Communication Design of Human-Hardware-In-the-Loop simulator (HHILs) for Steer-by-Wire Testing

Panus Iemvarapong[1], Nuksit Noomwongs, Ratchatin Chanchareon, and Sunhapos Chantranuwathana*

Department of Mechanical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand 10330
* Corresponding Author: E-mail: Supavut.C@chula.ac.th, Tel: 02 218 6636, Fax: 02 252 2889

Abstract

Steer-by-Wire (SBW) system offers a number of benefits over conventional steering systems such as handling performance and safety. This system uses the electrical connections between the steering wheel and the vehicle's wheels instead of mechanical ones. But there are still concerns such as reliability and its interactions with drivers. To test a SBW system, a test rig with both Hardware-In-the-Loop and Human-In-the-Loop is desirable. This can decrease error from complicated model such as the tire force generation and can test interaction between the driver and the proposed SBW system. This paper presents the development of Human-Hardware-In-the-Loop (HHIL) system which designed to test SBW systems. This system combines a driving simulator and a hardware that consists of tire, suspension, and steering system. Choosing and testing of the system architecture, especially the connections and communications between various components such as the steering hardware, a car dynamic simulator, the virtual environment simulation for the driver, a number of hardware controllers and sensors, is the main concerned in this paper. The system must also work with a low level hardware controller which is an NI CompactRIO system chosen for its I/O flexibility. Choices of the car's dynamic simulator are either a digital signal processor board (DSP board) or a PC running real-time program (xPC). A dedicate PC is also chosen for rendering visual environment for the driver and also to keep and supply data of the environment. Various system architectures are considered and two communication schemes which are the RS-232 and the TCP/UDP are tested. The final selected design with all components, signals, and communication methods are presented.

***Keywords**:* Steer-by-Wire (SBW), Hardware-In-the-Loop, Human-In-the-Loop, Human-Hardware-In-the-Loop (HHIL)

1. Introduction

Recent trend in automotive industry has been about replacing mechanical systems with electronic ones. Steer-by-Wire (SBW) is one example of this trend. This system uses the electrical connections between the steering wheel and the vehicle's wheels instead of mechanical parts. The steering column is replaced with electrical wire, actuators, and sensors. SBW offers a number of advantages but it is still in the development stage because there are still concerns such as cost, reliability, safety and its interactions with drivers. A typical SBW system is shown in Figure 1.

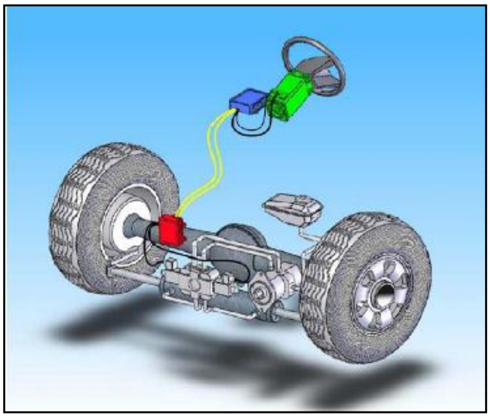

Fig. 1 A SBW system [1]

SBW has a number of advantages. SBW system can decrease fuel consumption; e.g., by reducing of weight of vehicle [2] and eliminate energy loss in hydraulic pump. Risk of injury from the steering column during a crash can also be eliminated [2]. In addition, SBW system can also offer improved handling performance and safety. Yih et al. presented a method to improve vehicle handling through SBW system [3]. For example, dynamic of the steering system can be cancelled allowing the wheel to follow the steering command closely. Safety can also be improved with an active steering which can augment the driver’s steering input to adjust the vehicle’s handling dynamics [4]. Active steering can also be used in the Direct-Yaw-Moment control (DYC) based on driver’s steering input [5]. Furthermore, the steering feel generated with force feedback motor can be adjusted electronically [1][6].

The improvement of steering feel is remarkable. With suitable torque feedback, the driver uses less effort while improving lane keeping ability. To examine force feedback of an SBW system, there are two main approaches. The first one is Hardware-In-the-Loop (HIL) [2][6] which allow the developer to test the actual system in a simulated environment. Using real parts instead of complicated mathematical models allows the test results to be more realistic. In this case, a real steering hardware is used with a simulated vehicle. The other is Human-In-the-Loop method [3] usually in a form of a driving simulator. It operates with a human driver, so the response and decision from human can be tested. This is important in many active safety systems that may be affected by the interaction between the system and the driver input. It is clear that the driver response cannot be completely modeled so Human-In-the-Loop testing will be needed.

In SBW system, combining Hardware-In-the-Loop and Human-In-the-Loop systems is a logical progress. Actual steering mechanism can be tested using real tire, eliminating complicated tire model and its interaction with a human driver can also be evaluated.

2. Human-Hardware-In-the-Loop simulator

The proposed HHIL simulator consists of two main parts. The first part is the hardware for testing the SBW which includes the suspension, steering mechanism, the tire, and the road surface simulator. The other is the driving simulator.

2.1 Hardware for Steer-by-Wire System

The SBW hardware consists of the tire, the suspension, and the steering system. The actual hardware of SBW is shown in the green part of Figure 2.

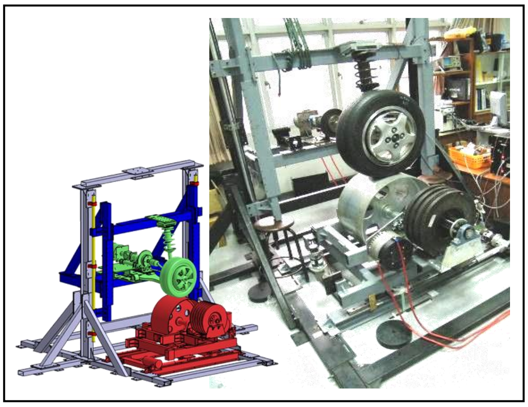

Fig. 2 The hardware of Human-Hardware-In-the-Loop simulator [7]

The system consists of 5 actuators. Realistic tire force are controlled with a wheel speed motor, road surface drum speed motor, and drum position and orientation motors. The road surface is simulated using cylindrical road drum (shown red in Figure 2) [7]. The knuckle was modified to measure tire forces. HIL is designed for running speeds up to 60 km/h and ±10 degree of wheel angle. The experiments related to the tire, suspension and steering system can be done by this HIL.

Two dedicate control systems were explored: NI CompactRIO and NI PXI systems. Due mainly to cost of I/O interface needed, the NI CompactRIO was chosen for controlling the 5 actuators while PXI system is used merely to process 6-axis force data as show in Figure 3.

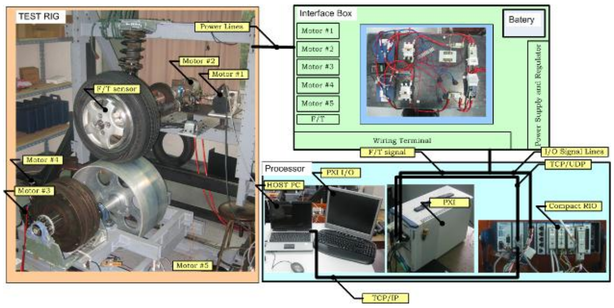

Fig. 3 The HIL control system

Verification of HIL has been done and is shown in Figure 4, the results of tire cornering stiffness testing. From Figure 4, side force and slip angle has an approximately linear relationship represented by $F_y = C_{\alpha_f} \times \alpha_f$ where F_y is the side force and α_f is the slip angle.

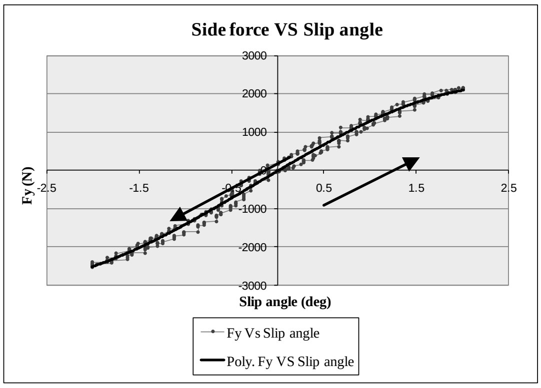

Fig. 4 Side force at various slip angles [7]

Although this processor was initially used also to simulate the dynamic of the vehicle, another option to be explored hereafter is to use an extra processor for simulation of vehicle dynamic which will allow a more complex dynamic model.

2.2 Driving Simulator

A driving simulator is used to simulate driving environment for the test driver. The selected driving simulator consists of three input devices: a steering wheel, an accelerator, and a brake pedal as shown in Figure 5. Furthermore, there is an electric motor connected to the steering wheel for generating torque feedback based mathematical model in the computer.

Fig. 5 The structure of driving simulator [8]

The controlled unit of simulator was dSPACE DS1104 card and computer that support the MATLAB® Simulink program [8]. dSPACE was chosen for simulating the bicycle model and visual environment while the course data (mainly road) is stored in the computer.

This driving simulator was used successfully to study effect of steering force feedback. The study used the vehicle's positioning error from the middle of the road to represent steering accuracy and used questionnaires on fifteen drivers to access steering comfort. The result is shown in Figure 6, Table 1, and Table 2.

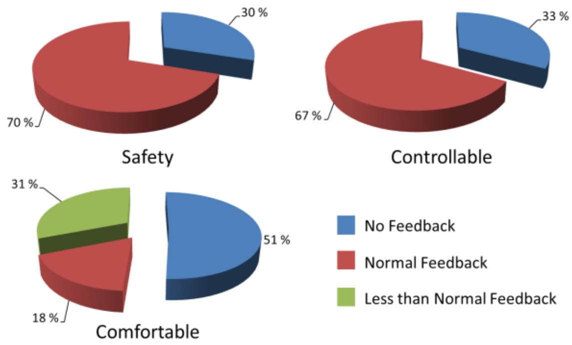

Fig. 6 The assessment of drivers' feeling [8]

From Figure 6, medium level (normal) torque feedback made most of drivers feel safe and feel that they can control the vehicle. Although it is most comfortable with no steering feedbacks, Table 1 and Table 2 indicate that normal torque feedback gives better steering accuracy both in straight line and curve.

Table 1 The numerical assessment from straight line driving [8]

Torque Feedback Characteristic	Max Error(m)	RMS Error(m)	SD. Error (m)
No Feedback	1.150	0.561	0.380
Less than Normal Feedback	0.711	0.324	0.234
Normal Feedback	0.344	0.158	0.113

Table 2 The numerical assessment from curve line driving [8]

Torque Feedback Characteristic	Max Error(m)	RMS Error(m)	SD. Error (m)
No Feedback	2.019	0.813	0.751
Less than Normal Feedback	1.657	0.694	0.652
Normal Feedback	1.579	0.684	0.578

However, this result was obtained with a mathematical model of the vehicle system. To test a SBW system, however, the author expects more realistic result if this system is combined with the Hardware-In-the-Loop SWB system. Real tire force can be used and actual SWB hardware and its controller may be tested.

2.3 Vehicle Dynamic Model

The author planned to use a bicycle model in the complete HHIL system. This model should be relatively accurate but simple enough for fast numerical simulation. This model was also used in both of the above system was a bicycle model with typical parameters of a small car assuming that the car is front wheel drive. In combining the above two system and since a more elaborate model may be need in the future, a dedicate simulator was proposed and eventually chosen in our final design.

2.3.1 Bicycle Model

A bicycle model is shown in Figure 7. In a bicycle model, left and right tire are combined together and no roll motion is modeled. Input to the model is the wheel steering angle (δ) and torque applied. An important output is the side slip angle (α_f) needed to be replicated in the HIL hardware. Other outputs include the linear speed (V), angular speed of the car (r) and linear speed of the drive wheel relative to speed of the road (V_f).

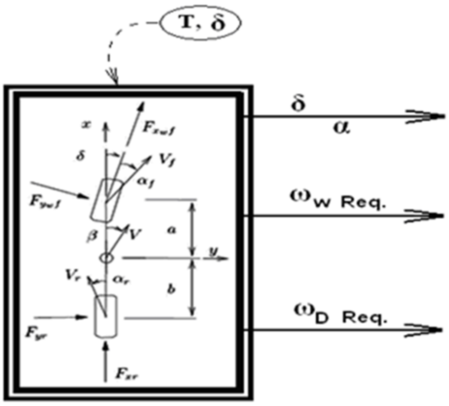

Fig. 7 The bicycle model [7]

The state equations of the bicycle model can be written as

$$\dot{u} = r \cdot v + \frac{1}{m} \cdot \{F_{xf} \cdot \cos\delta - F_{yf} \cdot \sin\delta + F_{xr}\}$$

$$\dot{v} = -r \cdot u + \frac{1}{m} \cdot \{F_{xf} \cdot \sin\delta + F_{yf} \cdot \cos\delta + F_{xr}\}$$

$$\dot{r} = \frac{1}{I_z} \cdot \{a \cdot F_{xf} \cdot \sin\delta + a \cdot F_{yf} \cdot \cos\delta - b \cdot F_{yr}\}$$

Where u is the longitudinal speed, v is the lateral speed, r is the yaw rate, m is the mass of the vehicle, F_x's are the forces along the wheels for the front wheel (subscript f) and the rear (subscript r), F_y's are the lateral forces of the wheels, I_z is the mass moment of inertia of the car, a and b are the distance from the center of mass to the front and rear wheels respectively. With the Hardware-In-the-Loop system tire force can be measured and used in place of the tire model.

2.3.2 Road Feeling

Road feeling is an important consideration is any steering system design. For the steering system as shown in Figure 8, steering feel is the torque around the wheel axis of the tire force. The relation of torque feedback and force acting on tire is shown in Equation 1.

$$T = (n_K + n_r)\, F_y \,/\, (i_L\, V_L) \tag{1}$$

where n_K is the mechanical trail, n_r is the pneumatic trail, i_L is the steering wheel ratio, and V_L is the ratio for active steering.

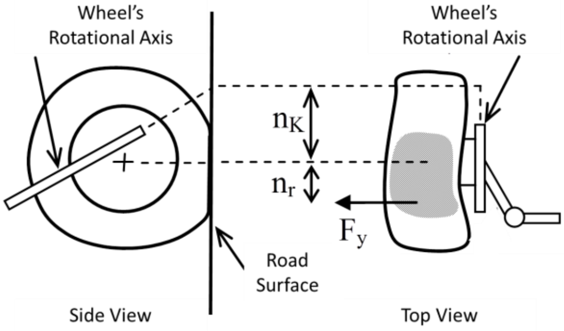

Fig. 8 Steering system mechanism [8]

The pneumatic trail can be approximated through experiment with the tire load and side slip angle. A typical approximation is given by Equation 2.

$$n_r = n_{r(3.6)} \times K \tag{2}$$

where $n_{r(3.6)}$ is n_r when the load on the tire is 3.6 kN (static load) and K is the compensator for the varying of load on tire.

$$n_{r(3.6)}=0.004|\alpha|^5-0.03|\alpha|^4+0.56|\alpha|^2+4.76|\alpha|+31.5 \tag{3}$$

$$K = 0.0008\, F_z^3+0.0207F_z^2+0.0876F_z+0.3763 \tag{4}$$

The relationship of the load on the tire, the side slip angle, and n_r are shown in Figure 9.

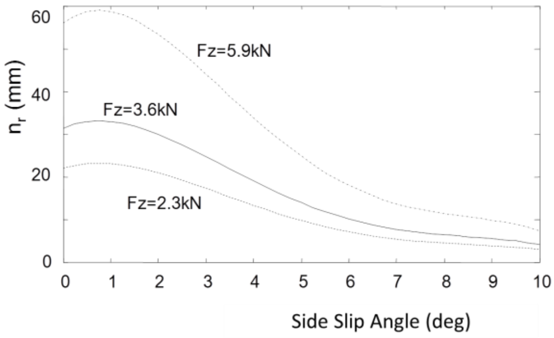

Fig. 9 Relationship of load on tire, side slip angle, and n_r [8]

Nevertheless, these four equations were used for simulated torque feedback when test driving simulator only. When the HHIL is complete, it will be possible to use real force measured from the actual tire.

3. Communication of HHIL Simulator

In this work HHIL system is proposed by combining the Hardware-In-the-Loop SBW system and the driving simulator. Connection, extra hardware, processor, distribution of processing loads and especially the communication between them must be designed with care. Initial decision was made that the motion may be controlled with the existing processors which are t he NI CompactRIO and the dSPACE DS1104, to minimize communication and a dedicate computer will be used to render the visual environment. All environment data will be stored in this computer. Course data elevation must be communicated to the dynamic simulator and the actual position and view point of the driver must be sent back for graphical rendering.

There are three main tasks to be executed: controlling the hardware, simulating the vehicle dynamics, and rendering the visual environment. Although all of these tasks can be accomplished with a single processor, the cost is prohibitive due to a number of requirements; e.g., the number of I/O type required, processing power to do both simulation and video rendering. Dividing the tasks by using multiple processors presents us with opportunities to choose a processor that suit for its task.

From the previous experiments, the processing's speed of HIL and driving simulator are about 500Hz and 2000Hz respectively. The original processing speed of the processors is shown in Table 3.

Table. 3 The properties of considered processors: NI CompactRIO, DSP board, and PC

	Processing	Transferred data's channels
NI CompactRIO (cRIO-9072/3/4) [9]	FPGA technology and 400 MHz Freescale MPC5200 processor	Supporting 8 modules of variant channel.
DSP Board (dSPACE DS1104) [12]	real time OS, Digital Signal Processing (DSP)	●A/D 8 channels ●D/A 8 channels ●Digital I/O 20-bit parallel ●Slave I/O PWM ●Inc 2 channels ●RS232, RS422, RS485
PC (Intel® PRO/1000 GT Desktop Adapter) [13]	Depend on PC	Network Communication rates 10/100/1000 Mbps, auto-negotiated

Due to the high processing speed available, it is expected the limiting factor would be the speed of communication between various hardware. Communication speed must be kept high by careful distribution of processing task which affects the amount of data needed to be transferred. Furthermore, the communication design of HHIL simulator is also depends on the processors and interface hardware available.

3.1 Processor

There are many kinds of processor to consider for the HHIL simulator. In this paper, the processors considered are the NI CompactRIO, the DSP board, and a PC.

3.1.1 NI CompactRIO

NI CompactRIO is a product of National Instruments Corporation. It consists of the I/O modules, reconfigurable field-programmable gate array (FPGA) chassis, and the real-time processor [9]. CompactRIO has high flexibility in choosing interface hardware in a relatively midrange price. Fast control loop can be implemented with the FPGA though this is difficult for complex algorithms. Its real time processor has relatively low speed, however. Initially, this system is used to both control the hardware and simulated the vehicle dynamics which allow fast interaction between the real hardware and the simulated one.

3.1.2 Digital Signal Processor Board (DSP board)

Important features of DSP boards include a fast processor and good communication channels as DSP boards need to collect and distribute data from/to many different sources [10]. dSPACE DS1104, one of the DSP board model from dSPACE Inc., is considered. dSPACE is fast and vehicle dynamic can be implement relatively easy with code in Matlab Simulink, however, it has limited interface port. In this case, communication is available through serial RS-232, digital I/O, or analog voltage.

3.1.3 PC Running Real-Time Program (PC)

A personal computer (PC) is considered because of its flexibility in choosing interface boards. It is decided that a real-time operating system is needed to simulate vehicle dynamic with the desired speed. Matlab Simulink can also be used to simulate vehicle dynamics [11].

Furthermore, cheap video card is readily available making it an ideal choice for rendering the visual environment. However, to take advantage of graphic software like OpenGL, real-time processor cannot be used. As a result, another dedicate PC will be used for this purpose only.

3.2 Interface Hardware

From these three processors, NI CompactRIO is chosen to be low level processor because of the flexible in signal I/O port. It can support 8 modules of various I/O ports which can be selected for the most appropriate signal transmission. Moreover, it has 4 types of communicative channels readily available; RS232, transmission control protocol (TCP), a digital I/O, and analog voltage.

Then, the design choices are narrow down to the followings. The NI CompactRIO will be used to control low level hardware. Either the DSP or the PC will be used to simulate the vehicle dynamic. Either RS-232 or TCP will be used to communicate between the processors.

There are three choices of communication design depends on the chosen processor shown in Figures 10-12.

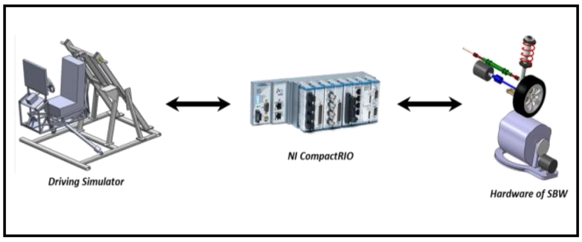

Fig. 10 Schematic view of the communication of HHILs using NI CompactRIO

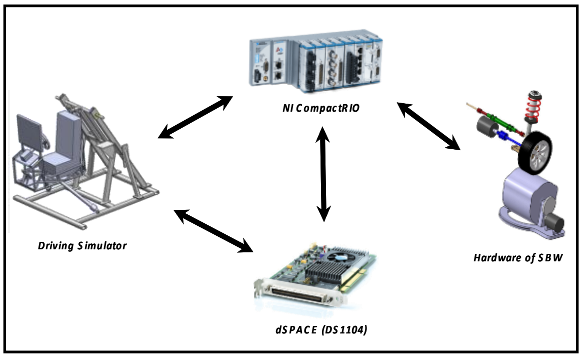

Fig. 11 Schematic view of the communication of HHILs using NI CompactRIO and DSP board

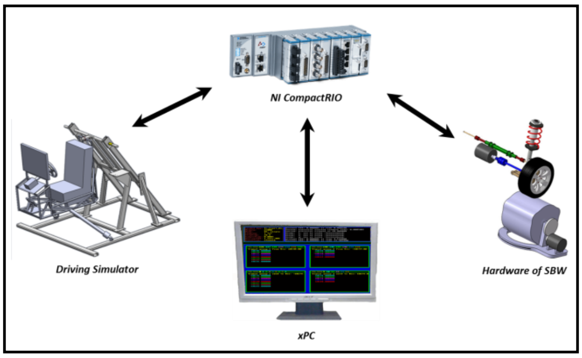

Fig. 12 Schematic view of the communication of HHILs using NI CompactRIO and a PC

Using NI CompactRIO alone which shows in Figure 10 was our initial choice but it was eventually rejected. In this case the CompactRIO must also simulate vehicle dynamics which made our initial test speed to be limited. In the second case which shows in Figure 11, RS-232 communication is chosen for the DSP board. Although, RS-232 is relatively slow, faster communication is also possible by routing the driver command from driving simulator directly to the DSP board for vehicle dynamic simulation. In Fig. 12, TCP (UDP) is chosen with NI CompactRIO as the central path way. This allows us to save cost since the PC need only a TCP Ethernet port.

4. Experiment and Result

The interested communication designs, RS-232 for the DSP board and NI CompactRIO and TCP/UDP for a PC and NI CompactRIO, are tested experimentally. According to the bicycle model, around 15 variables; as shown in table 4, are required to be transferred between processors for real-time simulation.

Test program is written in Matlab Simulink and NI Labview program. The program is designed to transmit and record the sequential number (1,2,3,…), size 64 bits for RS232 and 480 bits for TCP/UDP, via RS232 and TCP/UDP in different sampling time. Missing or duplicated data are used to indicate error in communication.

Table. 4 The required variable for for simulate bicycle model.

Parameter	Number	Note
1. Steering Angle	1	
2. Torque from acceleration or brake	1	
3. Force acting on tire	3	(x-y-z axis)
4. Torque acting on tire	2	(x-z axis)
5. Track Data	3	(x-y-z axis)
6. Angle speed of car	1	
7. Longitudinal and lateral speed	2	
9. yaw angle	1	
10. Slip Angle	1	

4.1 NI CompactRIO and DSP board

A sample record of the RS232 communication between the DSP board with the NI CompactRIO is shown in Table 5. Transmitted data at 1000 μs can get the accurate data but at lower than 1000 μs, some of number are missing. Moreover, at lower than 500 μs, system was unable to response.

Table. 5 The record of data transmitted between NI CompactRIO and DSP board

Sampling time (μs)	Written by NI Compact RIO	Read by DSP	Written by DSP	Read by NI Compact RIO
500	0 1 2 3 4 5 6 7 8 9 10 11 12 …	0 1 2 3 **4 4** 5 6 7 8 **9 9** **10 10** …	0 1 2 3 4 5 6 7 8 9 10 11 12 …	unable to response
800	0 1 2 3 4 5 6 7 8 9 10 11 12 …	0 1 2 3 **4 4** 5 6 7 8 9 **10** **10** …	0 1 2 3 4 5 6 7 8 9 10 11 12 …	0 1 2 3 4 5 6 7 8 9 **10** **12** …
1000	0 1 2 3 4 5 6 7 8 9 10 11 12 …	0 1 2 3 4 5 6 7 8 9 10 11 12 …	0 1 2 3 4 5 6 7 8 9 10 11 12 …	0 1 2 3 4 5 6 7 8 9 10 11 12 …

Hence, the speed of communication via RS232 can be calculated as

$$speed = \frac{64\ bits}{1000\ \mu s} \times \frac{10^6\ \mu s}{1\ s} \qquad eq.5$$

$$speed = 64000\ bps$$

Compared to the theory, RS232 can transmit data maximum 115200 bps. Although, the result shows the real speed is about half of its theoretical value.

With the bicycle model, 15 variables with 4 bytes each, the maximum data is 60 bytes per update. Then, from Equation 6, the simulation update rate via RS232 for HHIL will be 133.33 times per sec or 133.33 Hz.

$$speed = \frac{115200\ bits}{1\ s} \times \frac{1\ byte}{8\ bits} \times \frac{1\ time}{60 bytes} \quad eq.6$$

$$speed = 133.33\ s^{-1}\ (Hz)$$

This is much lower than the existing simulation speed. Thus, this design will reduce the simulation update rate and hence accuracy of the simulation. It is also important to note that the number of variable may be reduced to 10 and the number of byte for each variable may be reduced to 2 bytes, but this is still slower than that of the current simulation update rate.

4.2 NI CompactRIO and PC

Communication speed of an xPC target running on PC computer connected with NI CompactRIO by TCP/UDP is tested. UDP is chosen for its simplicity. A typical record is shown in Table 6. The record showed that data can be transmitted at 300 μs interval. At lower than 300 μs, some of number lost.

Table. 6 The record of data transmitted between NI CompactRIO and xPC target running on PC

Sampling time (μs)	Written by NI Compact RIO	Read by DSP	Written by DSP	Read by NI Compact RIO
200	0 1 2 3 4 5 6 7 8 9 10 11 …	0 **1 3** 4 5 6 **7 9** 10 11…	0 1 2 3 4 5 6 7 8 9 10 11 …	**0 2** 3 4 5 6 7 **8** **10** 11 …
300	0 1 2 3 4 5 6 7 8 9 10 11 …	0 1 2 3 4 5 6 7 8 9 10 11 …	0 1 2 3 4 5 6 7 8 9 10 11 …	0 1 2 3 4 5 6 7 8 9 10 11 …

Hence, the speed of communication via UDP will be

$$speed = \frac{480\ bits}{300\ \mu s} \times \frac{10^6\ \mu s}{1\ s} \qquad eq.7$$

$$speed = 1600000\ bps$$

Consider in the real situation; again, define each number to be each parameter, size 4 bytes, the maximum data is about 60 bytes per one update. Then, from Equation 8, the transmitted speed via UDP for HHIL will be 3333.33 times per sec or 3333.33 Hz.

$$speed = \frac{1\ time}{300\ \mu s} \times \frac{10^6\ \mu s}{1\ s} \qquad eq.8$$

$$speed = 3333.33\, s^{-1}\ (Hz)$$

5. Conclusion

This paper presents an HHIL simulator for SBW testing. The HHIL simulator consists of hardware of a SBW system and a driving simulator. System architecture and communication design of the HHIL simulator was the main concern in this paper. There are three patterns of communication design investigated which depend on processors; NI compactRIO, DSP board, and PC. Each processor has its own limitations that make their performance different. Using NI compactRIO alone can cause the processor to overwork and complexity of the dynamics model must be limited. The current update rate of the HHIL simulator is about 500Hz and should not be reduced. Using NI CompactRIO and DSP board allows complex model but the communication between processor is constrained to the slower RS232. From experiment, update rate would be limited to about 133Hz. Using NI CompactRIO and PC, communication between processors can be done via TCP/UDP. From experiment, update rate is 3333.33 Hz. Clearly, this offers better performance while allowing complex vehicle model to be implemented.

6. Acknowledgement

The work done in this paper was supported by National Metal and Materials Technology Center (MTEC) and Chulalongkorn University, Thailand.

7. References

[1] Bertacchini A., Tamagnini L., and Pavan P. (2006). Force Feedback in Steer-by-Wire Systems: Architecture and Experimental Results, paper presented in Industrial Electronics 2006, Montreal, Que, Canada.

[2] Jang, S.H., Park, T.J. and Han C.S. (2003). A Control of Vehicle Using Steer-by-Wire System with Hardware-In-the-Loop-Simulation System, paper presented in *the International Conference on Advanced Intelligent Mechatronics (AIM 2003)*.

[3] Paul Y., Jihan R., and J.Christian G. (2003). Modification of Vehicle Handling Characteristics via Steer-by-Wire, paper presented in *the American Control Conference*, Denvor, Colorado

[4] Paul Y. (2005). Steer-by-Wire: Implications for Vehicle Handling and Safety, Stanford University, California.

[5] Hayama, R., Nishizaki, K., Nakano, S., and Katou K. (2000). The Vehicle Stability Control Responsibility Improvement Using Steer-by-Wire, paper presented in *Intelligent Vehicles Symposium 2000*.

[6] Kim, C.J., Jang, J.H., Oh S.K., Lee J.Y., Han, C.S., and Hedrick J.K. (2008). Development of a Control Algorith foa a Rack-Actuating Steer-by-Wire System Using Road Information Feedback, paper presented in *the Institution of Mechanical Engineering, Part D: Journal of Automobile Engineering 2008*.

[7] Sookchaiyaporn S., Noomwongs N., Chanchareon R., Wannasuphoprasit W., and Chantranuwathana S., Tire-Suspension-Steering Hardware-In-the-Loop for Automotive development, paper presented in *The 6th International Conference on Automotive Engineering*, 2010, Bangkok, Thailand.

[8] Akapong T., Chayut L., Tosaporn L., and Tanapon, P., Development of a Driving

Simulator for Steering Force Feedback Study, paper presented in *The 21th The Conference of The Mechanical Engineering Network of Thailand*, 2007, Chonburi, Thailand.

[9] National Instruments Corporation. *Products & Services*, URL: http://www.ni.com, access on 5/11/2010

[10] GlobalSpec Inc., New York. *Products & Services: Learn More About DSP Boards*, URL: http://www.globalspec.com, access on 29/06/2011

[11] The MathWorks, Inc. *Products & Services*, URL: http://www.mathworks.com, access on 29/06/2011

[12] dSPACE GmbH. Downloads: *Product Information*, URL: http://www.dspace.aero/en/ltd/home.cfm, access on 1/11/2010

[13] Intel Corporation. *Products: Embedded & Communications*, URL: http://www.intel.com, access on 29/06/2011