

A PC-based Driving Simulator Using Virtual Reality Technology

H. S. Kang

Faculty of Mechanical Engineering
Universiti Teknologi Malaysia

M. K. Abdul Jalil*

Faculty of Mechanical Engineering
Universiti Teknologi Malaysia

Musa Mailah

Faculty of Mechanical Engineering
Universiti Teknologi Malaysia

Abstract

Driving simulators are frequently used for vehicle system development, human factor study, and vehicle safety research by enabling the reproduction of the actual driving environments in a safe and tightly controlled environment. Recent development of powerful desktop computer workstations can now support the computational requirement of small to medium-scale driving simulators. The driving simulator for this research is developed using PC-based workstations that are capable of producing high fidelity graphics at reasonable cost. The VR-based simulator gives a driver on board the impression that he drives an actual vehicle by predicting vehicle motion caused by the driver input and feeding back the corresponding visual, motion, audio and proprioceptive cues to the driver. This research is intended to provide a test bed for simulating driving related task using virtual reality technology.

Keywords: Real-Time Vehicle Simulation, Vehicle Dynamic Model, Visual Database.

1 Introduction

The term 'Virtual Reality' or VR has been used to describe a wide range of applications in computer graphics and human-computer interaction, some of which have little connection with the term's original meaning. A virtual reality, or as we normally prefer, a virtual environment, is a system which allows a user to feel immersed in a computer-generated environment. VR technology has been widely used in engineering and scientific visualization due to its ability to create a life-like simulation.

A virtual driving simulator is a virtual reality device that allows its user to feel a life-like experience of driving an actual vehicle. The driving simulator is effectively used for studying the interaction of a driver and vehicle and for developing new vehicle systems.

During the design of an automobile, detailed mockups of the interior are built to study the design and evaluate the ergonomic

*e-mail: kasim@fkm.utm.my

issues. However, these physical prototypes are expensive, time consuming, and difficult to modify. It is more effective to use a virtual prototype replacing the physical mockup for the design and analysis aspects such as layout and packing efficiency, visibility of instruments, controls and mirrors, reachability and accessibility. Thus, a driving simulator is needed to enable the analysis on driving characteristics, and interaction between visual database and vehicles. Recent development of powerful desktop computer workstations can now support the computational requirement of small to medium-scale driving simulators.

Many factors should be carefully considered in developing and applying full-scale driving simulators effectively, such as constructions costs, application areas and target performance. A prototype simulator, down-scaled, yet consisting of all the necessary subsystems can be used effectively for evaluation of the factors mentioned above and design of the full-scale simulators.

The driving simulator for this research is developed using PC-based workstations that are capable of producing high fidelity graphics at reasonable cost. The VR-based simulator gives a driver on board the impression that he drives an actual vehicle by predicting vehicle motion caused by driver input and feeding back the corresponding visual, audio and proprioceptive cues to the driver. This research project provides the initial groundwork for future research development of driving simulator in Malaysia. The long-term objective of this research is to provide a test bed for simulating driving related task and for virtual environment technology research in the country. This paper describes a first research attempt to develop a static base driving simulator based on virtual environment technology to create an immersive driving environment for the simulator user. The knowledge gained from this project will be further used to develop a full-scale driving simulator for automotive and transport safety research in Malaysia.

2 Related Work

Research related to the development and application of driving simulator is not new. Driving simulators, having their roots on flight simulators in early 1900s [Repa and Wierwille 1976], have begun to appear in primitive forms in the 1970s. With the advent of computer technologies, Daimler-Benz launched a high-fidelity driving simulator in 1980s [Drosdol and Panik 1985], which created wide interests throughout the world. Since then, many automotive makers and research institutions worldwide have developed their own simulators.

There are various level of complexity of driving simulators ranging from a simple static base simulator to the most advanced simulator that are capable of simulating the dynamic motion and scenes of an actual vehicle. Various institutions and companies have developed driving simulators for research purposes including National Advance Driving Simulator (NADS), Leeds Advanced Driving Simulator (LADS) [Bailey et al. 1999], and

Copyright © 2004 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.
© 2004 ACM 1-58113-884-9/04/0006 \$5.00

the Swedish National Road and Transport Research Institute. A typical advanced driving simulator, such as NADS in Iowa, normally consists of a visual system, angular and translational motion systems with 6 DOF, vehicle cab system, control feel system, auditory system, and vehicle dynamics. An advanced simulator is capable of producing a high fidelity simulation output. However the construction cost of this system is very high. On the other hand, a static base simulator is a low cost and may be sufficient for applications that do not require very high-fidelity simulation. There are also companies that are involved in construction and supply of driving simulators such as AutoSim™ from Norway. The only setback of a ready-made system is that it may not fulfill a very specific research requirement that needs high level of equipment customization. Therefore, for fundamental research purposes, developing a simulator from scratch is much preferred. The VR-based driving simulator developed in this research will lay the groundwork for future research in transport safety, vehicle design and ergonomic evaluation, intelligent vehicle highway systems and driver education and testing.

3 System Configuration

The setup of the driving simulator developed in this research is shown in Figure 1. The simulator consists of a static base driving cab with projected visual on a wide screen display to simulate the driving environment, and the visual and audio database. The simulator is controlled and operated on two personal computers connected by Ethernet. The driver on board the simulator controls the driving tasks using steering wheel, brake pedal and accelerator. These controllers feed signals to the simulation system using TCP/IP network protocol. The first PC acts as the graphic rendering engine and the second PC manages the signals transfer between the rendering engine and the simulation controllers, i.e. the steering wheel, the accelerator and the brake pedals. The distribution of the tasks into two different PC's is necessary to ensure that the driving simulation be done in real-time.

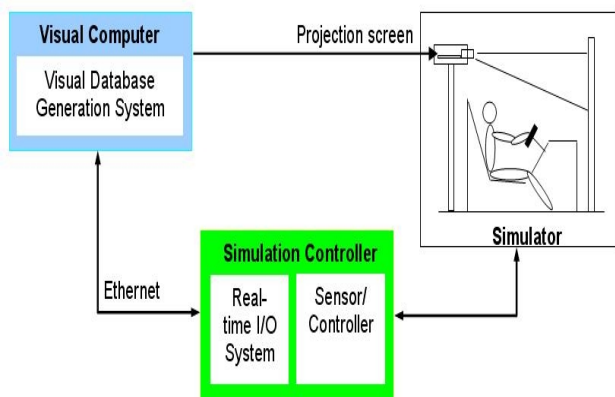


Figure 1 Functional Diagram of the Static Simulator

The cockpit or the driving cab consists of a half-cut car to give the simulator user the actual feel of driving an actual vehicle. The simulation controller consists of an acceleration pedal, a brake pedal and a steering wheel that are connected to sensors to detect the change of signals and feed them to a PC that acts as a server. These signals are then processed and fed back to the

simulation environment. Two PC's are used to handle the entire task to facilitate real-time interaction between the driving simulator and the visual database system. The visual scenes are projected to a wide-angle screen using two LCD projectors that are connected to the second PC that controls the visual rendering. The visual scenes can also be run on a two flat panel LCD monitors. Stereographic glasses can also be used to immerse the simulator user in a virtual environment to produce a sense of realism while driving in the virtual scenes. Figure 2 illustrates the hardware system of driving simulator cockpit.



Figure 2 The Driving Simulator Cockpit

4 System Architecture

Figure 3 shows the system architecture of the virtual reality based driving simulator. The driving simulator consists of a simulator cockpit, client-server network system, and driver interface. Three sensors are mounted on the steering wheel, the accelerator, and the brake pedal of the simulator cockpit. The input data to the I/O data acquisition board are the steering angle, δ_w , throttle displacement, δ_T and brake pedal displacement, δ_B respectively. The NI-DAQ PCI 6024E I/O adapter board receives these data and passes them to a MATLAB program in server computer that computes the vehicle dynamic model output such as the vehicle's surge (X_o), sway (Y_o), yaw (ϵ_o) and longitudinal velocity (v_o). These output data generated from the vehicle dynamic model are sent to the client computer using TCP/IP socket. The visual database of the virtual driving environment, which is located on the client computer, was developed using WorldToolKit (WTK) programming language. It consists of functions written in C language that makes the process of linking the visual database and the network system easy. The output data from the vehicle dynamic model computed by MATLAB are used to simulate the driving actions in the virtual environment.

The visual database developed for this project consists of a flat terrain roadmap of Universiti Teknologi Malaysia (UTM) campus. The user of the simulator is able to drive in the virtual environment by controlling the steering wheel, the accelerator and the brake paddle.

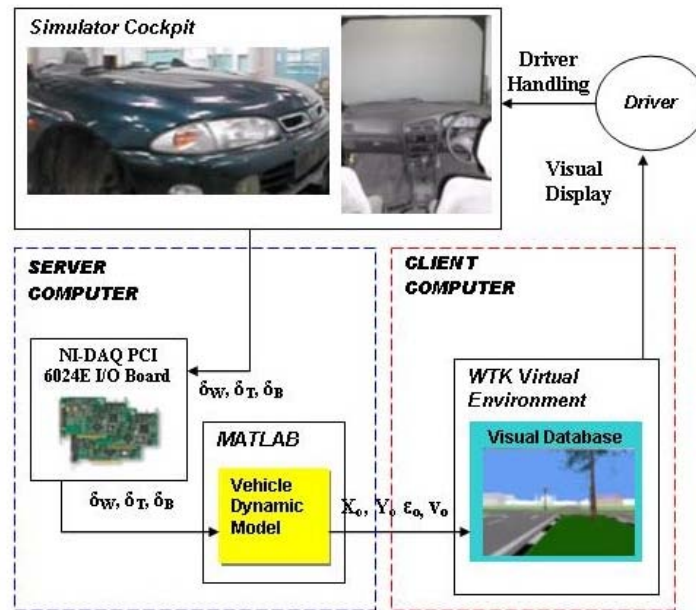


Figure 3 System Architecture of the Simulator

5 Visual Database

The visual database was developed using PC based rendering engine. The visual database system generates high fidelity driving scenes that can be displayed on multiple display systems that simulate the driver's view when a vehicle is in motion. The visual system is composed of a high-speed graphics accelerator running on Intel Pentium 4, 2 GHz processor. The virtual driving scenes for this project are based on the roadmap and landscape of Universiti Teknologi Malaysia. The virtual driving scenes were generated using WorldToolKit (WTK) software that consists of functions written in C language that supports virtual environment development and control. The toolkit includes facilities for real-time rendering and animation, and extensive functions for the creation and management of 3D graphic objects and external sensors. The visual scene rendering includes full texturing, shading, fog, and lighting effects to create a realistic driving environment. The visual system is capable of producing high-resolution simulation environment spread across two computers or projected screens. A set of stereographic glasses can be used to view the visual scene to give user the sense of immersiveness while driving in the virtual environment. Figure 4 shows a sample of driving scenes developed in this research.

The computer-generated environment or visual scene can be divided into three parts: static universe, dynamic objects, and the interior of the driver's vehicle. The static environment includes roads, trees, buildings, and other elements that remain stationary during the simulation. The dynamic objects can be controlled independently and can react to the motion of the user's vehicle such as traffic lights. The virtual driving environment is controlled using a steering wheel, accelerator and brake pedal that provide the driver's input to the simulator.

The models used in developing the visual database were created using AutoCAD, 3D Studio and neutral file formats (NFF), which are readable by WorldToolkit (WTK). These simulation models, as shown in Figure 4, were then arranged in a

hierarchical order in WTK environment, which is called scene graph. The scene graph is the structure that holds all of the current elements of the scene, such as geometries, lights, fog, and positional information. Figure 5 illustrates the structure of the scene graph in the visual database.



Figure 4 The UTM Campus Virtual Driving Scenes

The universe refers to the created virtual world in WTK. It contains all the WTK objects including geometries, sensors, lights, viewpoints, serial ports, and paths. Under the universe, there is a scene graph ancestry known as root node. The scene graph is an ordered collection of nodes, in the form of a directed acyclic graph, which holds hierarchical scene data.

The scene graph of the virtual driving simulator consists of three elements under the root node, namely the light nodes, geometry nodes, and transformation nodes. The light nodes contain the

lighting information to illuminate some or all of the geometries in virtual scene. The geometry nodes include static geometries files loaded into WTK using WTK's file import function and dynamic geometries created within WTK at the polygon and vertex levels. The transformation nodes contain positional information associated with geometries and lights read in from a file, and dynamic positional information created and managed within WTK. It describes where particular elements (geometries and lights) should be placed in the scene, either in relation to another object, or in relation to the scene as a whole. The scene graph provides a very powerful scene structure for real-time 3D simulation. Specifically, scene graph provides the hierarchical framework for easily grouping objects together spatially using parent-child relationship. This is essential for maintaining performance in scenes that contain many individual objects. Because objects can be grouped together in a positional hierarchy, the scene graph can be used to easily construct and maintain efficient simulations that contain individual moving parts.

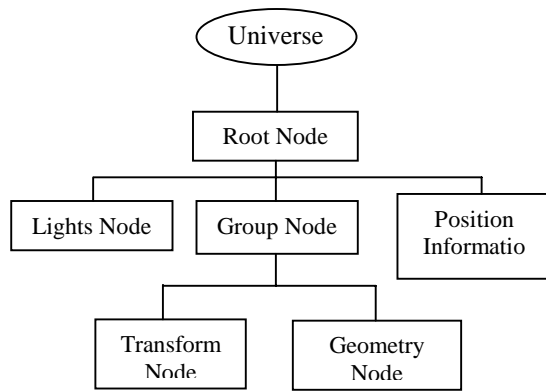


Figure 5 The Structure of Scene Graph

The scene graph enhances performance of the WTK's rendering stage (drawing the scene) because it facilitates spatial culling of the scene. In other words, WTK calculates which parts of the scene (or scene graph) are visible from the current viewpoint, and quickly rejects non-visible geometry before drawing begins. There is another important aspect besides using scene graph concept in creating the virtual visual database. The graphic rendering speed depends on the complexity of the models in front of the user viewpoint, i.e. the number of polygons used to create the models and the texture mapping applied to these models. In this research project, most of the static objects in the visual database are minimally rendered using large polygon size. For example, a visible facet of a building containing window frames and wall are created using single polygon with wall textures image mapped on the polygon. Another technique that is used to enhance the graphic rendering speed is the use of level-of-detail (LOD) concept. A group of objects can be classified under a single node or object called LOD node. LOD node is a specialized 'switch' node that selects its currently active child automatically based on its distance from the viewpoint. LOD node can be used to dynamically select between a set of different representations, each of which is a different level of detail. For example, suppose the simulator driver passes close to a building and then recedes into the distance. WTK allows us to specify the distance at which the LOD node could "swap in" a new or less expensive representation. As the driver recedes into the distance, simpler models (which require less and

less computational effort to process) are progressively swapped in, freeing up memory and system resources.

6 Network Data Transmission

The Transmission Control Protocol/Internet Protocol (TCP/IP) was employed as the data transmission protocol. This protocol ensures the reliability of data transfer system because TCP is a connection-based protocol and requires the establishment of a session before data is transmitted between two computers. TCP is designed to verify that all packets sent by a computer are received on the other end. TCP packets are delivered to Socket. A Socket is a connection between two computers in a TCP/IP network that sends or receives packets of data across a network. Sockets are highly useful in client-server models.

Figure 6 illustrates the conceptual schematic diagram of Windows Socket (Winsock) data transmission. The TCP client socket was created as an additional part in the WTK programming structure and the TCP server socket was constructed as a part of MATLAB Integrated Visual C program. By creating of the client socket, it connects the virtual reality programming to the server, and receives the data computed from MATLAB vehicle dynamic model. In order to establish a socket connection to transfer these data, the client will create a socket and connect it to the server. On the server side, once the server socket is created, it is bound to a specific endpoint and told to listen indefinitely for an attempt by the client to connect. The server socket then accepts the requests of the client for connection. Once the communication channel between these two machines is established, the server will send the vehicle dynamic model output data, and then close the connection. Client socket will receive these data and process them in WTK to control the virtual environment driving simulation.

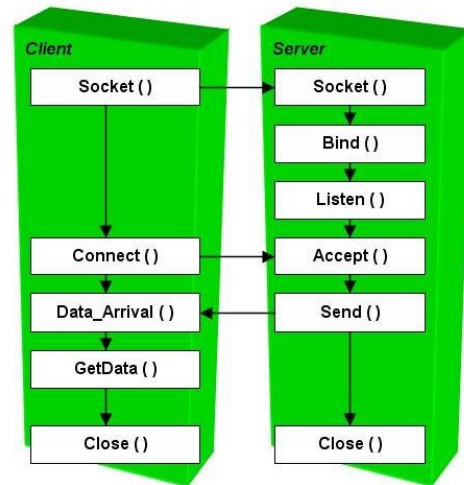


Figure 6 Conceptual Schematic Diagram of Winsock Data Transmission

7 Vehicle Dynamic Model

Figure 7 illustrates the vehicle dynamic model of the driving simulator. Currently, the vehicle dynamic model of the driving simulator is based on a linearized handling model. The model is

based on three degree-of-freedom (3 DOF) of the mass center of the vehicle. On the simulator hardware, the driver manipulates the cockpit and provides three input values into the system: the steering wheel angles, δ_{sw} , the throttle displacement, δ_T , and the brake pedal displacement, δ_B .

These input values from the potentiometers (sensors) are sent to the MATLAB vehicle dynamic model after being converted into wheel angles, δ_w , throttle torque, T_p , and braking torque, T_B respectively. The Linearized Handling Model will simulate the behaviors of actual vehicle handling system and generate vehicle's surge (X_o), sway (Y_o), yaw (ϵ_o) and longitudinal velocity (v_o) values.

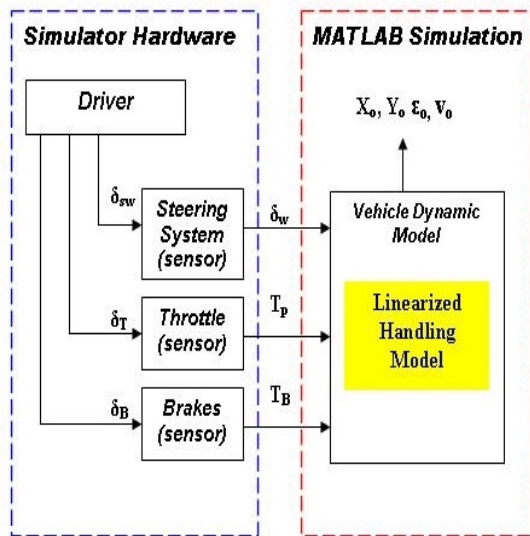


Figure 7 The Simulator Vehicle Dynamic Model

8 Future Work and Implementation

For the future development and implementation, the vehicle dynamic model will be expanded and will include the tire model, suspension model, aerodynamic model and road inputs. Hence, a vehicle model dynamic of six degree of freedom will be employed.

The visual database of driving simulator will be improved to include a more realistic road condition, such as road bumps, dividers and slopes. The actual vehicle orientation in virtual scene does not only depend on these input data but also the integration factor with the virtual road condition. Therefore, a new variable called road input will be introduced. The road input from the simulation collected by WTK internal sensor function will be compared to the vehicle dynamic data to simulate the actual motion of the driving simulator in action.

For a six degree of freedom vehicle dynamic model, the input values from the driving simulation will be passed into the wheel model to generate the wheel spin velocity. These variables are then transferred to the tire model to generate the output forces of x , y , z directions from or on the tire (F_x , F_y , and F_z), slip angle and slip ratio to the vehicle dynamic model. The road input from road condition and road curvature affecting the suspension model will generate the variables of spring force, P_F damping force, P_D , and spring deflection, e for the vehicle dynamic model. The vehicle dynamic model will also include the

aerodynamic forces and aerodynamic friction (F_{wx} and F_{wy}) that will be used to produce the outputs of vehicle's surge (X_i), sway (Y_i), heave (Z_i), roll (Φ_i), yaw (ϵ_i), pitch (θ_i), longitudinal velocity (v_i) and lateral velocity (u_i) respectively. These new series of vehicle's surge (X_i), sway (Y_i), heave (Z_i), roll (Φ_i), yaw (ϵ_i), pitch (θ_i), longitudinal velocity (v_i) and lateral velocity (u_i) will be used to update the vehicle orientation in the virtual scene. These data (X_i , Y_i , Z_i , Φ_i , ϵ_i , θ_i , v_i , u_i) will be looped back into MATLAB vehicle dynamic model for data re-computation and analysis to develop a real time force feedback model for the driving simulator.

9 Conclusion

A low-cost yet effective static base simulator has been developed using virtual-reality technology. This project has laid the groundwork for the development of a full-scale driving simulator to be used for road safety and vehicle related research. The ongoing research work involves the development of a more realistic visual database system, an advanced vehicle dynamic model and integration of audiovisual effect. Future development will also include a motion base integration into the existing system to simulate the dynamic response of a vehicle in motion.

Acknowledgement

The authors wish to acknowledge support of this work under IRPA grant 03-02-06-0087EA001 and RMC-UTM grant 71861.

References

- BAILEY, A.C., JAMSON, A.H., PARKES, A.M. AND WRIGHT, S. 1999. Recent and Future Development of the Leeds Driving Simulator. Presented at DSC99, Paris.
- DROSDOL, J. AND PANIK, F. 1985. The Daimler-Benz Driving Simulator, SAE Paper 850334.
- ENGINEERING ANIMATION INC. 1999. WorldToolKit Reference Manual, Release 9.
- GENTA, G. 1997. Motor Vehicle Dynamic. World Scientific Co. Pte. Ltd.
- HUEI, P. AND MASAYOSHI, T. 1990. Lateral Control of Front-Wheel-Steering Rubber-Tire Vehicles. University of California.
- JAMSON, A.H., PARKES A.M. AND WRIGHT, S. 1999. Recent and Future Development of the Leeds Driving Simulator. Presented at DSC99, Paris.
- NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION AND THE UNIVERSITY OF IOWA. 2002. NADS: National Advanced Driving Simulator, The Most Sophisticated Driving Simulator in the World.
- REPA, B. AND WIERWILLE, W. 1976. Driver Performance in Controlling a Driving Simulator with Varying Vehicle Response characteristics, SAE Paper 760779.