

Development of a Microcontroller-Based Control System with a Hardware-in-the-Loop (HIL) Method for Control Education using MATLAB/Simulink/xPC Target*

WAHYUDI, MOMOH J.E. SALAMI and ABDULGANI ALBAGUL

Department of Mechatronics Engineering, International Islamic University Malaysia, Jalan Gombak, 53100, Kuala Lumpur, Malaysia. E-mail: wahyudi@iiu.edu.my

This paper discusses the development of a microcontroller-based control system with hardware-in-the-loop (HIL) for enhancing the teaching of control engineering. This system consists of both hardware and software. The software tools MATLAB/Simulink/xPC Target by MathWorks, Inc and a C++ compiler are used to simulate the physical system (plant) to be controlled, while the hardware include a microcontroller as a controller and interfacing circuits to allow communication between the simulated plant and the real controller. This proposed system is inexpensive and allows students to carry out extensive experimental investigations, as well as the design, implementation, performance evaluation and comparative studies of controllers. A case study of the controller design and implementation for an active suspension system is presented to illustrate the application of the proposed system.

INTRODUCTION

STUDENTS ENCOUNTERING control systems for the first time often find the subject too abstract and theoretical in nature; that is, too many mathematical equations as well as block diagrams. It is well known that the control laboratory experiment is an important way of exposing students to the practical applications of control theory so as to overcome the above-mentioned difficulty. Hence real hands-on experiments or design problems are an alternative way of augmenting the conventional way of teaching control theory, as they can be related to real engineering applications, such as modeling, controller design, and implementation [1, 2].

A typical laboratory for control education needs three main pieces of equipment: analysis and simulation software, a target controller and an experimental set-up. The software may include controller design/analysis tools, real-time code generators and a compiler. Moreover, the target hardware for controller implementation may be based on a digital signal processor (DSP) or other low-cost alternatives such as PC-based controllers or microcontrollers, while the experimental set-up is the hardware for capturing the real life of industrial control systems. However, achieving this requires complex installation and/or machinery, which are both costly and inflexible, as well as in some cases potentially dangerous to students.

On the other hand, the use of simulation only will not provide sufficient illustration of the real physical application of using control theory in solving engineering problems.

The above shortcomings and problems can be overcome by adopting the concept of the hardware-in-the-loop (HIL) method [3, 4] as an experimental part of control engineering; it has been developed at the Department of Mechatronics Engineering, International Islamic University Malaysia. The HIL method uses a simulation model of the controlled object (plant) and a microcontroller as the real target hardware for controller implementation. Instead of using a costly experimental set-up, the simulation model in a computer provides all the plant signals in real time which are next converted into analog signals by a digital-to-analog converter (DAC). The control signal produced by the controller is processed by an analog-to-digital converter (ADC) and then sent to the simulated model. By using the HIL approach, students can learn how to design a controller and then implement it physically using a microcontroller. Furthermore, the HIL method allows students to use different controlled objects, such as the active suspension system, DC motor control system, temperature control system and gantry crane system, as well as selecting a controller such as the phase lag/lead compensator, lag-lead compensator, state feedback controller and fuzzy logic controller without any additional cost.

The paper is organized as follows. In the first section the concept of the hardware-in-the-loop system is explained. Next, the proposed

* Accepted 3 July 2005.

microcontroller-based control system using the HIL method is described. Finally, the application of the proposed system to realize a process of modeling, controller design and implementation of an active suspension system is discussed.

CONCEPT OF THE HARDWARE-IN-THE-LOOP SYSTEM

Laboratory sessions are usually included in the control system course to offer students an environment whereby they can design control systems, starting from desired performance specification, right through to plant modeling, controller design, controller implementation and performance analysis, and controller re-design, if necessary. The lab sessions attempt to reflect the real industrial applications of the control system. This classical control experiment requires direct connection with the plant, as shown in Fig. 1. A real plant, or its laboratory model, delivers physical control output and generates the plant response (output) measured by the sensor. Various plants have been used in the control laboratory, such as the inverted pendulum, a helicopter model, a magnetic bearing and simple robots [5]. However, more complex models of industrial plants are too expensive for control education and, in some cases, they are potentially dangerous for the students. In addition, the classical control experiment often lacks flexibility. Furthermore, it would be costly and time intensive if the lecturer wanted to conduct experiments with different plants, especially as the lab work is expected to be completed within a stipulated time.

The problem of cost and experimental flexibility can be overcome by adopting the concept of hardware-in-the-loop simulation (HILS). The basic principle of the HILS is that a specific part of the real system (hardware) is replaced by a real-time simulation model. There are many possibilities for replacing the real system (hardware) of a control system with a simulated model for different applications [6], as shown in Fig. 2. This figure shows that the simulated model can be used to replace controller hardware, real plant or sensors. HILS based on real plant replacement is a useful method for dealing with the problems that arise in many control experiments. The HILS method offers several advantages compared with the classical control experiment. Control system experiments based on the HILS method are less expensive

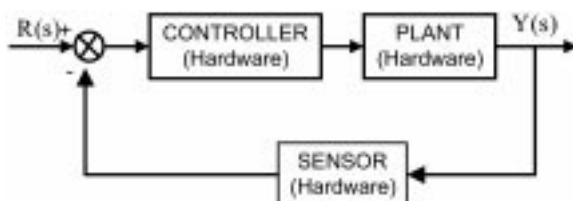


Fig. 1. Classical control experiment.

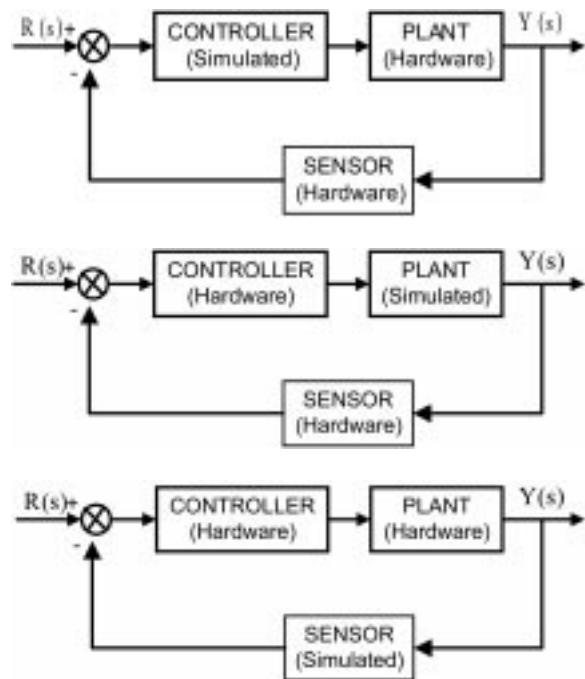


Fig. 2. Hardware-in-the-loop simulation (HIL): (a) simulated-controller HILS; (b) simulated-plant HILS; (c) simulated-sensor HILS.

and more flexible. The plant in the HILS method can be changed easily and quickly, since one just changes the transfer function of the plant.

DEVELOPMENT OF A MICROCONTROLLER-BASED CONTROL SYSTEM WITH HILS

The proposed microcontroller-based control system with HILS is shown in Fig. 3 and its schematic representation is depicted in Fig. 4. As shown in both diagrams, the proposed system consists of a microcontroller, two personal computers and interfacing circuits. The real system (hardware) of this system is a controller, which is implemented using a microcontroller, while the controlled object is replaced by a real-time simulation model, which is located in the target computer. Another computer, the host computer, is needed for generating both the simulation model and microcontroller code. This is the arrangement used, since the main purpose of the development of the proposed system is to demonstrate to students the real-time implementation of the controller design, as well as to evaluate its performance.

In the proposed system, the sensor feedback from the actual plant output is simulated in the computer-based plant model, and the output signal is then sent to the controller. The controller will receive the output signal of the plant and then generate a control output based on the controller algorithm. The control output is processed and then used to control the plant. In order to interface between controller and plant, two analog-to-digital converters (ADC) and two digital-to-analog

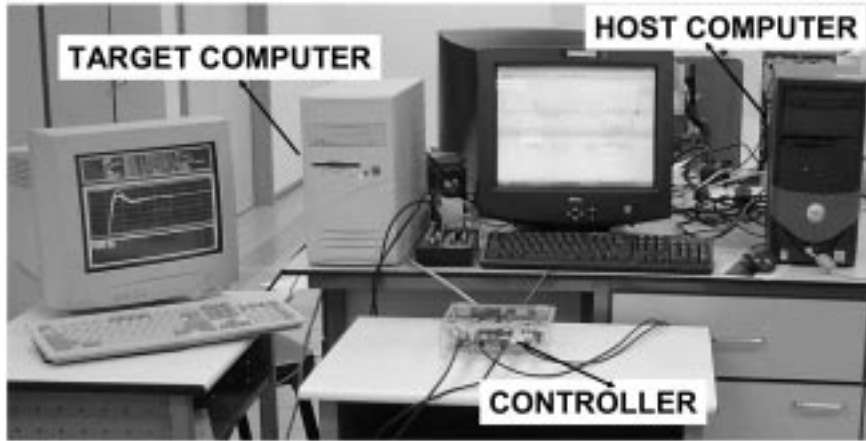


Fig. 3. Proposed HILS.

converters (DAC) are adopted. The interfacing circuits are used to transfer and convert the signals from/to a simulated plant in the target computer to/from the controller in the microcontroller.

Hardware configuration

The hardware elements of the proposed systems are a microcontroller and interfacing circuits, as shown in Fig. 5. The microcontroller used for controller implementation is the BASIC Stamp II (BS2), from Parallax Inc., due to its availability in the Mechatronics Engineering Department and its simplicity in programming. Table 1 shows the specification of the BASIC Stamp II. It is a hybrid

microcontroller designed to be programmed in a version of the BASIC programming language called PBASIC [7]. The BASIC Stamp Editor, which is provided free by Parallax Inc., is an easy application software that is used to write, edit, and download the PBASIC programs for the BASIC Stamp.

The other hardware elements are interfacing circuits. As discussed above, two ADCs and DACs are required for developing the proposed system. A unit of the ADC and DAC is built in the data acquisition (DAQ) card of PCI-6024-E from National Instruments [8]. This DAQ card is installed in the target computer to receive the control output from the microcontroller and to

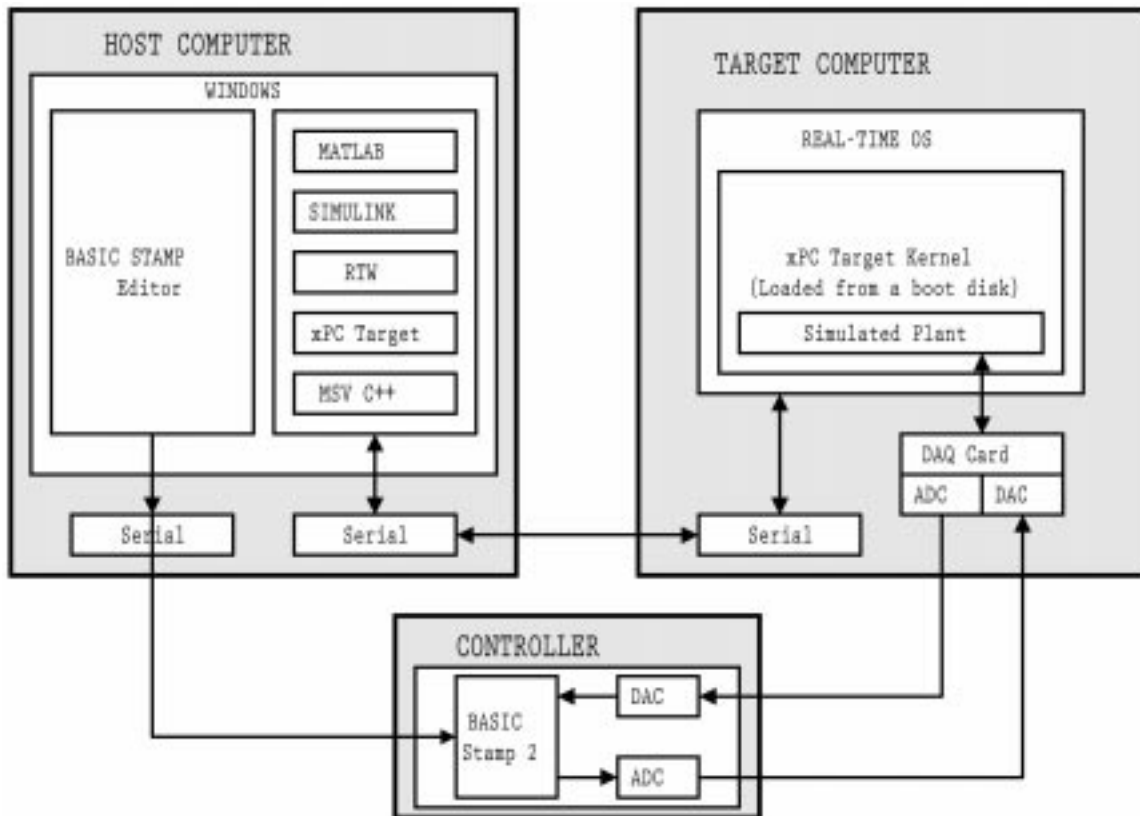


Fig. 4. Architecture of the proposed HILS.

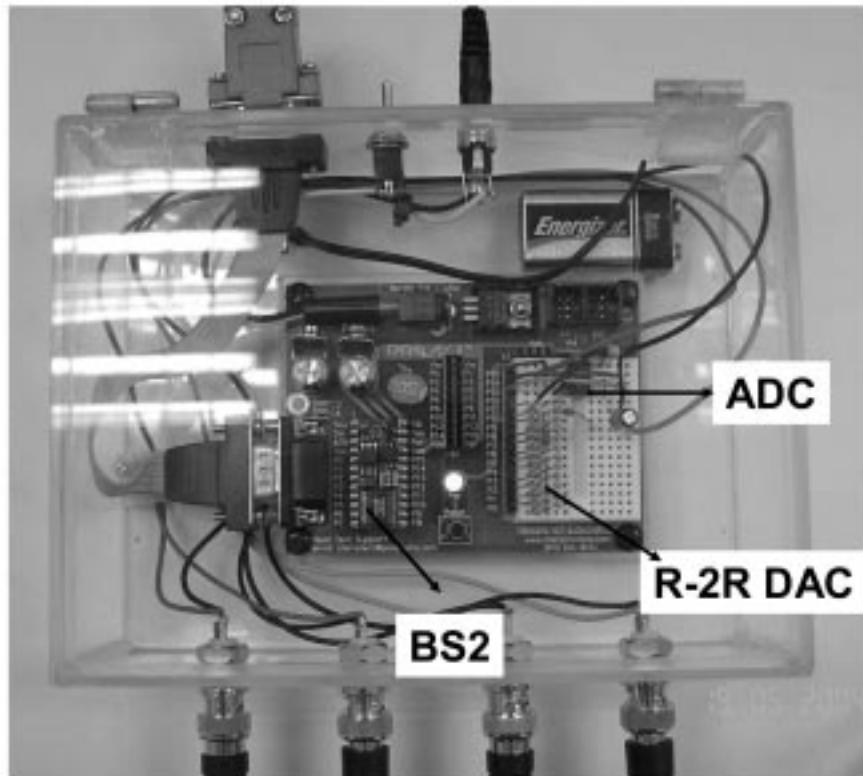


Fig. 5. Microcontroller-based hardware element.

send the simulated actual plant output to the microcontroller. The other ADC and DAC cards are directly connected to the microcontroller, as shown in Fig. 5. The ADC connected directly to the microcontroller, as shown in Fig. 6, is built using a 12-bit LTC1298 ADC chip manufactured by Linear Technology. Meanwhile, the DAC circuit is connected directly to the microcontroller and used to send the control output to the simulated plant; its design is based on the R-2R ladder network DAC, as shown in Fig. 7.

Table 1. BASIC Stamp II specifications

BASIC Stamp 2 IC Item	Specification
Package	24—Pin DIP
Package Size (L W H)	1.2" × 0.6" × 0.4"
Environment	0–70°C (32–158°F)
Microcontroller	Microchip PIC16C57c
Processor Speed	20 MHz
Program Execution Speed	~4000 instruction/sec
RAM Size	32 Bytes (6 I/O, 26 Variables)
Scratch Pad RAM	N/A
EEPROM (Program) Size	2K Bytes, ~500 instructions
Number of I/O Pin	16 + 2 Dedicated Serial
Voltage Requirements	5–15 VDC
Current Draw @ 5V	8 mA Run/100 μ A Sleep
Source/Sink Current per I/O	20 mA/25 mA
Source/Sink Current per Unit	40 mA/50 mA per 8 I/O pins
PBASIC Commands	36
PC Programming Interface	Serial Port (9600 baud)
DOS Text Editor	STAMP2E.EXE
Window Text Editor	Stampw.exe (v1.04 and up)

Software configuration

The MathWorks MATLAB/Simulink tool is used to present a simulated model in the computer, as most engineering students are introduced to MATLAB and its various toolboxes at an early stage in their university education. Furthermore, Simulink, together with Real-Time Workshop (RTW) and xPC Target, makes it possible to obtain the HILS model from the Simulink model [9–11]. The RTW environment provides an easy and quick way to generate code for discrete-time, continuous-time (fixed-step), and hybrid systems. The MATLAB-based RTW can convert a Simulink model to C code. However, the use of the RTW still requires the development of customized interface programs for real-time implementation, as well as correct communication with multifunction I/O boards. To overcome this problem, xPC Target is included in the software configuration. Basically, xPC Target is a solution for prototyping, testing, and deploying real-time systems using standard PC hardware. It is an environment that uses a target PC that is separate from a host PC for running real-time applications. The overall host-target control system architecture is shown in Fig. 3.

Combining RTW and xPC Target makes it unnecessary for students to write a low-level programming language to realize a controller and/or access other components such as DAQ boards. In particular, the xPC Target software supports and provides built-in drivers for many standard DAQ cards [11], including National

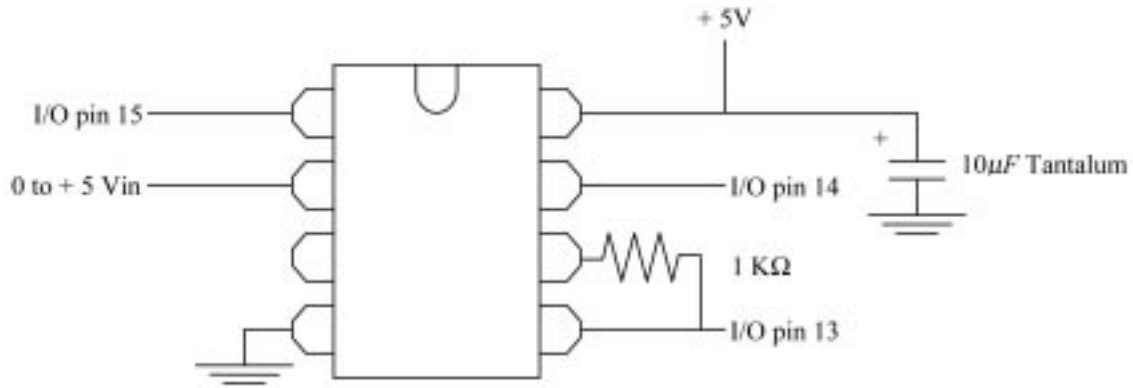


Fig. 6. ADC circuit.

Instruments' PCI-6024E DAQ card, which was used in the proposed system. This combination of devices provides a unique and complete environment for prototyping, testing and deploying a

simulated plant model or real-time digital controller using standard hardware components. The Windows operating system and other required software are run in the host computer. The plant model is developed in Simulink using its block diagrams, and is then built so as to generate the C code. Once this is compiled, a real-time executable code is finally generated and downloaded to the target computer.

The target computer is another personal computer, which is booted using an xPC boot floppy disk that loads the xPC Target real-time kernel [11]. The generated real-time executable code is subsequently downloaded to the target computer via selected communication protocol without requiring the students or the lecturer to write any low-level code. The connection between the host computer and target computer is accomplished either through serial (RS-232) or network (TCP/IP) communications. The communication interface is defined during the xPC set-up process in MATLAB, since the communication protocol definition is required to create the xPC boot floppy disk for the target computer. Serial communications are used in the proposed system, as they are inexpensive, easy to install and only require a cable for connecting the serial ports of the host computer and target computer.

Finally, the students initiate the communication of the simulated plant model, start the real-time code and run the proposed system using pull-down menus on the host computer. The Simulink environment allows students to perform experiments interactively, and in real time, by modifying any of the block parameters in the Simulink model, without the need to rebuild and re-download the model to the target computer.

Furthermore, the host computer is not only used for building the Simulink model of the simulated plant and transferring the generated code of the model to the target computer but is also used for generating the code of the controller algorithm and then downloading it to the microcontroller. For this purpose, the BASIC Stamp Editor, which is provided free by Parallax Inc., is used to write, edit, and download the PBASIC programs for the BASIC Stamp II microcontroller. The PBASIC

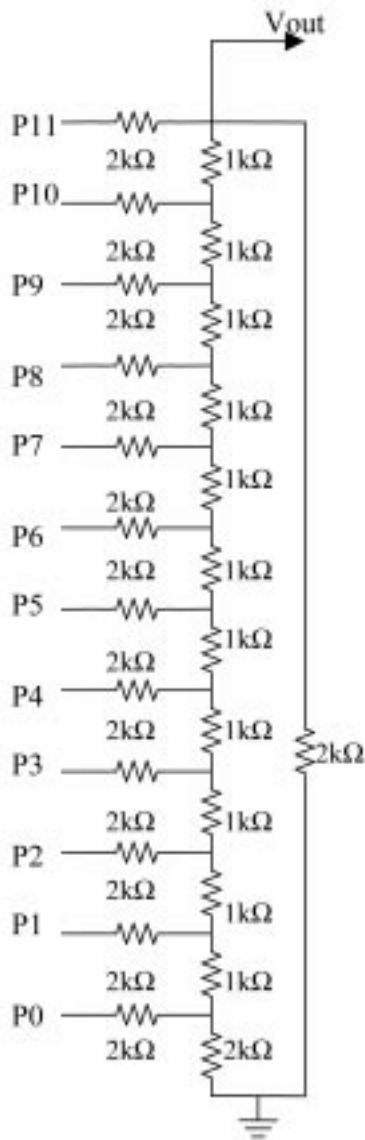


Fig. 7. The R-2R ladder network DAC circuit.

code is converted to the symbolic format and then transmitted through serial cable and stored in the EEPROM of the BASIC Stamp II [7]. Consequently, it is necessary to connect a serial cable between BASIC Stamp II and the computer's serial communication port (RS 232).

A summary of the software tools used in the proposed system and their corresponding versions are as follows:

1. MATLAB 6.5, Simulink 5.0, RTW Workshop 5.0, xPC Target 2.0 and Microsoft Visual (MSV) C++ 6.0 are used for generating the simulated plant model, as well as providing the interface between the simulation and hardware part of the proposed HILS.
2. BASIC Stamp Editor 2.0 is used to write, edit and download the control algorithm in PBASIC language to the BASIC Stamp II microcontroller.

CASE STUDY: ACTIVE SUSPENSION SYSTEM

In order for students to gain experience in controller design and implementation, an active suspension system was selected as a case study and this is discussed here. The active suspension system is selected because the automotive industry is seriously considering implementation of active suspension systems on automobiles [12]. The suspension system refers to the use of front and rear springs and damper to suspend a vehicle's frame, body, engine and power train above the wheels. The main functions of the suspension system are to support the vehicle weight, to isolate the vehicle body from road disturbances, and to maintain the traction force between the tyres and the road surface. Moreover, the suspension system provides the necessary comfort and safety for passengers and driver.

Most vehicles currently use a passive suspension system. In passive suspension design, comfort and road handling are difficult to achieve because the design parameters are fixed. Therefore, the performance of the passive suspension is only good under limited conditions. To overcome the problems of the passive suspension system, an active suspension system has been developed [13]. The goal of the active suspension is to improve comfort and road handling. Comfort and safety improvement using the active suspension system requires a careful control mechanism of the suspension system so that its parameters can adapt to the driving conditions.

Plant modeling

An active suspension system can be realized on the proposed HILS method if the model of the suspension system is known. The relatively heavy assemblies constitute what is known as 'sprung' weight. 'Unsprung' weight, on the other hand, includes wheels and tyres, brake assemblies and other structural members which are not supported

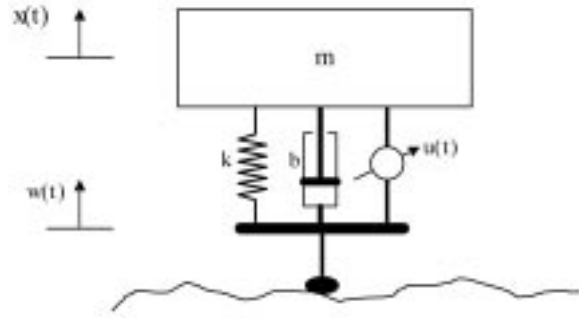


Fig. 8. Quarter suspension system and its simple model.

by the springs. The current trend is to have four independent suspensions on a single car and, therefore, a quarter model system is reasonable enough to represent the actual suspension system [14].

Figure 8 shows a simple schematic diagram of a quarter model of the suspension system. In this model, three assumptions are made: unsprung mass (tyre mass) is negligible compared with sprung mass, the tyre is rigid and a hydraulic actuator is used. Applying Newton's second law, a quarter car model is expressed as:

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = b\dot{w}(t) + kw(t) + u(t), \tag{1}$$

where

- m = mass of a quarter car body (sprung mass)
- k = spring constant of suspension system
- b = damping constant of suspension system
- u(t) = hydraulic actuator force output
- x(t) = sprung mass displacement
- w(t) = road profile displacement

The actuator force u(t) is simply modeled as directly proportional to the output voltage v(t), that is u(t) = cv(t), where c represents the actuator gain. Hence Equation (1) becomes:

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = b\dot{w}(t) + kw(t) + cv(t), \tag{2}$$

Taking the Laplace of Equation (2) yields

$$X(s) = \frac{bs + k}{ms^2 + bs + k_1} W(s) + \frac{c}{ms^2 + bs + k_1} V(s) \tag{3}$$

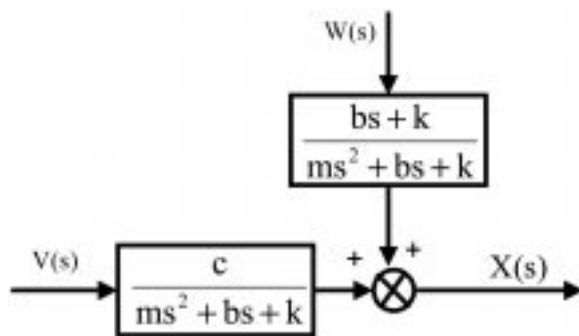


Fig. 9. Block diagram of the suspension system model.

Table 2. Simulated suspension parameters

Parameter	Symbol	Value	Unit
Sprung mass	M	253	Kg
Suspension damping	B	1500	N/m/s
Suspension stiffness	K	12000	N/m
Actuator gain	c	1	N/V

Figure 9 shows the block diagram of an active suspension system which can be easily built in Simulink. Furthermore, the simulated parameters of the suspension model are also needed. Table 2 shows the simulated suspension model parameters which have been adopted from a quarter-car test rig discussed in reference [15].

Controller design

A proportional plus derivative (PD) controller is adopted to control the active suspension system so that

$$V(s) = (K_P + K_D s)E(s) \quad (4)$$

where K_P and K_D are proportional and derivative gains respectively. The PD controller is mainly designed to reject disturbance input due to road profile displacement $w(t)$. The root locus technique [16], which is discussed earlier in the class, is used for designing the PD controller. As an example, students are required to design the PD controller to achieve the following criteria:

- Settling time of the closed-loop system is less than 1 second.
- Maximum percent overshoot is less than 10 %.

Finally, based on the root locus method, the PD controller parameters K_P and K_D are 1 and 500, respectively.

Implementation and result

The Simulink block diagram implemented for the HILS-based active suspension system is shown in Fig. 10. The Simulink block diagram is used to build the simulated model of the suspension and then downloaded to the target computer using xPC Target toolbox. Furthermore, it is necessary to download the designed PD controller to the BASIC Stamp II microcontroller. A PBASIC language program is written in the Stamp Editor environment to implement the PD controller

algorithm and then downloaded to the BASIC Stamp II microcontroller through serial communication between the host computer and the microcontroller. Once the simulated plant and PD controller are on the respective target, some experiments are performed in real time. Parameter changes may be performed from the Simulink and their effects on the HILS-based active suspension system can be monitored directly.

Furthermore, the performance of the active suspension system controlled by the PD controller is evaluated through the developed HILS-based active suspension system. A simple type of road disturbance profile is used in the experiment; namely, a 0.1 m step disturbance road profile as shown in Fig. 11. Figure 12 shows the result of the preliminary experiment when a 0.1 m step disturbance road profile is applied. The figure shows that the active suspension system results in a lower overshoot and shorter settling time than that of the passive system. Hence, from the experiment performed using the proposed HILS, one can conclude that the active suspension system performs better than the passive suspension system. Figure 12 also shows that an active suspension system is better able to recover from road disturbances than the passive suspension one. Moreover, there are slight differences between the experimental results obtained using the proposed HILS and the simulated one using Simulink, as depicted in Fig. 12. Such results allow discussion between the lecturer and students with respect to solving practical problems in the control system, such as actuator saturation, noise and the effect of sampling time in digital controller implementation. Additional experimental work is in progress to tune the controller gains, adopt a more complex model of the suspension system and apply other road disturbance profiles such as sinusoidal and square wave profiles. Future improvements will include the implementation of other controller types, such as the PID, state feedback and intelligent controllers.

DISCUSSION

The proposed HILS method offers several benefits in comparison with the classical control experiment. Firstly, the control system experiment based

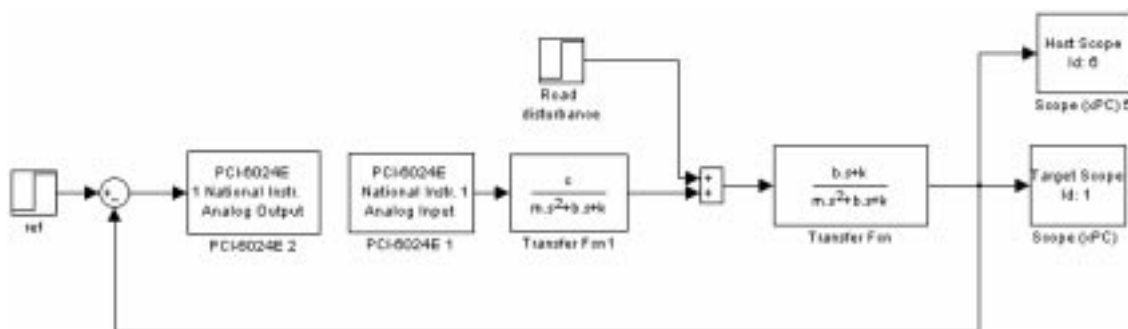


Fig. 10. Simulink model of HILS-based active suspension system.

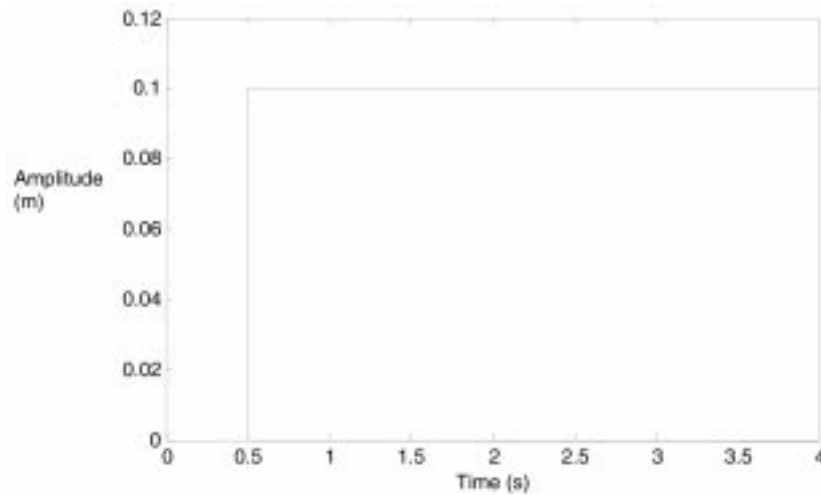


Fig. 11. The 0.1 step disturbance road profile.

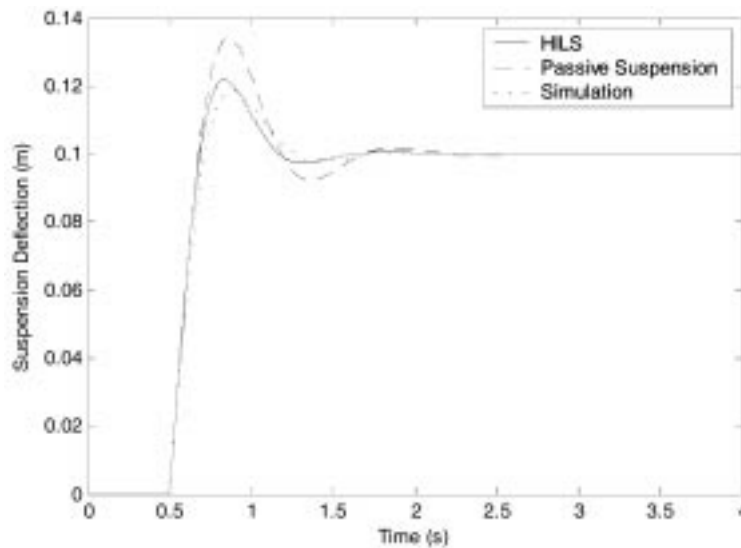


Fig. 12. Response to a 0.1m step disturbance road profile.

on the HILS method is less expensive in the development of the experimental set-up as well as in terms of maintenance. Secondly, the HILS implementation of the control system experiment offers greater flexibility and the plant in the HILS method can be changed easily and quickly, since one only needs to change the transfer function of the plant in the Simulink model. In addition, the students gain experience in controller design that closely resembles controller design in real industrial application. It is well known that the concept of HILS is becoming more popular in industry for controller design and development.

However, the application of the HILS method for the control system experiment also suffers from significant drawbacks. The use of the HILS method for control experiments reduces the real level of illustration. In the experiments, students will not be exposed to the real effects of the controller implementation, such as moving parts, noise and temperature changes, which can be observed when the real experiments are conducted.

Therefore, further improvements will also include this level of illustration by using a multimedia approach, such as plant animation and artificially generated sounds from the plant.

CONCLUSIONS

An interactive microcontroller-based control system with the HIL approach has been developed, implemented and demonstrated for educational purposes, particularly teaching control engineering. The proposed system was implemented using standard personal computers and data acquisition cards. The physical part of the proposed system is a controller which is implemented using the BASIC Stamp II microcontroller, while the controlled object is replaced by a real-time simulation model which is located in the target computer. The real-time simulation in the target computer is carried out by using the xPC Target and other software tools from MathWorks, such as

MATLAB, Simulink and a C++ compiler. As a case study, the controller design and implementation of active suspension systems have been discussed. The use of MATLAB-based software in the proposed system provides a quick experi-

mentation and performance evaluation. Furthermore, the ease of performing experiments allows students to focus on the task of understanding, developing and testing control algorithms for many specific industrial applications.

REFERENCES

1. A. Leva, A hands-on experimental laboratory for undergraduate courses in automatic control, *IEEE Transaction on Education*, **46**(2) (2003), pp. 263–272.
2. P. E. Wellstead, Teaching control with laboratory scale models, *IEEE Transaction on Education*, **33**(3) (1990), pp. 286–290.
3. I. R. Kendall and R. P. Jones, An investigation into the use of hardware-in-the-loop simulation testing for automotive electronics control, *Control Engineering Practice*, **7** (1999), pp. 1343–1356.
4. R. Iserman, R. Schaffnit and S. Sinsel, Hardware-in-the-loop simulation for the design and testing of engine control systems, *Control Engineering Practice*, **7** (1999), pp. 643–653.
5. J. C. Juang, Controller rapid prototyping and its incorporation in control education, *Proceedings of the 4th IFAC Symposium on Advances in Control Education*, Istanbul (1997), pp. 301–306.
6. D. Shetty, and R. A. Kolk, *Mechatronics System Design*, PWS Publishing (1997).
7. *BASIC Stamp Editor V 2.0 User's Guide*, Parallax Inc. (2004).
8. *6023E/6024E/6025E Families of DAQ Cards*, National Instruments, Austin, TX (2000).
9. *Using MATLAB*, Mathworks, Inc., MA (1999).
10. *Using Simulink*, Mathworks, Inc., MA (1999).
11. *xPC Target User's Guide*, Mathworks, Inc., MA (1999).
12. D. Hrovat, Optimal active suspension structure for quarter-car vehicle models, *Automatica*, **26**(5) (1990), pp. 845–860.
13. T. Aburaya, M. Kawanishi, H. Kondo and T. Hamada, Development of an electronic control system for active suspension, *Proceedings of the 29th IEEE Conference on Decision Control* (1990), pp. 2220–2225.
14. C. Kim and P. I. Ro, Reduced order modeling and parameter estimation for a quarter car suspension system, *Proceedings of the Institute of Mechanical Engineers Part D, Journal of Automotive Engineering*, **214** (2000), pp. 851–864.
15. H. Peng, R. Strathearn and A. G. Ulsoy, A novel active suspension design technique: Simulation and experimental Results, *AACC* (1997).
16. R. D. Dorf and R. H. Bishop, *Modern Control Systems*, 9th edition, Prentice-Hall, New Jersey (2001).

Wahyudi was born in Indonesia in July 1970. He received a B.Eng. and M.Sc. in Mechanical Engineering from the Institute of Technology Bandung, Indonesia, in 1994 and 1997, respectively, and a Ph.D. in Precision Machinery Systems from Tokyo Institute of Technology, Tokyo, Japan, in 2002. He is currently an Assistant Professor at the Mechatronics Engineering Department of the International Islamic University, Malaysia. His research fields are motion control, mechatronics and intelligent systems. Dr. Wahyudi is a member of IEEE.

Momoh Jimoh E. Salami received a B.S.E.E. from the University of Ife, Ile-Ife, Nigeria, in June 1977 and a Ph.D. in Electrical Engineering from the University of Calgary, Calgary, Canada, in 1985. He has been involved in teaching and research activities at university level in Nigeria, Saudi Arabia and Malaysia since completing his graduate studies. He joined IIUM in September 1996 and is currently Professor and Deputy Dean of Postgraduate and Research in the Faculty of Engineering. He has written more than 70 publications for both local and international journals and conference proceedings, as well as being one of the contributors in a recently published book entitled *The Mechatronics Handbook*, edited by Professor Bishop. His research interests include digital signal and image processing, intelligent control system design and instrumentation. Dr. Momoh is a senior member of IEEE.

Abdulgani Albagul received a B.Sc. in Electronic Engineering from the Higher Institute of Electronics in Baniwalid, Libya, in 1989, an M.Sc. in Control Engineering from the University of Bradford in 1993 and a Ph.D. in Electrical Engineering from the University of Newcastle upon Tyne in 2001. He is currently an Assistant Professor in the Department of Mechatronics Engineering, Faculty of Engineering, International Islamic University Malaysia. His research interests are control systems, system dynamics and modeling, smart sensors and instrumentation, robotics and automation. He is a member of IEEE and has frequently been published.