ITcon

www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

USING ROBOT KITS FOR TEACHING RAILWAY ENGINEERS

PUBLISHED: September 2012 at http://www.itcon.org/2012/28
GUEST EDITOR(S): Xiangyu Wang and Robert Klinc

Yung Cheng Lai, Assistant Professor,
Transportation Group, Department of Civil Engineering, National Taiwan University.
yclai@ntu.edu.tw

Hung Lin Chi, Ph.D. Student,
Computer-Aided Engineering Group, Department of Civil Engineering, National Taiwan University.
hlchi@caece.net

Wei Choung Cheng, Ph.D. Student,
Computer-Aided Engineering Group, Department of Civil Engineering, National Taiwan University.
anthony@caece.net

Shih Chung Kang, Associate Professor,
Computer-Aided Engineering Group, Department of Civil Engineering, National Taiwan University.
sckang@ntu.edu.tw

SUMMARY: *This paper describes a methodology that uses robot kits for teaching railway engineering. Current railway engineering courses focus more on the theories of railway design and management. The lack of hands-on experience of students may result in design defects in practice. Automation and Robotics, an optional course designed for senior students in the Department of Civil Engineering at National Taiwan University, was specifically designed as a 4-week courseware for training future railway engineers. In addition to studying the theory of railway control systems, students were required to implement a railway control system using the robot toolkit LEGO Mindstorm NXT, and the robot platform Microsoft robotics developer studio (MSRDS). After the 4-week course, as a final project students were divided into six teams and demonstrated their automatic train control (ATC) systems. From the project demonstration, we found that the designs of all six teams were conceptually very similar, differing only in certain characteristics. Four of the six teams successfully delivered stable ATC systems. According to feedback obtained from the questionnaires, students had a very positive learning experience. Therefore, the authors conclude that the incorporation of these hands-on training techniques into advanced design courses will be very successful.*

KEYWORDS: *Railway Education, Robotic, Automatic Train Control.*

REFERENCE: *Yung Cheng Lai, Hung Lin Chi, Wei Choung Cheng, Shih Chung Kang (2012) Using robot kits for teaching railway engineers, Journal of Information Technology in Construction (ITcon), Vol. 17, pg. 416-433, http://www.itcon.org/2012/28*

1. EDUCATION OF RAILWAY ENGINEERS

Railway transportation is the more energy efficient compared to highway and air transportation. Therefore, it was deemed a key solution to the growing transportation needs, and tight environmental requirements for the 21st century (FRA, 2009). Unfortunately, as the demand for rail transportation increases, the industry faces a significant shortage of engineers, due to the lack of an adequate railway education infrastructure (Barkan, 2008). A large portion of rail employees are approaching retirement age. As a result, new employees are often required to assume job responsibilities soon after joining the industry. Consequently, if these engineers are sufficiently educated on railways, they can adapt faster to the industry, and perform better.

In terms of railway education, universities are typically responsible for providing to future engineers fundamental railway knowledge. A competent railway engineer should have a general knowledge of every element in a railway system, including infrastructure, rolling stock, traffic control and operations, and network service design. Because all these elements interact closely, they often have to be considered together in the planning, operation, and management processes. For example, service design aims in designing appropriate railway services to accommodate customer demands. To do so, design managers need to account for available resources, such as capacity resources from the infrastructure and train control systems, and also the available rolling stock. Therefore, a railway education curriculum is designed according to the important elements in each area. To receive certification, students are required to take and pass several introductory courses, such as Railway Transportation Engineering, professional courses, including Track Engineering, Railway Traffic Control, and Signaling Systems, as well as system courses, such as High Speed Rail Engineering, and Mass Rapid Transit System Engineering.

Most railway courses utilize a standard university lecture format. While this style may be appropriate for some courses, students sometimes have difficulties fully comprehending the logic and concepts of other courses, especially Railway Traffic Control and Signaling Systems. Railway signaling is a system used to safely control railway traffic by preventing collisions. Because the movement of trains has only a single degree of freedom and trains usually operate at speeds that do not allow drivers to stop them within sighting distance of a signal, the appropriate design of a signaling system is crucial to ensure safe operations. Besides safety considerations, traffic control systems also dictate how much capacity the infrastructure can carry, and how efficiently the system is used. Engineers are responsible for designing the most suitable control system according to demand.

Railway Traffic Control and Signaling Systems is a class covering the fundamentals of rail traffic control. Students in this class obtain a clear understanding of train movement authority, train position monitoring systems, train control systems, and special considerations in interlocking design, operation, and control. Some of these elements and logics are too complex to be explained by a lecture-style of teaching. An interactive teaching style providing students with hands-on skills on train control will be significantly more effective. For instance, modern metro systems are often equipped with automatic train control (ATC) systems, which is a framework consisting of three main components: automatic train protection (ATP), automatic train operation (ATO), and automatic train supervision (ATS) (as shown in Figure 1).

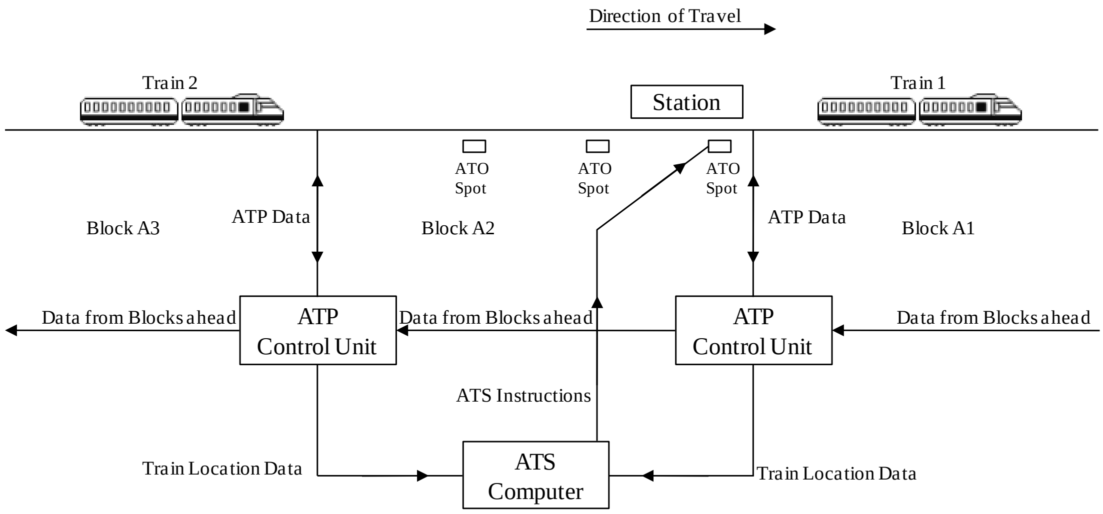

FIG. 1: Architecture of an Automatic Train Control System (Railway Technical Web Pages, 2010).

In the ATC framework, ATP is the primary means of keeping trains a safe distance apart. The ATP control units, which are installed in every signal block, receive data from the blocks ahead and convert it into a speed limit for the block they control. The speed limit data is then transmitted to the track. When trains enter this block, they receive the data and follow the speed limit.

ATS is another component in the ATC framework, which is a system for supervising and controlling the movement of trains. It monitors the speed and location of trains, and then compares the data with the timetable to check whether trains are running on schedule. If it is necessary to adjust the timing of a train, the ATS sends commands to the ATO spots located along the track.

ATO is the non-safety part of train operation related to station stops and starts. ATO spots send data concerning the time and location the train should stop, and if a train speed adjustment is required, it may instruct the train how fast to travel to the next station.

As can be seen, the framework for a railway traffic control system can be quite complex; without hands-on training it is difficult to understand and follow these inbuilt system logics. A lack of understanding may cause design defects when these students face real problems in practice. A better course plan should include opportunities for students to design and implement these control logics in a model railway system. This way, students can validate their design concepts and realize the complex logics behind the scene (Lindsay, 2008). Consequently, there is a need for an educational tool to achieve these objectives.

2. TEACHING CHALLENGES

Providing engineers with sufficient skills for dealing with the design, maintenance, and management problems of a railway system is the most important topic in the field of education. This topic mainly concerns three challenges: complexity, feasibility, and proficiency. To overcome them, some modifications are required in the railway engineering courses.

Complexity: Unlike solving problems on paper, in practice, when designing or managing a railway system, a railway engineer will encounter problems with very complex control mechanisms. The problems may contain numerous decisions with respect to the selection of train types, and the schedule and route plans of trains. The scale of a practical railway system is typically much larger than the theoretical models presented in class. As a result, theoretical models in textbooks may not be sufficient for students to fully understand the complexity of railway systems. If these course elements can materialize and be rendered operable (Blank et. al., 2007), the teaching can be much more effective, and there is a better chance for students to understand how the system works.

Feasibility: In addition to the complexity issue of railway education, there is a gap between the theoretical models and practical railway systems, which is not currently emphasized. For example, it is difficult for students to fully understand the software-hardware integration issues without hands-on experiences. In reality, the malfunction of sensors, traffic signals, and mechanics in railway systems can cause delays or even disasters. Reliability is one of the most important issues that these future railway engineers should address in class. Furthermore, they should design concepts for error handling and management, before they have to deal with a real case where they are required to obtain feasible results.

Proficiency: In addition to complexity and accuracy issues, students without hands-on experiences may also be overwhelmed in practice due to the lack of opportunities to test and implement railway system designs. Current courses related to railway engineering focus more on the concepts of railway design and management. They provide little opportunities for students to design a real system, not mentioning opportunities for the implementation and validation of design concepts. The lack of hands-on experience may cause design defects in practice. As a result, it is necessary to introduce efficient and flexible teaching aids to increase the duration of hands-on training during courses.

To overcome these challenges, the goal of this project is to propose integrated robot kits as teaching aids for railway engineering courses. Robot kits are widely used and discussed in the field of computer science education (van Lent, 2004; Burhans et. al., 2006), and have been integrated into undergraduate (Wolfe et. al., 2003; Dias, 2007) and graduate education (CMU, 2003; Matarić, 2004). These robot kits may have a great potential for integration with engineering courses. The authors designed a 4-week courseware for training railway engineers. To evaluate the feasibility of applying this new teaching style to other engineering courses, we collected and analyzed feedback and performance results from students.

3. COURSE DESIGN

Automation and Robotics, an optional course designed for senior students in the Department of Civil Engineering at National Taiwan University, includes a 4-week courseware. The courseware provides theoretical lessons, robot kit instructions, and term project scenarios. Based on these, the students prototype and implement the main control mechanisms of a railway system. Through this hands-on process, students become familiar with design concepts and realize the difference between simulation models and real situations. In the following sections, the authors will describe the preparation of the teaching aids and the schedule of the course.

3.1 Teaching Aids Preparation

To provide a hands-on learning environment with operable flexibility to teach railway engineering topics, the authors use robot kits. Specifically, we use the LEGO Mindstorm NXT package for hardware components, and Microsoft robotics developer studio (MSRDS) as the software platform for this course. These tools assist with the visualization of concept models for railway control theories so that students can easily understand them.

LEGO Mindstorm NXT is a hardware product from LEGO Corporation, in collaboration with the Massachusetts Institute of Technology (MIT), and released for robot education and development purposes (LEGO Corporation, 2010). It incorporates sensing, motion, and control components to provide robots with a high degree of flexibility, and allows structural designs enabling the rapid construction of an intelligent robot prototype (Cliburn, 2006; Workman and Elzer, 2009). For the students in the class who do not have a strong background in electrical and mechanical engineering, this robot kit can serve as a prototyping tool for them to demonstrate their designs of railway systems. Moreover, it can be used by teachers to build their lessons and demonstrate to students working theoretical models. For these reasons, this tool has been used in class.

MSRDS (Bruyninckx, 2007; Choi et. al., 2009) is a software product in the field of robotics, introduced by Microsoft in 2006, and supports Coordination and Concurrency Runtime (CCR) and Decentralized Software Services (DSS). These features allow us to decouple the binding relationships between each component of the robot system, and permit the system to retain workability even when some components have malfunctioned (Microsoft Corporation, 2008). As shown in Figure 2, it also supports Microsoft's visual programming language (VPL) environment. Unlike other robot platforms, such as the OROCOS project (Bruyninckx, 2001; Markou and Refanidis, 2009), VPL provides a high-level graphical interface that is easily accessible to students who have limited software engineering experience, allowing them to easily integrate various software modules .

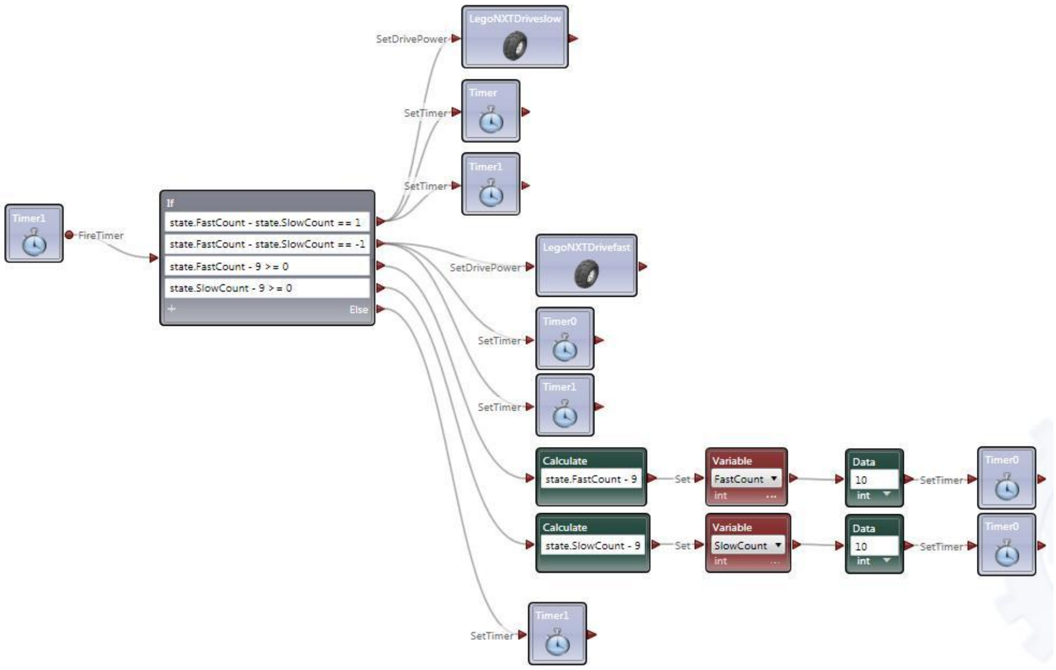

FIG. 2: Interface of Microsoft Visual Programming Language.

By combining the LEGO Mindstorm NXT and the MSRDS platforms, students in the course can easily construct railway models and program internal mechanisms for controlling them. In addition to these tools, the course also provides on a website the references, videos, and technical reports related to railway engineering. Students can download the materials before every class and practice the example programs using the robot kits.

3.2 Course Schedule

In the 4-week course, the essential elements of an ATC system are arranged into four lessons: track guidance, blocking mechanisms, passing movement, and system integration. In the first lesson, the topics cover basic knowledge concerning the tracks of the ATC system, such as the introduction of track types, track components, and so on. Template programs for track guidance using robot kits are also included as teaching material. Students can follow the templates presented in every lesson to build their own system. The second lesson is about blocking, which is a control mechanism for preventing train collisions by setting blocks on the tracks, and localizing every train in the railway system. In the third lesson, students are taught a common strategy in ATC systems, called passing movement, which allows a fast train to pass a slow train for the sake of efficiency. In the final lesson, the topics of the previous three lessons are integrated, and students are asked to develop their own railway system design and implement it. These lessons are taught in lectures and through hands-on practice, according to the 4-week course schedule presented in Figure 3.

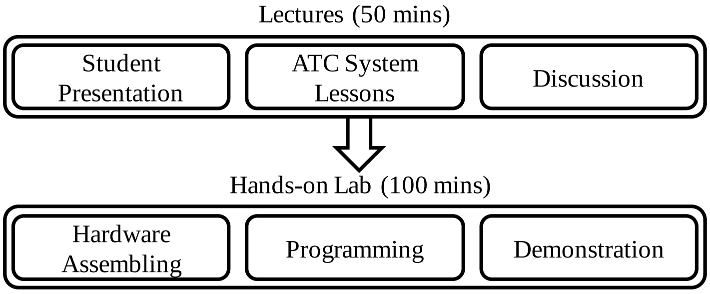

FIG. 3: Flow of each lesson of the 4-week course.

The weekly class is divided into two sections. The first section is approximately 50 minutes long. It covers basic knowledge of railway design; including track mechanisms, train controls, station management, and the four main components mentioned above. Moreover, each week a student presents to the class a paper provided on the course website. After the course, the presentation slides are submitted to the course website for students to share the knowledge they acquired. The second section is approximately 100 minutes. It focuses on implementing railway designs. Each week, the teacher teaches only one or two components of the railway system. Then, these are implemented by a robot toolkit, LEGO Mindstorm, and a robot platform, MSRDS. At the end of each class, students need to complete the components and test their performance.

4. PROJECT DESIGN

After the 4-week course, students were divided into six teams and conduct group design projects. At the end of the project, they were required to demonstrate a prototype railway system with ATC (Murphy, 2001).

4.1 Project Description

ATC refers to the whole system, including ATC functions and a degree of manual intervention. In this project, each team developed a small-scale ATC system for a particular scenario. They were required to study the design criteria, such as traveling time, train types, and rules for defining ticket prices. There were two major parts in this project: (1) to design the ATC system, and (2) to implement it.

The required ATC scenario was a simple loop railway system with three stations and two trains. As managers of the railway, each team attempted to maximize the total revenue by selecting the ticket prices for each origin-destination pair (OD pair), number of types of trains to operate, and the stopping pattern of each train. The conceptual model of the railway can be seen in Figure 4.

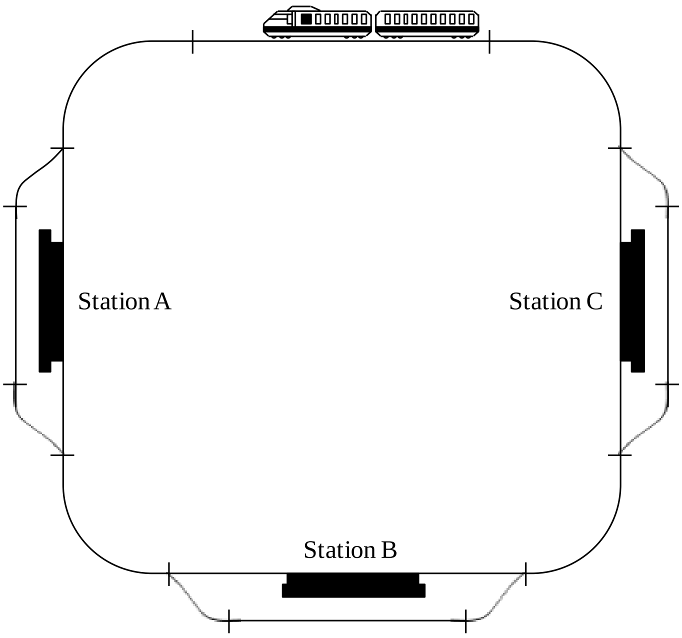

FIG. 4: Scenario of the small-scaled ATC system.

The characteristics of two types of trains, fast train (A) and slow train (B), are shown in Table 1. The relationship between price (P) and demand (D) (passengers per hour) of each OD are also provided in Table 2.

TABLE 1: The characteristics of the two types of trains.

Category	Number of Stops	Possible Stopping Patterns	Capacity (Passengers/Train)	Average Speed (kph)
Type A Train (Fast)	2	One of the following patterns: A-B, B-C, or C-A.	700	60
Type B Train (Slow)	3	A-B-C	700	30

The following are the requirements and assumptions of this project: all trains should run in a counterclockwise direction; every station should be served by at least one train; if required, a Type A train can pass a Type B train at any station; both the station dwell time and operating cost is ignored to keep the problem simple; price is independent of distance, and thus, different prices can be charged for different links with the same distance.

TABLE 2: Relationship between price and demand.

OD Pair	Type(A)	Type(B)
AB	D= −6P + 240	D= −120P + 1200
BC	D= −4P + 120	D= −100P + 1000
CA	D= −5P + 200	D= −90P + 900
AC	D= −12P + 720	D= −15P + 300
BA	D= −10P + 600	D= −10P + 200
CB	D= −9P + 540	D= −12P + 240

After designing the ATC system, each project team began implementing their designs, which included four essential elements. (1) Track and Train Integration: each team should design and implement the integration mechanism of trains. The track needs to be carefully prototyped using papers and tapes. Track templates were provided for students in the class to ensure a constant curvature. (2) Block Signaling: adjacent trains should be controlled by the mechanism of block signaling to avoid collisions. Students may choose one of the methods mentioned in class. (3) Passing Movements: in order to let fast train overtake slower ones, trains are capable of performing passing movements in stations. (4) Extra Design: extra designs regarding railway control are welcomed to assist system implementation.

4.2 Project Materials

The materials required to conduct the final course project were provided. Materials included track and train templates, tapes, papers, and so on. Students could follow these templates to develop their own trains and tracks. Similarly, the paths of the designed railway were implemented by placing the black tapes on the papers. Students could follow the scenario requirements of this project, or implement different types and shapes of tracks. The use of these materials was not limited, and students were encouraged to alter designs to complete their projects.

As shown in Figure 5, the basic structure and cover of trains was provided. Basically, the authors used LEGO Mindstorm kits to create a walking machine called Railbot. It contains two motors for controlling the wheels in each side, and two light sensors for detecting the tracks. The covers of the train were made by paper, and were provided to students for decorating their Railbots.

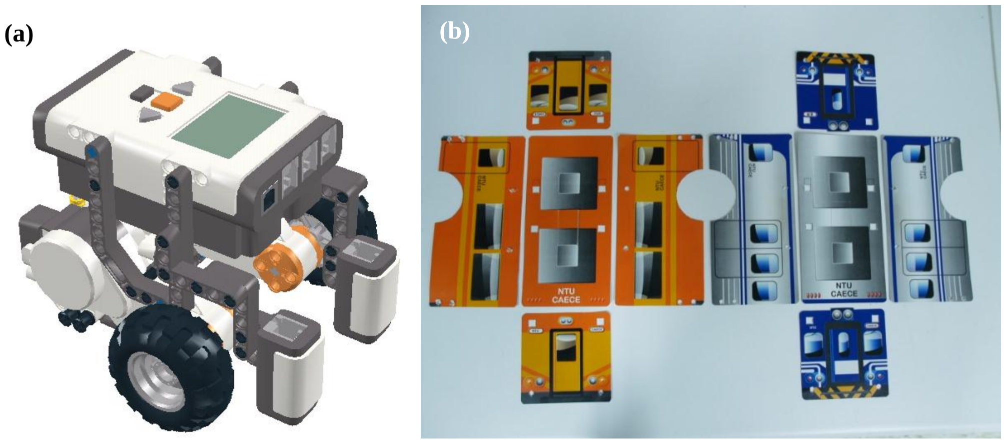

FIG. 5: Schematic of Railbot: (a) structure template; (b) cover.

The recommended movement strategy of the Railbots in the railway system is presented in Figure 6. Two light sensors mounted at the front were used to detect the path of the track. The light sensors received different intensity of light reflected from black tapes and white paper, and the Railbots were able to react appropriately and follow the track. Learning these control mechanisms was also part of the 4-week coursework.

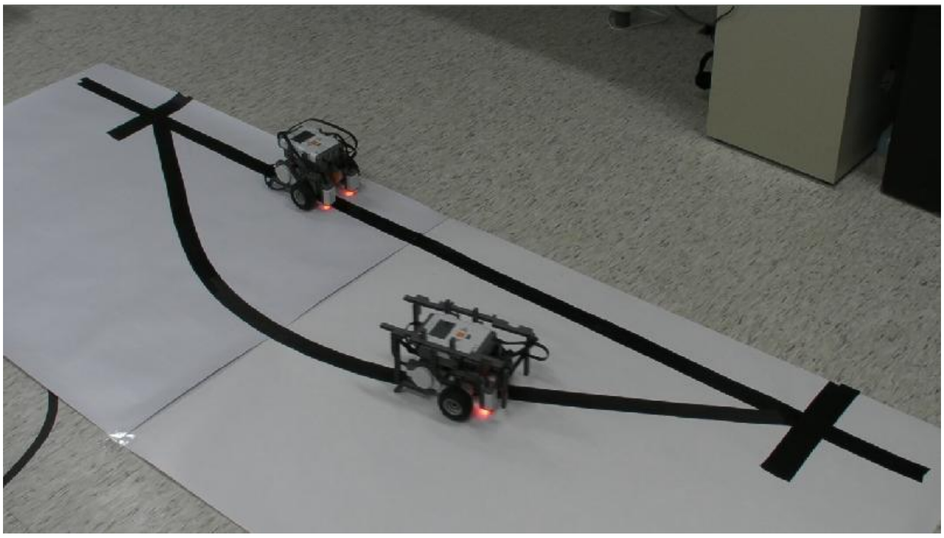

FIG. 6: Movement of the Railbot.

Both hardware and software designs were evaluated in terms of effectiveness, performance, and creativity. Grading was divided into two parts: demonstration and reporting. Following the instructions of the project assignment, students were required to design and implement a capable railway system with robust performance, adhering to a profitable train schedule. They also were required to demonstrate the system, and show the major functions in 10 minutes. In the second part, the students prepared a report outlining both hardware and software system designs.

5. EXAMPLE OF A PROJECT IMPLEMENTATION

After working for two weeks on designing and implementing the ATC system, students presented details of their implementation in reports and demonstrated their system designs in class. The authors present as an example, the project implementation that was the most complete and efficient.

5.1 Ticket Strategy

The requirement of the first part of this project was to plan a strategy for operating a simple loop railway system with three stations. In order to achieve the goal of maximizing revenue, students had to select the ticket price for each origin-destination pair, the number of types of trains to operate, and the stopping pattern of each train.

According to the predefined assumptions, operating costs can be ignored. Thus, the revenue was given by:

$$\textit{Revenue}\ (R) = \textit{Price}\ (P) \times \textit{Demand}\ (D) \quad (1)$$

This equation depicts the common relationship between price and demand. The higher the price is, the less the demand (i.e. passengers).

Because the demand of each OD pair is a linear function of price, the revenue of each OD pair is a quadratic function. Thus, the team of our example obtained the maximum revenue and corresponding ticket price by obtaining the root of the derivative of the revenue. The results for fast trains are shown in Table 3.

TABLE 1. Result of obtaining maximum revenue for Type A trains.

OD Pair	Demand (D)	Revenue (R)	Maximum Revenue	Price	Demand (passengers/hr)
AB	−6P + 240	$-6P^2 + 240P$	2,400	20	120
BC	−4P + 120	$-4P^2 + 120P$	900	15	60
CA	−5P + 200	$-5P^2 + 200P$	2,000	20	100
AC	−12P + 720	$-12P^2 + 720P$	10,800	30	360
BA	−10P + 600	$-10P^2 + 600P$	9,000	30	300
CB	−9P + 540	$-9P^2 + 540P$	8,100	30	270

The results for slow trains are shown in Table 4.

TABLE 2. Result of obtaining maximum revenue for Type B trains.

OD Pair	Demand (D)	Revenue (R)	Maximum Revenue	Price	Demand (passengers/hr)
AB	−120P + 1200	$-120P^2 + 1200P$	3,000	5	600
BC	−100P + 1000	$-100P^2 + 1000P$	2,500	5	500
CA	−90P + 900	$-90P^2 + 900P$	2,250	5	450
AC	−15P + 300	$-15P^2 + 300P$	1,500	10	150
BC	−10P + 200	$-10P^2 + 200P$	1,000	10	100
CB	−12P + 240	$-12P^2 + 240P$	1,200	10	120

Based on these results, the total revenue and maximum number of passengers per hour for each route option were obtained, and are presented in Table 5.

TABLE 3. Route options.

Route	Revenue	Max. passengers/hr	Exceed Capacity?
A-B-A (fast)	11,400 (AB + BA)	300	No
B-C-B (fast)	9,000 (BC + CB)	270	No
C-A-C (fast)	12,800 (CA + AC)	360	No
A-B-C-A (slow)	11,450 (Sum. of all OD pair)	870 (A-B segment: AB + AC + CB)	Yes

Because demand exceeded the capacity of the train, students tried to adjust the prices for some OD pairs to meet the criteria. When the price of AB was raised to 6.42, and the price of BC was raised to 5.5, the re-calculated passenger demand did not exceed 700 per hour, and the revenue dropped only to 11,184, which was less than the A-B-A fast train route.

As a consequence, this team chose as their solution the strategy of using two fast trains (A-B-A and C-A-C), as shown in Table 6.

TABLE 4. Ticket prices and operation strategy

Types of train	Stopping pattern	Ticket price	Max. total revenue
Fast Train 1	A-B-A	20 (AB) and 30 (BA)	11,400
Fast Train 2	C-A-C	20 (CA) and 30 (AC)	12,800

5.2 Hardware Design

5.2.1 Train design

This team built the two trains by basically following the instructions provided in the course. Each train was comprised of one NXT microcomputer, two servomotors, and two light sensors. An example of a finished train is shown in Figure 7.

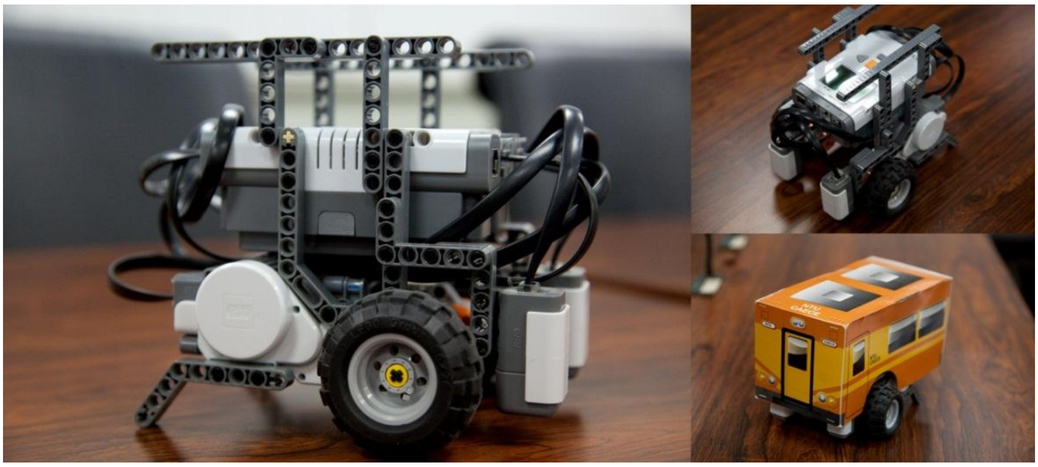

FIG. 7: Train assembled using LEGO MINDSTROM NXT.

- Sensors

The train was equipped with two light sensors to distinguish between dark and light areas. The two sensors were mounted at the front of the train to detect the signals, or to determine whether the train was moving along the track.

- Drive

The train was equipped with two servomotors to drive the two front wheels, respectively. However, the rear wheel was removed from the original design to stabilize the direction when the train was moving at low speeds.

- Microcomputer

The microcomputer brick on the train was linked to the sensors and servo motors using cables. The management computer used a Bluetooth interface, instead of USB, to establish a wireless connection to transmit data, states, and commands.

5.2.2 Track design

- Rail

Rails were represented using 30-mm-wide black lines on a white surface; this width is a little smaller than the distance between the two light sensors of the train. In order to achieve a uniform rail width and smooth curvature, this team did not use the tapes to construct the tracks, but instead drew them using graphic software, and printed them using a large format printer. By combining the printed track components together, the whole track layout, including passing loops, signals and main (single) tracks between stations, was finally constructed. The track layout was approximately 220×200 mm. Figure 8 illustrates the track layout and locations of stations.

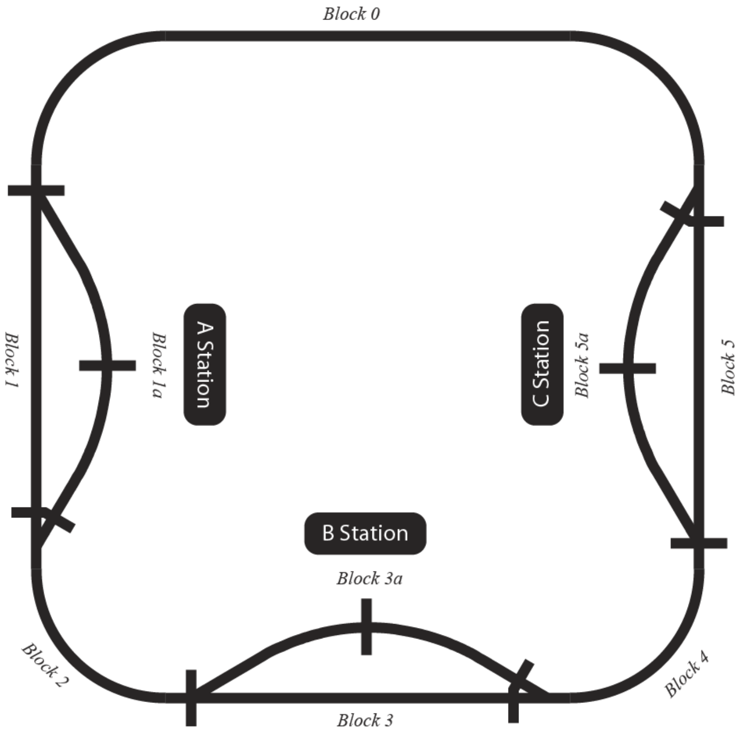

FIG. 8: Track layout and locations of stations.

- Passing loop

The passing loops used in this project permit a train to overtake another if necessary. The team designed the passing loop to have turnouts at either end, while the station is located on the loop line. Therefore, trains expected to pass through stations can run on the main line at full speed, rather than reducing their speed on the curve. In this project, trains are only allowed to run in a counterclockwise direction around the loop railway system. The turnout at the entrance of the passing loop was a left-hand lateral turnout, whose diverging track went to the left. A train that intends to stop at the station must turn left by 30 degree at this switch, and then rejoin the main line at the other end, which is a right-hand lateral turnout. Conversely, a train scheduled to pass through the station, must go via the straight route at the switch. The length of the passing loops of the model is approximately 112 mm.

- Signal

The team used as signals short lines that were perpendicular to the rails. Signals were placed in three different locations of the track system: at switch points at the entrance of passing loops, at stations, and at switch points at the exit of passing loops. These signals not only notified trains about the presence of turnouts or stations, but were also used as block signals that governed the movement of trains entering the blocks. Furthermore, the short lines were tested to ensure that they were thick enough to be detected by the light sensors installed on the trains.

5.3 Software Design

There are four main parts in the software design of this team: initializing, track following, passing, and blocking.

- Initializing

Before trains were allowed to move, several initial parameters needed to be set in order to define the operational strategy. These parameters include the current location of trains, stations to stop for each train, state of each train, train speed, time to stop at a station, and frequency of collecting sensor data measurements when following the track. Because some train behaviors, such as speed and stopping pattern, could be parameterized through the software program, the team in addition to realizing their solution (two fast trains), they also realized different strategies and stop patterns.

- Track Guidance

Track guidance was accomplished by continuously monitoring the measurements obtained from the sensors. First, if both light sensors detected light areas, the computer commanded the train to go straight ahead. Conversely, if only one light sensor detected a light area, the computer commanded the train to revise its direction by turning left or right in order to follow the track. Finally, if both light sensors detected dark areas, which indicate that a signal is encountered, the computer decided the next move of the train according to its current location and blocking status of the track ahead. Nevertheless, near signals and switches trains moved in a fixed pattern, instead of following the track.

- Blocking

The blocking behavior of this team's design was implemented by using the location and state of each train, rather than using the blocking status of every track segment. In a two-train scenario it is easier to design the dataflow using the location and state of each train. When a train encountered a signal, the computer compared the locations and states of both trains, and determined which track it was allowed to enter.

- Passing

Because all trains should run in the same direction, and this team designed only one platform on a diverging track, passing was only possible in the station when a train that was not planned to stop could overtake another train that was stopping. To prevent the two trains from colliding near the switch, both tracks were blocked until one of the two trains left the passing loop. The only exception was when a train was waiting at the station, in which case the passing line was unblocked. Figure 9 shows the part of the program that implements the passing behavior.

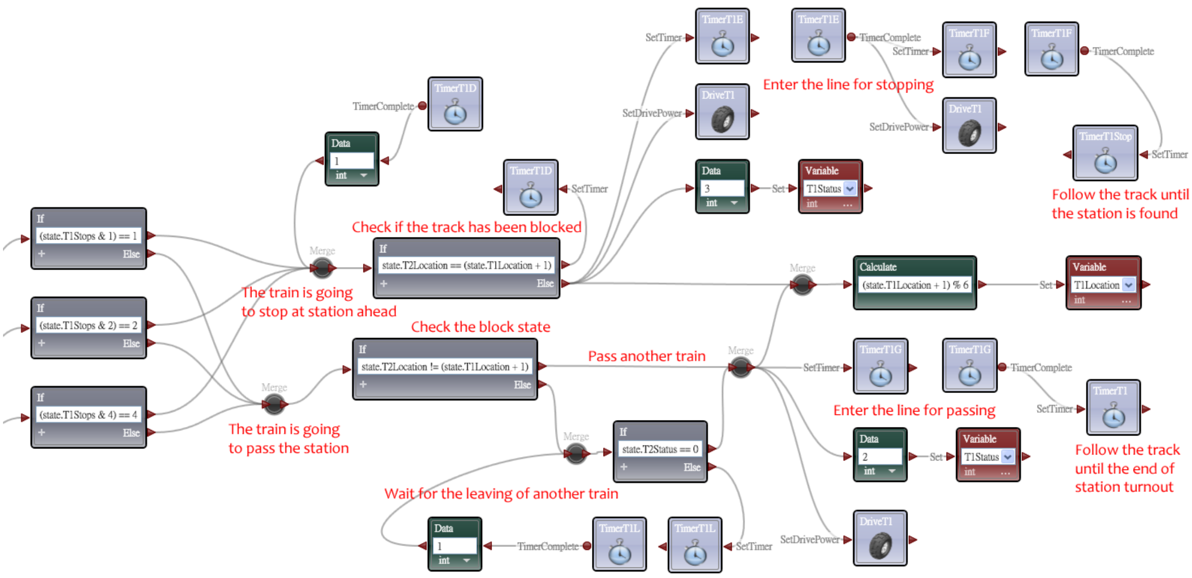

FIG. 9: Dataflow for a train entering a block that has a station.

6. DISCUSSION OF PROBLEMS ENCOUNTERED

The selected team encountered some problems when designing the hardware and implementing the ATC system. It took the team a substantial amount of time to identify the sources of the problems and correct them. However, in certain cases they were forced to change their design, because the problems identified were due to hardware or software limitations. These problems are discussed in the following sections.

6.1 Robot Control

Even though LEGO MINDSTORMS NXT has a rich set of inbuilt functions and various add-on components that can be used to fulfill the needs of a complex design, there are still some hardware limitations that should be considered during implementation.

First, the readings of light sensors were heavily influenced by the environmental lighting conditions. Every time the lighting or location changed, the team had to calibrate the parameters in their application. In addition, the method used to mark rails and the paper that was used for the track background affected the stability of sensor readings, which consequently affected the efficiency of track following.

Second, the remote control of the trains was achieved by linking the microcomputer of the robot and the management computer through a serial interface, via a Bluetooth wireless connection. The performance of real-time control, especially for continuous track guidance, depended largely on the efficiency of transmitting data through the wireless connection (provided that the performances of the processor of the robot microcomputer and the management computer running the control application were good enough). However, the team discovered that a notable latency existed in the transmitting process, which affected the performance of track following. Fine tuning the polling frequency of the light sensors reduced the problem, but this limitation had to be addressed in the design of the application to avoid unexpected train movements.

Finally, the team found that the servomotors in the LEGO MINDSTORMS NXT package delivered adequate mechanical power to drive the robot, and provided good control of the movement. However, students needed to drive the train at low speeds, because the motors did not respond immediately to changes in the reading of the sensors, as mentioned in the previous section. Moreover, because the servomotors could not provide a constant speed at low-speed situation, it was difficult to precisely control train movement. Adding some gear wheels to alter the gear ratio might be a solution, but these parts would require additional space, and it would not be feasible to maintain the original size of the train.

6.2 Software Development

Before the final project, the students learned how to control a train assembled using a robot kit. In particular, they had practiced basic behaviors, such as simple track guidance, blocking, or passing. The VPL programs implementing these behaviors were not complex. However, in the final project, simultaneously controlling the movements of two trains was actually challenging. Several problems in developing the program were encountered by this team.

First, in VPL there is a limitation in handling the notification outputs of activities. To modularize VPL applications, developers typically create custom activities to reuse common dataflow sequences. However, this team realized that they were not allowed to receive notifications from any service implemented in custom activities. This limitation made a considerable impact on the design of the program, because the movement control of the trains, especially for track guidance, relied heavily on receiving notifications from the sensor's monitor and timers to decide the next move. As a result, students had to implement most of the dataflow in the main diagram, which was difficult to maintain and debug as the program grew larger.

Second, the team found that programmers should be careful when using timer services in VPL, to prevent them from interfering with each other. In order to control two trains to simultaneously perform different behaviors, multiple timer services were simultaneously used. However, students found that during program execution, a running timer occasionally interfered with another when they used a request of the timer service called 'Wait' to wait for a certain time interval. The running timer interference resulted in a temporary unavailability of the service for other timers. Therefore, the corresponding dataflow sequences of the other timers did not run on expected time intervals, thus causing irregular movements of trains. To overcome this problem, the team had to create more timer instances, and wait for the notification of completion from each timer. Nevertheless, this alternative method required more computer resources, and it was hard to modularize the program by integrating similar actions into custom activities.

Finally, the students thought that the current version of Microsoft VPL is not suitable for developing complex large-scale programs. The graphical dataflow-based programming model is targeted for beginner programmers, such as engineers with only a basic understanding of programming. Though it is much easier for engineers to learn and implement robot control applications using VPL, students found that the user interface of the development environment was slow to respond during development in this project. Moreover, when running the program on a computer with mediocre speed, delayed responses of activities often caused abnormal moving behaviors of trains. The complexity or the size of the program was the reason for the poor performance. Furthermore, it is inconvenient for programmers to manipulate large amounts of structural data, due to the lack of appropriate functions. In short, robotics programmers should consider implementing complex logic within MSRDS using C# or VB.NET. Alternatively, other platform, such as LabVIEW, should be considered for developing robotic applications.

6.3 Complexity

In this project, students realized the complexity of developing a robot system capable of performing complex behaviors. The uncertainty of the responses of the robot was significant when controlling multiple robot trains simultaneously in a system. For example, it was difficult to reproduce identical train movements, even though the parameters for controlling the robots were unchanged. Moreover, a minor modification to the controlling sequence of a train might alter the behavior of the other train. In addition, environmental conditions, the sensitivity of sensors, the output stability of actuators, and the state of the asynchronous operations of the control application, all influenced the behaviors of the robot trains. As a result, it was difficult to test and debug such a complex system, and students needed to address the uncertainty and make their system as reliable as possible to accomplish their course objectives.

7. RESULTS

7.1 Demonstration Performance

From the project presentations, the authors found that the scenario designs of all six teams were identical, but the implementation styles were quite different. Four of the six teams successfully delivered a stable system. Two of them encountered hardware and software integration problems, and could only demonstrate part of their systems. In this section, the authors list the integration problems identified by the students:

- Overall, each team implemented a very different type of railway system. Not only are the designs of the track different, but also the control mechanisms used. Even though each team derived almost the same answer for optimizing the given scenario, the implemented systems exhibit quite a bit of variation in the final results.

- During the presentation, many teams encountered unexpected situations. For example, the Railbots would go out off-track, or the sensors would miscount the block, causing system instability. Most teams had built check mechanisms, such as voice warnings or counting dialogs to detect such situations. These mechanisms helped them understand and explain the causes of these unexpected situations.

- Students indicated that discovering the problems in their systems was a difficult task. They realized that railway systems are complex, and even small problems, such as inaccuracy of the Railbots' speed, sensor’s communication frequency and so on, can cause the whole system to fail.

- The two teams that suffered from integration problems indicated that they incorrectly estimated the time required to implement their system. The time spent programming and modifying the algorithms prior to obtaining stable results was much longer than originally predicted.

- With respect to the implementation of the project, students indicated that finding an optimized solution to gain maximum total revenue for the assigned scenario was relatively easier than actually implementing it. They were forced to address hardware limitations and errors, and they found some solutions to avoid system crashes.

7.2 Questionnaire

At the end of the course, students were required to answer a questionnaire to evaluate this course. Their feedback can be seen in Table 7. In general, students responded positively to every part of the teaching method, especially the increasing of students' hands-on skills. From the students, 75% of them strongly agreed that the course helped them develop their implementation skills. When compared to traditional teaching methods, 87% of the students agreed or strongly agreed that the method of integrating the robot kits was better than traditional methods. From the students, 62% thought the course helped them think like engineers and made them want to take a similar course in the future. Unlike other questions which recorded no negative responses, almost 46% of them thought it was difficult to finish the assignments of this course.

TABLE 5. Feedback of students on the course

Questions	Strongly Agree (%)	Agree (%)	Neither Agree nor Disagree (%)	Disagree (%)	Strongly Disagree (%)
1. Course content is strongly related to engineering issues.	58	42	0	0	0
2. Course is well-prepared; I can understand the content and follow up.	50	41	9	0	0
3. Lecture describes the content clearly and can help me implement the project.	47	37	16	0	0
4. Assignment is too difficult for me or has some parts I am really unable to complete.	8	38	29	25	0
5. The project effectively enhances the understanding and implementation ability of all students.	50	37	13	0	0
6. This course is somehow better than the traditional railway teaching methodology; I can learn easier from it.	37	50	13	0	0
7. This course helped me develop my hands-on skills.	75	25	0	0	0
8. This course helped me learn how to think like an engineer.	62	38	0	0	0
9. I will attend similar courses in the future.	62	25	13	0	0

7.3 Lesson Learnt

Based on the results of the project performance and feedback, it was concluded that the robot kits are very effective tool for educating future railway engineers on railway signaling systems and control. A list of lessons learnt is presented below.

- Clear Understanding of the Control Logics: By using robot kits as teaching aids to prototype the conceptual model of the railway system, students can easily understand the theory taught in class. From the results of the presentation, students can easily describe problems encountered in complex railway system due to the time spent implementing and testing their projects.

- Consideration of Uncertainty: From the hands-on project, students understood the complexity of the railway system, and experienced the difference between theory and practical situations. This was evident by the integration problems encountered by the projects teams, in spite that most of them had no problem theoretically solving the design problems.

- Practice Opportunities: The course helps students examine their design, and at the same time solve practical problems, like building error handling mechanisms. This will be highly beneficial when students face these tasks in practice.

- Consideration of Integration Issues: By observing the results of each team, we noticed that integration issues between hardware and software when developing a railway system were encountered by all students. This is a fundamental issue that students need to address when designing or implementing similar projects in practice.
- Disadvantage of the Course: For a course of this duration, there were too many project materials and not enough instructions for the students. Next time this should be improved by providing a better project description, extending teaching hours, and designing appropriate scenarios.

8. CONCLUSIONS AND FUTURE WORKS

This research focused on developing a methodology that uses robot kits for teaching railway engineering. A 4-week course for implementing an ATC system was conducted. Students were required to implement the systems using LEGO Mindstorm NXT and Microsoft Robotics Developer Studio. After the 4-week course, the students were divided into six teams to demonstrate their works as a final project. All six teams produced very similar solution concepts, but differed in their implementation characteristics. Overall, each team implemented a very different type of railway system.

From the performance of the final project, and feedbacks obtained through the questionnaires, the authors found that students were very positive towards their learning experiences. From the students, 75% of them strongly agreed that the course helped them develop their implementation skills. The authors also found that using the robot kits was particularly helpful for training railway engineers. Because during the implementation part of the project students realized and modified their designs, they experienced the entire design processes of a railway system. Hence, they realized the complexity and difficulty associated with railway control. This will help them face more complex cases in practice.

In the future, this teaching method for railway engineering may be improved by providing more appropriate scenarios and clear instructions for enhancing the implementation experiences. The robot kits can also be used in teaching, to enhance advanced design courses involving automation and control systems.

9. REFERENCES

Barkan, C. P. L. (2008). Building an Education Infrastructure for Railway Transportation Engineering: Renewed Partnerships on New Tracks, *TR News*, 257, Transportation Research Board of the National Academies, Washington, D.C., 18-23.

Blank, D., Kumar, D., Marshall, J., and Meeden, L. (2007). Advanced Robotics Projects for Undergraduate Students, *AAAI Spring Symposium - Technical Report*, SS-07-09, 10-15.

Bruyninckx, H. (2001). Open Robot Control Software: The OROCOS Project, *IEEE International Conference on Robotics and Automation*, 3, 2523-2528.

Bruyninckx, H. (2007). Microsoft Robotics Studio: Expected Impact, Challenges & Alternatives, *Panel Presentation at the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10-14.

Burhans, D. T., Meyer, R. M., VanVerth, P., Puehn, D., Steck, V., Wiejaczka, J. P. (2006). Introductory Computer Science with Robots, *AAAI '06*, Boston, Massachusetts, Jul. 16-20.

Choi, B. W., Lee, S. H., and Yi, S. Y. (2009). Internet Coordinated Pet Robot Simulator Based on MSRDS, *ICCAS-SICE - ICROS-SICE International Joint Conference 2009*, Fukuoka, Aug. 18-21.

Cliburn, D. C. (2006). Experiences with LEGO Mindstorms throughout the Undergraduate Computer Science Curriculum, *ASEE/IEEE Frontiers in Education Conference 2006*, San Diego, CA, Oct. 27-31.

CMU. (2003). Educational Robotics, <http://www.education.rec.ri.cmu.edu> (Oct. 16, 2009).

Dias, M. B., Browning, B., Mills-Tettey, G. A., Amanquah, N., and El-Moughny, N. (2007). Undergraduate Robotics Education in Technologically Underserved Communities, *IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, Apr. 10-14.

FRA (2009). Vision for High-Speed Rail in America, Federal Railway Administration (FRA), <http://www.fra.dot.gov/rpd/passenger/31.shtml > (Dec. 13, 2010).

LEGO Corporation (2010). LEGO Mindstorms, <http://mindstorms.lego.com/en-us/default.aspx> (Mar. 15, 2010).

Lindsay, E., Munt, R., Rogers, H., Scott, D. and Sullivan, K. (2008). Making Students Engineers, *Engineering Education: Journal of the Higher Education Academy Engineering Subject Centre*, 3(2), 28-36.

Markou, G. and Refanidis, I. (2009). MSRS: Critique on its Usability via a Path Planning Algorithm Implementation, *5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI) 2009*, Thessaloniki, Apr. 23-25.

Matarić, M. J. (2004). Robotics Education for All Ages, *AAAI Spring Symposium - Technical Report*, 1, 14-16.

Microsoft Corporation (2008). Microsoft CCR and DSS Toolkit 2008 - Customer Solution Case Study, <http://go.microsoft.com/fwlink/?LinkId=130995> (Jan. 13, 2009).

Murphy, R. R. (2001). "Competing" for a Robotics Education, *IEEE Robotics and Automation Magazine*, IEEE, 8(2), 44-55.

Railway Technical Web Pages (2010). Automatic Train Control (ATC), ATC, <http://www.railway-technical.com/sigtxt6.shtml> (Feb. 28, 2010).

Wolfe, D., Gossett, K., Hanlon, P.D., and Carver, C.A. (2003) Active Learning Using Mechatronics in a Freshman Information Technology Course, *ASEE/IEEE Frontiers in Education Conference*, Westminster, Colorado, Nov. 5-8.

Workman, K. and Elzer, S. (2009). Utilizing Microsoft Robotics Studio in Undergraduate Robotics, *Journal of Computing Sciences in Colleges*, 24 (3), 65-71.

van Lent, C. E. (2004). Using Robot Platforms to Enhance Concept Learning in Introductory Computer Science Courses, *AAAI Spring Symposium - Technical Report*, SS-04-01, 115-117.