International Journal of the Korean Society of Precision Engineering Vol. 2, No. 4, November 2001.

A Driving Simulator of Construction Vehicles

Kwon Son[1], Sang-Hwa Goo[2], Kyung-Hyun Choi[3], Wan-Suk Yoo[1], Min-Cheol Lee[1] and Jang-Myung Lee[4]

[1] School of Mechanical Engineering, Pusan National University, Pusan, Korea
[2] Agency for Defence Development, Changwon Proving Ground, Changwon, Korea
[3] School of Mechanical Engineering, Cheju National University, Cheju, Korea
[4] School of Electrical and Computer Engineering, Pusan National University, Pusan, Korea

ABSTRACT

Vehicle driving simulators have been used in the development and modification of models. A simulator can reduce cost and time through a variety of driving simulations in the laboratory. Recently, driving simulators have begun to proliferate in the automotive industry and the associated research community. This paper presents the hardware and software developed for a driving simulator of construction vehicles. This effort involves the real-time dynamic analysis of wheel-type excavator, the design and manufacturing of the Stewart platform, an integrated control system of the platform, and three-dimensional graphic modeling of the driving environments.

Keywords: Driving simulator, Stewart platform, Dynamic analysis, Driving environment

1. Introduction

A simulator plays an important role in developing and evaluating various kinds of vehicles. The vehicle driving simulator is a virtual reality device from which a driver can obtain realistic driving feelings through movements, visual displays, and sound effects in virtual driving situations by implementing real-time simulation over the vehicle. Use of simulators enables researchers to try in advance easier evaluations of a vehicle model by reproducing real driving situations in a laboratory. The driving simulator has been widely used to evaluate how design variables and changes in driving situation affect the driving performance and ride comfort. The simulator becomes an essential equipment in the development of new models, as it is impossible to implement all necessary experimentations using a new vehicle model whenever model changes are required at each development stage. Moreover, the simulator is indispensible to the test for such driver's critical situations as in a collision, a crash and a sliding.

Major vehicle manufacturers in the advanced countries have designed and utilized their own vehicle simulators for virtual tests of newer models[1-5]. The leading companies have developed subsystem models of an engine, a transmission, an actuator, a suspension and steering device, and a dynamic model, as well as driving resistance models testing rolling resistance and air resistance in a variety of driving situation cases. These companies have developed virtual driving environments and specific topographical models for terrain tests using a high performance parallel- processed computer or a group of server computers for real-time graphics and dynamics. More than three image generators are being used for high resolution graphic projection. Motion platforms of large capacity, handling 8,000 kg to 18,000 kg of payload, have been installed for movement simulations. These simulators can be used to investigate driver's motion in various situations, to develop a newer vehicle or its components, to study effects of the road and traffic environments, etc. Though simulators can be employed for a wide range of purposes, they can not be afforded to general researchers. Since high cost blocks

an easier access to a simulator, it is necessary to develop a more affordable simulator even though its application fields are somehow limited.

A few companies in Korea, on the other hand, possess and attempt to use some simulators mainly imported from abroad. Some of the research centers and universities have tried to study over vehicle simulators in recent years. Their researches are restricted in certain fields, therefore, more intensive studies are required in order to put them into practical use. This study presents a driving simulator which provides a virtual driver with driving situations of a wheel-type excavator as the first step into a construction vehicle simulator. A distinct method using the velocity transformation is applied to real-time calculations in vehicle dynamics and a 6-DOF hydraulic stewart platform, capable of a load up to 500 kg, was designed and installed to reproduce movements of the excavator. A graphic program using a public software OpenGL was coded to give the driver a driving velocity perception and to easily display a driving situation. The whole system can effectively be controlled using an integrated control system which was developed in order to evaluate the simulator itself, related hardware components and graphic softwares.

2. Simulation and Modeling

The driving simulator consists of the Stewart platform, a hydraulic compressor, a controller, an image projector, and computers as shown in Fig. 1. The Stewart platform developed in this research has six degrees of freedom parallel kinematics and is operated by the six hydraulic cylinders whose maximum strokes are 500 mm. On the Stewart platform a cap of the real excavator is installed and used as a driving room. For the controller, an 80C196KC microprocessor which has high speed input/output ports is utilized. This Intel 16 bits microprocessor has advantages of low power-consumption with high performance and has memories for input and output ports. Through the input ports, the microprocessor receives the digital data from the manipulation part of the simulator, and according to the input signals, it sends output signals through the output ports to control the cylinders. There are two major memories: the program memory which governs the global motion states of the simulator and the data memory which stores the data from the individual sens

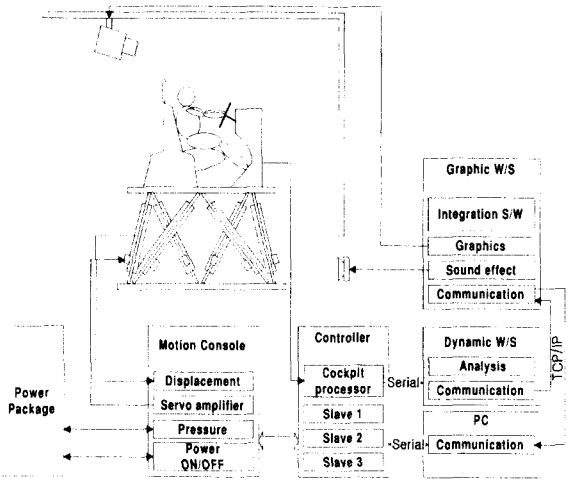

Fig. 1 Driving simulator of construction vehicles

ors. A dynamic workstation analyzes the driving states using a simplified model for the real-time analysis while the degrees of the freedom for the tire model and the excavator remain high enough to duplicate the real system. A graphic workstation executes the integrated programs to operate the simulator while the computer performs its own graphic processing of the excavator and surrounding environments.

2.1 Vehicle dynamics

For a real-time simulation which is essential to the simulator, the computer modeling of the excavator is simplified without degrading the motion. Since a well developed commercial program like DADS or ADAMS are applicable to general problems, it may take longer simulation time as a special program suitable to simulators. Therefore, a special program using velocity transformation technique is developed in this research and used for dynamic simulation of the excavator.

The velocity transformation technique for dynamic simulation combines the generality of the absolute coordinates and the efficiency of relative coordinates[6]. The technique transforms the relative velocities to the absolute velocities via a velocity transformation matrix.

The equations of motion using the absolute coordinates can be written as[6]

$$M\ddot{y} + \Phi_y^T \lambda = Q \qquad (1)$$

where y, M, Φ_y and Q are the absolute coordinate vector, the mass matrix in terms of the absolute coordinates, the Jacobian matrix of the system and the generalized force vector, respectively. Using the velocity transformation matrix B, the equations of motion in terms of the relative coordinates q can easily be transformed to the following equations:

$$\overline{M}\dot{q} + \Phi_y^T\lambda = \overline{g} \quad (2)$$

$$\overline{M} = B^T M \quad (3)$$

$$\overline{g} = B^T(f - M\dot{B}\dot{q} - h) \quad (4)$$

where $\overline{M}$ and $\overline{g}$ are the generalized mass matrix and the modified generalized force vector, respectively.

For a closed loop, constraints should be composed of cut-joint constraints and driving constraints. The driving constraints are the imposed constraints to drive the system. When a driving constraint has a complicate expression in terms of the absolute coordinates, it is sometimes difficult to transform that driving condition in terms of the relative coordinates. The driving condition for the wheel excavator in this research, however, was a simple one for the longitudinal motion, and thus it was easily transformed in the relative coordinates.

The driving constraint in terms of the relative coordinates can be written as[7]

$$\Phi^d \equiv \Phi^d(q, t) = 0 \quad (5)$$

where the superscript d means driving. Differentiating equation (5) with respect to time gives the following relations:

$$\Phi_q^d \dot{q} = -\Phi_t^d \quad (6)$$

$$\Phi_q^d \ddot{q} = -(\Phi_q^d \dot{q})_q \dot{q} + 2\Phi_{qt}^d \dot{q} + \Phi_{tt}^d \equiv \gamma \quad (7)$$

where γ and Φ_q^d mean the right-hand side term of the acceleration equation and the Jacobian matrix of the constraint equations, respectively. Therefore, the final equations of motion including the constraint equations can be written as

$$\begin{bmatrix} B^T M B & \Phi_q^{dT} \\ \Phi_q^d & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q \\ \gamma \end{bmatrix} \quad (8)$$

The excavator model used here contains a chassis, a front axle, and four tires. Three revolute joints and two universal joints between bodies make the excavator a 13 degrees of freedom system. Using the program developed, the dynamic responses are obtained and compared to those calculated by DADS program. The two sets of results were in a good agreement and, moreover, the computational times can be reduced up to 1/10 of the commercial program in the J-turn simulation and the lane change simulation[8].

2.2 Control of motion platform

Motions of the excavator should be transformed to those in a link coordinate system in order to move the Stewart platform as obtained in the space coordinate system by the vehicle dynamics. For this reason the inverse kinematics was used to calculate input values of link strokes and a controller was designed to control the cylinder strokes.

The coordinate system of the Stewart platform is depicted in a simple geometric form as shown in Fig. 2. The coordinate systems {P} and {B} have their origins at the centroids of the upper and the base plates, respectively. Six connecting points on the upper and lower plates are denoted by p_i and b_i ($i=1, \cdots, 6$)

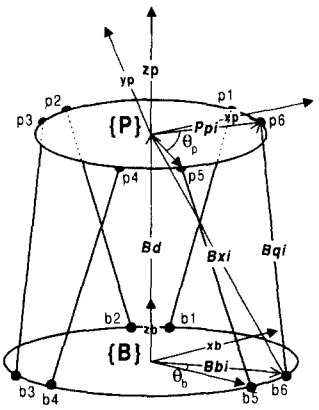

Fig. 2 Coordinate systems of Stewart platform

The motion of the payload plate can be described with respect to the base plate. Vector B_d pointing the origin of {P} from that of {B} and vector B_{qi} denoting connecting cylinders are expressed as

$$B_d=[x \quad y \quad z]^T \tag{9}$$

$$B_{qi}=[q_{ix} \quad q_{iy} \quad q_{iz}]^T \tag{10}$$

$$B_{xi}=B_{qi}-B_{pi} \tag{11}$$

where B_{xi} denotes the distance vector from point b_i to the origin of {P} which is fixed on the upper plate, and B_{pi} denotes vector P_{pi} expressed in the coordinate system {B}. Vector B_{xi} can be written as

$$B_{xi}=B_d-B_{bi} = \begin{bmatrix} x-b_{ix} \\ y-b_{iy} \\ z-b_{iz} \end{bmatrix} = \begin{bmatrix} x-b_{ix} \\ y-b_{iy} \\ z \end{bmatrix} = \begin{bmatrix} \overline{x_i} \\ \overline{y_i} \\ \overline{z_i} \end{bmatrix} \tag{12}$$

and vector B_{pi} can be expressed as

$$B_{pi}={}^P_B R \; P_{pi} \tag{13}$$
$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{bmatrix} = \begin{bmatrix} r_{11}p_{ix}+r_{12}p_{iy} \\ r_{21}p_{ix}+r_{22}p_{iy} \\ r_{31}p_{ix}+r_{32}p_{iy} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}$$

where matrix ${}^P_B R$ is the transformation matrix representing the direction of the coordinate system {P} with respect to the coordinate system {B}. The transformation matrix can be expressed in the Bryant angles α, β and γ as[8]

$${}^P_B R = \begin{bmatrix} C_\beta C_\gamma & -S_\gamma C_\beta & S_\beta \\ C_\gamma S_\alpha S_\beta + C_\alpha S_\gamma & -S_\alpha S_\gamma S_\beta + C_\alpha C_\gamma & -S_\alpha C_\beta \\ -C_\gamma C_\alpha S_\beta + S_\alpha S_\gamma & C_\alpha S_\beta S_\gamma + S_\alpha C_\gamma & C_\alpha C_\beta \end{bmatrix} \tag{14}$$

where C and S denote cos and sin, respectively.

The cylinder length $l_i (=|B_{qi}|)$ can be obtained using equations (10), (11) and (12) as

$$l_i=\sqrt{q_{ix}^2+q_{iy}^2+q_{iz}^2} = \sqrt{(\overline{x_i}+u_i)^2+(\overline{y_i}+v_i)^2+(\overline{z_i}+w_i)^2} \tag{15}$$

The controller for the platform consists of one main controller and three slave controllers. Each slave controller is assigned to handle a couple of hydraulic cylinders. The main controller manages the setup and the communication. A setup program was developed to start with the functions related to 80C196KC of the main processor of slave controllers and to initialize the hardware. The communication program developed sends and receives data between the main controller and each slave controller.

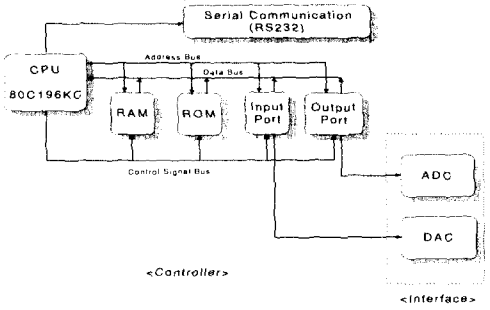

Fig. 3 Slave controller

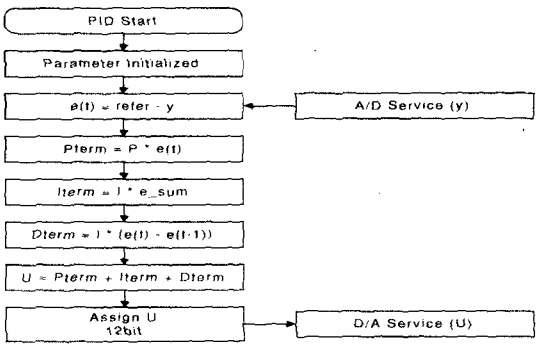

Fig. 4 PID control logic

The processor for a slave controller used was a micro-controller 80C196KC of CHMOS type with 16 bits. The slave controllers consist of input port, output port, and memory as shown in Fig. 3. The input port receives the digital signal transformed from the analog stroke measured from each actuator of the platform.

The output port sends commands for the cylinder stroke to follow the reference input. The memory consists of a program memory for managing the motion and a data memory for storing sensor data.

A PID control algorithm was used to control the cylinder strokes as shown in Fig. 4. The control program was coded using the assembler of the 80C196KC.

2.3 Design of motion platform

The motion of 6 degrees of freedom is necessary to generation of a driving status of the excavator. The Stewart platform used in this study is one of the parallel manipulators capable of three-dimensional rigid body dynamics. The target performance of the Stewart platform was chosen as listed in Table 1 with reference to MIL-STD-1558 specifications for flight simulators[9].

The platform was designed to move the maximum payload of 750 kg based on the mass of the upper plate and the excavator cab to be placed on it. The maximum velocity attainable was set at ± 61 cm/sec in consideration of the safety factor in heave motion. The six hydraulic cylinders were connected to the two plates with their maximum strokes of ± 250 mm. The diameters of each cylinder head and piston rod were selected as 40 mm and 30 mm, respectively, by considering the actuating power and bucking problem of the rod. The hydraulic pressure value was determined on the basis of the maximum payload and the cross-sectional area of the cylinder head. The servo-valve was chosen to have a regular flow rate of 15 gpm considering the maximum flow to lift the platform in the vertical direction. Fig. 5 and Fig. 6 show the assembled Stewart platform and hydraulic unit, respectively.

The workspace of the Stewart platform was identified through a simulation using the relation suggested by Pennock and Kassner[10]. A structure of flexible four links was assumed because it is very difficult to directly derive the workspace of this kind of parallel manipulator[11]. As listed in Table 2, the workspace falls short of the desired requirements except for the roll. The reasons for this might be insufficient cylinder strokes and inadequate spacings of six universal joints on each plate.

2.4 Graphic processing

A graphic display for the simulator has its meaning

Table 1 Motion platform requirements

6 DOF motion	Maximum displacement	Maximum velocity	Maximum acceleration
Roll	20 deg	20 deg/sec	60 deg/sec^2
Pitch	25 deg	20 deg/sec	60 deg/sec^2
Yaw	30 deg	20 deg/sec	60 deg/sec^2
Surge	432 mm	610 mm/sec	0.6 mm/sec^2
Sway	432 mm	610 mm/sec	0.6 mm/sec^2
Heave	432 mm	610 mm/sec	0.6 mm/sec^2

Table 2 Workspace of Stewart platform

Motion	Workspace
Roll	+25 deg ~ -25 deg @ Heave = 0 mm
Pitch	+23 deg ~ -25 deg @ Heave = 0 mm
Yaw	+20 deg ~ -20 deg @ Heave = 0 mm
Surge	+480 mm ~ -380 mm @ Heave = 0 mm
Sway	+400 mm ~ -400 mm @ Heave = 0 mm
Heave	+320 mm ~ -320 mm @ Yaw = 0 deg

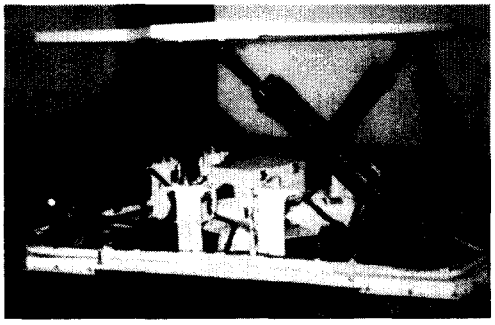

Fig. 5 Stewart platform

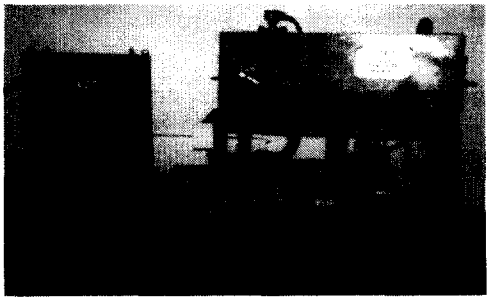

Fig. 6 Hydraulic unit

only if it is processed in real time. As a personal computer is unable to catch up with a larger amount of data and the screen processing speed, a workstation (SGI Indigo2 Impact R4000) is used for graphic processing. An SGI is selected for its excellent graphic treatment. Since the SGI offers graphic library called GL or OpenGL, all graphic objects can be expressed using the library and C compiler provided.

2.4.1 Object modeling

The graphic modeling of objects is largely divided into the simplest wire frame model and the solid model. OpenGL manipulates three-dimensional data, so the wire frame can be easily composed by defining coordinates of vertices and corners in case of a flat surface. The wire frame model composed with only lines is simple and fast in its construction within the limitations of degraded reality. Therefore, a solid model with shading effect is formed for each object modeled in this research.

While the nature of real world is made up of analog signals, the computer monitor deals with digital ones. Therefore, drawing lines and surfaces on the computer may cause an aliasing phenomenon. Aliasing was eliminated for some degrees by controlling surrounding gradation to obtain natural images as in the real world.

GluPerspective function of GLU library was employed to obtain perspective effect. As it is hard to get the perspective effect for distant scenery when a view point is fixed, three-dimensional data processing is not adequate due to its lower efficiency. A two- dimensional texture model is introduced to express distant scenery.

Shading is known as the process which takes most of the time for graphic calculation and it can be obtained using various kinds of algorithm. In addition, speed and resolution in shading depend on the used algorithm. This research employes the soft shading process to gain the light effect and uses continuous values of the surface direction to make each surface look realistic.

2.4.2 Increasing graphics speed

Graphic processing is a part which needs a lot of data and calculations of this program. A huge amount of data and calculations decrease the processing speed. In order to increase the graphics speed, a real-time simulator is programmed as follows. During the process of simulation, objects are displayed continually merely by changing their locations. Three-dimensional models repeatedly presented are stored in the fastest retrieving space of the memory, and then are generated on the computer. The program execution is done by reading the data, storing the model data in memory and then generating a display list. Once a display list is generated, each identification number is given and the program calls out this number to retrieve the data stored. When displaying the model a second time on the screen, the processing speed is increased compared with the first time as the data stored in memory is directly used for graphic processing. This technique of memory use increases the speed varying with the shape of model and the frequency of use. A noticeable speed increase can be observed in case of complicated models and frequent use of the data.

The number of objects to be modeled should be exceedingly large in order to provide the driving simulator with reality. Though the objects within the visual field are to be displayed, the objects out of view must be calculated during the graphic process, which spends much time in processing. This research improves the graphic processing speed by excluding the objects existing but out of the visual field from calculating. Fig. 7 illustrates a driving environment block seen from above. A square block composed with three layers is the region to be searched for the locations of environmental objects. Assume that the excavator is located at point a on the street marked with thick line and the driver's view point is facing to the arrow direction. The triangle abc is the area within the view and only the objects occupying this area are to be displayed as a driving environment at this moment. The basis location of visual field is placed at the center of rectangle adef which includes the three vertices of the triangle; the basis location of the surrounding environment at the center of the environment block.

The basis location of the visual field, V.basis, and the field size, V.size, can be expressed as

$$V.basis = \frac{1}{2}(Max(a, b, c) + Min(a, b, c)) \quad (16)$$

$$V.size = |Max(a, b, c) - Min(a, b, c)| \quad (17)$$

The basis location and the size of driving environment

are fixed in the block and their values are denoted by P.basis and P.size, respectively. Their relation to the visual field can be expressed as

$$|P.basis - V.basis| \leq \frac{1}{2}(P.size + V.size) \quad (18)$$

Inclusion of calculation is determined to see the visual field and the basis location of driving environment satisfy inequality (18). If they fail to satisfy the inequality, those objects which prove to be out of view are excluded from calculation. In order to reduce the time to check the basis location, an inclusion test is performed for the first layer of block having the biggest dimension, then the second layer is used in the test once the objects are included in the first layer, i.e., the largest square shown in Fig. 7. By repeating this way, the size of block becomes smaller. Finally, graphic processing is done for all the smallest blocks surrounded by the rectangle adef in Fig. 7.

2.5 System communication

The overall communication system is structured to operate the Stewart platform as shown in Fig. 8. Dynamics are calculated by the dynamics workstation and are transferred to the graphic workstation, where the Stewart platform images are displayed in real time. The communication media for this channel is an ethernet-based LAN. At the same time, the data for the cylinder strokes and gains of the PID control algorithm are sent to the personal computer through the hub. For the communication between the personal computer and the controller, an enhanced parallel port (EPP) mode which provides bi-directional parallel communication is implemented as in Fig. 9.

The controller is designed for controlling the manipulator according to the control parameters determined by the control loop. In the controller, the multi-process communication service is designed for the communication among the master and slave processes with the polling-allocation to prevent the collision and to allow the independent control. The asynchronous multi-drop serial communication is adapted for the internal communication of the controller, that is, the communication between the master controller and the slaves 1, 2 and 3.

The processor of master controller generates an interrupt whenever it receives a signal in mode 3, while the processor of the slave receives addresses in mode 2 and changes to mode 3 when they are recognized as for itself and continues the communication in the polling-allocation. The advantages of this communication scheme are the feasibility in writing the communication program, availability of various algorithms, and the feasibility in changing the program as needed to control the manipulator.

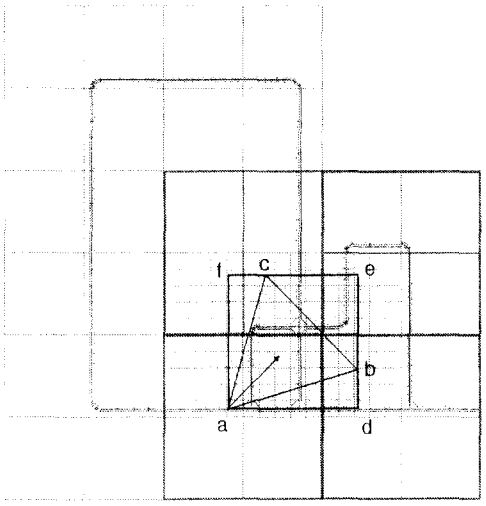

Fig. 7 Driving environment block

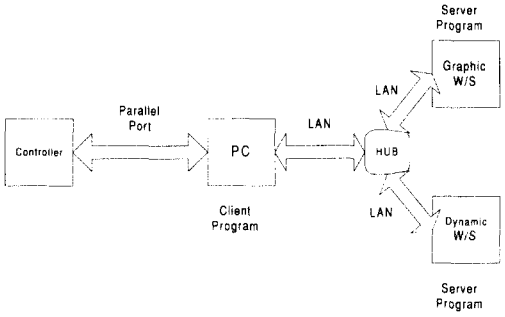

Fig. 8 Organization of communication system

3. Integrated program

An integrated program is developed and used to operate the simulator in three distinct ways as shown in Fig. 10.

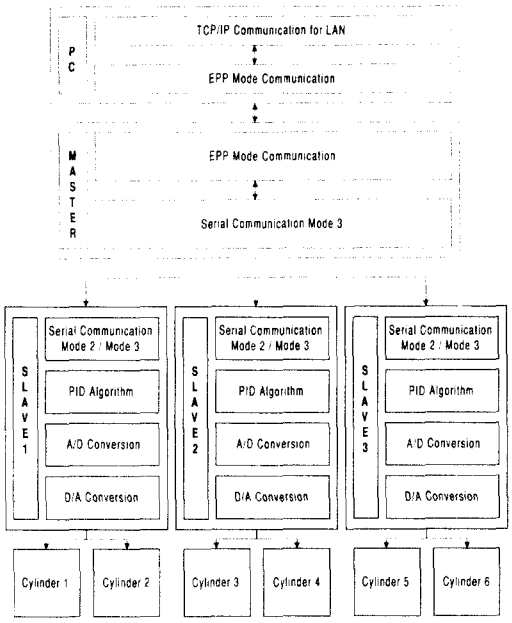

Fig. 9 Structure of communication network

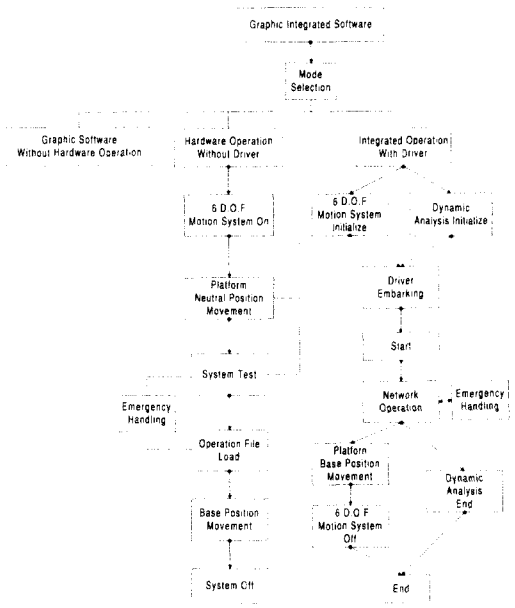

Fig. 10 Tree of integration program

The program displays the graphic models of driving environments and excavator itself in an optimum fashion according to each special purpose. The three ways include the graphic simulation mode without hardware operation, the hardware operation mode without driver, and the integrated operation mode with a driver. For safety purpose, users are recommended to run the simulator in the second mode, hardware operation mode without driver, before they test the integrated operation with a driver.

The integrated program is coded using Motif in Windows environment so that users can easily execute the program just by clicking icons or menu bars without using any keyboard. Motif is one of the object-oriented program libraries, which consists of various functions that can be run in X-Windows environment. Motif has excellent compatability to hardware environment so that it may be compatible with X and Xt libraries. Graphics can be exhibited using OpenGL in a drawing region generated on Motif window.

3.1 Pure graphic simulation mode

The pure graphic simulation mode can be chosen for a graphic simulator without any hardware operation. Two screens are basically generated in this mode: the one is for showing driving environment as in Fig. 11 and the other is for interfacing with the driver as in Fig. 12. Data obtained from the dynamic analysis are used for graphic simulations in this mode.

3.2 Hardware operation mode without driver

This mode is used for moving the Stewart platform in order to evaluate its posture and motion. The platform movement is displayed on the monitor or projected on the screen in this mode based the calculated dynamics of the platform. As shown in Fig. 13, setting of PID gains and cylinder strokes, controlling and testing of the hardware can be done on the computer with graphical representations. The values of PID gains and cylinder strokes can be input by either text or file format. Buttons for easier editing the text are also provided to save the time to set those input values.

3.3 Integrated operation mode with driver

This mode manages the networking among hardwares and two computers independently used for graphics and for dynamics. Users select in this mode a window to obtain scenes to be displayed from a driver's viewpoint or another to manage the integrated operation of the simulator. In the window for managing the integrated

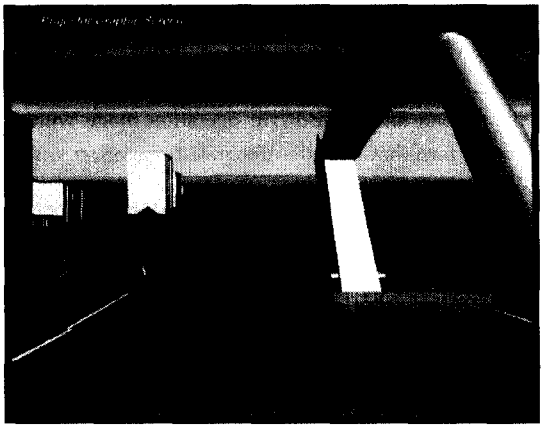

Fig. 11 Screen of driving environment

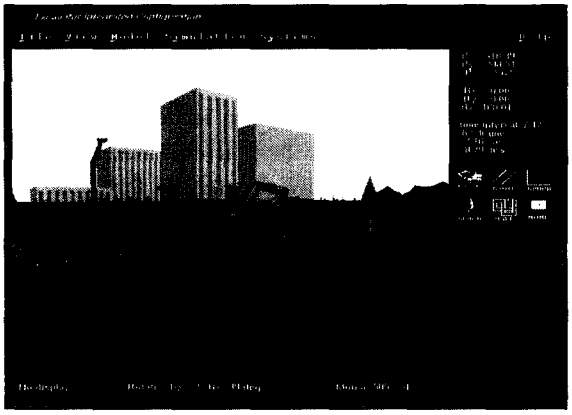

Fig. 12 Screen for driver interface

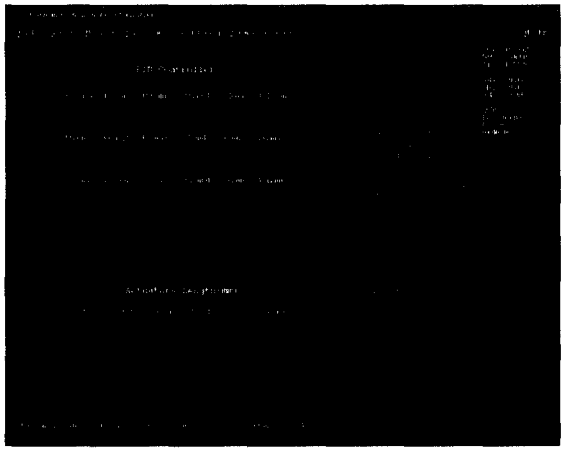

Fig. 13 Setting of PID gains and cylinder strokes

operation, widgets of various functions are formed by allocation or release of memory. A widget either for text or graph can be generated by the users' selection.

The dynamic analysis results in three rectilinear displacement components and three angles of the Stewart platform on which a driver's seat of the excavator is placed. Each of six cylinder strokes can then be calculated through the inverse kinematics. On the computer used for graphics, graphical displays of the excavator and Stewart platform are obtained based on the data obtained by the dynamic analysis.

To simulate driving the excavator using the Stewart platform with a limited workspace, both synchronization and coupling are needed between the graphic display and platform motion. For this reason, a washout algorithm is designed so that the platform may take charge of driving accelerations while the graphic display takes charge of driving speeds. The graphics computer not only sends the values of actuator strokes to the controller but also makes a corresponding graphic frame of the driving environment to the simultaneous dynamic analysis.

3.4 Simulation results

The results stored are arranged in a graphic form after each completion of a driving simulation. Results obtained from the dynamic analysis are saved to a file so that any data to be throughly investigated can be exhibited graphically from this file. Fig. 14 shows one of the graphs showing the vertical acceleration of the center of mass of the excavator. Fig. 15 shows how to use a pop-up menu with this kind of graph displayed on the monitor. The pop-up menu is activated by clicking the right button of the mouse. A desired item and an axis are selected as shown in this pop-up menu. The selection of the axis is followed by the selection of the desired item, then the program automatically finds out the corresponding maximum and minimum values so as to make a graph in a proper scale. In this mode users can examine the time history of each component of displacement, velocity, and acceleration of the mass center of the excavator or the Stewart platform, and the time history of difference between the input and output values of each cylinder stroke.

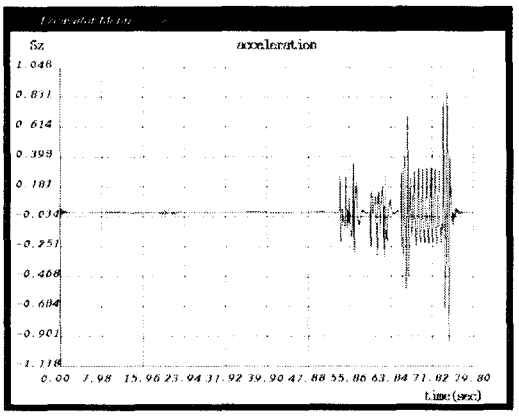

Fig. 14 Displacement of z-direction

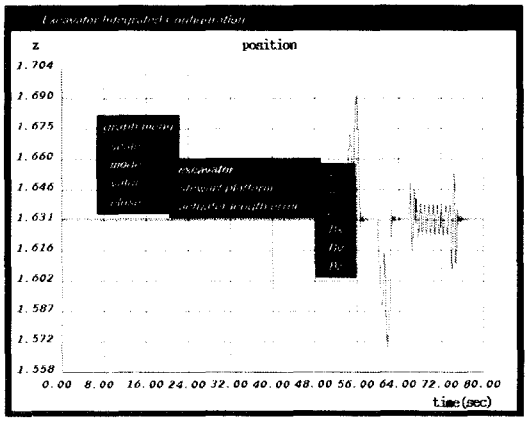

Fig. 15 Menus for graph setting

4. Conclusions

This study presents the hardware and software developed for a driving simulator of construction vehicles and draws the following conclusions:

(1) An effective dynamic analysis is suggested by applying the velocity-transformation method to a simplified excavator model. The proposed approach can reduce the computational time to approximately one-tenth of the case of the general commercial program.

(2) The platform designed and installed has its maximum stroke of 320 mm and maximum rotation of 20 degrees and can move a payload of up to 500 kg using the hydraulic power.

(3) LAN is used in communication among the two workstations for dynamics and graphics, and a personal computer. Communication between PC and controller is embodied as either-way parallel type to which EPP mode is applied.

(4) The shading model, antialiasing technique, and perspective effect are used for a realistic graphic model. Also speed increase in graphics can be attained by accessing computer memories, and removing objects through basis location tests.

(5) The integrated program consists of three modes: the pure graphic simulation mode, the hardware operation mode without driver and the integrated operation mode with driver.

Driving fidelity of the motion platform is being improved by experiments based on the evaluation of driving feeling through a human sensibility ergonomics approach. Analyses of the relationships between platform components and sensory expressions can enhance the reality in simulation using the platform. For more effective simulators, further studies will be done to find the fastest algorithm for vehicle dynamics analysis, to include specific construction tasks, and to develop real-time graphics processing schemes and to devise more robust control algorithms.

Acknowledgement

This research was funded by Korea Science and Engineering Foundation (Basic Research Grant No. 97-0200-1001-5) and School of Mechanical Engineering at Pusan National University.

References

1. Käding, W., and Hoffmeyer, F., "The Advanced Daimler-Benz Driving Simulator," SAE Paper 950175, 1995.
2. Bartollini, B., "The General Motors Driving Simulator," SAE Paper 940176, 1994.
3. Greenberg, J. A., and Park, T. J., "The Ford Driving Simulator," SAE Paper 940179, 1994.
4. Repa, B., and Wierwille, W., "Driver Performance in Controlling a Driving Simulator with Varying Vehicle Response Characteristics," SAE Paper 760779, 1976.
5. Freeman, J. S., "The Iowa Driving Simulator: An Implementation and Application Overview," SAE Paper 950174, 1995.
6. Kim, S. S., and Vanderploeg, M. J., A State Space

Formulation for Multibody Dynamic Systems Subject to Control, The University of Iowa Technical Report, No. 84-20, 1984.

7. Nikravesh, P. E. Computer-Aided Analysis of Mechanical Systems, Prentice-Hall, 1988.
8. Kim, K. S., Yoo, W. S., Lee, M. C., Son, K., Lee, J. M., Choi, D. H., Park M. G., and Park, H. H., "Suggestion of Cutoff Frequency in the Washout Filter for a Wheel Type Excavator," Journal of the Korean Society of Precision Engineering, Vol. 16, No. 5, pp. 19-28, 1999.
9. Six Degree-of-Freedom Motion System Require- ments for Aircrew Member Training Simulators, MIL-STD-1558, Department of Defense, 1974.
10. Pennock, G. R. and Kassner, D. J., "The Workspace of a General Geometry Planar Three-Degree-of-Freedom Platform-Type Manipulator," ASME J. of Mechanical Design, Vol. 115, pp. 269-276, 1993.
11. Goo, S. and Son, K. "Study on Forward Kinematics of Stewart Platform Using Neural Network Algorithm together with Newton-Raphson Method," Trans. of Korea Society of Automotive Engineers, Vol. 9, No. 1, pp. 156-162, 2001.