International Journal of Automotive Technology, Vol. 7, No. 6, pp. 687–695 (2006)

DEVELOPMENT OF A NETWORK-BASED TRACTION CONTROL SYSTEM, VALIDATION OF ITS TRACTION CONTROL ALGORITHM AND EVALUATION OF ITS PERFORMANCE USING NET-HILS

J. RYU[1)], M. YOON[1)] and M. SUNWOO[2)*]

[1)]Department of Automotive Engineering, Graduate School, Hanyang University, Seoul 133-791, Korea
[2)]Department of Mechanical Engineering, Hanyang University, Seoul 133-791, Korea

(Received 2 January 2006; Revised 14 June 2006)

ABSTRACT–This paper presents a network-based traction control system (TCS), where several electric control units (ECUs) are connected by a controller area network (CAN) communication system. The control system consists of four ECUs: the electric throttle controller, the transmission controller, the engine controller and the traction controller. In order to validate the traction control algorithm of the network-based TCS and evaluate its performance, a Hardware-In-the-Loop Simulation (HILS) environment was developed. Herein we propose a new concept of the HILS environment called the network-based HILS (Net-HILS) for the development and validation of network-based control systems which include smart sensors or actuators. In this study, we report that we have designed a network-based TCS, validated its algorithm and evaluated its performance using Net-HILS.

KEY WORDS : HILS (Hardware-In-the-Loop Simulation), Network-based TCS (Traction control system), Net-HILS (Network-based HILS), CAN (Controller Area Network)

1. INTRODUCTION

Recently, the importance of vehicle system safety has received much attention. In particular, the electronic control technology that ensures vehicle stability in dangerous situations has been in the spotlight. However, competition in the automotive industry is fierce, meaning that car companies need to balance economic considerations with the continued demand for the development of new vehicle safety technologies. Consequently, rules have been established governing the development period and cost reduction schemes of electronic control technology.

Typically, during the development of electronic control systems, numerous tests need to be repeated, from testing of control algorithms to proving ground tests. These tests are necessary to verify the performance, stability and reliability of vehicle systems. Therefore, companies have had to spend considerable time and expense in optimizing electronic control systems.

To overcome the various problems encountered in the development of control systems, Hardware-In-the-Loop simulation (HILS), a real-time computational simulation method, has been developed (Jo *et al.*, 2006). The HILS environment has a single closed loop formation that connects the real electronic control unit (ECU) to accurate models of a target plant, and is simulated on a computation node in real time. This provides a virtual test environment for the development of ECUs in electronic control systems (Ryu *et al.*, 1999; Yoon *et al.*, 2005).

One of the main advantages of using HILS is to reduce the development time of control systems, since it is possible to develop a system to the stage that ECUs can be applied directly to a real plant. This is one reason for accomplishing signal interface through the real-time simulation between real ECUs and computation nodes. Additionally, HILS can contribute to the reduction of research costs because HILS is based on computer simulation which adds no extra expense because simulations are infinitely repeatable. This also makes it possible to simulate the many possible serious situations a vehicle may encounter without having to subject a real vehicle to such stresses.

In this research, we constructed a HILS environment to evaluate the performance of a traction control system (TCS) and validate its control algorithm. As mentioned, the implementation of electronic control systems in vehicles is rapidly increasing. At the same time, ECUs in electronic control systems involve the exchange of enormous amounts of data to control a vehicle toward the target state. This requirement brings about the complicated

**Corresponding author.* e-mail: msunwoo@hanyang.ac.kr point-to-point wiring between ECUs, and consequently makes it difficult to maintain the electronic control system in a vehicle. In order to solve these problems, the network-based control system has been recently introduced. Integrating ECUs into a network system to replace the traditional point-to-point wiring has enormous advantages, including lower cost, reduced weight and power, simpler installation, higher reliability, ease of system diagnosis and maintenance, and increased system agility (Zhang and Wang, 2002). With these factors in mind, we have developed a network-based TCS which has several ECUs and a controller area network (CAN) protocol. Additionally, we propose a new HILS environment concept, called Net-HILS.

This paper is organized as follows. In Section 2, two main computational models of an engine and vehicle chassis will be explained. Section 3 includes the control algorithm based on the sliding mode control concept for regulating the wheel slip. In Section 4, the system partitioning of the overall power-train and CAN networking configuration are presented to embody the control algorithm in the network-based TCS. Section 5 explains the distributed HILS components and introduces the Net-HILS to validate the network-based control system. In Section 6, the simulation results for four vehicle driving situations will be shown and discussed. Finally, Section 7 presents the conclusions determined by this research.

2. SYSTEM MODELING

To validate the control algorithm and evaluate the performance of a vehicle system, we needed to construct an overall vehicle system that configures all of the drive-train components, such as a throttle body, engine, transmission and chassis.

2.1. Engine Model

The engine model adopted in this study is taken from (Yoon and Sunwoo, 1999). This engine model is represented by three input values (throttle position, ignition timing, fuel flow) and three state values (intake manifold pressure, fuel mass of fuel film, engine rotational speed) and one disturbance (load torque).

It is assumed that the engine always operates on the exact stoichiometric air-fuel (A/F) ratio and MBT (Minimum spark advance for Best Torque) set by the control action of an engine ECU. Under this assumption, the engine model can be configured simply using MATLAB/ SIMULINK®.

2.2. Vehicle Chassis Model

In order to fully validate the performance of the traction control algorithm, the vehicle model with eight degrees of freedom (DOF) is used(Smith and Starkey, 1995). (In general, most research has focused on low-level DOF vehicle models, including longitudinal maneuvering and two wheel rotational speed). In addition, the tire model by Pacejka, which is sufficiently accurate to be used in TCS, is adopted in the vehicle model (Bakker *et al.*, 1989). And the vehicle is driven by front wheels and a flat road is only considered as a simulation load condition.

3. TRACTION CONTROL ALGORITHM

The main objective of the traction control algorithm is to design wheels that meet a target slip threshold. In designing the algorithm, the range in which maximum traction force can be generated while maintaining vehicle stability is determined. The control procedure is processed as follows. First, a controller calculates the desired traction torque to govern the target wheel slip, after which the desired throttle position can be determined in terms of desired engine torque. In this paper, two control stages are suggested and developed throughout the sliding mode control algorithm (Slotine and Li, 1991).

3.1. Slip Control Algorithm

In the first control stage, the wheel torque for tracking the desired wheel slip ratio is calculated using a sliding mode control algorithm. The desired traction torque can be obtained by equation (1) as follows (Kang *et al.*, 2004).

$$T_w=\frac{I_w\omega^2}{v}\left(R\dot{\lambda}_d+\frac{\dot{v}}{\omega}-R\eta_1 sat\left(\frac{s_1}{\phi_1}\right)\right)+T_L \qquad (1)$$

Table 1. Nomenclature.

T_w	Desired wheel torque [Nm]
T_e	Desired engine torque [Nm]
T_L	Load torque [Nm]
I_w	The effective inertia of wheel [kg/m^2]
ω	Wheel speed [rad/s]
v	Longitudinal velocity [m/s]
R_w	Wheel radius [m]
TC	Throttle chracteristics
PRI	Effect of pressure ratio
MA	Maximum flow rate as shape of the throttle valve
V_{man}	Manifold volume [m^3]
C_D	Discharge coefficient [–]
T_{man}	Intake manifold air temperature [K]
$\dot{m}_{ap}$	Air mass flow rate into the intake port [kg/s]
R	Gas constant

3.2. Engine Torque Control Algorithm

Similar to the slip control stage, at the second control stage and throughout the procedure, the throttle characteristic equation for deciding the desired throttle position is obtained by the following equation (2) (Kang *et al.*, 2004).

$$TC=\frac{V_{man}}{C_D MAPRI\times RT_{man}}\times\left(\frac{RT_{man}}{V_{man}}\dot{m}_{ap}+\dot{P}_{desired\_man}+\eta_2 sat\left(\frac{s_2}{\varepsilon_2}\right)\right) \qquad (2)$$

4. CONTROL SYSTEM PARTITIONING AND CAN CONFIGURATION

4.1. Control System Partitioning

The network-based TCS designed in this research consists of four ECUs which are integrated into a CAN protocol. In order to construct a network-based TCS, the overall power-train is derived as shown in Figure 1, and each part is controlled by a corresponding ECU.

From the left side of Figure 1, the overall power-train consists of an electronic throttle body, an engine, a transmission, a chassis platform composed of four wheels, a wheel shaft, and suspensions. These have their respective electronic throttle controller, engine controller, transmission controller and vehicle dynamics controller. We were thus able to achieve progress in evaluation of the performance of the ECUs and the control algorithm using a HILS environment. The HILS environment for research-oriented ECUs will be explained in detail in the next section, Section 5.

Furthermore, use of a smart sensor for the measurement of wheel speed was assumed. The smart sensor has a network module which allows it to communicate with other devices in the network.

4.2. Network Configuration

In this research, the network environment of TCS has been constructed using a CAN protocol developed for the in-vehicle communication network. CAN is a standard protocol used in the automotive industry and can support a maximum communication speed of 10 Mbit/s (Lawrenz *et al.*, 1997).

When a control system is networked, i.e., when the ECUs are connected with a single bus using CAN, they are able to not only share mutual data, but also their

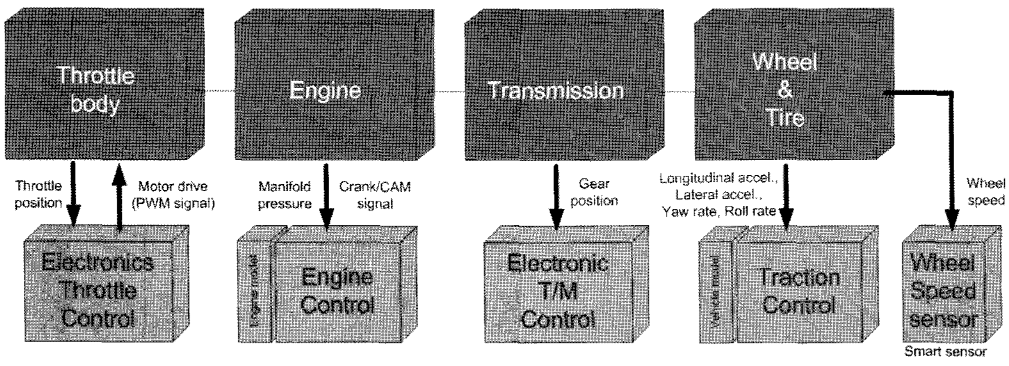

Figure 1. Control system partitioning.

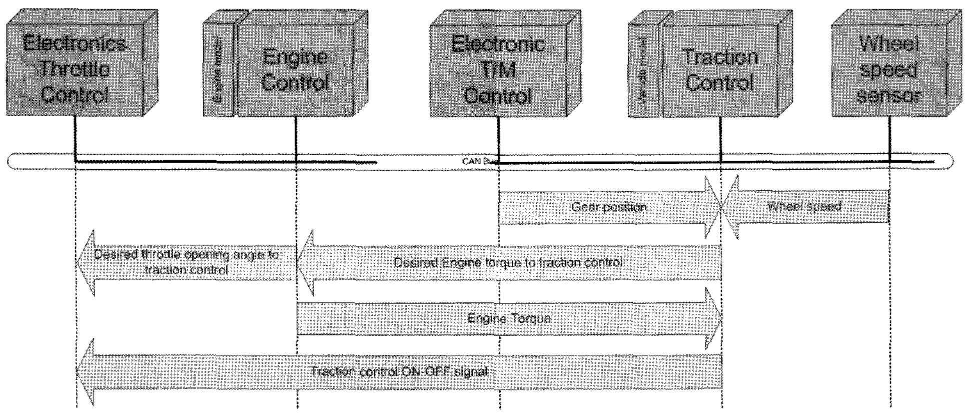
Electronics Throttle Control
Engine model
Engine Control
Electronic T/M Control
Vehicle model
Traction Control
Wheel speed sensor
CAN Bus
Gear position
Wheel speed
Desired throttle opening angle to traction control
Desired Engine torque to traction control
Engine Torque
Traction control ON-OFF signal

Figure 2. Control system networking.

Table 2. CAN Network datas.

Controller	Transfer data	CAN ID	Receive data
Electonic throttle controller	x	x	Desired throttle position, traction control ON/OFF
Engine controller	Desired throttle position	3	Desired engine torque
Transmission controller	Gear position	4	x
Vehicle dynamics controller	Desired engine torque	2	Gear position
	Traction control ON/OFF	1	

functionalities can be expected to extend throughout the modulation of the control system.

Figure 2 represents the flow of data between ECUs through CAN bus and sensor/actuators signals that have to be processed in the ECUs. Additionally, the CAN IDs of moving data have been arranged in Table 2.

5. HILS ENVIRONMENT

To run research-oriented ECUs for a network-based TCS without using a real engine or vehicle, it is necessary to plausibly simulate the electrical signals of the sensors and actuators. In order to create HILS, a powerful computing system is necessary which can emulate the behaviors of the controlled process accurately. This system should be interfaced with the actual controller in real-time through the signals of sensors and actuators. The controlled object and auxiliary components in the control loop are replaced by real-time simulations of their behavior. The only real component in the test setup may be the ECU itself. The real part of an actuator is not adopted into the developed HILS because the ECUs are obviously divided from the actuator.

Sufficiently complex system dynamics may overwhelm even the fastest processor. The solution is to distribute the simulation and to perform parallel processing. This HILS environment is typically referred to as a distributed HILS platform.

The platform also consists of a widely used PC and commercial-off-the-shelf (COTS) I/O boards. Furthermore, this environment can be integrated seamlessly into the modern development process including rapid control prototyping (RCP). These features make the HILS equipment more cost-effective and flexible. The HILS uses an automatic code generation extension, RTW® of the MATLAB® tool-chain, and this helps the control system developers to more easily handle the controlled-object model and to test the control system in a more comfortable and time-effective manner (Lee *et al.*, 2003).

5.1. Net-HILS

In order to effectively develop the network-based TCS,

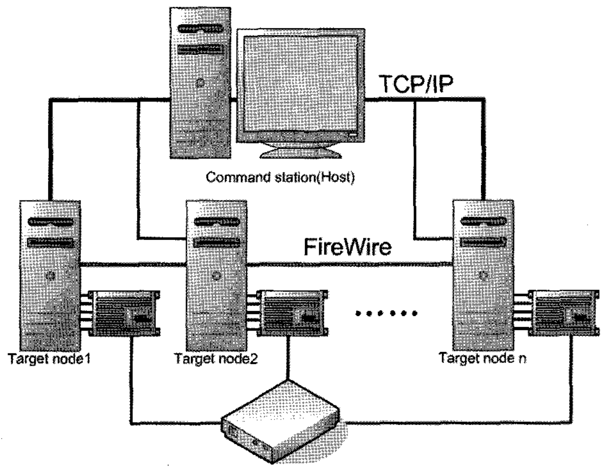

Figure 3. Distributed HILS environment.

we propose in this study a new concept for the HILS environment which is based on the integration of a general HILS and a distributed, real-time control system. This HILS environment is named the Network-based HILS (Net-HILS).

Net-HILS is particularly useful when a smart sensor or smart actuators are included in the entire control system loop, being networked with several ECUs to exchange current sensing values or the desired set-point of a control plant.

Figure 3 shows a distributed HILS environment and Figure 4 shows a Net-HILS environment developed in this research. In a general distributed HILS environment, all actuator or sensor signals must be simulated as real electric signals. This requires the complicated wiring of signal lines between several I/O boards and ECUs. Also, noise rejection circuits have to unfortunately be added to the I/O boards. These complicated wiring tasks can be a substantial trouble for the control system developers.

However, Net-HILS can reduce these complex tasks. It is assumed that some smart actuators or sensors are included in the control network phase, and these are loaded in the computation PC clusters as a model and simulated in real-time. Therefore, the output data of some

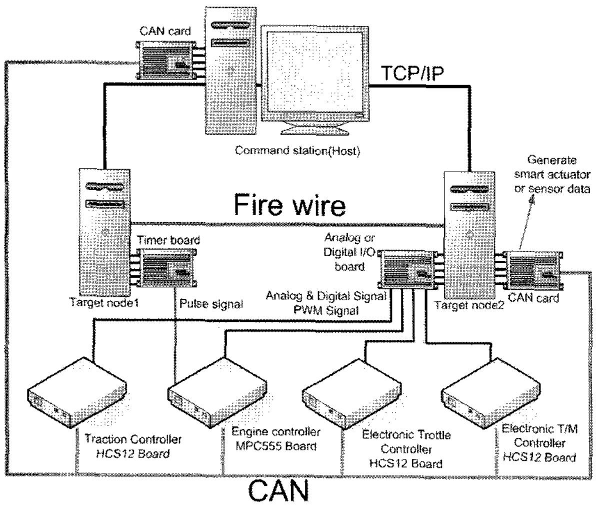
CAN card
TCP/IP
Command station(Host)
Generate smart actuator or sensor data
Fire wire
Timer board
Analog or Digital I/O board
Target node1
Pulse signal
Analog & Digital Signal PWM Signal
Target node2
CAN card
Traction Controller HCS12 Board
Engine controller MPC555 Board
Electronic Trottle Controller HCS12 Board
Electronic T/M Controller HCS12 Board
CAN

Figure 4. Net-HILS and network-based TCS block-diagram.

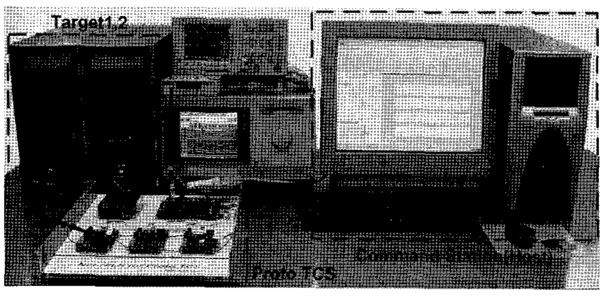

Figure 5. Net-HILS environment.

smart actuators or sensors are connected with the network line, made up of several ECUs, thereby eliminating much of the wiring task.

Additionally, a developer can advance the research on control performance using a Net-HILS, because, a developer can be free enormous wiring tasks. The entire Net-HILS environment constructed in this research is laid out in Figure 5.

5.1.1. Hardware architecture

As seen in Figure 4, the physical configuration of the simulation system is divided into the host (command station) and the target sides (target nodes). On the host side, engineers design their model and run off-line simulations, and are able to contact the target side to observe the real-time simulation of the models. On the target side, several microprocessors are connected via a high-speed communication network and serve in the execution of simulation distribution.

The target side is implemented by two PC clusters which use a Pentium IV® 1.8-GHz processor and 512 MB DDR RAM in order to simulate the engine and vehicle

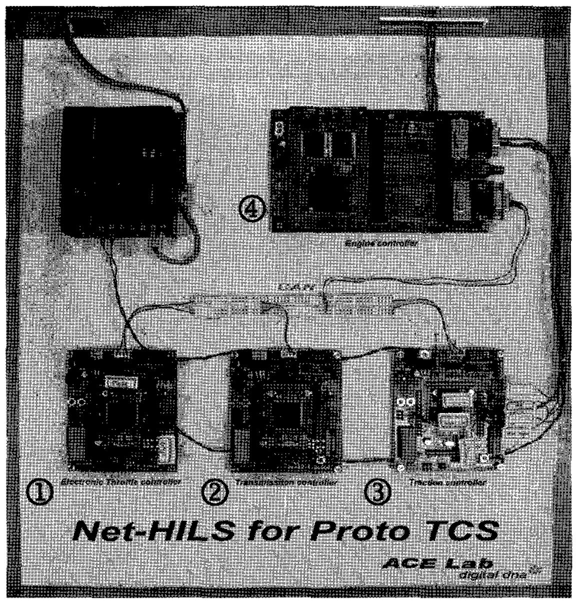

Figure 6. Proto TCS.

models. Each PC cluster corresponds to target nodes 1 and 2, as identified in Figure 4.

The engine and vehicle models are downloaded in target node 2 and simulated in real-time. In the target node1, the Crank/Cam signals of the engine are generated as square pulse signals, based on events such that they are generated in exact timing with the crank rotational angle.

Figure 6 is a prototype of the Network-based TCS. As discussed above, we divided the power-train into four components which have individual controllers (ECUs) and which are connected with a CAN protocol. The four ECUs in the prototype of Figure 6 and the power supply provide DC voltage.

As shown in Figure 6, two kinds of controllers are used. Number 1, 2 and 3 indicate the electronic throttle controller, transmission controller and vehicle dynamics controller, respectively. These are T-boards which have freescale's MC9S12DP256 micro-processor and several peripheral devises such as a display LED, buzzer, power regulator, etc. Number 4 is an engine controller, which is realized as a Wruz Board using Motorola's MPC555 micro-processor.

5.1.2. Software architecture

The RT-LAB® environment enables distributed simulation and parallel processing over a PC cluster, making it useful software in a distributed HILS. RT-LAB® from Opal-RT Inc. is an industrial grade software package for engineers who use mathematical block diagrams for simulation, control and related applications, and is selected for the distributed simulation and advanced digital signal generation/capture of the HILS platform.

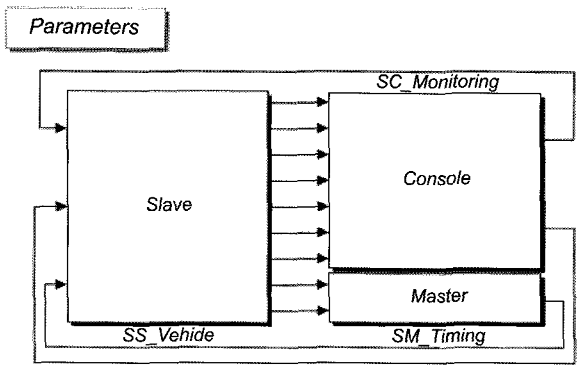

Figure 7. Overall SIMULINK® model.

In addition, RT-LAB® works with the standard SIMULINK® block diagram languages and code generators to enable parallel, real-time execution of simulations and input/output interfacing.

5.2. SIMULINK® Models

5.2.1. Console

The console is configured to observe real-time data from target nodes and to transfer user commands using SIMULINK®. The SIMULINK® model of the console is given in Figure 7.

5.2.2. Master node

In Figure 7, the master node corresponds to the target node1, in which the SIMULINK® model is included to generate a crank/cam signal with a Dcc20p® board that can be controlled by RT-EVENT (in the RT-LAB toolbox, offered by the RT-LAB® software). In order to create exact transitions of the square pulse for the crank/cam signals, two 'Event generator' (Opal-RT Technologies Inc., 2000) blocks in the RT-EVENT library are used. In addition, the duty ratio of the PWM signal generated by the electronic throttle controller is detected using the 'PWM input' (Opal-RT Technologies Inc., 2000). Here, the PWM signal is the driving input of a motor for controlling throttle position. Figure 8 shows a SIMULINK® model called 'Timing I/O subsystem' in the master node.

5.2.3. Slave node

In Figure 7, the slave node corresponds to target node 2, in which the engine, vehicle chassis and tire models are included to simulate an engine operating state and vehicle maneuvers in real-time. Simple electronic throttle body and transmission models are also inserted into the slave node. Each model is shown in Figure 9.

In this paper, it is assumed that the smart sensor is the wheel speed sensor and that smart actuators do not exist. We assume that wheel speed is detected accurately; wheel speed sensor transfers wheel speed calculated from vehicle dynamics model directly having fixed period which is sampling time of sensor. In this case, the sampling time must be within the model parameter of the smart sensor. Figure 10 gives the appearance of the CAN subsystem, which has a wheel speed sensor model and a subsystem for monitoring the simulation state. These are each enabled for every given period.

To control a vehicle toward desired states, various

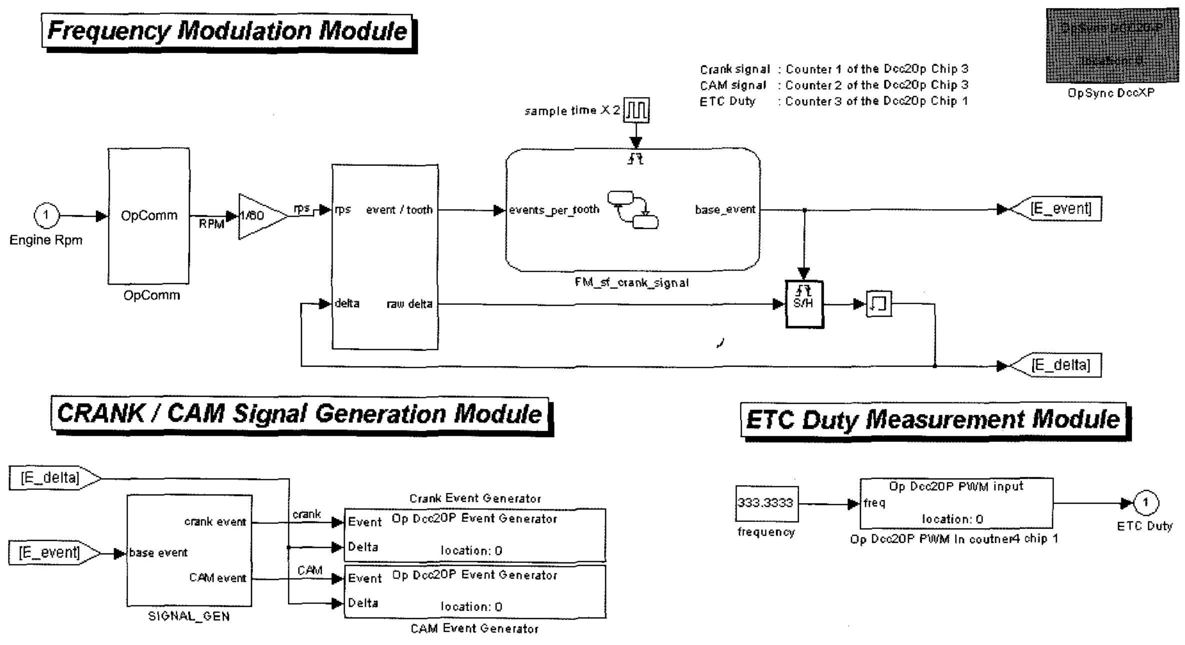

Figure 8. Timing I/O subsystem.

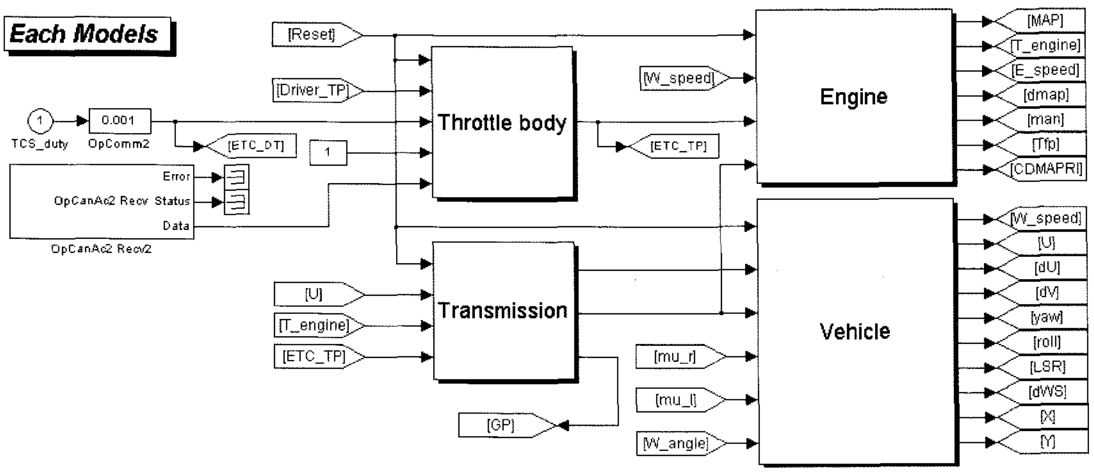

Figure 9. Each models.

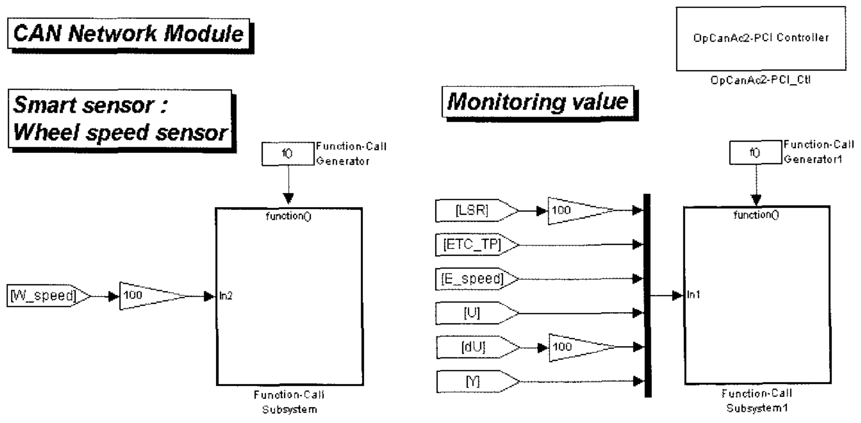

Figure 10. CAN network module.

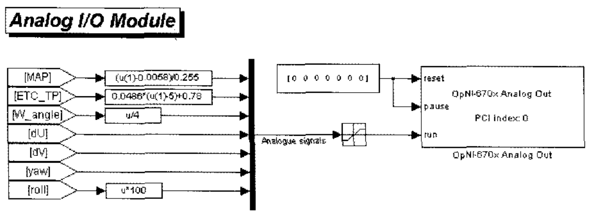

Figure 11. Analog I/O subsystem.

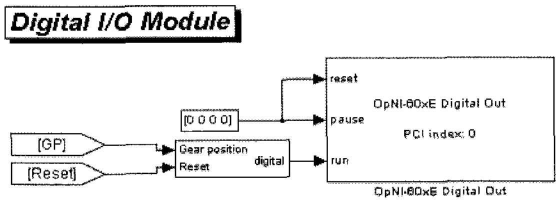

Figure 12. Digital I/O subsystem.

sensors such as the longitudinal/lateral acceleration sensor, the yaw rate sensor, throttle position sensor, manifold pressure sensor and others are installed in a real vehicle. In order to simulate the signals of these sensors, an analog I/O subsystem and digital I/O subsystem are also contained within the slave node, with the exception of the several models explained above.

The analog I/O subsystem and digital I/O subsystem are configured using the SIMULINK® and RT-LAB toolbox, and real analog and digital signals are created from the NI-6703® board and NI-6025E® board, respectively. These boards are manufactured by National Instrument. Figures 11 and 12 show the analog I/O subsystem and the digital I/O subsystem, respectively.

6. NET-HILS RESULTS

Simulation was performed for four different driving situations, which are: slippery road conditions, abrupt change in road conditions, split road conditions and turning on a slippery road. A solid line and dashed line correspond to the activation and deactivation of the network-based TCS, respectively.

6.1. Slippery Road Condition

In order to verify the performance of the network-based TCS on slippery roads, such as a snowy or frozen road, it is assumed that the friction coefficient of the slippery road is 0.3 and that the driver fully pushes the accelerator to the 70° throttle position at initial start-up. Simulation results are shown in Figure 13.

As seen by a solid line in the first graph of Figure 13, the driver's intent (throttle position: 70°) is ignored and the throttle position is then regulated between 10° and 20° by the electronic throttle controller. Accordingly, the wheel slip ratio is controlled to a target value (slip ratio: 0.125) as demonstrated in the second graph of Figure 13. On the other hand, the slip ratio is increased to 0.7 when TCS is deactivated. Finally, when the network-based TCS is activated, the acceleration performance is enhanced over that of the other case. (Arithmetic average: 1.0651 [m/s^2] > 0.7851 [m/s^2])

6.2. Abrupt Change of Road Condition

Driving situations in which an abrupt change of road condition is encountered are similar to those of slippery road conditions. However, the friction coefficient of the road only changes from 0.8 to 0.3 suddenly and then returns to 0.8 after 2 seconds. In other words, an abrupt change of road condition occurs at initial start-up. Several simulation results are represented in Figure 14.

In the first graph of Figure 14, when an instantaneous change of road condition occurs, the throttle position closes to track the target slip ratio. The second graph shows that the slip ratio follows the target slip ratio with some fluctuation. In terms of vehicle acceleration performance, when the network-based TCS is activated in this case (arithmetic average: 1.31274 [m/s^2]), performance is improved over the alternative situation (arithmetic average: 1.2421 [m/s^2]).

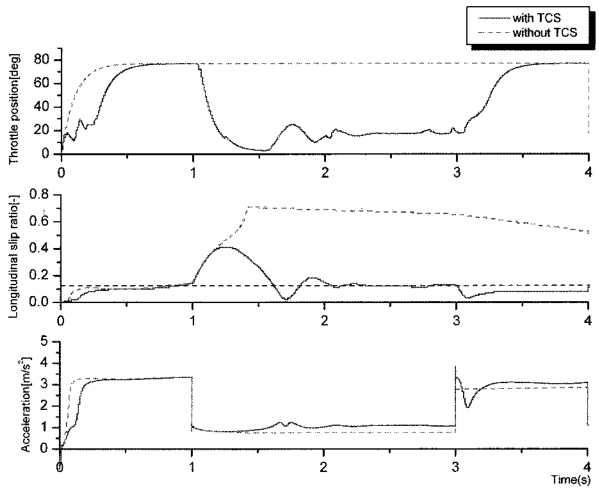

Figure 14. Results for abrupt change of road condition.

6.3. Split Road Condition

The two driving situations above are applied to consider the respective performance of the network-based TCS in a longitudinal maneuver. In order to predict a lateral vehicle maneuver when the network-based TCS is activated, added simulations are carried out for situations when a car is either going straight and encounters a split in the road or is turning on a slippery road.

Figure 15 shows the results of the simulation for the split road scenario. In the first and second graph, throttle position is regulated between 10° and 20°, and the slip ratio tracks the target ratio well. As seen in the third graph, when the network-based TCS is activated, it is

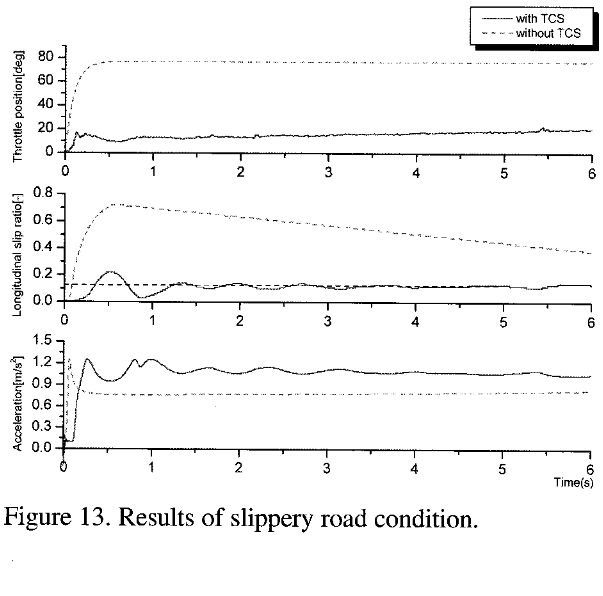

Figure 13. Results of slippery road condition.

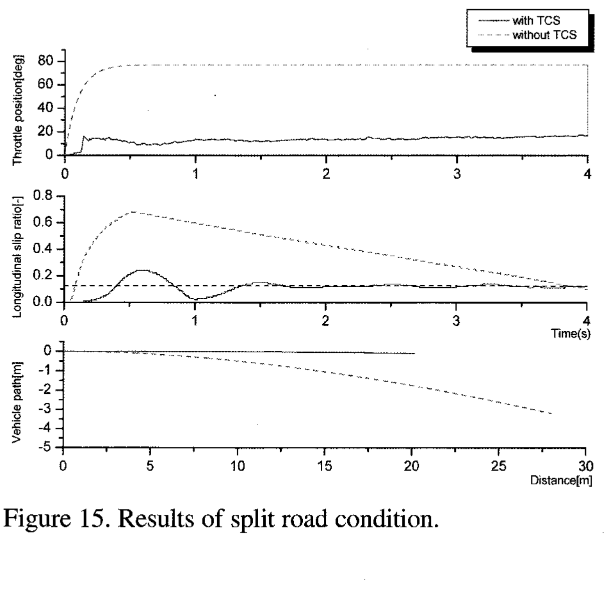

Figure 15. Results of split road condition.

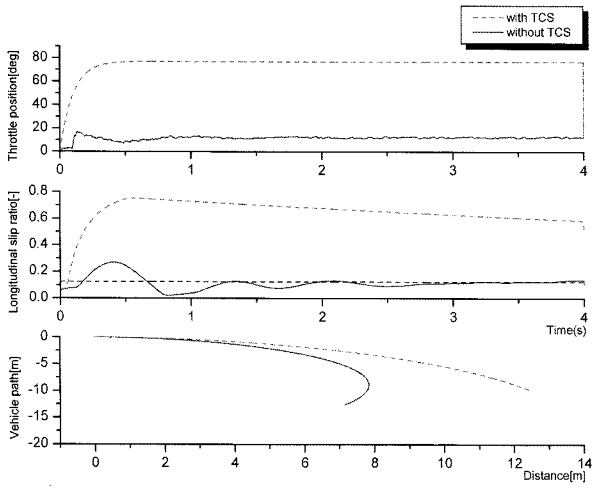

Figure 16. Results for turning in slippery road condition.

possible to guarantee the longitudinal motion ability, in contrast to when the TCS is not activated. This is caused by a difference in traction force of each of the two tractive wheels. Namely, the traction force on the unslippery side is much higher than on the slippery side. Therefore, yaw moment is generated on the vehicle.

6.4. Turning in Slippery Road Conditions

Figure 16 shows the results of the simulation of turning in slippery road conditions. Similar to the results above, throttle position is regulated between 10° and 20° and the slip ratio tracks the target ratio well, shown in the first and second graphs of Figure 16.

In addition, for the case in which the network-based TCS is deactivated, under-steer motion is described with a dashed-line in the third graph. However, when the network-based TCS is activated, the vehicle does not stray from the intended curving path; therefore, the under-steer motion can be kept off. These results come from the changes in side force generated by the tires. Side force is increased with decreasing increments of the wheel slip ratio.

7. CONCLUSION

In this paper, we present the development of a network-based TCS considering the network protocol (CAN). In addition, we propose a new concept of HILS, Net-HILS, for developing network-based control systems that include smart sensors or smart actuators. Using this novel development environment, we have developed the network-based TCS.

Therefore, Using Net-HILS, we were able to efficiently develop the network-based TCS without a wiring harshness as the prototype previous to the proving ground test and its algorithm and performance could be validated. As a result, the network-based TCS performed well in guaranteeing the acceleration and stability of a vehicle.

ACKNOWLEDGEMENT–This research is supported in part by MOST (Ministry of Science and Technology) under the National Research Laboratory program, and part of the project 'Development of Partial Zero Emission Technology for Future Vehicle'. We are grateful for their financial support.

REFERENCES

Bakker, E., Pacejka, H. B. and Lidner, L. (1989). A new tire model with an application in vehicle dynamics studies. *SAE Paper No.* 890087. 439–451.

Jo, H. Y., Lee, U. K. and Kam, M. S. (2006). Development of the independent-type steer-by-wire system using HILS. *Int. J. Automotive Technology* **7, 3,** 321–327.

Kang, S. M., Yoon, M., Sunwoo, M. (2004). Engine control TCS using throttle angle control and estimated load torque. *Trans. Korean Society of Automotive Engineers* **12, 2,** 139–147.

Lawrenz, W. (1997). *CAN System Engineering from Theory to Practical Applications.* Springer. Germany. 26–39.

Lee, W. T., Yoon, M., and Sunwoo, M. (2003). A cost- and time-effective hardware-in-the-loop simulation platform for automotive engine control system. *Proc. Institution of Mechanical Engineers* **217**, 41–52.

Opal-RT Technologies Inc. (2000). *RT-EVENT User's Guide.* USA.

Ryu, J. H., Noh, K. H., Kim, J. H. and Kim, H. S. (1999). Development of a steering HILS system. *Trans. Korean Society of Automotive Engineers* **7, 9**, 105–111.

Slotine, J. J. E. and Li, W. (1991). *Appiled Nonlinear Control.* Prentice-Hall. New Jersey. 276–310.

Smith, D. E. and Starkey, J. M. (1995). Effects of model complexity on the performance of automated vehicle steering controllers: Model development, validation, and comparison. *Vehicle System Dynamics*, **24**, 163–181.

Yoon, M., Lee, W. and Sunwoo, M. (2005). Development and implementation of distributed hardware-in-the loop simulator for automotive engine control system. *Int. J. Automotive Technology* **6, 2,** 102–117.

Yoon, P. J. and Sunwoo, M. (1999). A nonlinear dynamic engine modeling for controller design. *Trans. Korean Society of Automotive Engineers* **7, 7,** 167–180.

Zhang, J. M. and Wang, S. Q. (2002). Networked control system design and implementation. *Proc. 1st Int. Conf. Maching Learning and Cybernetics*, 750–753, Beijing.