclusions written by the research assistant and presented here in a condensed form. The research assistant provided the following qualitative assessments (testimonials) for the sensors and motors.

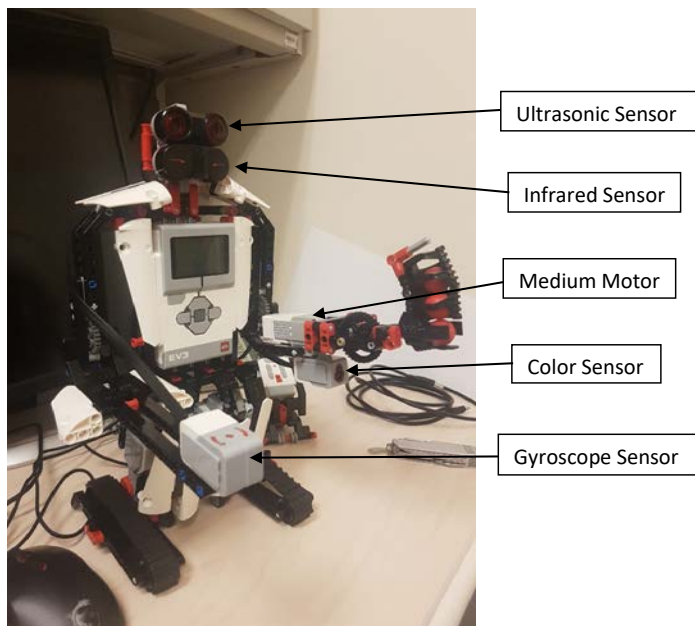
- “The motors and touch sensor were easily connected to MATLAB and programs were written to utilize them. The functions associated with the motors and touch sensor were straightforward to use and would make good preliminary assignments for first-year students.”
- “The touch sensor has two states, pressed and not pressed and is controlled by a function that responds to the sensor’s state. The touch sensor could be used at the beginning of a program as a trigger, or as a bumper detection or as a ledge detector. The touch sensor was used to control the motion of the robot.”
- “The infrared sensor has three modes: i) proximity mode to detect objects, ii) beacon mode and iii) remote mode, which utilize functions to make use of the sensor. This sensor was easily connected to MATLAB and programs written to utilize it. The functions associated with this sensor are straightforward and should be easy for first-year engineering students to use with practice.”

- “The ultrasonic sensor has two modes: i) presence mode and ii) measure mode. It would be possible to collect data and write the data to files for later analysis. This sensor was easily connected to MATLAB and programs could be written to utilize it. This sensor also makes use of functions that are straightforward for use in our programming course.”
- “The color sensor has three modes: i) color mode, ii) reflected light intensity mode and iii) ambient light intensity mode. The color mode can detect various colors, while the other two modes can detect light intensity in two different situations. The color sensor was easy to connect to MATLAB and use of the functions was straightforward.”

The conclusion reached by the research assistant was that all motors and sensors were straightforward and easy to use, and that data could be collected and stored to files for later analysis using MATLAB.

The sensor/motor evaluation was conducted at the very beginning of the feasibility study to determine whether the LEGO® Mindstorms robot satisfied the ease of use concept. The research assistant also concluded that first-year engineering students would be capable of downloading the Mindstorms platform and creating MATLAB code to use the sensors and motors. This was completed in approximately one week. When this was completed we proceeded to assess the combined Mindstorms/MATLAB platform in terms of our first-year computing course. Figure 2 shows the robot setup with various motors and sensors. The sensors are easily attached and connected to the robot using USB cables. The two EV3 robots were used in this study were referred to as “Fred” and “Barney”.

**Figure 2: Mindstorms EV3 robot, Fred, with connected motors and sensors**



A small sample of MATLAB code developed by the research assistant for the testing of the color sensor is presented in Figure 3 to illustrate the simplicity of using the sensor functionality.



**Figure 3: MATLAB code for using the color sensor**

```
%Color Sensor
%Has three modes:
%1) color mode - recognizes 7 colors
%2) reflected light intensity mode - measures intensity of light reflected
%back from its red light-emitting lamp
%3) ambient light intensity mode - measures the strength of light that
%enters the window from its environment, ie sunlight or flashlight
%creates a handle for the touch sensor
%by inputting 3, you have plugged the color sensor into port 3
mycolor = colorSensor(Fred,3)

%if the port number is unknown use:
%mycolor = colorSensor(Fred)
%to automatically find the sensor port with the lowest number and set up
Color sensor

%outputs the color as a string
%can recognize 7 colors: black, blue, green, yellow, red, white, and brown
readColor(mycolor)

%ambient refers to mode, choice of ambient or reflected; defaults to ambient
%outputs light intensity from 0 (very dark) - 100 (very light)
readLightIntensity(mycolor, 'ambient')
```

## Introductory Engineering Computing Course

This next section deals specifically with our computing course. The first-year engineering computing course is a 13-week introductory programming course in MATLAB, with a problem-solving based methodology. The course consists of 3 hours of lecture and 3 hours of lab each week. Table 1 summarizes the learning outcomes for the course and cross-references the learning outcomes with Bloom's taxonomy [56] for cognitive level. To apply the cognitive levels to curriculum mapping we used the work of Britto and Usman [57], [58] who applied Bloom's cognitive domain to the learning outcomes and curriculum mapping for a software engineering course. In this case, Britto and Usman [58] considered only the cognitive domain. We use the same six levels of objectives for the cognitive domain that have been presented in by Britto et al. summarized from the earlier work of Benjamin Bloom [56].

We present Bloom's levels as: Knowledge (1), Comprehension (2), Application (3), Analysis (4), Synthesis (5) and Evaluation (6). The progression is usually described as hierarchical moving from knowledge to evaluation as complexity increases. In this course we did not expect to reach the highest level (6) for any of our learning outcomes, but we did expect to reach the synthesis (5) level for several outcomes. The cognitive levels in Table 1 are the maximum levels achieved by the end of the first-year level course for each specific learning outcome. The learning outcomes used in this study are standard compared with other similar general programming courses except for LO5, LO6, and LO15. These learning outcomes are applied to engineering specific applications to help engage students in the course.

**Table 1: Learning Outcomes (LO) for first-year engineering computing course**

<b>LO Label</b>	<b>Learning Outcome (LO)</b>	<b>Cognitive Level Achieved</b>
LO1	Use basic programming language syntax	3
LO2	Develop algorithms using flowcharts and pseudocode	5
LO3	Use fundamental data types (Boolean, integers, floating point, characters)	3
LO4	Employ scalar operators (Math, Conditional, Logical)	4
LO5	Employ array operators	3
LO6	Illustrate data visualization using programs	5
LO7	Develop programs using decision constructs	5
LO8	Formulate program using repetition constructs	5
LO9	Synthesize array input using data analytics	5
LO10	Model programs using functions	4
LO11	Describe the limitations and inaccuracies of numerical computations	2
LO12	Create data structures (structures, cell arrays)	5
LO13	Use standard and file input/output	3
LO14	Apply verification and validation principles to programs	3
LO15	Discuss the uses of programming in real world applications	2

Next, the topical order of course content was examined, and course topics were cross referenced in Table 2 with learning outcomes and the weekly assignments. The learning outcomes are associated with material covered in the lecture. It should be noted that the learning outcomes may not be completed in a certain week, such as data types (LO3) which are progressively taught throughout the course. As can be seen in Table 2, the weekly assignments are staggered such that course material is presented prior to assigning independent assessment activities. Table 2 was then used to create the curriculum map in Table 3 and was also used to determine the feasibility of using the LEGO<sup>®</sup> Mindstorms robot to satisfy all learning outcomes for the course, and to determine if there were shortcomings with the combined robotics/programming platform.

The proposed learning activities developed in this feasibility study were mapped to the anticipated learning outcomes to ensure that all learning outcomes could be satisfied. In addition, we addressed potential limitations of using the robotics platform (active learning) to complement the learning outcomes, which were specific to the MATLAB programming language. At the same time, we assigned Bloom's cognitive levels to the learning outcomes to get a sense of the cognitive levels one might expect by the end of the course. It should be noted that curriculum maps are normally applied to assessment of entire degrees or programs, but we have applied it to assess the feasibility of using active learning using LEGO<sup>®</sup> Mindstorms, while still achieving the goals and learning outcomes of our course. The curriculum map presented in Table 3 illustrates that the content of the course and the learning outcomes are satisfied for our first-year engineering computing course. The challenging part was to determine if the course would be improved by introducing active learning in the form of LEGO<sup>®</sup> Mindstorms robotics and how this would be different than the status quo which uses more traditional algorithmic programming exercises.

**Table 2: Course concepts and Learning Outcomes for first-year engineering computing course**

Week	Main Topic	Subtopics	Learning Outcomes	Asgn. #
1	Course outline and introduction	Introduction to computers: hardware and software, introduction to MATLAB environment	LO1, LO3, LO13	1
2	Problem solving and expression assignment	Pseudo-code, flow charts, program structure, variable assignment, math operators, precedence, built-in functions	LO1, LO3	2
3	Arrays and plotting	Array initialization, indexing, operations, plotting	LO1, LO2, LO3, LO4, LO15	3
4	Logical data types and selection flow control	Logical data types, if-else, select	LO1, LO3, LO4, LO5, LO6, LO9, LO15	4
5	Repetition flow control	Control mechanisms: for, while, convergence	LO1–LO5, LO7, LO13, LO15	5
6	Review		LO1–LO9	
7	Functions	M-file, anonymous, and recursive functions	LO1–LO4, LO7–LO9, LO11, LO15	6
8	Numeric data types	Integers, floating point, precision vs. round-off error	LO1–LO5, LO7–LO10, LO14, LO15	7
9	Data types: Character and heterogeneous	Characters, structures, cell arrays	LO1, LO3, LO5–LO7, LO9, LO13, LO15	8
10	Input / Output	Standard and file I/O	LO1–LO5 LO7, LO8, LO10, LO13, LO15	9
11	Validation and verification	Testing and debugging, variable checking	LO1–LO8, LO10, LO12–LO15	10
12	Engineering problems	Various possible topics	LO1–LO10, LO13, LO15	11
13	Review		LO1–LO15	12

**Table 3: Curriculum Mapping: Learning Outcomes with Assignment Order using Bloom’s Cognitive Levels**

Asgn. #	Learning Outcome														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	-	1	-	-	-	-	-	-	-	-	-	1	-	-
2	2	-	1	-	-	-	-	-	-	-	-	-	-	-	-
3	3	2	2	2	-	-	-	-	-	-	-	-	-	-	1
4	3	-	2	2	2	3	-	-	1	-	-	-	-	-	1
5	3	3	2	2	2	-	3	-	-	-	-	-	1	-	2
6	3	3	3	3	-	-	3	3	2	-	2	-	-	-	2
7	3	3	3	3	3	-	5	5	3	3	-	-	-	1	2
8	3	-	3	-	3	5	5	-	4	-	-	-	2	-	2
9	3	5	3	3	3	-	5	5	-	4	-	-	2	-	2
10	3	5	3	4	3	5	5	5	-	4	-	3	2	1	2
11	3	5	3	4	3	5	5	5	5	4	-	-	3	-	2
12	3	5	3	4	3	5	5	5	5	4	2	5	3	3	2

**Learning Activities Assessment Method**

The proposed learning activities obtained from the curriculum map are developed in this section where a brief description of potential robotics assignments for the first-year engineering computing course is provided. There are 12 proposed weekly assignments to fit within the 13-week duration of the course. These assignments were created and tested by the research assistant who had just completed our first-year engineering course without robotics. A template rubric was created for each assignment which compared the current assignments to the new assignments to ensure that the course objectives and learning outcomes were met with the proposed activities. The research assistant then created MATLAB code for all the proposed activities with ease of use criterion foremost in testing the activities. A sample MATLAB code from Assignment #4 is included in Figure 4, to demonstrate level of code complexity and which learning outcomes were satisfied. In this code the Mindstorms robot “Fred” will move forward for 3 seconds if three criteria are met simultaneously: 1) ultrasonic sensor reads an object between 30 and 100 cm, 2)

the touch sensor is pressed, and 3) the color sensor reads an ambient light of 15 or higher. Otherwise, “Fred” will make a noise and not move.

**Figure 4: Sample MATLAB code illustrating the use of control statements using a robotics activity satisfying learning outcomes LO1, LO2, LO3, LO7**

```
% If Statements
mysonicsensor = sonicSensor(Fred, 4)
mytouch = touchSensor(Fred,4);
mycolor = colorSensor(Fred,1);
myLeg1 = motor(Fred, 'B');
myLeg2 = motor(Fred, 'C');
dis = double(readDistance(mysonicsensor))
if dis >= 30 & dis < 100
    distance = true;
else
    distance = false;
end
tou = readTouch(mytouch)
if tou == 1
    touch = true;
else
    touch = false;
end
int = double(readLightIntensity(mycolor, 'ambient'))
if int >= 15
    intensity = true;
else
    intensity = false;
end
if distance==1 & touch==1 & intensity==1
    myLeg1.Speed = 30;
    myLeg2.Speed = 30;

    start(myLeg1);
    start(myLeg2);

    pause(3);

    stop(myLeg1);
    stop(myLeg2);
else
    playTone(Robot, 5000, 3, 10)
end
```

The development of the learning activities to match the course objectives is shown in Table 4. In this table we defined a goal that was consistent with the learning objectives and considered various robotics assignment tasks that would satisfy those goals. If we were able to satisfy the

**Table 4: Learning Goal with Assignment Tasks**

<b>Asgn.</b>	<b>Goal</b>	<b>Assignment Tasks</b>
1	Introduction to MATLAB environment and syntax	Interfacing robot with MATLAB – output to screen
2	Exploring MATLAB features: i.e. syntax, built-in functions, operators, and data types	Interfacing robot with MATLAB – exploring use of motors and sensors
		Rewrite expressions in MATLAB, use built-in functions, learn syntax
3	Solution procedure – simple programs	Creating simple programs using robot’s motors and sensors, and creating flow charts and pseudo code
4	Using arrays and plotting	Gather data using a sensor, plot data Vision - Image Analysis using Pixy Cam with the robot
5	Logic – decisions, conditional statements	Object detection and avoidance Explore use of infrared sensor Braitenberg Vehicles – complex behaviors using simple sensors
6	Repetition	Automatic Motion with Object avoidance (Random Walk) Arm movement using sensors and motors
7	Functions	Dead Reckoning or other applications using functionality of robot
8	Numeric Data types	I2C communication – understanding digital data transfer
		General base conversions
9	Character	Send string instructions to robot to execute various tasks
10	Standard and File I/O	Rover - collect data and periodically sync with computer to download
11	Engineering Application - Structures	Building advanced robot functionality through structures
12	Advanced	Allow students to develop complex programming project to encompass all learning outcomes – i.e. maze solving algorithms, color tracking

learning objectives using LEGO® Mindstorms, there is only one cell in the assignment task column. Those cells that are split indicate that the robot activities were somewhat limited in satisfying the outcomes. However, there were only two assignments (Assignments #2 and #8) which needed supplemental activities that were unrelated to robotics. This was not enough to affect our final conclusions for the feasibility analysis. The proposed assignment content and testing methodology is described in the following paragraphs.

- For Assignment #1, the research assistant downloaded the MATLAB programming toolkit and became familiar with the LEGO® Mindstorms EV3 robot. The intent of this assignment is to introduce the basics of MATLAB and the robotics platform. Literature suggests the robotics kits should be built prior to this course [4]. The research assistant built two robots during the project but suggested that building the robots should not be a part of the assignments. Initially a basic configuration for the robot was used.
- Next, in Assignment #2, the research assistant explored the sensors and motors for the LEGO® Mindstorms EV3 robot. For this assignment extra activities using MATLAB were necessary to meet the learning outcomes. Since Assignment #2 in the current set is quite light compared to other assignments, this assignment was used to concurrently gain experience using the sensors.
- Assignment #3 deals with developing simple pseudo code, flowcharts and MATLAB expressions. At this stage in the course, the research assistant coded simple programs to utilize the robot's motors and sensors to perform simple tasks while simultaneously using flow charts to illustrate their problem-solving skills.
- Assignment #4 is intended to use arrays and plotting. In this case the robot had ample capability to produce data through its sensors, both spatially and temporally, which was plotted using MATLAB. With multiple sensors, it would be easy to generate a wide variety of plotting exercises. Alternatively, an external non-standard sensor, Pixy Cam, could be used to capture images that can be displayed and/or modified by creating simple MATLAB programs.
- For Assignment #5 the research assistant used the sensors to produce logic-based decision making to satisfy the learning outcomes described earlier. The code for one of these activities was given in Figure 4. Other activities could be accomplished using the infra-red sensor for object detection and avoidance. Another possibility would be to connect various sensors to create Braitenberg vehicles which model complex behaviors using simple sensors.
- Following with Assignment #6, the research assistant proposed the use of repetition to perform automated tasks for the robot incorporating MATLAB constructs to satisfy the learning outcomes described earlier. The research assistant tested a variety of activities for these learning outcomes and concluded that it would be easy to develop new robot activities from year to year.
- Assignment #7 was designed to explore the use of functions to develop modularized code. The robot itself has internal sensors that can be used to determine distance traveled which can then be used to create functions that determine a change in position using a dead reckoning algorithm. Several activities were proposed with sensors and motors to move the robot and collect sensor data by creating MATLAB functions to perform certain tasks. However, this assignment was not completely tested due to time constraints. The research student concluded that it would not be difficult to create robot activities that would satisfy the learning outcomes associated with Assignment #7.
- Next, for Assignment #8, the research assistant practised the more abstract concept of numeric data types. This was the most difficult learning outcome to satisfy using robotics. In this case, even in the current curriculum students test their base conversion skills using pencil and paper. The practical application of digital data transfer could be

used to illustrate the I2C communication protocol used in the EV3 input ports of the robot.

- The learning outcomes associated with characters and strings were accomplished in Assignment #9, where string input in MATLAB was sent to the robot. The research assistant programmed the robot to act as a calculator, displaying different characters on the LCD screen. Also, different strings were used to execute various tasks for the robot, creating a “game” environment.
- Standard and file input/output was practised in Assignment #10 using the robot to collect and store data. Both these assignments were tested by the research assistant who was able to collect, store and display data on the robot display. The data was then synthesized using data analytics in MATLAB.
- The next assignment, Assignment #11, deals with data structures. The intent is to create MATLAB structures that will mimic those that exist in LEGO® Mindstorms MATLAB toolbox [59] and will augment those structures with additional functionality that would develop by the students using the programming skills developed in the course.
- Lastly, Assignment #12, uses the skills acquired throughout the term where students will choose a more complex programming project such as maze solving or color tracking algorithms to demonstrate all learning outcomes. The research assistant began the development of MATLAB code for a maze solver. This assignment is the most open-ended of all the assignments and will likely vary dramatically from year to year. Complete testing of this assignment was not achieved by the end of this project due to the time constraints. However, we concluded from the literature survey and from activities conducted during the project, that it would not be difficult to fulfill the learning outcomes for this assignment.

Activities for all assignments except for Assignment #6 were created, tested and validated against our learning outcomes by a first-year engineering student (research assistant) who was familiar with the learning outcomes and course objectives. Comparative templates were used for each assignment to maintain consistency between the new and old assignments (status quo). Each template considered: i) objectives for the assignment based on the existing current assignment, ii) overall objectives in terms of learning outcomes, iii) proposed Mindstorms activities and lastly iv) proposed problems (activities) to satisfy objectives and learning outcomes. For each assignment we considered the learning outcomes, created the new assignment activity using the active learning concept (robotics) and validated the assignments by having the research assistant develop code and provide testimonials regarding the ease of use and what was learned. These templates were too long to include in this paper. The templates were based on content and learning outcomes from the existing curriculum. The results of this comparative analysis are provided in Table 4 which was discussed previously. These results and ease of use were instrumental in making the conclusions from our feasibility study.

## **Discussion**

The results of the feasibility study for LEGO® Mindstorms EV3 with MATLAB were positive. The combined platform offers the flexibility needed to create the first-year engineering computing course described in this study. The functionalities of the sensors and motors that are available for the kits offer many opportunities to demonstrate programming concepts, and to



create different assignments from year to year. The literature and results of the feasibility study suggest that it would be easy to vary the assignments by considering different features of the robots and perhaps different designs. Table 4 shows limitations in Assignments #2 and #8 where the use of robotics would not be appropriate to satisfy the learning outcomes for those assignments. Adding the robotic complement to Assignment #2 was beneficial in this case as previous experience had indicated that Assignment #2 was relatively easy compared with other assignments in the course. For Assignment #8, the use of MATLAB itself to demonstrate base conversions is also limited due to the abstraction of this course concept, so being unable to use robotics for this learning outcome was not problematic. It is possible with further research there may be some activity with the robot that could demonstrate base conversions with a “hands-on” approach. Cost is a main limitation, but the literature suggests that having students work in groups can alleviate the cost issue.

The results summarized in Table 4 also show that the final proposed assignments for this course are ideally suited to the higher level of Bloom’s cognitive domain through a reasonably complex final project testing the use of the Mindstorms robot. All the “active learning” (robotics) assignments that were proposed herein (except Assignment #6) have been tested to ensure that a first-year engineering student could develop the MATLAB code. At the same time, we have shown that these activities satisfy the learning outcomes and show potential for improving student outcomes. In addition, there are a wide variety of activities that can be used to create these “complex” assignments as the robots have seemingly endless configurations. There are many ways that complexity can be tested from having the robot negotiate mazes, to having the robot make advanced calculations, using MATLAB as the purveyor of knowledge moving the students’ cognitive abilities to the levels of comprehension through to synthesis.

In the beginning of the course Bloom’s knowledge level is satisfied: acquiring knowledge about sensors/motors, and programming environments. Then as the courses progresses, students begin to comprehend the material. The earlier assignments provide knowledge, the middle assignments work toward comprehension and application. The later assignments develop students’ analytical skills through active learning activities with the robots. As the assignments progress in complexity, it becomes easier for the robot to satisfy the learning outcomes and the synergy between the active learning concept (robot) and the programming platform is enhanced. The latter two assignments then focus on more complex engineering applications where the students begin to combine different aspects of programming and move towards Bloom’s evaluation level, increasing their skills and understanding thereby satisfying our proposed learning outcomes. Midterm, final exams and projects would be used to assess the progress of the students at various stages in the course much as they are used in the current course configuration. Exam assessment was not considered in this study.

## **Conclusions**

The LEGO® Mindstorms EV3 robot combined with MATLAB programming language would be feasible for a first-year programming course. The next step in the consideration would be to look at budget restrictions and how this platform would be introduced into an engineering program, and whether students would engage in collaborative or cooperative learning. Students typically have access to licensing to MATLAB so there would be no substantial change from the status

quo in our course, since the MATLAB EV3 toolbox has no additional charge beyond the cost of the MATLAB licensing fee. To date, we have not typically used teams in our first-year computing course so if this learning approach were to be considered we would have to carefully contemplate the implications. To date, assignments have been given to individual students with individual assessments. In addition, incorporating robots in this new curriculum would necessitate some learning for faculty teaching the course, but this learning would be the application of the LEGO® Mindstorms toolbox. The ease of use discussed earlier for this platform suggests that learning for faculty would not be overly onerous.

## Acknowledgements

This research study was funded through a Project Grant from MacEwan University.

## References

- [1] A. Behrens, L. Tarof, R. Schwann, J. Ballé, T. Herold and A. Telle, “First Steps into Practical Engineering for Freshman Students Using MATLAB and LEGO® Mindstorms Robots”, *Acta Polytechnica*, Vol. 48, No. 3, pp.44-49, 2008.
- [2] A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Ballé, T. Herold, A. Telle, T.G. Noll, K. Haymeyer and T. Aach, “MATLAB Meets LEGO® Mindstorms – A Freshman Introduction Course into Practical Engineering”, *IEEE Transactions on Education*, vol. 53, no. 2, pp. 306-317, 2010.
- [3] A. Álvarez and M. Larrañaga, “Using LEGO® Mindstorms to Engage Students on Algorithm Design”, in *2013 IEEE Frontiers in Education Conference (FIE)*, pp. 1346-1351, 2013.
- [4] A. Álvarez and M. Larrañaga, “Experiences Incorporating LEGO® Mindstorms Robots in the Basic Programming Syllabus: Lessons Learned”, *J. Intell. Robotics Syst.*, 81, 1, pp. 117-129, 2016.
- [5] M. Aznar, J. Zacarés, J. López, R. Sánchez, J.M. Pastor and J. Llorca, “Interdisciplinary Robotics Project for First-Year Engineering Degree Students”, *Journal of Technology and Science Education*, pp. 151-165, 2015.
- [6] C.A. Berry and D. Chang, “From LEGO® to Arduino: Enhancement of ECE Freshman Design with Practical Applications”, in *Proceedings of ASEE 123rd Annual Conference & Exposition, 2016*. <https://doi.org/10.18260/p.26972>, [Date Accessed: July 25<sup>th</sup>, 2018].
- [7] J. Boender, E. Currie, M. Loomes, G. Primiero and F. Raimondi, “Teaching Functional Patterns through Robotic Applications”, J. Jeuring and J. McCarthy (Eds.): *Trends in Functional Programming in Education (TFPIE) EPTCS 230*, pp. 17-29, 2016.
- [8] D.T. Butterworth, “Teaching C/C++ Programming with LEGO® Mindstorms”, *RiE 2012, Prague*, pp. 61-65, 2012. <http://www.ksi.mff.cuni.cz/rie2012/proceedings/2012RiE-08.pdf>, [Date Accessed: July 25<sup>th</sup> 2018].
- [9] Q. Chen, Y. Tang, L. Li, G. Yang, M. Yang, Z. Xie, H. Chen, F. Xiao, Y. Li and R. Huang, “A Practice on LEGO® Mindstorms for Computer Science Freshman Experimental Education”, in *2016 International Conference on Advanced Education and Management Engineering (AEME 2016)*, pp. 17-21, 2016.
- [10] A. Chraith and K. Kelly, “LEGO® Mindstorms as a Teaching Tool in Engineering Education”, in *Proceedings of the 27<sup>th</sup> International Manufacturing Conference (IMC 27)*, pp. 386-394, 2010.

- [11] D. C. Cliburn, “Experiences with the LEGO® Mindstorms throughout the Undergraduate Computer Science Curriculum”, in *Proceedings 36th FIE Conference*, pp. 1–6, 2006.
- [12] A. Cruz-Martin, J.A. Fernández-Madriral, C. Galindo, J. González-Jiménez, C. Stockmans-Daou and J.L Blanco-Claraco, “A LEGO® Mindstorms NXT Approach for Teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems Undergraduate Courses”, *Computers & Education*, vol. 59, issue 3, 2012.
- [13] M.P. Cuéllar and M.C. Pegalajar, M.C., “Design and Implementation of Intelligent Systems With LEGO® Mindstorms for Undergraduate Computer Engineers”, *Computer Applications in Engineering Education*, vol. 22, no. 1, pp. 153-166, 2014.
- [14] K.Y. (Daisy) Fan, “Teaching Computer Programming to First-Year Undergraduates with a MATLAB Based Robot Simulator”, <https://www.mathworks.com/company/newsletters/articles/teaching-computer-programming-to-first-year-undergraduates-with-a-MATLAB-based-robot-simulator.html>, [Date Accessed: July 25<sup>th</sup>, 2018].
- [15] E. Danahy, J. Wang, A. Brockman, B. Carberry, B. Shapiro and C.B. Rogers, “LEGO®-based Robotics in Higher Education: 15 Years of Student Creativity”, *International Journal of Advanced Robotic Systems*, 2014.
- [16] C. De Vries, Y. Wu, M.R. Ford and O. Onipede Jr., “Using LEGO® MINDSTORMS in a Control Systems Lab to impact next generation engineers”, in *122<sup>nd</sup> ASEE Annual Conference & Exposition*, paper ID #13324, 2015.
- [17] E. J. Delp and Y.H. Lu, “The Use of Undergraduate Project Courses for Teaching Image and Signal Processing Techniques at Purdue University”, in *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, Teton National Park, WY, 2006, pp. 281-284, 2006.
- [18] E.M. Eke, “Using Simulink, Matlab, and LEGO® Mindstorms to Teach a Project-Based Control Systems Design Course”, in *Proceedings of 2015 ASEE Annual Conference & Exposition*, Seattle, Washington. 10.18260/p.25016, 2015.
- [19] B. Fagin and L. Merkle, “Measuring the effectiveness of robots in teaching computer science”, *ACM SIGCSE Bulletin*, vol. 35, pp. 307-311, 2003.
- [20] C. Fernández Panadero, J. Vilena Román and C. Delgado Kloos, “Impact of Learning Experiences Using LEGO® Mindstorms® in Engineering Courses”, in *2010 IEEE Educon Xplore Conference: Education Engineering*, Madrid, pp. 503-512, 2010.
- [21] D. Gabauer, T. Bayse, J. Terpenney and R. Goff, “Improving Undergraduate Engineering Design Instruction Through Lessons Learned Mentoring First Lego League”, in *Proceedings of 2007 ASEE Annual Conference & Exposition*, Honolulu, Hawaii. <https://peer.asee.org/2672>, 2007.
- [22] A. Gage, and R.R. Murphy, “Principles and Experiences in Using LEGO®s to Teach Behavioral Robotics”, in *33rd ASEE/IEEE Frontiers in Education Conference*, vol. 2, session F4E, pp. 23-28, 2003.
- [23] E. A. Gandy, S. Bradley, D. Arnold-Brookes and N.R. Allen, “The Use of LEGO® Mindstorms NXT Robots in the Teaching of Introductory Java Programming to Undergraduate Students”, *Innovation in Teaching and Learning in Information and Computer Sciences*, 9:1, pp. 2-9, 2015.
- [24] M. A. Garcia and H. Patterson-McNeill, “Learning how to Develop Software Using the Toy LEGO® Mindstorms”, in *Proceedings 32<sup>nd</sup> Frontiers in Education Conference*, vol. 3, p. 7-10, 2002.

- [25] L. Folan, D. Doucette, and G. Georgi, "Introducing Design Throughout the Curriculum", in *Proceedings of 2002 ASEE Annual Conference*, Montreal, Canada. <https://peer.asee.org/10990>, 2002.
- [26] W. Grega and A. Pilat, "Real-time Control Teaching Using LEGO® MINDSTORMS NXT Robot", *International Multiconference on Computer Science and Information Technology*, pp. 625-628, 2008.
- [27] E. B. Gyebi, M. Hanheide and G. Cielniak, G. "The Effectiveness of Integrating Educational Robotic Activities into Higher Education Computer Science Curricula: A Case Study in a Developing Country", *Advances in Intelligent Systems and Computing*, vol. 560, pp. 73-87, 2017.
- [28] R.W. Hasker, "An Introductory Programming Environment for LEGO® Mindstorms Robots," in *Midwest Instruction and Computing Symposium*, Eau Claire, Wisconsin, 2005.
- [29] A. Hirst, J. Johnson, M. Petre, B.A. Price and M. Richards, "What is the Best Environment-Language for Teaching Robotics Using LEGO® MindStorms?", 2002. <http://mcs.open.ac.uk/bp5/papers/AROB2002/2002-AROB-Hirst.pdf>, [Date Accessed: July 25<sup>th</sup>, 2018].
- [30] J. Katz, "Active Learning in Academic Environment", 2010. [https://www.usma.edu/cfe/Literature/Katz\\_10.pdf](https://www.usma.edu/cfe/Literature/Katz_10.pdf), [Date Accessed: July 25<sup>th</sup> 2018].
- [31] S. Kim and J. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO® Mindstorms", *IEEE: Transactions on Education*, vol. 52, no. 1, pp. 99-108, 2009.
- [32] Y. Kim, "Control Systems Lab Using a LEGO® Mindstorms NXT Motor System", *IEEE Transactions on Education*, vol. 54, no. 3, pp. 452-461, 2011.
- [33] F. Klassner and S.D. Anderson, "LEGO® Mindstorms: Not Just for K-12 Anymore", *IEEE*, vol. 10, pp. 12-18, 2003.
- [34] F. Klassner, J.C. Peyton-Jones and K. Lehmer, "Genetic Algorithms with LEGO® Mindstorms and MATLAB", *Florida Artificial Intelligence Research Society Conference*, 2012.
- [35] P. Lau, M. Portsmore, S. McNamara and C. Rogers, "Lego Robotics in Engineering", in *Proceedings of 2001 ASEE Annual Conference*, Albuquerque, New Mexico. <https://peer.asee.org/9511>, 2001.
- [36] P.B. Lawhead, E. Duncan, C.G. Bland, M. Goldweber, M. Schep, D.J. Barnes and R.G. Hollingsworth, "A Road Map for Teaching Introductory Programming Using LEGO® Mindstorms Robots", *ACM SIGCSE Bulletin*, vol. 35, no. 2, pp. 191-201, 2002.
- [37] H. Liang, C. Fleming, K. Man and T. Tillo, "A First Introduction to Programming for First-year Students at a Chinese University using LEGO® Mindstorms", *IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, Bali Dynasty Resort, Kuta, Indonesia, pp. 233-238, 2013.
- [38] Y. Liu, W. Wang and O. Ezenwoye, "Introducing Software Engineering to the Freshman Student", in *Proceedings of the 44<sup>th</sup> Annual Midwest Instruction and Computing Symposium*, Duluth, Minn., April 8-9, 2011.
- [39] G. López-Nicolás, A. Romeo and J. J. Guerrero, "Project Based Learning of Robot Control and Programming", 2009. [https://www.researchgate.net/publication/228973257\\_Project\\_Based\\_Learning\\_of\\_Robot\\_Control\\_and\\_Programming](https://www.researchgate.net/publication/228973257_Project_Based_Learning_of_Robot_Control_and_Programming), [Date Accessed: July 25<sup>th</sup> 2018].

- [40] T. Sharpe, R. Maher, J. Peterson, J. Becker and B. Towle, "Development and Implementation of a Robot Based Freshman Engineering Course", in *Proceedings of 2005 ASEE Annual Conference*, Portland, Oregon. <https://peer.asee.org/15301>, 2005.
- [41] R.M. Meyer and D.T. Burhans, "Robotran: A Programming Environment for Novices Using LEGO® Mindstorms Robots", *American Association for Artificial Intelligence (AAAI)*, pp. 321-326, 2007.
- [42] F. Michaud, "Engineering Education and the Design of Intelligent Mobile Robots for Real Use", *Intelligent Automation and Soft Computing*, vol. 13, no. 1, pp. 19-28, 2007.
- [43] A. C. Murillo, A. R. Mosteo, J.A. Castellanos and L. Montano, "A Practical Mobile Robotics Engineering Course using LEGO® Mindstorms", from book *Research and Education in Robotics - EUROBOT 2011: International Conference*, Prague, Czech Republic, pp. 221-235, 2011.
- [44] N.C. Özüorçun and H. Bicen, "Does the Inclusion of Robots Affect Engineering Students Achievement in Computer Programming Courses?", *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 13, pp. 4779-4787, 2017.
- [45] H. Patterson-McNeil and C. L. Binker, "Resources for Using LEGO® Mindstorms", *Journal of Computing Sciences in Colleges*, vol. 16, no. 3, pp. 48-55, 2001.
- [46] J.V. Ringwood, K. Monaghan and J. Maloco, "Teaching Engineering Design Through LEGO® Mindstorms", *European Journal of Engineering Education*, 30:1, pp. 91-104, 2007.
- [47] A. Valera, M. Vallés, L. Marín, A. Soriano, A. Cervera and A. Giret, "Application and Evaluation of LEGO® NXT tool for Mobile Robot Control", *Proceedings of the 18<sup>th</sup> World Congress the International Federation of Automatic Control (IFAC)*, Milano, Italy, 2011.
- [48] M.B.R. Vallim, J. M. Farines and J. E. R. Cury, "Practicing Engineering in a Freshman Introductory Course", *IEEE Transactions on Education*, vol. 49, no. 1, pp. 74-79, 2006.
- [49] J. LaCombe, E. Wang, and C. Rogers, "Using Lego Bricks to Conduct Engineering Experiments", in *Proceedings of 2004 ASEE Annual Conference*, Salt Lake City, Utah. <https://peer.asee.org/13880>, 2004.
- [50] A. B. Williams, "The Qualitative Impact of Using LEGO® MINDSTORMS Robots to Teach Computer Engineering", *IEEE Transactions on Education*, vol. 46, no. 1, pp. 205, 2003
- [51] M. Christie and E. de Graaff, "The Philosophical and Pedagogical Underpinnings of Active Learning in Engineering Education", *European Journal of Engineering Education*, vol. 42, pp. 5-16, 2016.
- [52] R. M. Lima, P. Hammar Andersson and E. Saalman, "Active Learning in Engineering Education: a (re)introduction", *European Journal of Engineering Education*, vol. 42, no. 1, pp. 1-4, 2017.
- [53] A. Oddie, P. Hazelwood, S. Blakeway and A. Whitfield, "Introductory Problem Solving and Programming: Robotics Versus Traditional Approaches", *Innovation in Teaching and Learning in Information and Computer Sciences*, vol. 9, pp. 1-11, 2010.
- [54] P. Pardjono, "Active Learning: The Dewey, Piaget, Vygotsky, and Constructivist Theory Perspectives", *Journal of Science Education*, vol. 9, no. 3, pp. 163-178, 2002.
- [55] M. Prince, "Does Active Learning Work? A Review of the Research", *Journal of Engineering Education*, vol. 93, pp. 223-231, 2004.
- [56] B.S. Bloom, (Ed.). M.D. Engelhart, E.J. Furst, W. H. Hill and D.R. Krathwohl, "Taxonomy of Educational Objectives", *Handbook I: The Cognitive Domain*. New York: David McKay Co Inc., 1956.

- [57] M. Usman, R. Britto, J. Börstler and E. Mendes, “Taxonomies in Software Engineering: A Systematic Mapping Study and a Revised Taxonomy Development Method”, *Information and Software Technology*, Volume 85, pp. 43-49, 2017. ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2017.01.006>.
- [58] R. Britto and M. Usman. “Bloom's Taxonomy in Software Engineering Education: A Systematic Mapping Study”, in *Frontiers in Education Conference (FIE), 2015 IEEE*, pp. 1-8, 2015. DOI:10.1109/FIE.2015.7344084
- [59] [https://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3) , [Date Accessed: February 1, 2019].
- [60] <https://pixycam.com/pixy-cmucam5/> , [Date Accessed: February 4, 2019].
- [61] V. Braitenberg, “*Vehicles: Experiments in synthetic psychology*”, Cambridge, MA: MIT Press.