軌道自動化技術實作（自動化與機器人技術課程深化）成果完整報告

Yung-Cheng Lai, Shih-Chung Kang

1. Abstract

This project conducts a teaching methodology that uses robot kits for railway engineering. Current railway engineering courses focus more on the theories of railway design and management. The lack of hands-on experience may cause design defects for students in practice. Automation and Robotics, an optional course designed for senior students in the Department of Civil Engineering, has specifically designed a 4-week courseware for training these future railway engineers. Besides studying the theory of the railway control system, students were required to implement the railway control systems using a robot toolkit, LEGO Mindstorm NXT, and a robot platform, Microsoft Robotics Developer Studio, MSRDS. After the 4-week course, the students were divided into six teams to demonstrate their automatic train control (ATC) systems as a final project. From the project demonstration, we found that the designs of all six teams are conceptually very similar in the concept, differing only in certain characteristics. Four of the six teams successfully delivered stable ATC systems. According to feedback from the questionnaires, students were very positive towards the learning experiences. We therefore conclude that the incorporation of these hands-on elements into advanced design courses will be a great success.

2. Introduction

Railway is the more energy efficient transportation mode compared to highway and air. It was therefore 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 infrastructure in railway education in the past (Barkan, 2008). A large portion of rail employees are approaching retirement age. As a result, new employees are often required to get ready for their job soon after joining this industry. Consequently, the more railway education these engineers can obtain beforehand, the faster they can adapt to this industry, and the better their performances.

In terms of railway education, universities are usually responsible for providing the fundamental railway knowledge to future engineers. 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. All these elements interact closely so they often have to be considered together in the planning, operation, and management processes. For example, service design aims to design appropriate railway services to accommodate customer demands. To do so, the design manager needs to account for the available resources, such as the capacity resources from the infrastructure and train control systems, and also the available rolling stock. A railway education curriculum is therefore designed according to these important elements in this area. To receive the certification, students are required to take and pass several introductory courses, such as Railway Transportation Engineering, professional courses, including Track Engineering, and Railway Traffic Control and Signaling Systems, along with system courses such as High Speed Rail Engineering, and Mass Rapid Transit System Engineering.

Most of the railway courses utilize a standard university lecture format. While this style may be appropriate for some courses, students sometimes have difficulties fully comprehending the logics and concepts of other courses, especially Railway Traffic Control and Signaling Systems. Railway signaling is a system used to safely control railway traffic, especially by preventing collisions. Since the movement of trains has only a single degree of freedom and they usually operate at speeds that do not enable them to stop within sighting distance of the driver, the appropriate design of a signaling system is crucial to ensure safe operations. Besides safety considerations, the types of 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 from this class should have 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 comprehended by a lecture-style of teaching. An interactive teaching style providing students with a hands-on experience of train control that will be significantly more effective. For instance, the modern metro systems are often equipped with Automatic Train Control (ATC) systems, which is a framework including three main components: Automatic Train Protection (ATP), Automatic Train Operation (ATO), and Automatic Train Supervision (ATS) (as shown in Figure 1).

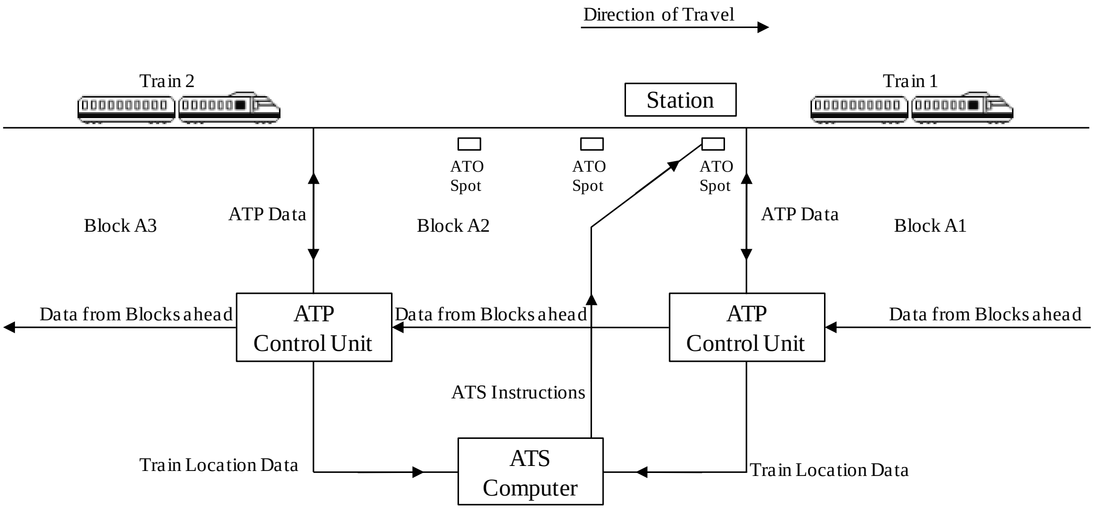

Figure 1. The Architecture of Automatic Train Control System (Railway Technical Web Pages, 2010)

In the ATC framework, ATP is the primary means of keeping trains a safe distance apart. The ATP control units, installed in every signal block, receive data from the blocks ahead, which they convert into a speed limit for the block it controls. The speed limit data is then transmitted to the track. The train entering this block then picks up the data and follows 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 if trains are running late or early. If an adjustment in the train’s timing is necessary, the ATS will send commands to the ATO spots located along the track.

ATO is the non-safety part of train operation related to station stops and starts. The ATO spots send data about the time and location the train should stop and may tell it how fast to go to the next station if any adjustment in train speed is required.

As can be seen, the framework for a railway traffic control system can be quite complex; without hands-on experience these inbuilt system logics are not easily understood or followed. This 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. In 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 accomplish these opportunities.

3. Course Design

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

3.1. Teaching Aids Preparation

For providing a hands-on learning environment and containing operable flexibility in teaching the railway engineering topics, we use robot kits, including LEGO Mindstorm NXT package as hardware components, and Microsoft Robotics Developer Studio (MSRDS) as software platform in the course. These tools assist with the visualization of the concept model for railway control theories so students can easily understand them.

LEGO Mindstorm NXT is the 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 equip the robots with a high degree of flexibility and allow structural designs to rapidly construct 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 knowledge, this robot kit can serve as a prototyping tool for demonstrating their design of railway systems. It is even accessible by teachers, who can build their lessons and show the working results of theoretical models to students. For these reasons, this tool has been used in class.

Microsoft Robotics Development Studio (MSRDS) (Bruyninckx, 2007; Byoung et. al., 2009) is a software product that Microsoft introduced to the field of robotics in 2006, which has the following features. It supports Coordination and Concurrency Runtime (CCR) and Decentralized Software Services (DSS). These features decoupled the binding relationships between each component of the robot system and permit the system to retain workability when some components have malfunctioned (Microsoft Corporation, 2008). As shown in Figure 2, it also supports the Microsoft Visual Programming Language (VPL) environment. Unlike other robot platforms, such as the OROCOS project (Bruyninckx, 2001; Markou and Refanidis, 2009), this provides a high-level graphical interface that is very accessible to engineering students who have limited software engineering experience, allowing them to easily integrate various software modules .

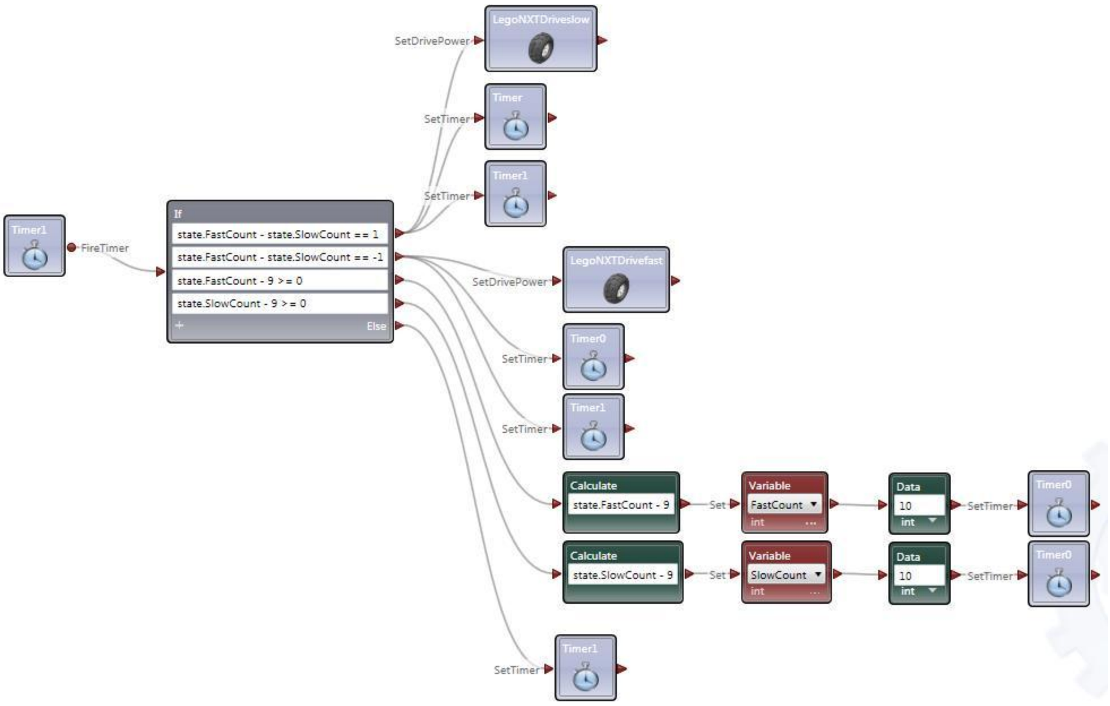

Figure 2. The interface of Microsoft Visual Programming Language

Combining LEGO Mindstorm NXT and MSRDS platform, students in the course can easily build up their railway models and program the internal mechanisms for controlling them. Besides these tools, the course also provides the references, videos and technical reports related to railway engineering on the course website. Students can download those materials before every class.

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, we cover basic knowledge concerning the tracks of the ATC system, such as the introduction of track types, track components and so on. We have also included a template program for track guidance by using robot kits. Students can follow these kinds of templates presented in every lesson to build their own system. The second lesson is about blocking. It is a control mechanism for preventing train collisions by setting blocks on the tracks and localizing every train among the railway systems. In the third lesson, students are taught a common strategy used frequently in ATC systems, called passing movement. It allows a fast train to come across a slow train for the sake of efficiency. In the final lesson, we integrate the elements of the previous three lessons, and ask students to develop their own railway system design and implementation. These lessons have been taught by lectures and hands-on practice according to the schedule of the 4-week course. The schedule of the 4-week course can be seen in Figure 3.

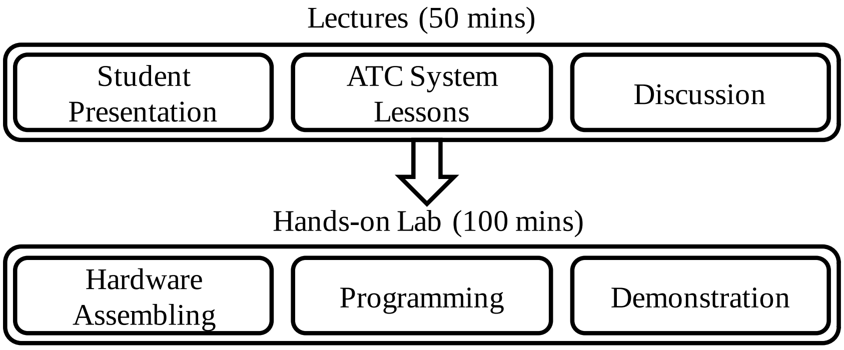

Figure 3. The flow of each lesson in the 4-week course

The class in each week is divided into two sections. The first section is approximately 50 minutes. It covers basic knowledge of railway design, including track mechanisms, train controls, station management, and the four main components mentioned above. Also, every week one of the students needs to study papers provided on the course website and make a presentation to everyone. The second section is approximately 100 minutes. It focuses on implementation of the railway designs. Each week, we teach only one or two components of the railway system. 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 to conduct group design projects. They were required to demonstrate a prototype railway system with ATC at the end (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 the given scenario. They need to study the design criteria, such as the traveling time, train types, and the rules of defining ticket prices. There are two major parts in this project: the first part is to design the ATC system and the second is to implement it.

The required ATC scenario is a simple loop railway system with three stations and two trains. As managers of the railway, each team has to first decide the ticket price for each origin-destination pair (OD pair), number of types of trains to operate and the stopping pattern of each train in order to maximize the total revenue. The conceptual model of this railway can be seen as Figure 4. The characteristics of two types of trains, fast train (A) and slow train (B), are shown as Table 1. The relationship between price (P) and demand (D) (passengers per hour) of each OD are also provided as

Table 2.

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

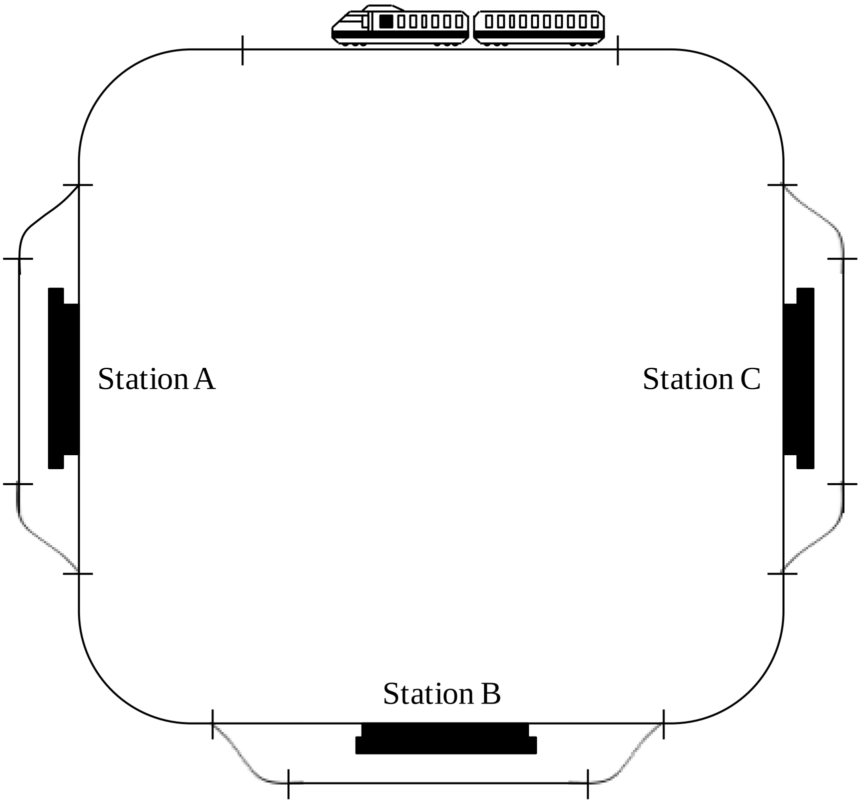

Figure 4. The scenario of the small-scaled ATC system.

Table 1. The characteristics of two type of trains

	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

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, the project teams should start implementing their designs which include the following 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 made by prototyping with papers and tapes. The track template is provided for students in the class to ensure a constant curvature; (2) Block Signaling: the adjacent trains should be controlled by the mechanism of block signaling to avoid collisions. Students may choose one of the methods mentioned in the class; (3) Passing Movements: the train is capable of performing passing movements on the stations in order to let fast train overtake slower ones. (4) Extra Design, any extra design regarding railway control is welcome to assist system implementation.

4.2. Project Materials

The materials required to conduct the final project were provided in the course. They include track and train templates, tapes, papers and so on. Students can follow these templates to develop their own trains and tracks. Similarly, we used the black tapes on the papers as the path of the designed railway. Students can implement a different type and shape of the track as well as following the scenario requirements of this project. The use of these materials is not limited and we encourage students to design an appropriate way to finish their projects.

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

(a) (b)

Figure 5. The appearance of the Railbot: (a) structure template; (b) cover.

The recommended movement strategy of the Railbots in the railway system can be seen in Figure 6. Two light sensors mounted at the front are used to detect the path of the track. By receiving the different intensity of light reflected from black tapes or white paper using light sensors, the Railbots are able to identify different conditions and do the relative reactions to keep them following the track. Learning these control mechanisms are also part of the 4-week course.

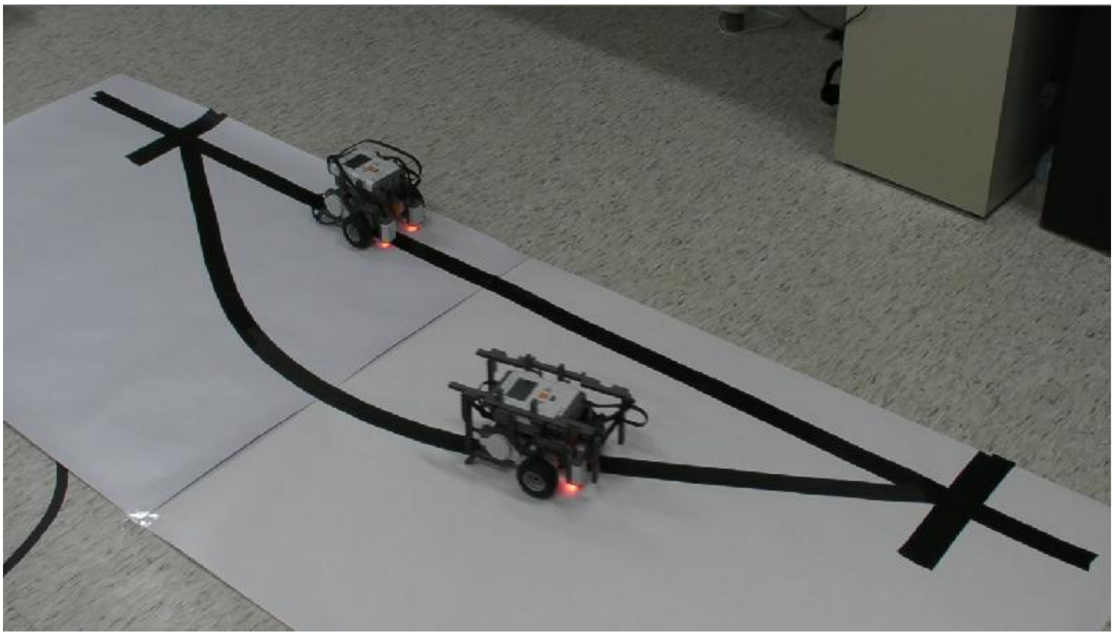

Figure 6. The movement of the Railbot

Both hardware and software designs have been evaluated in terms of effectiveness, performance and creativity. The grading is separated into two parts. One is demonstration. By following the instructions of the project assignment, students need to design and implement a railway system capable of a robust performance and adhering to a profitable train schedule. They also need to demonstrate the system and show the major functions in 10 minutes. The other is a report, in which the teacher reviews system designs in both the hardware and software components.

5. An Example of Project Implementation

After working on designing and implementing the ATC system for two weeks, students presented the details of their implementation in the reports and demonstrated their system designs in the class. One work was the most complete and efficient, and it was selected as an example of the project implementation. (Figure 7)

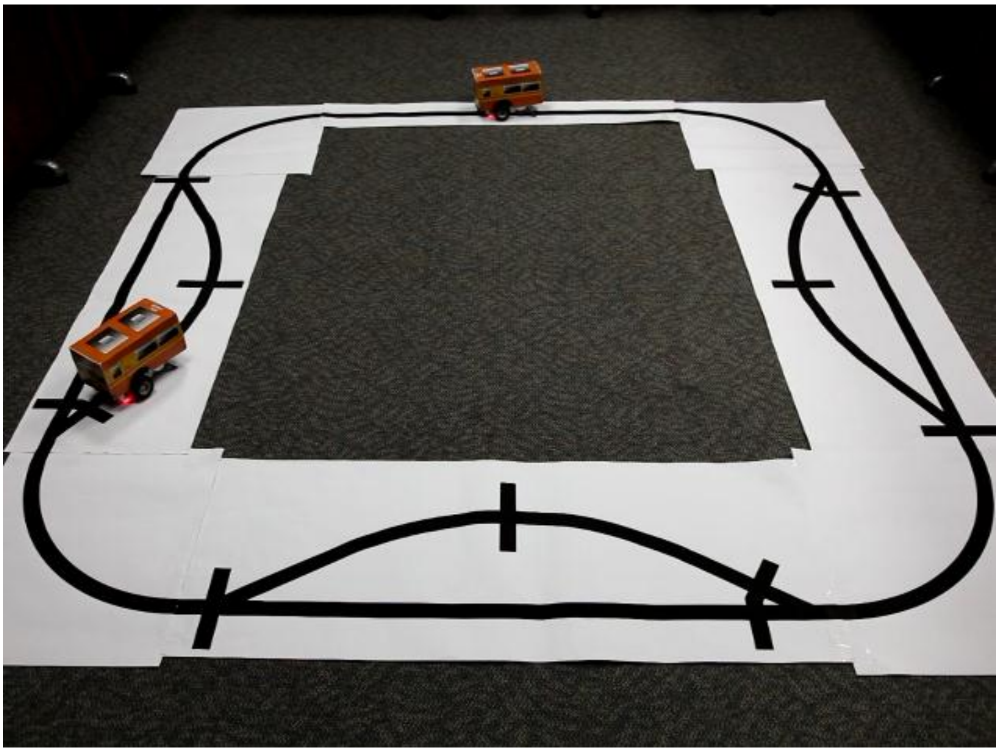

Figure 7. An Example of Project Implementation.

5.1. Ticket Strategy

The requirement of the first part of this project is 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 decide 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 assumption, operating costs can be ignored. Thus, the revenue was given by:

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

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

Since the demand of each OD pair is a linear function of price, the revenue of each OD pair becomes a quadratic function. As a result, this team obtained the maximum revenue and corresponding ticket price by obtaining the root of the derivative. The results for fast and slow trains are shown in Table 3 and Table 4, respectively.

Table 3. Result of obtaining maximum revenue for Type A train

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

Table 4. Result of obtaining maximum revenue for Type B train

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

Then, the total revenue and maximum passenger per hour of each route option were obtained, as per Table 5.

Table 5. Route options

Route	Revenue	Max. passengers / hr	Exceed Capacity?
A-B-A (fast)	11400 (AB+BA)	300	No
B-C-B (fast)	9000 (BC+CB)	270	No
C-A-C (fast)	12800 (CA+AC)	360	No
A-B-C-A (slow)	11450 (Sum. of all OD pair)	870 (A-B segment: AB+AC+CB)	Yes

Since the demand exceeded the capacity of the train, students tried to adjust the prices for some OD pairs to meet the criteria. If the price of AB is raised to 6.42 and the price of BC is raised to 5.5, the re-calculated passenger demand will not exceed 700 per hour, but the revenue would drop to only 11184, which was less than the A-B-A fast train route.

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

Table 6. 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 given instructions in the course. Each train was comprised of one NXT microcomputer, two servomotors, and two light sensors. A finished train is shown in Figure 8.

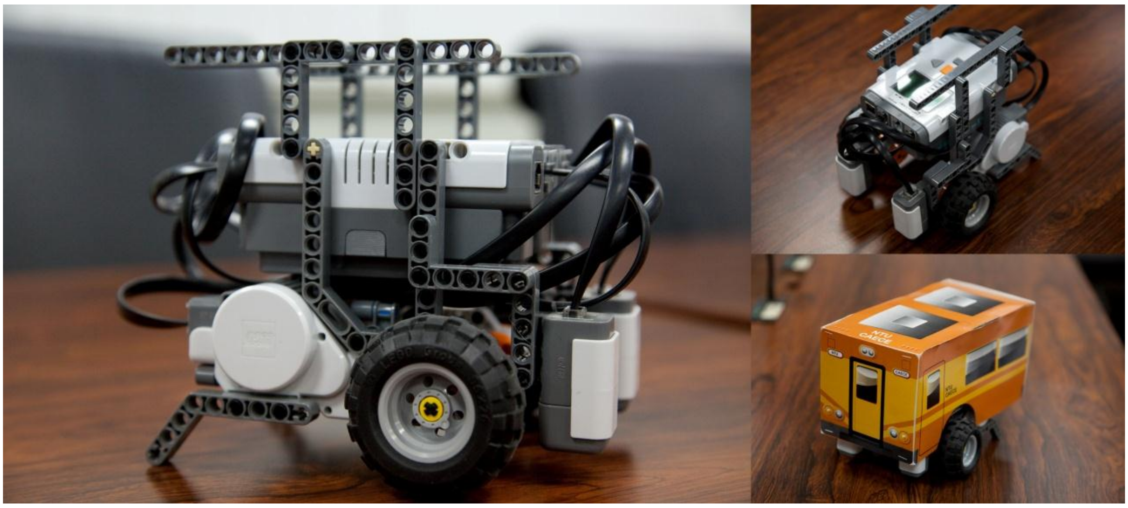

Figure 8. Train assembled using LEGO MINDSTROM NXT

- Sensors

The train was equipped with two light sensors to distinguish between the 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 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 speed.

- Microcomputer

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

5.2.2. Track design

- Rail

The rail was represented using a black line 30mm in width, which is a little smaller than the distance between the two light sensors of the train, on a white surface. In order to achieve uniform rail width and smooth curvature, this team did not use the tapes to make up the tracks, but drew them using graphic software and print them using large format printer instead. By combining the printed track components together, the whole track layout, including passing loops, signals and main (single) tracks between stations, was finally constructed. It occupied about 220mm x 200mm space. Figure 9 illustrates the track layout and location of stations.

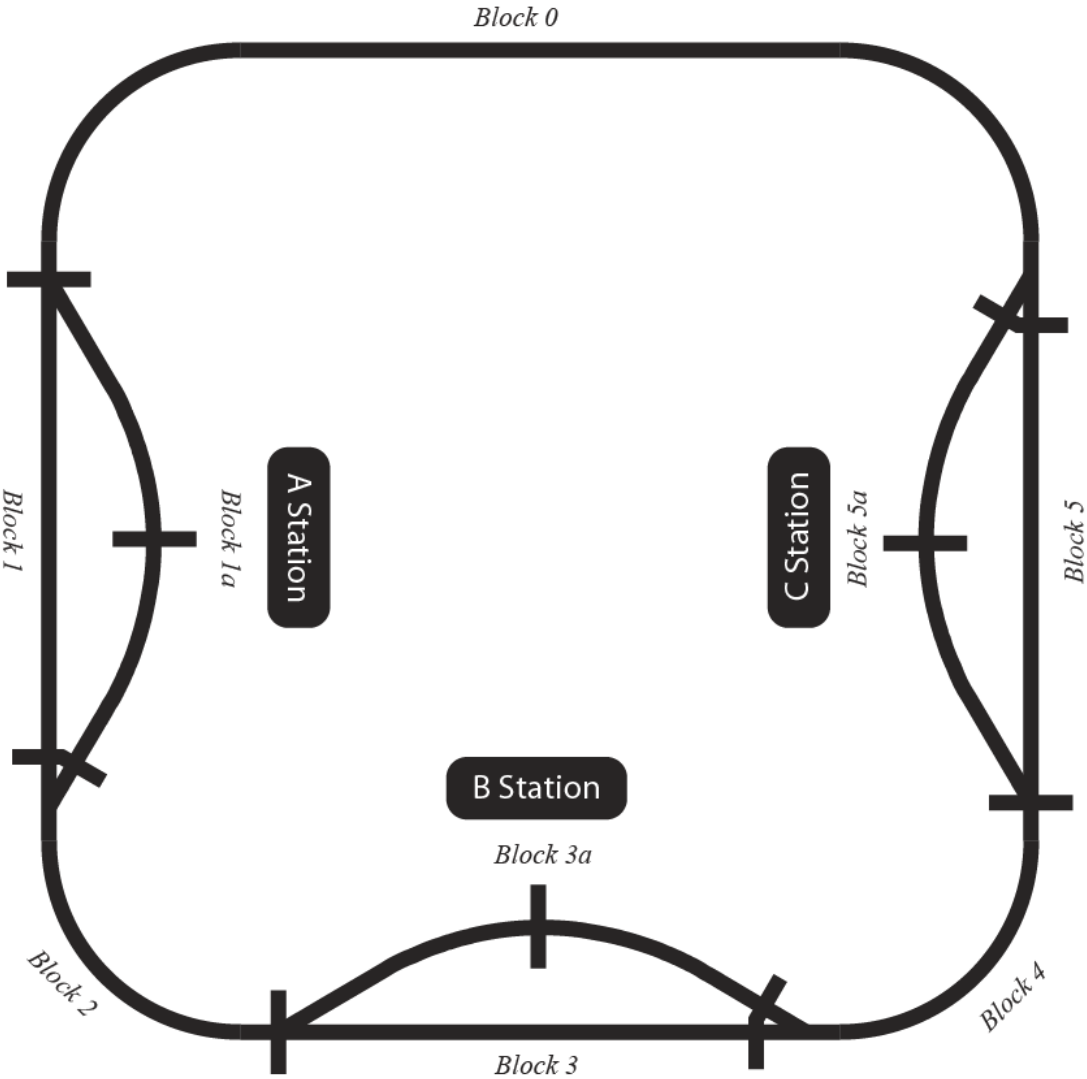

Figure 9.Track layout and location of stations

- Passing loop

The passing loops used in this project permit a train to overtake another if necessary. The design of the passing loop by this team has turnouts at either end, while the station is located on the loop line. Therefore, trains expected to pass through the station can run on the main line at full speed, rather than reduce 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 designed by this team was a left-hand lateral turnout, whose diverging track went to the left. A train that was going to stop at the station must turn left by 30 degree at this switch, and then rejoin to the main line at the other end, which is a right-hand lateral turnout. On the other hand, a train which was going to pass through the station must go via the straight route at the switch. The length of the passing loops model is about 112mm.

- Signal

This team used short lines, which are perpendicular to the rails, as signals in three different locations of the track system: at switch points of passing loop entrance, at stations, and at switch points of the passing loop exit. These signals not only notified trains of the arrival of turnouts or stations, but were also used as block signals that governed trains entering the blocks. Furthermore, the short lines were tested to ensure that they were thick enough to be discovered by the light sensors set on the running train.

5.3. Software design

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

- Initializing

Before the trains are allowed to move, several initial parameters need 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 interval of polling sensors' data when following the track. As some behaviors of a train, such as speed and stopping pattern, were parameterized, this team used the program to realize the train strategy not only using the team's solution – two fast trains, but also using other different strategies and stop patterns.

- Track Guidance

Track guidance was accomplished by repeatedly monitoring the readings polled from the sensors. First of all, if both of the light sensors are located in light areas, the computer will control the train to go straight ahead. Alternatively, if only one of the light sensors is located in a light area, the computer will control the train to revise its direction by turning left or right in order to follow the track. Finally, if both of the light sensors are located in dark areas, which indicate that a signal is encountered, the computer will decide what the next move of a train is according to its current location and blocking status of the track ahead. Nevertheless, trains moved in a fixed pattern near the signals and switches, instead of following the track.

- Blocking

The blocking behavior of this team's design was performed by keeping the location and state of each train, rather than keeping the blocking status of every track segment. It is easier to design the dataflow in such a two-train scenario. When a train encountered a signal, the computer looked at the locations and states of both trains to determine which track it was allowed to enter.

- Passing

Since all trains should run in the same direction and this team designed only one platform on a diverging track, passing was possible only on the station when one train that was not planned to stop, overtook another train that was stopping. In order to prevent the two trains from colliding near the switch, both tracks were blocked until any train left the passing loop. The only exception was that the passing line was unblocked when a train was waiting at the station. Figure 10 shows the part of the program that includes the passing behavior.

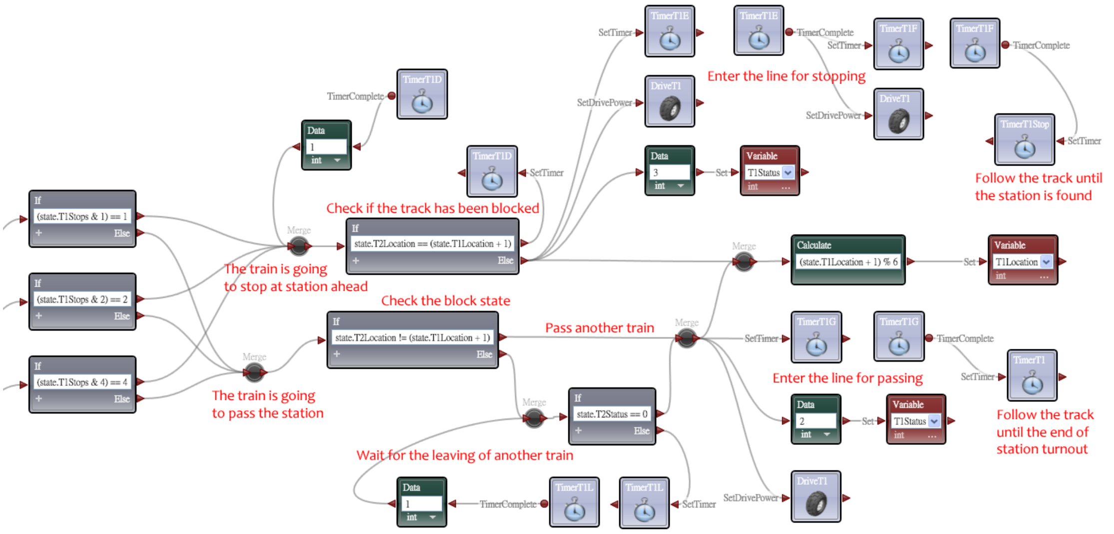

Figure 10. Dataflow of a train when entering a block that has a station

5.4. Problems Encountered

This team encountered some problems when designing the hardware and implementing the ATC system. Those problems took them quite a while to identify the sources and fix them. However, they sometimes had to change their design instead because the problems found during the implementation of the system were due to the limitation of the hardware or software. These problems are discussed in the following sections.

5.4.1. Robot Control

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

First, the reading of a light sensor was influenced heavily by environmental lighting conditions. This team had to calibrate the parameters in their application each time when lighting or location was changed. In addition, the method of marking rails and the material properties of the paper that made up the track background affected the stability of sensor readings, and then consequently affected the efficiency of track following.

Second, this team used a serial interface via a Bluetooth wireless connection to link the robot microcomputer and management computer for remote control in this project. The brightness data acquired by two light sensors was transmitted to the NXT controller. The controller then used the established serial interface to send to the control application to process. After completing the process, the application issued corresponding commands back to the controller and sensors using the same serial interface. The performance of this kind of real-time control, especially for continuous track guidance, depended largely on the efficiency of transmitting data through the wireless connection if the processor of the robot microcomputer and performance of the management computer running control application were good enough. However, this team found that a notable latency existed in this transmitting process and actually affected the performance of track following. Fine tuning the polling frequency of the light sensors could reduce the problem, but this limitation must still be considered in the design of the application to avoid the train running unexpectedly.

Finally, this team found that the servomotors in the LEGO MINDSTORMS NXT package delivered mechanical power adequate enough to drive the robot and provided good control of the movement during the implementation. However, students needed to drive the train at low speed in this project because the motors would not immediately respond to the reading changes of the sensors, as referred to in the previous section. Besides, the servomotors could not provide constant speed in such a low speed situation; thus, it was difficult to precisely control the train movement. Adding some gear wheels to alter the gear ratio might be a solution, but that would need additional space for those parts and it would not be feasible to keep the train in its original size.

5.4.2. Software Development

Before the final project, all students had learned how to control a train assembled using a robot kit to practice each basic behavior, such as simple track guidance, blocking or passing. The VPL programs for implementing such behaviors were not complex. In the final project, however, to control the movements of two trains simultaneously was actually a challenge. Several problems in developing the program were encountered by this team.

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

Second, this team found that programmers should carefully use timer services in the VPL to prevent timers from interfering with each other. To precisely control the movement of the trains, the timer service was vital within the program of this team. For example, the students used timers to periodically read data from sensors or to control the schedule of stopping at stations. They also used timers to control the train to move in a specific pattern when encountering a signal. In order to control two trains to perform different behaviors simultaneously, multiple timer services were used at the same time. However, the students found that a running timer occasionally interfered with another when executing the VPL program if students used a request of the timer service called 'Wait' to wait for a certain interval; which resulted in its temporary unavailability for the other timers. Therefore, the corresponding dataflow sequences of the other timers did not run on expected time, causing irregular movement of the trains. To avoid this problem, this team finally had to create more timer instances and wait for the notification of completion from each timer to achieve the original purpose. Nevertheless, this alternative method needs more computer resources, and it was hard to modularize the program by integrating similar actions into custom activities.

Last of all, the students thought that the current version of Microsoft VPL is not suitable for developing a complex and large-scale program. This 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 the robot control application using VPL, students found that the user interface of the development environment was slow in response when developing the program for this project. Moreover, delayed responses of activities within the program often caused abnormal moving behaviors of trains when running on a computer with mediocre speed. The complexity or the size of the program was the reason for the poor performance. Besides, it is inconvenient for programmers to manipulate a large amount of structural data due to the lack of such functions. In short, robotics programmers should consider implementing their complex logic using C# or VB.NET with MSRDS. Alternatively, other platform, such as LabVIEW, for developing robotics applications, can be taken into account.

5.4.3. Complexity

In this project, the students realized the complexity of developing a robot system capable of performing complex behaviors. The uncertainty of the robot response was particularly significant when controlling multiple robot trains simultaneously in a system. For example, it was hard to reproduce a movement identical to a previous one, even though the parameters for controlling the robots were unchanged. Besides, 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 a difficult job to test and debug such a complex system and students needed to deal with this uncertainty and make their system as reliable as possible to accomplish their course mission.

6. Results

6.1. Demonstration Performance

From the project presentations, we found that the scenario designs of all six teams are identical, but the implementation styles are quite different. Four of the six teams successfully deliver a stable system. Two of them suffered from the integration of hardware and software so can only demo part of the system. In this section, the integration problems indicated by students are listed as follows:

- Overall, each team implemented a very different type of railway system. Not only the design of the track but also the control mechanism they used are different to each other. Although they almost figure out the same answer for optimizing the given scenario, the implemented system shows quite a bit of variation in the final results.
- During the presentation, many teams encountered unexpected situations. For example, the Railbots may go out off-track, or the sensors may miscount the block, causing system instability. Most teams have built up check mechanisms, such as voice warning or counting dialogs to figure out what has occurred. It helped them think and explain the reasons behind the unexpected situations.
- Students indicated that discovering the problems behind their system is a tough issue in this project. They realized the railway system is complex and any tiny problems, such as inaccuracy of the Railbots' speed, sensor’s communication frequency and so on, can cause the whole system to fail in a minute.

- Two of the teams who suffered integration problems indicated that they incorrectly estimated the time required to implement their system. The time spent programming the algorithms and modifying them prior to obtaining stable results could be much longer than originally predicted.

- About the project implementation, students indicated that finding an optimized solution in order to gain maximum total revenue for the assigned scenario is relatively easier than its actual implementation. They are forced to consider the hardware limits and errors, and they really found some solutions to avoid the system crashing at the implementation stage.

6.2. Questionnaire

At the end of the course, students are required to fill in a questionnaire to evaluate this course. Their feedback can be seen in **Error! Reference source not found.**. In general, students responded positively to every term of the teaching method, especially the increasing of students' hands-on ability. 75% students strongly agree the course assists the development of their implementation skills. When compared to traditional teaching methods, 87% students agree or strongly agree that the method of integrating the robot kits is better than traditional methods. 62% thought the course is strongly helpful for them to think like engineers and makes them willing to take a similar course in the future. Unlike the other questions which recorded no negative responses, almost 46% thought it was difficult to finish the assignments of this course.

Table 7. Students Feedbacks on the course.

Questions	Strongly Agree (%)	Agree (%)	Neither Agree or 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 to 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. Project effectively enhances the understanding and implementation ability of all the 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 is helpful for me to develop my hands-on ability.	75	25	0	0	0
8. This course is helpful for me to 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

6.3. Lesson Learnt

According to the results of the project performance and feedback, it was evident that the robot kits are a very effective tool for educating future railway engineers on railway signaling systems and control. A list of lessons learnt is presented as follows.

- *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 theorems taught in the class. From the results of the presentation, students can easily describe the problems among the complex railway system due to the implementation and testing time spent on their works.
- *Consideration of Uncertainty*: From the hands-on project, students can notice the complexity of the railway system and uncertainty between theorems and practical situations. It can be observed from the integration problems among the projects teams, although most of them have no problem solving the design question theoretically on the given scenario.
- *Practice Opportunities*: The course helps students have the opportunities to examine their design and at the same time figure out the practical problems for building further error handling mechanisms. This will be highly beneficial when they face these tasks in practice.
- *Consideration of Integration Issues*: From observing the results of each team, the integration issues and difficulties between hardware and software when developing a railway system have been brought up by all students. It will become a foundational concept when the students need to design or implement relative works in practice.

- *Disadvantage of the Course*: Too many project materials and not enough instructions for may be difficult for students to handle. It should be improved by providing better project description and designing appropriate scenarios next time.

7. Lesson Learnt

According to the results of the project performance and feedback, it was evident that the robot kits are a very effective tool for educating future railway engineers on railway signaling systems and control. A list of lessons learnt is presented as follows.

- *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 theorems taught in the class. From the results of the presentation, students can easily describe the problems among the complex railway system due to the implementation and testing time spent on their works.

- *Consideration of Uncertainty*: From the hands-on project, students can notice the complexity of the railway system and uncertainty between theorems and practical situations. It can be observed from the integration problems among the projects teams, although most of them have no problem solving the design question theoretically on the given scenario.

- *Practice Opportunities*: The course helps students have the opportunities to examine their design and at the same time figure out the practical problems for building further error handling mechanisms. This will be highly beneficial when they face these tasks in practice.

- *Consideration of Integration Issues*: From observing the results of each team, the integration issues and difficulties between hardware and software when developing a railway system have been brought up by all students. It will become a foundational concept when the students need to design or implement relative works in practice.
- *Disadvantage of the Course*: Too many project materials and not enough instructions for may be difficult for students to handle. It should be improved by providing better project description and designing appropriate scenarios next time.

8. Conclusions and Future Works

From the performance of the final project and feedbacks through the questionnaires, we found that students are very positive towards their learning experiences. We also found that using the robot kits is particularly helpful for training a railway engineer. Because students can realize and modify their designs by using programmable robot kits, they can experience the entire design processes of a railway system.

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 transfer to enhance those 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*, pp. 18-23, Transportation Research Board of the National Academies, Washington, DC.

Blank, D., Kumar, D., Marshall, J., and Meeden, L. (2007). "Advanced robotics projects for undergraduate students," *AAAI Spring Symposium - Technical Report*, SS-07-09, pp. 10-15.

Bruyninckx, H. (2001). "Open robot control software: The OROCOS project," Proceedings of IEEE International Conference on Robotics and Automation, 3, pp. 2523-2528.

Bruyninckx, H. (2007). " Microsoft Robotics Studio: Expected impact, Challenges & Alternatives," Panel presentation at the IEEE International Conference on Robotics and Automation.

Burhans, D. T., Meyer, R. M., VanVerth, P., Puehn, D., Steck, V., Wiejaczka, J. P. (2006). "Introductory Computer Science with Robots," Proceedings of AAAI '06, AAAI Press.

Byoung, W. C., Lee, S. H., and Yi, S. Y. (2009). "Internet Coordinated Pet Robot Simulator Based on MSRDS," ICCAS-SICE 2009 - ICROS-SICE International Joint Conference 2009, Fukuoka, 18 August 2009.

Cliburn, D. C. (2006). " Experiences with LEGO Mindstorms throughout the Undergraduate Computer Science Curriculum," Proceedings of ASEE/IEEE Frontiers in Education Conference 2006, T2F-1 - T2F-6 IEEE Computer Society.

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

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).

FRA (2009). "Vision for High-Speed Rail in America," Federal Railway Administration (FRA), US Department of Transportation, Washington, D.C..

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

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

Markou, G. and Refanidis, I. (2009). "MSRS: Critique on its usability via a path planning algorithm implementation," *IFIP International Federation for Information Processing*, 296, pp.311-320.

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

Microsoft Corporation (2008). Microsoft CCR and DSS Toolkit 2008, Retrieved Jan. 13, 2009, from http://go.microsoft.com/fwlink/?LinkId=130995.

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

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

Wolfe, D., Gossett, K., Hanlon, P.D., and Carver, C.A. (2003) " Active Learning Using Mechantronics in a Freshman Information Technology Course," ASEE/IEEE Frontiers in Education Conference, pp. 24-28.

Workman, K. and Elzer, S. (2009). "Utilizing Microsoft robotics studio in undergraduate robotics," *Journal of Computing Sciences in Colleges*, 24 (3), pp. 65-71.

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