

# IMPLEMENTATION OF A DRIVING SIMULATOR BASED ON A STEWART PLATFORM AND COMPUTER GRAPHICS TECHNOLOGIES

Hung-Lung Tseng and I-Kong Fong

## ABSTRACT

This paper describes works related to the development of a land vehicle driving simulator, which consists of a Stewart platform as its motion device and a computer graphical system as its visual component. The main task is to enable the pilot-experience the sensation of motion while driving the vehicle under the constraint of the platform's finite working space. Involved are works such as establishment of the vehicle dynamical model, analysis of the forces acting on the pilot, and application of the washout algorithm and human motion sensation models. With the results from these works, appropriate motion trajectory commands for the platform can be generated. Besides the physical motion part, a computer graphical system is installed to generate scenes that give visual cues. While following the pilot's steering signals and matching the platform's motion, the visual system creates animation scenes of the environment, which are shown on a large screen in front of the pilot through an LCD projector. To test the completed system, different pilots operate it and give subjective assessments. Also, gyros are mounted on the platform to measure its responses to motion commands. Here some experimental results are summarized and discussed.

**Keywords:** Driving simulator, Stewart platform, computer graphics, control system.

## I. INTRODUCTION

Nowadays, vehicle driving simulators that provide motion and visual cues and have interactive capability are widely employed in pilot training and entertainment utilities. They are convenient and safe to use because they can be operated indoors and are unaffected by environmental factors, such as weather conditions. Moreover, for some large and expensive vehicles, the training scenarios of simulators can be richer than real process, so the training cost can often be reduced and effects improved.

This paper describes works related to the development of a land vehicle driving simulator, which consists of a Stewart platform as its motion device and a computer graphical system as its visual component. The Stewart

platform [1], as shown in Fig. 1, is a mechanical structure which has an upper platform and a lower base platform connected by six variable length links. When the length of each link is controlled properly, the upper platform is

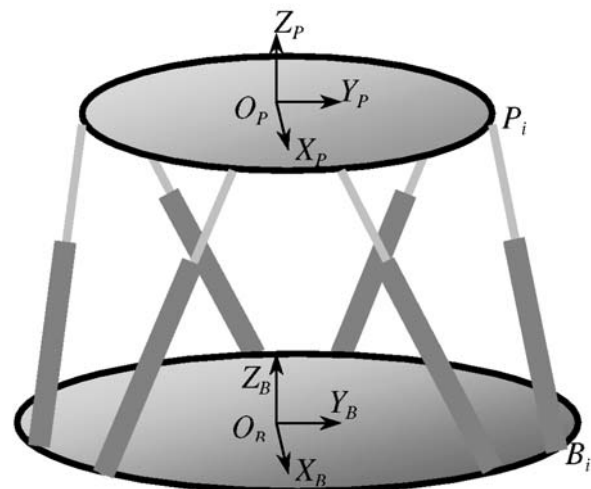


Fig. 1. A Stewart platform.

Manuscript received July 29, 1999; revised October 20, 1999; accepted January 31, 2000.

Hung-Lung Tseng and I-Kong Fong are with Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617, Republic of China.

capable of performing six degree-of-freedom motion within its working space. There are many research topics concerning this mechanical structure, including its construction, singular points and working space determination, forward and inverse kinematic problem solution, dynamic forces computation, motion control, and practical applications [2-12]. Here, we use the hydraulic Stewart platform and its control system reported in [11] and [12] as the basic apparatus to develop a land vehicle driving simulator.

In addition to the motion generation device, a visual system is the other essential component in a driving simulator. It is needed not only because a virtual environment must be created in which the pilot of the simulated vehicle can drive around, but also because visual cues can greatly enhance the motion sensation of the pilot. Fortunately, computer graphics technologies are quite mature today, and many tools can be utilized to help build a reasonably good yet economical visual system. Foley, *et al.* [13] provided complete coverage of computer graphics theory, including topics such as application development, geometric transformation, model construction, devices, interactive skills, and the use of relevant software packages. Trujillo [14] discussed the background and use of the software Direct3D, a component of Microsoft DirectX. With the aid of this software, we have developed a visual system for our simulator, which creates animation scenes of the environment and shows them on a large screen in front of the pilot through an LCD projector. The scenes are designed to be synchronized with the pilot's steering signals and the platform's motion.

Below, we present the development details of the two main subsystems: the Stewart platform-based motion component and the computer graphics-based visual component. First, we define our symbols and conventions for three-dimensional coordinate frames in Section II, which will be used throughout the paper. Then, we briefly introduce the inverse kinematics of the Stewart platform in Section III and derive formulas for link lengths that will be needed to provide the specific upper platform position and orientation needed in the process of motion simulation. In Section IV, a simplified vehicle dynamic model is described, which mainly considers the influence of the vehicle suspension system. Based on this model, the motion of the pilot inside the vehicle is analyzed in Section V, and the part that the pilot is supposed to feel is extracted by means of a classical washout algorithm explained in Section VI, which generates the motion commands for the Stewart platform. After introducing the motion subsystem, we introduce in Section VII the visual component, which includes a big screen, an LCD projector, and a computer with an appropriate algorithm for generating suitable graphics. Finally, in Section VIII, we use a picture and a block diagram to present our experimental setup, and

give some experimental results obtained using the integrated system. Conclusions are given in Section IX.

## II. RELATED COORDINATE FRAMES

First, we will define three right-hand orthogonal coordinate frames,  $F_f$ ,  $F_C$ , and  $F_V$ , needed to discuss the vehicle dynamics. For  $f = I, C$ , and  $V$ , let  $X_f, Y_f$ , and  $Z_f$  be the three principle axes of  $F_f$  with unit vectors  $\bar{I}_f, \bar{J}_f$ , and  $\bar{K}_f$ , respectively. The inertial frame  $F_I$  is land-fixed, and its z-axis is against the direction of gravity. The origin and the other two axes of  $F_I$  are on flat ground on which the land vehicle moves. We assume that, initially, the vehicle is at rest and pointed toward the positive direction of  $X_I$ . The origin of the vehicle frame  $F_C$  is attached to the mass center of the vehicle, with its  $X_C-Y_C$  plane always parallel to the  $X_I-Y_I$  plane. The relation between  $F_C$  and  $F_I$  is a translation equal to the displacement of the vehicle's center of mass, plus a rotation of angle  $\theta_z$  about the  $Z_C$  axis, depending on the yaw motion of the vehicle, so that the positive direction of  $X_C$  keeps to be the vehicle moving forwards. The rotation matrix from  $F_C$  to  $F_I$  is, thus, given by

$$R_C^I = R_{z, \theta_z} = \begin{bmatrix} cz & -sz & 0 \\ sz & cz & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where  $cz$  and  $sz$  denote  $\cos \theta_z$  and  $\sin \theta_z$ , respectively. Similar notations will be used in the following. The origin of the coordinate frame  $F_V$  is the same as that of  $F_C$ , but its  $X_V$  and  $Y_V$  axes coincide with the forward longitudinal and lateral body axes of the vehicle, respectively. More precisely,  $F_V$  can be obtained by rotating  $F_C$  by an angle  $\theta_x$  about the  $X_C$  axis, followed by rotating it by an angle  $\theta_y$  about the  $Y_C$  axis, depending on the roll and pitch motion of the vehicle body, respectively. Obviously, the rotation matrix from  $F_V$  to  $F_C$  is

$$\begin{aligned} R_V^C = R_{y, \theta_y} \cdot R_{x, \theta_x} &= \begin{bmatrix} cy & 0 & sy \\ 0 & 1 & 0 \\ -sy & 0 & cy \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cx & -sx \\ 0 & sx & cx \end{bmatrix} \\ &= \begin{bmatrix} cy & sxsy & cxsy \\ 0 & cx & -sx \\ -sy & sxcy & cxcy \end{bmatrix}, \quad (2) \end{aligned}$$

and the rotation matrix from  $F_V$  to  $F_I$  is

$$R_V^I = R_C^I \cdot R_V^C = \begin{bmatrix} cz & -sz & 0 \\ sz & cz & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cy & sxsy & cxsy \\ 0 & cx & -sx \\ -sy & sxcy & cxcy \end{bmatrix}$$

$$= \begin{bmatrix} cycz & sxsycz - cxsz & cxsycz + sxsz \\ cysz & sxsysz + cxcz & cxsysz - sxcz \\ -sy & sxcy & cxcy \end{bmatrix}. \quad (3)$$

In the sections that follow, we shall assume that with respect to  $F_P$ , that  $F_V$  has an angular velocity

$$\bar{\omega}_{IV} = \dot{\theta}_x \bar{I}_C + \dot{\theta}_y \bar{J}_C + \dot{\theta}_z \bar{K}_C. \quad (4)$$

At any fixed point  $A$  inside the vehicle body, we can further define a vehicle-fixed frame  $F_A$  by translating the origin of  $F_V$  to the point  $A$ . This frame will be useful when we analyze the specific forces acting on the pilot and compute the pilot's rotational velocity.

Finally, we need to mention two coordinate frames pertaining to the Stewart platform. In Fig.1, the base frame  $F_B$  and the platform frame  $F_P$  are defined with their origins located at the center of the base and the upper platform of the structure, respectively. The X-Y planes of the two frames coincide with the base and upper platform planes, respectively.

### III. INVERSE KINEMATICS OF THE STEWART PLATFORM

The inverse kinematics problem for the Stewart platform is to determine link lengths given the relative translation and rotation of the upper platform with respect to the fixed base. Ignoring the translational difference for now, let the frame  $F_P$  be obtained by rotating  $F_B$  by an angle  $\gamma$  about the  $X_B$  axis, followed by rotating it by an angle  $\beta$  about the  $Y_B$  axis, followed by rotating it by an angle  $\alpha$  about the  $Z_B$  axis. The rotation matrix from  $F_P$  to  $F_B$  is given by

$$R_P^B = R_{z,\alpha} \cdot R_{y,\beta} \cdot R_{x,\gamma}$$

$$= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \quad (5)$$

Suppose the position vector from  $O_B$  to  $O_P$ ,  $\overrightarrow{O_B O_P}$ , has the following coordinate vector in the frame  $F_B$ :

$$\overrightarrow{O_B O_P}^B = [x \ y \ z]^T \quad (6)$$

Note that, here, we adopt the convention of using a superscript after a vector to represent its coordinate vector in a certain coordinate frame. This convention will also be followed below, except for the superscript  $T$ , which is reserved for the transpose of column and row

vectors. From Fig. 2, we have

$$\begin{aligned} \overrightarrow{B_i P_i}^B &= \overrightarrow{O_B O_P}^B + \overrightarrow{O_P P_i}^B - \overrightarrow{O_B P_i}^B \\ &= \overrightarrow{O_B O_P}^B + R_P^B \cdot \overrightarrow{O_P P_i}^P - \overrightarrow{O_B P_i}^B, \end{aligned} \quad (7)$$

in which  $\overrightarrow{O_P P_i}^P$  and  $\overrightarrow{O_B P_i}^B$  are constant vectors determined by the physical dimensions of the structure. Hence, the length of the  $i$ -th link is

$$l_i = \sqrt{\|\overrightarrow{B_i P_i}^B\|_2}. \quad (8)$$

Note that the right side of (8) is a nonlinear function of the position and attitude variables  $(x, y, z, \alpha, \beta, \gamma)$  of the upper platform. In other words, given any position and attitude of the upper platform within its working space, we can use (8) to determine all six link lengths that are required. For the Stewart platform in a driving simulator,  $(x, y, z, \alpha, \beta, \gamma)$  of the upper platform must follow the trajectory commands generated by the system and the pilot. Thus, (8) is the key equation for motion realization, as link length changes can be implemented by controlling the hydraulic cylinder links appropriately.

### IV. VEHICLE DYNAMIC MODEL [15]

It is important to predict the vehicle dynamic responses to the pilot's inputs in an interactive driving simulator. In this section, we will introduce a simplified vehicle dynamic model, which considers mainly the dynamic characteristics of the vehicle suspension system. For simplicity, all other sources of dynamic forces, such as engine vibration, structural flexibility, transmission actions etc., are ignored. Consider the vehicle at rest on the  $X_P$ - $Y_P$  plane as shown in Fig. 3.

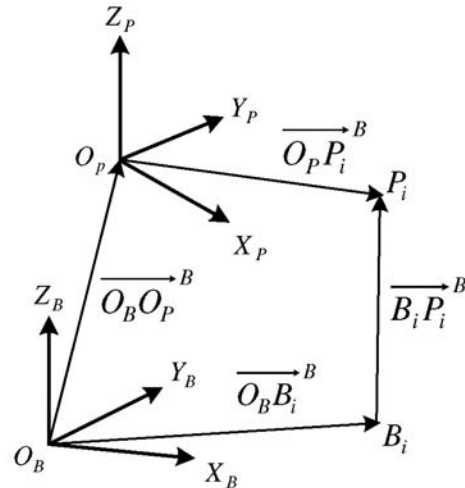
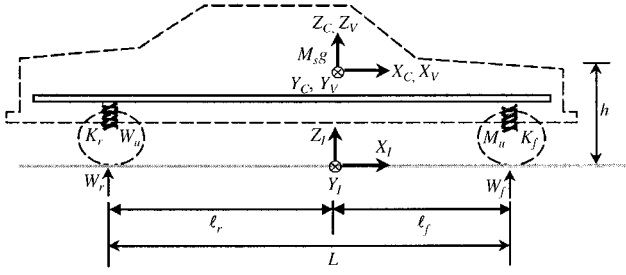


Fig. 2. Inverse kinematics of a Stewart platform.

Fig. 3. Side view of the vehicle at rest on the  $X_f$ - $X_r$  plane.

When the vehicle is not moving, the static forces exerted by the ground on the front and rear wheels are

$$W_f = M_u g + M_s g \frac{\ell_r}{L}, \quad (9)$$

$$W_r = M_u g + M_s g \frac{\ell_f}{L}, \quad (10)$$

respectively, where  $M_s$  is the mass of the vehicle body,  $M_u$  is the mass of two suspension systems and wheels,  $g$  is the gravitational acceleration, and  $\ell_f$ ,  $\ell_r$  and  $L$  are lengths shown in Fig. 3.

When the vehicle has forward acceleration  $a_x$  at time  $t$ , we can take moments about points  $E_1$  and  $E_2$  in Fig. 4 to obtain the normal reaction forces on the front and rear wheels as

$$W_f = M_u g + M_s g \frac{\ell_r}{L} - \frac{M_s a_x (h + s_z)}{L}, \quad (11)$$

$$W_r = M_u g + M_s g \frac{\ell_f}{L} + \frac{M_s a_x (h + s_z)}{L}, \quad (12)$$

respectively, where  $h$  is the height of the vehicle's center of mass when it is at rest, as shown in Fig. 3, and  $s_z$  is the shift of the vehicle's center of mass along the positive direction of  $Z_C$  at time  $t$ . From (9)-(12), we know that the deflection of front and rear suspension is, respectively,

$$\Delta_f = \frac{M_s a_x (h + s_z)}{L K_f}, \quad (13)$$

$$\Delta_r = \frac{M_s a_x (h + s_z)}{L K_r}, \quad (14)$$

where  $K_f$  and  $K_r$  are the stiffness coefficients of the front and rear suspension systems, respectively. At the next instant  $t + \Delta t$ , where  $0 < \Delta t \ll 1$ , the shift of the vehicle's center of mass due to pitching of the vehicle body is

$$s_{z1} = \frac{\ell_r}{L} \Delta_f - \frac{\ell_f}{L} \Delta_r, \quad (15)$$

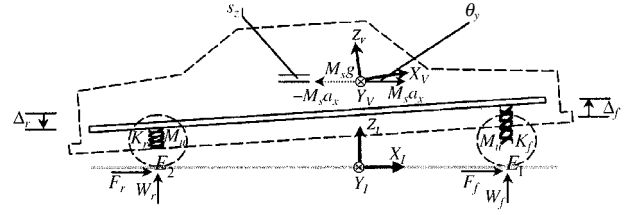


Fig. 4. Side view of the vehicle during acceleration.

and the rotation angle of  $F_V$  along the  $Y_C$  axis with respect to  $F_C$  is

$$\theta_y = -\frac{\Delta_f + \Delta_r}{L}. \quad (16)$$

For the lateral motion of the vehicle, consider Fig. 5. When the vehicle is not moving, the static forces exerted by the ground on the left and right wheels are, respectively,

$$W_l = M_u g + M_s g \frac{d_{rt}}{D}, \quad (17)$$

$$W_{rt} = M_u g + M_s g \frac{d_l}{D}, \quad (18)$$

where  $d_{rt}$ ,  $d_l$ , and  $D$  are shown in Fig. 5. Suppose the vehicle has a velocity  $v_x$  in the positive direction of  $X_C$  and an angular velocity  $\dot{\theta}_z$  in the positive direction of  $Z_C$ ; then, the centripetal acceleration is

$$a_y = \frac{v_x^2}{\rho} = v_x \dot{\theta}_z. \quad (19)$$

Via similar derivation for the longitudinal case, we can obtain relative quantities at time  $t$  as

$$W_l = M_u g + M_s g \frac{d_{rt}}{D} - \frac{M_s a_y (h + s_z)}{D}, \quad (20)$$

$$W_{rt} = M_u g + M_s g \frac{d_l}{D} + \frac{M_s a_y (h + s_z)}{D}, \quad (21)$$

$$\Delta_l = \frac{M_s a_y (h + s_z)}{D K_l}, \quad (22)$$

$$\Delta_{rt} = \frac{M_s a_y (h + s_z)}{D K_{rt}}. \quad (23)$$

Consequently, we can determine the shift of the mass center at time  $t + \Delta t$  due to the rolling of the vehicle body as

$$s_{z2} = \frac{d_{rt}}{D} \Delta_l - \frac{d_l}{D} \Delta_{rt} \quad (24)$$

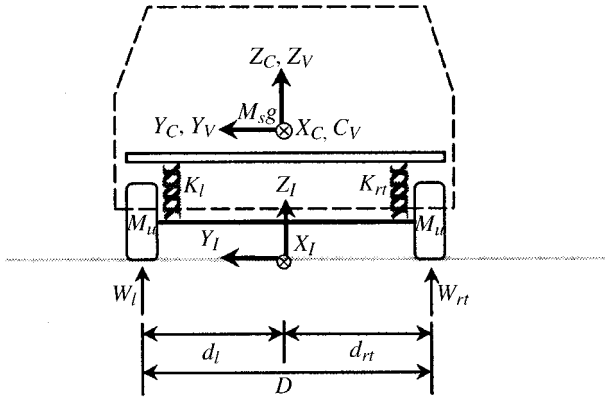


Fig. 5. Rear view of the vehicle at rest.

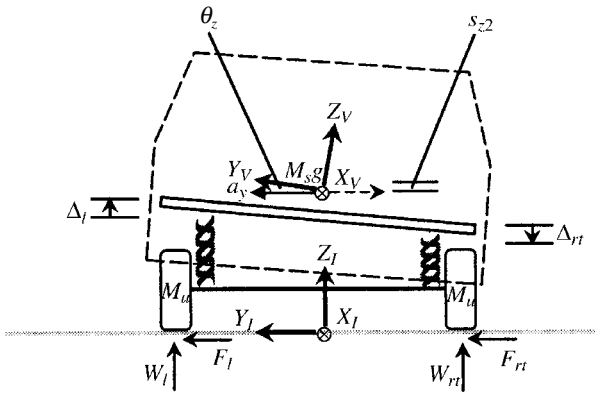


Fig. 6. The vehicle turning right.

and the rotation angle of  $F_V$  along the axis  $X_C$  with respect to  $F_C$  as

$$\theta_x = \frac{\Delta_l + \Delta_{rt}}{D}. \quad (25)$$

Thus, at time  $t + \Delta t$ , the total shift of the mass center is

$$s_z = s_{z1} + s_{z2}. \quad (26)$$

## V. PILOT MOTION ANALYSIS

In this section, we will analyze the motion of a fixed point  $A$  inside the vehicle body. In our simulator, this point will be the center of the pilot's seat in the simulated vehicle, so we shall be able to describe the motion that the pilot will experience in appropriate frames. Let  $\vec{r}_{VA}$  be the position vector from the origin of  $F_V$  to point  $A$ ; thus,  $\vec{r}_{VA}^V$  is a constant vector. Coordinate transformation gives

$$\vec{r}_{VA}^V = R_I^V \cdot \vec{r}_{VA}^I. \quad (27)$$

Hence,

$$\dot{\vec{r}}_{VA}^V = R_I^V \cdot \dot{\vec{r}}_{VA}^I + \dot{R}_I^V \cdot \vec{r}_{VA}^I = \vec{0}. \quad (28)$$

Symmetrically, we have

$$\dot{\vec{r}}_{VA}^I = R_I^V \cdot \dot{\vec{r}}_{VA}^V + \dot{R}_I^V \cdot \vec{r}_{VA}^V = \dot{R}_I^V \cdot \vec{r}_{VA}^V. \quad (29)$$

According to the kinematics of non-inertial frames, we also have

$$\dot{\vec{r}}_{VA}^I = \Omega_{IV}^I \cdot \vec{r}_{VA}^V, \quad (30)$$

where

$$\Omega_{IV}^I = \begin{bmatrix} 0 & -\omega_{IV3}^I & \omega_{IV2}^I \\ \omega_{IV3}^I & 0 & -\omega_{IV1}^I \\ -\omega_{IV2}^I & \omega_{IV1}^I & 0 \end{bmatrix} \quad (31)$$

is skew-symmetric and  $[\omega_{IV1}^I \ \omega_{IV2}^I \ \omega_{IV3}^I]^T = \vec{\omega}_{IV}^I$  from (4). Based on (29) and (30), we see that

$$\dot{R}_V^I \cdot \vec{r}_{VA}^V = \Omega_{IV}^I \cdot \vec{r}_{VA}^V = \Omega_{IV}^I \cdot R_V^I \cdot \vec{r}_{VA}^V. \quad (32)$$

Since  $\vec{r}_{VA}^V$  is arbitrary, it can be concluded that

$$\dot{R}_V^I = \Omega_{IV}^I \cdot R_V^I, \quad (33)$$

$$\Omega_{IV}^I = \dot{R}_V^I \cdot R_V^V. \quad (34)$$

Now, if we treat  $F_V$  as the fixed frame and  $F_I$  as the rotating one with angular velocity  $-\vec{\omega}_{IV}^I$ , we get

$$\dot{R}_I^V \cdot \vec{r}_{VA}^I = -\Omega_{IV}^V \cdot \vec{r}_{VA}^V = -\Omega_{IV}^V \cdot R_I^V \cdot \vec{r}_{VA}^I, \quad (35)$$

$$\dot{R}_I^V = -\Omega_{IV}^V \cdot R_I^V. \quad (36)$$

The orthogonality of  $F_I$  and  $F_V$  implies that

$$R_V^I = (R_I^V)^T = (R_I^V)^{-1}, \quad (37)$$

$$\dot{R}_V^I = (\dot{R}_I^V)^T. \quad (38)$$

Substitution of (36), (37), and (38) into (34) yields

$$\Omega_{IV}^I = (\dot{R}_I^V)^T \cdot R_I^V = (-\Omega_{IV}^V \cdot R_I^V)^T \cdot R_I^V = R_V^I \cdot \Omega_{IV}^V \cdot R_I^V, \quad (39)$$

which in turn can be substituted into (33) to produce

$$\dot{R}_V^I = (R_V^I \cdot \Omega_{IV}^V \cdot R_I^V) \cdot R_V^I = R_V^I \cdot \Omega_{IV}^V. \quad (40)$$

From (29), (33), and (39), we get

$$\dot{\bar{r}}_{VA}^I = \Omega_{IV}^I \cdot \bar{r}_{VA}^I = (R_V^I \cdot \Omega_{IV}^V \cdot R_I^V) \cdot \bar{r}_{VA}^I, \quad (41)$$

which can be simplified as

$$R_I^V \cdot \dot{\bar{r}}_{VA}^I = \Omega_{IV}^V \cdot \bar{r}_{VA}^V. \quad (42)$$

The position vector  $\bar{r}_{IA}^I$  pointing from the origin of frame  $F_I$  to fixed point  $A$  can be expressed as the sum

$$\bar{r}_{IA}^I = \bar{r}_{IV}^I + \bar{r}_{VA}^I, \quad (43)$$

where  $\bar{r}_{IV}^I$  is the position vector pointing from the origin of frame  $F_I$  to the origin of frame  $F_V$ . If the vehicle has both translation and rotation motions in frame  $F_I$ , then the first and second order time derivatives of the position vector  $\bar{r}_{IA}^I$  expressed in  $F_I$  are, respectively,

$$\dot{\bar{r}}_{IA}^I = \dot{\bar{r}}_{IV}^I + R_V^I \cdot \Omega_{IV}^V \cdot \bar{r}_{VA}^V, \quad (44)$$

$$\ddot{\bar{r}}_{IA}^I = \ddot{\bar{r}}_{IV}^I + R_V^I (\dot{\Omega}_{IV}^V \cdot \bar{r}_{VA}^V + \Omega_{IV}^V \cdot \Omega_{IV}^V \cdot \bar{r}_{VA}^V), \quad (45)$$

where (37), (40), and (42) are used to compute these derivatives. By taking  $R_V^I$  on (45) to the left side of the equation, (45) can also be used to express the second derivative of  $\bar{r}_{IA}^I$  in  $F_V$ :

$$R_I^V \cdot \ddot{\bar{r}}_{IA}^I = R_I^V \cdot \ddot{\bar{r}}_{IV}^I + (\dot{\Omega}_{IV}^V + \Omega_{IV}^V \cdot \Omega_{IV}^V) \cdot \bar{r}_{VA}^V, \quad (46)$$

where the first term on the right side can be changed to

$$R_I^V \cdot \ddot{\bar{r}}_{IV}^I = \ddot{\bar{r}}_{IV}^V + \Omega_{IV}^V \cdot \dot{\bar{r}}_{IV}^V \quad (47)$$

by taking the derivatives of both sides of the equation

$$\dot{\bar{r}}_{IV}^I = R_V^I \cdot \dot{\bar{r}}_{IV}^V \quad (48)$$

and using (40).

For our simulator development problem, we can use (46) and (47) as well as (50) and (51) given in the next Section to compute the specific force and angular velocity on point  $A$ , the pilot's seat. Note that the right hand sides of (46) and (47) are the vehicle angular and translational velocities  $\Omega_{IV}^V$  and  $\dot{\bar{r}}_{IV}^V$  and the acceleration  $\dot{\Omega}_{IV}^V$  and  $\ddot{\bar{r}}_{IV}^V$  represented in frame  $F_V$ . These quantities can be easily determined from the information provided by the vehicle dynamic model introduced in Section IV.

## VI. THE CLASSICAL WASHOUT ALGORITHM

To generate the sensation of motion for pilots driving a simulator, a standard approach is to compute the specific force [16] acting on the pilot in the vehicle, which is defined as the nongravitational force per unit mass. Suppose from the vehicle dynamic model and the method described in Section V, we know that at a certain instant, the acceleration of the center of the pilot's seat is  $\bar{a}$ . Then the specific force acting on the pilot is

$$\bar{f} \equiv \bar{a} - \bar{g}. \quad (49)$$

Clearly, due to the motion capability and working space constraints, the Stewart platform can not re-create the specific force faithfully in the entire operational process. Thus, some adjustments are necessary so that only the part that can be sensed by the pilot is re-created. The principle human sensing organs for rotation motion and the specific force are the semicircular canals and otolith, respectively. The mathematical response models for the semicircular canals and otolith are shown in Fig. 7 and Fig. 8, and are active within the frequency region of 0.2–10 rad/sec and 0.2–2 rad/sec, respectively. Exploiting the threshold and band-pass properties of the human motion sensation, the washout algorithms suggests how we can create a sensation of motion by performing translation and rotation of certain frequency bands only, and by replacing the sustained linear translation with a tilt motion.

Consider the seat center point  $A$  on which the pilot sits. Because  $F_A$  is merely a translation of  $F_V$ , the specific force acting on point  $A$  has the same representation in both frames:

$$\bar{f}_A^A = \bar{f}_A^V = R_I^V \cdot \ddot{\bar{r}}_{IA}^I - R_I^V \cdot \dot{g}^I, \quad (50)$$

and the angular velocity of point  $A$  with respect to  $F_I$  is equal to that of the vehicle's center of mass, i.e.,

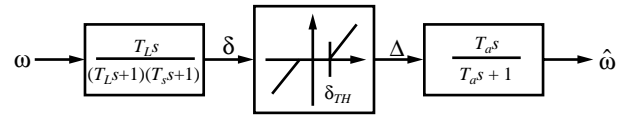


Fig. 7. Mathematical response model of the semicircular canals [17,18].

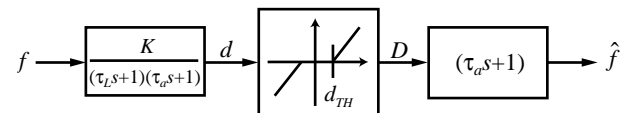


Fig. 8. Mathematical response model of the otolith [17,18].

$$\Omega_{IA}^A = \Omega_{IA}^V = \Omega_{IV}^V. \quad (51)$$

Note that the second equality in (50) comes from the results derived in Section V. To extract from (50) and (51) the parts that both need to and can be implemented on the Stewart platform, the classical washout algorithm [9,18] shown in Fig. 11 employs three functions. First of all, it generates platform translational motion commands  $x$ ,  $y$ , and  $z$  by putting  $\vec{f}_A^A$  through a nonlinear attenuator of the characteristics shown in Fig. 9 and a filter  $H_1(s)$ . The nonlinear attenuator is used to simulate the threshold effect of human sensation and to scale the amplitudes of the motion commands. The filter is used to eliminate the low frequency content of the motion commands to which the pilot is not sensitive. Secondly, it generates some of the platform rotational motion commands  $\alpha$ ,  $\beta$ , and  $\gamma$  by putting  $\vec{\omega}_{IA}^A$  from  $\Omega_{IA}^A$  in (51) through a nonlinear attenuator and a filter  $H_2(s)$ . Thirdly, it generates the remaining platform rotational motion commands. The purpose here is to implement the sustained linear trans-

lational acceleration by means of tilt-coordinates shown in Fig. 10, where the relative tilt angles  $\vec{\phi} = [\phi_x \ \phi_y \ \phi_z]^T$  are

$$\phi_x = \tan^{-1}\left(\frac{f_{L2}}{f_{L3}}\right), \quad (52)$$

$$\phi_y = -\tan\left(\frac{f_{L1}}{f_{L3}} \cdot \cos(\phi_x)\right), \quad (53)$$

$$\phi_z = 0, \quad (54)$$

and  $\vec{f}_L^T = [f_{L1} \ f_{L2} \ f_{L3}]^T$  is obtained by putting  $\vec{f}_A^A$  through a nonlinear attenuator in series with a low-pass filter  $L(s)$ . Obviously,  $\vec{f}_L^T$  is the specific driving force for the sensible part of the sustained linear translational acceleration, and if we tilt the upper platform with angles  $\phi_x$  and  $\phi_y$  around the axes  $X_p$  and  $Y_p$ , respectively, then the gravity force will produce the first two

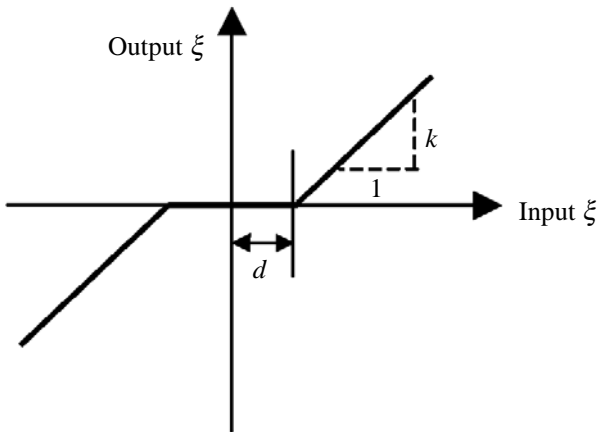


Fig. 9. Nonlinear attenuator of the washout algorithm.

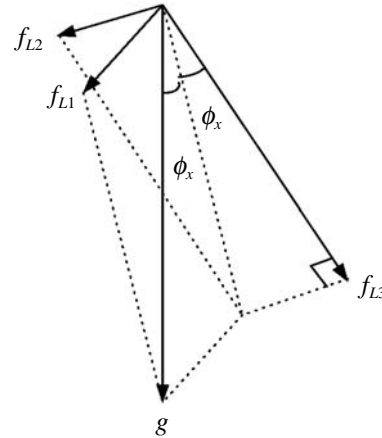


Fig. 10. Tilt coordinates.

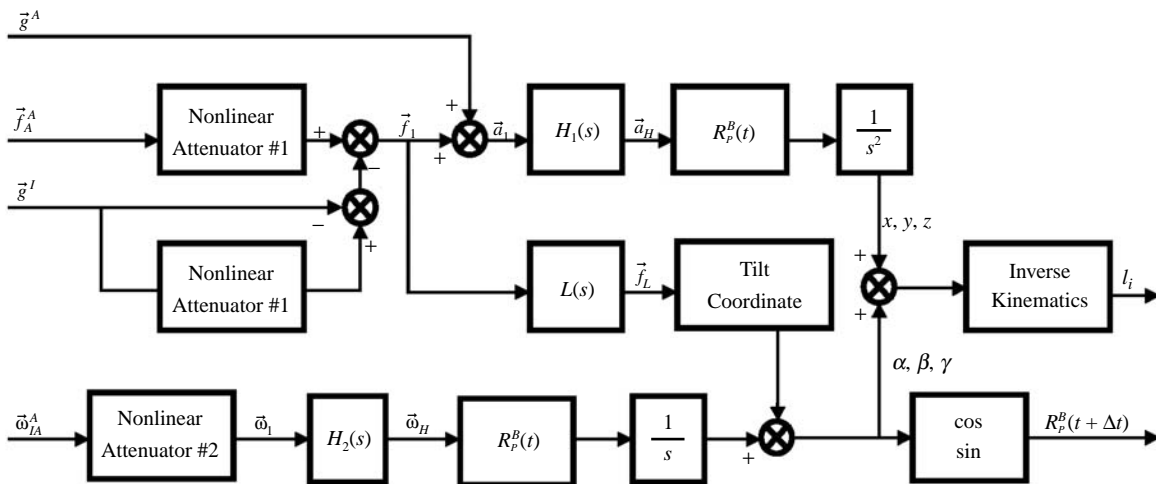


Fig. 11. Classical washout algorithm.

**Table 1. Parameters of the filters.**

| Transfer function | $\frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$ |          |         |
|-------------------|---|----------|---------|
|                   | $H_1(z)$  | $H_2(z)$ | $L(z)$  |
| $a_1$             | -1.9187   | -1.9112  | -1.8890 |
| $a_2$             | 0.9244  | 0.9150   | 0.8949  |
| $b_0$             | 0.0378  | 0.9565   | 0.0015  |
| $b_1$             | 0   | -1.9131  | 0.0029  |
| $b_2$             | -0.0378   | 0.9565   | 0.0015  |

**Table 2. Parameters of nonlinear attenuator #1.**

|  | Surge<br>(x-axis) | Sway<br>(y-axis) | Heave<br>(z-axis) |
|--|-------------------|------------------|-------------------|
| Threshold value $d$<br>(m/sec <sup>2</sup> ) | 0.17              | 0.17             | 0.28              |
| Amplitude ratio $k$                          | 0.4               | 0.4              | 0.4               |

**Table 3. Parameters of nonlinear attenuator #2.**

|                                  | Roll<br>(x-axis) | Pitch<br>(y-axis) | Yaw<br>(z-axis) |
|----------------------------------|------------------|-------------------|-----------------|
| Threshold value $d$<br>(deg/sec) | 3.0              | 3.6               | 2.6             |
| Amplitude ratio $k$              | 0.7              | 0.7               | 0.7             |

components of  $\vec{f}_L^T$  for us. The related parameters of the filters and nonlinear attenuators are listed in Tables 1, 2 and 3, respectively, in which the transfer functions of the filters are expressed by the z-transform because they need to be implemented by the digital computer.

## VII. VISUAL SYSTEM

The visual system is needed not only to create a virtual environment for the pilot of the simulated vehicle to drive around in, but also to produce visual cues which greatly reinforce the sensation of motion experienced by the pilot. For our system, the image generation software was developed using the graphics tool package Direct3D on the operating system Windows 95. The environmental scenes, such as trees, road, and road signs, were constructed using the method of texture mapping [13,14], as shown in Fig. 12. A typical scene like that shown Fig. 13, is displayed on a large screen in front of the pilot using an LCD projector. When the pilot inputs steering signals into the simulator through a joystick, we decide what the pilot should see by computing the position and orientation of the view port marked in front of the vehicle as shown in Fig. 12. In our system, two computers are

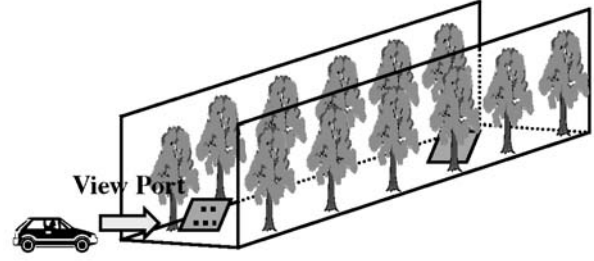


Fig. 12. Scene construction.

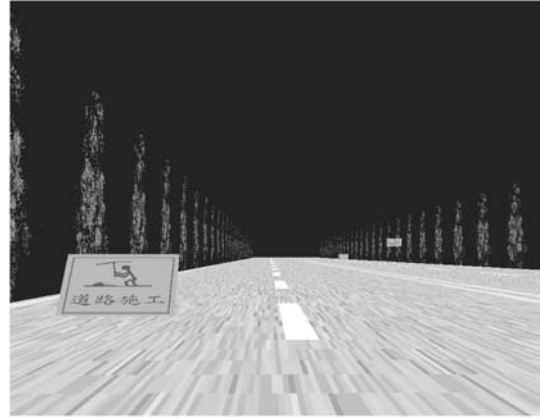


Fig. 13. A typical scene projected by the simulator.

employed. The master computer receives/transforms commands, determines the view port, generates images in real time, and passes the transformed commands to the other computer, the slave computer, for the vehicle dynamics, utilizing the washout algorithm, and controlling the Stewart platform.

Here, we will briefly describe how the position and orientation of the view port are computed. In response to steering signals, which are the axial acceleration  $a_x$  and the front wheel turn angle  $\varphi$ , the master computer first determines the yawing velocity of the vehicle based on the relation given in Fig. 14, where  $L$  is defined in Fig. 3,  $dx$  is an infinitesimal translational displacement,  $d\theta_z$  is an infinitesimal yaw angle, and

$$dx \cdot \sin(\varphi) = L \cdot \sin(d\theta_z) \approx L \cdot d\theta_z. \quad (55)$$

An infinitesimal time  $dt$  is divided on both sides to obtain

$$\frac{dx}{dt} \cdot \sin(\varphi) = L \cdot \frac{d\theta_z}{dt} \quad (56)$$

or

$$\omega_{IC3}^C = \frac{\sin(\varphi)}{L} \cdot v_{IC1}^C, \quad (57)$$



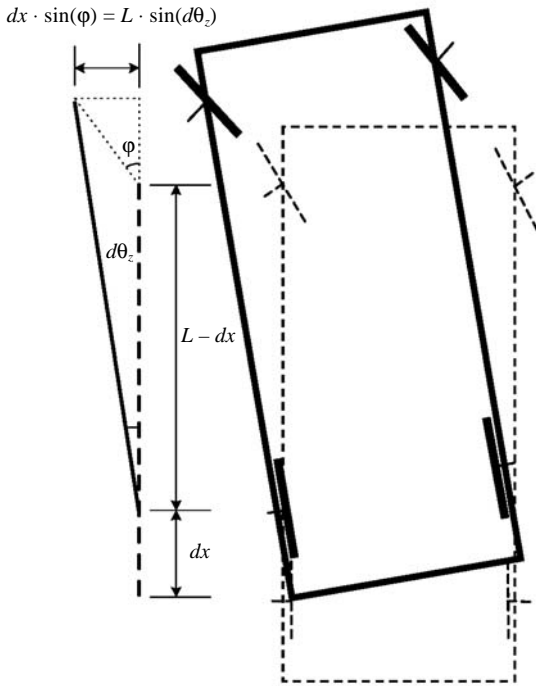


Fig. 14. Yawing velocity of the vehicle.

where  $\omega_{IC3}^C$  is the angular velocity of the vehicle about the axis  $Z_C$ , and  $v_{IC1}^C$  is the axial velocity of the vehicle. Note that  $v_{IC1}^C$  can be obtained by integrating the axial acceleration  $a_x$ , which is the first component of  $\vec{a}_{IC}^C$ , i.e.,  $a_{IC1}^C$ . Clearly,  $\omega_{IC3}^C$  and  $v_{IC1}^C$  together are sufficient to define the position of the view port. As to the orientation of the view port, we use a simplified scheme to speed up the computation and graphics display. In Direct3D, every object in the scene belongs to some frame, and the view port is no exception. Also, a frame may be the “child” frame of some “parent” frame, which in turn may be the “child” frame of some other “parent” frame and, eventually of the “root” frame. We describe the translational and yawing motion of the view port using a frame Z, which is the “child” frame of the “root” frame. Then, we describe the pitching motion of the view port using a frame Y, which is the “child” frame of the frame Z. Finally, we describe the rolling motion of the view port using a frame X, which is the “child” frame of the frame Y. Figure 15 shows the frame structure of the view port except that, actually, frames X, Y, and Z have the same origin. They are separated in Fig. 15 to facilitate presentation. For the view port, the pitching angle  $\theta_y$  is the rotation angle of the Y-axis of the frame Y and is largely proportional to the axial acceleration  $a_{IC1}^C$  according to (13), (14), and (16). Therefore,

$$\theta_y = \sigma_y \cdot a_{IC1}^C \quad (58)$$

where  $\sigma_y$  is a constant. Similarly, the rolling angle  $\theta_x$  is the rotation angle of the X-axis of the frame X and is

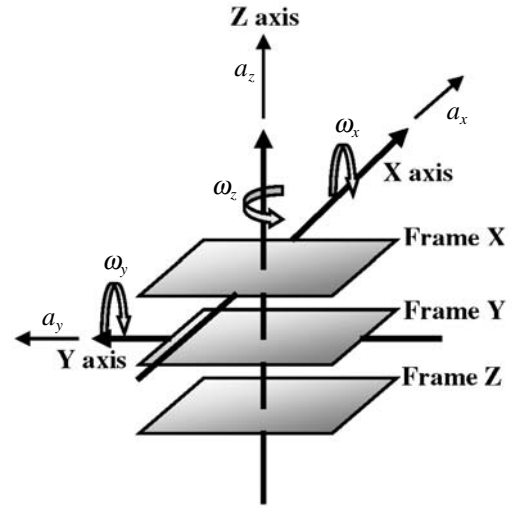


Fig. 15. Frame structure of the view port.



Fig. 16. Stewart platform driving simulator.

largely proportional to the centripetal acceleration determined in (19) according to (22), (23), and (25). Therefore,

$$\theta_x = \sigma_x \cdot v_{IC1}^C \cdot \omega_{IC3}^C, \quad (59)$$

where  $\sigma_x$  is a constant. Lastly, the yawing angle  $\theta_z$  is the rotation angle of the Z-axis of the frame Z and can be obtained by integrating  $\omega_{IC3}^C$ .

## VIII. EXPERIMENTAL RESULTS

Figure 16 shows the driving simulator assembled in the Advanced Control Laboratory of the Department of Electrical Engineering, National Taiwan University. The block diagram of its system configuration is shown in Fig. 17. In the system, the master computer generates

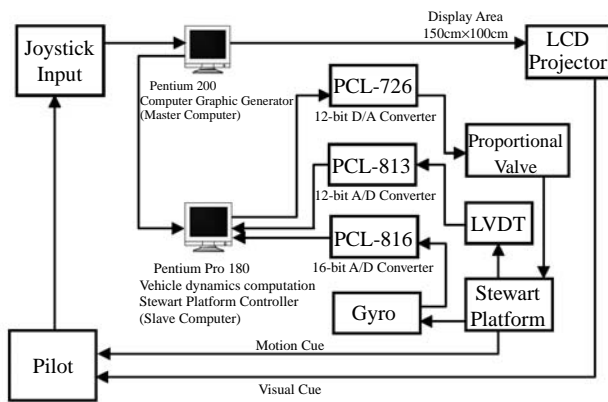


Fig. 17. Block diagram of the driving simulator.

images at a rate of 25 pictures per second while the slave computer controls the motion of the Stewart platform based on the vehicle model at a rate of 400 Hz. This method of computational load sharing successfully satisfies the real time operational requirements of the system. Also, because the steering commands received from the pilot are received by the master computer and used for signal processing, and are then passed to the slave computer which controls the motion part, visual cues and motion cues are well synchronized. Besides the computer part of the system, the mechanical part also plays an important role. As we mentioned above, the controller is based on the results reported in [11] and [12], where only LVDTs are meaning unclear feedback link lengths. Here, we add gyroscopes to record the real time orientation responses of the platform but do not feed the signals back to the controller, thus avoiding the bias effect of the gyroscopes. Use of the hydraulic Stewart platform enables us to carry heavy loads, such as human pilots, but also restricts the operational speed, compared with those of Stewart platforms driven by electric motors. However, the response time of the hydraulic actuators and controller adopted here is fast enough for this application project, as the test results to be presented below show.

Here, we will present three groups of experimental results obtained using this setup. In the first experiment, without putting a pilot on the platform, we let the master computer itself generate a set of steering signals shown in Fig. 18. These command signals asked the simulated

vehicle to accelerate constantly for 10 seconds without making any turns, and to cruise with zero acceleration afterwards. The resulting command signals for the platform orientation and position are the dark lines in each diagram in Fig. 19. Apparently, these command signals are correct, as the commands for  $\alpha$ ,  $\gamma$  and  $y$  are all zero, and the commands for  $\beta$ ,  $x$  and  $z$  reflect the motion that the pilot would experience at the beginning and end of a constant translational acceleration. Figure 19 also includes the orientation responses (light lines) for  $\alpha$ ,  $\beta$ , and  $\gamma$  of the platform measured by gyroscopes. These responses contain some small tracking errors, which may be due to the bias of the gyroscope and to coupling between different degrees of freedom of platform motion. The second experiment was similar to the first one except that this time the simulated vehicle was required to make a turn while accelerating forwards. The steering signals are shown in Fig. 20. Note that it is the yawing velocity  $\omega_{C3}^C$  that is shown, not the front wheel turn angle  $\phi$  received from the joystick. Again, we see that the resulting command signals for the platform orientation and position represented by the dark lines in Fig. 21 are reasonable, and that the orientation responses (light lines) measured by the gyroscope contain only small tracking errors. Finally, in the third experiment, a pilot tested the platform by steering it using the joystick and produced the steering commands displayed in Fig. 22. It is worth noting that even with these almost random inputs, the platform responded quite well, as the gyroscopic outputs in Fig. 23 show. More detained results of these experiments can be found in [19].

## IX. CONCLUSION

We have constructed a driving simulator by integrating a realistic vehicle dynamic model and virtual reality technologies. The system has been tested thoroughly, and the performance has been found to be quite satisfactory. A video clip showing operation of the simulator and its components can be viewed at the Web site <http://acl.ee.ntu.edu.tw>. For more complex vehicles such as ships and aircraft, it will be possible to use the same approach to build simulators. More complex factors, such as road roughness, air resistance etc., will be introduced into our system in the future.

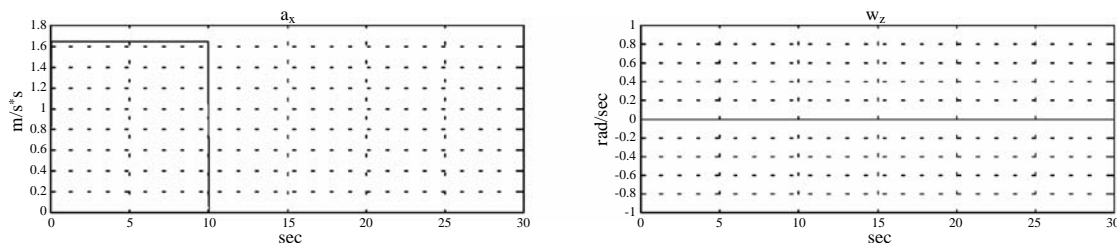


Fig. 18. Steering commands.

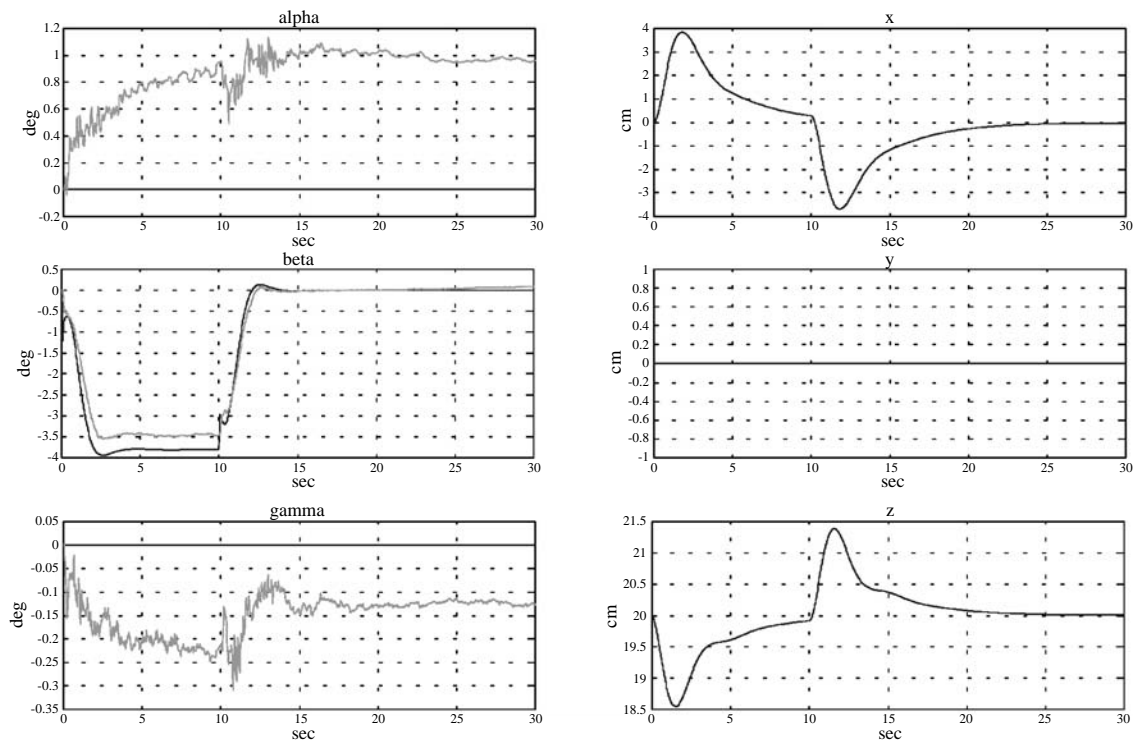


Fig. 19. Platform orientation/position commands and orientation responses.

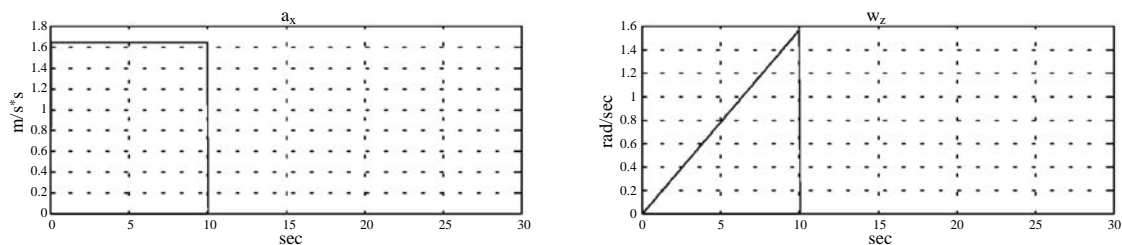


Fig. 20. Steering commands.

## REFERENCES

- Stewart, D., "A Platform with Six Degrees of Freedom," *Proc. Inst. Mech. Eng.*, Vol. 180, Part 1, No. 5, pp. 371-386 (1965).
- Yang, D.C.H. and T.W. Lee, "Feasibility Study of a Platform Type of Robotic Manipulators from a Kinematic Viewpoint," *J. Mech., Transm. Autom. Design*, Vol. 106, No. 1, pp. 191-198 (1984).
- Fichter, E.F., "A Stewart Platform Based Manipulator: General Theory and Practical Construction," *Int. J. Rob. Res.*, Vol. 5, No. 2, pp. 157-182 (1986).
- Do, W.Q.D. and D.C.H. Yang, "Inverse Dynamic Analysis and Simulation of a Platform Type of Robot," *J. Rob. Syst.*, Vol. 5, No. 3, pp. 209-227 (1988).
- Liu, K., "Modeling and Control of a Stewart Platform Manipulator," *Amer. Soc. Mech. Eng., Dyn. Syst. Contr. Division*, Vol. 33, No. 1, pp. 83-89 (1991).
- Liu, K., J.M. Fitzgerald and F.L. Lewis, "Kinematic Analysis of a Stewart Platform Manipulator," *IEEE Trans. Ind. Electron.*, Vol. 40, No. 2, pp. 282-293 (1993).
- Reid, L.D., "Motion Algorithm for Large-Displacement Driving Simulator," *Transpo. Res. Record*, No. 1403, pp. 98-106 (1993).
- Salcudean, S.E., P.A. Drexel, D. Ben-Dov, A.J. Taylor and P.D. Lawrence, "A Six Degree-of-Freedom, Hydraulic, One Person Motion Simulator," *Proc. IEEE Int. Conf. Rob. Autom.*, San Diego, CA, USA, pp. 2437-2443 (1994).
- Grant, P.R. and L.D. Reid, "Motion Washout Filter Tuning: Rules and Requirements," *J. Aircraft*, Vol. 34, No. 2, pp. 145-151 (1997).
- Pouliot, N.A. and C.M. Gosselin, "Motion Simulator Capabilities of Three-Degree-of-Freedom Flight Simulators," *J. Aircraft*, Vol. 35, No. 1, pp. 9-17 (1998).
- Wen, C., C.-C. Hsu and I.-K. Fong, "Modeling and

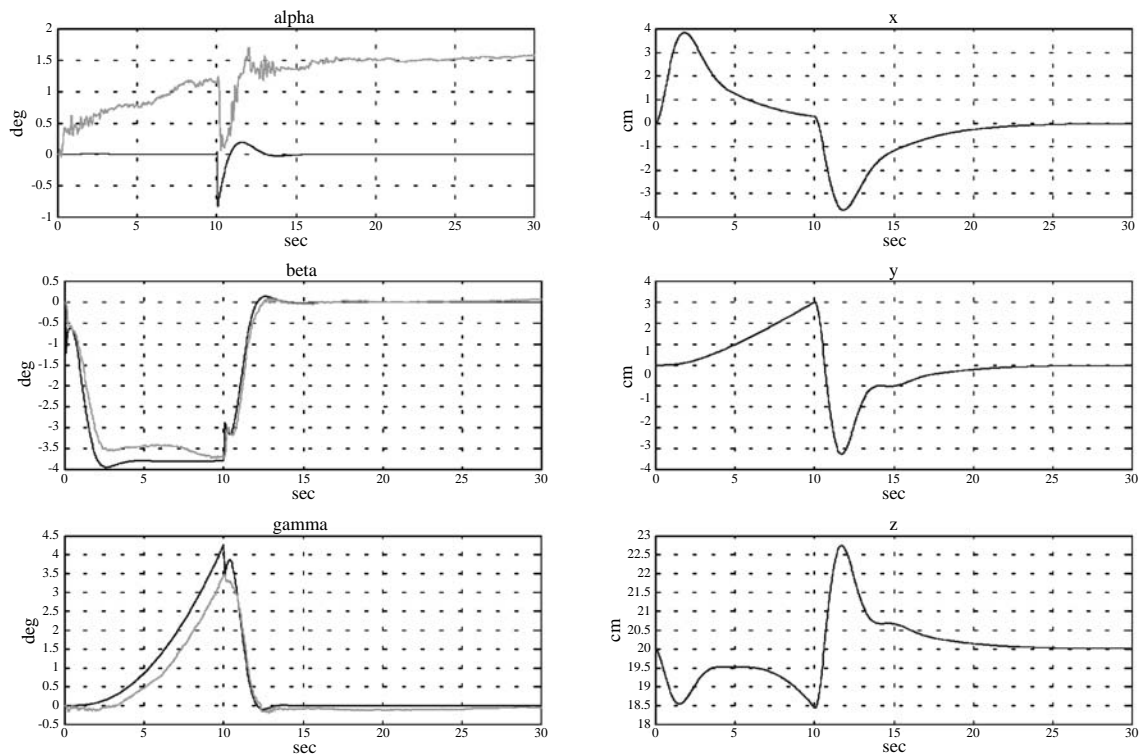


Fig. 21. Platform orientation/position commands and orientation responses.

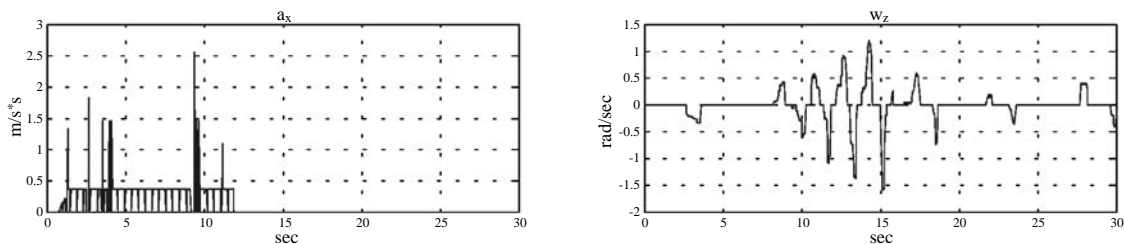


Fig. 22. Steering commands.

- Control of a Hydraulic Stewart Platform, Part I: Kinematics and Dynamics Analysis," *J. Contr. Syst. Technol.*, Vol.6, No.3, pp. 177-184 (1998).
12. Wen, C., C.-C. Hsu and I.-K. Fong, "Modeling and Control of a Hydraulic Stewart Platform, Part II: Modeling and Control of the Hydraulic Links," *J. Contr. Syst. Technol.*, Vol.6, No.3, pp.185-192 (1998).
  13. Foley, J.D., A. van Dam, S.K. Feiner, J.F. Hughes and R.L. Phillips, *Introduction to Computer Graphics*, Addison Wesley, New York (1994).
  14. Trujillo, S., *Direct 3D Programming*, The Coriolis Group, Scottsdale (1996).
  15. Dukkupati, R.V., M.O.M. Osman and S.S. Vallurupalli, "Adaptive Active Suspension to Attain Optimal Performance and Maintain Static Equilibrium Level,"

*Int. J. Vehicle Design*, Vol. 14, No. 5, pp. 471-496 (1993).

16. Pitman, G.R., *Inertial Guidance*, John Wiley, New York (1962).
17. Zacharias, G.L., "Motion Cue Model for Pilot-Vehicle Analysis," AMRL-TR-78-2 (1978).
18. Reid, L.D. and M.A. Nahon, "Flight Simulator Motion-Base Drive Algorithms: Part 1 – Developing and Testing the Equations," *UTIAS Report No.296*, CN ISSN 0082-5255 (1985).
19. Tseng, H.-L., "Implementation of a Driving Simulator Based on Stewart Platform and Computer Graphics Technologies," Master Thesis (in Chinese), Institute of Electrical Engineering, National Taiwan University (1999).

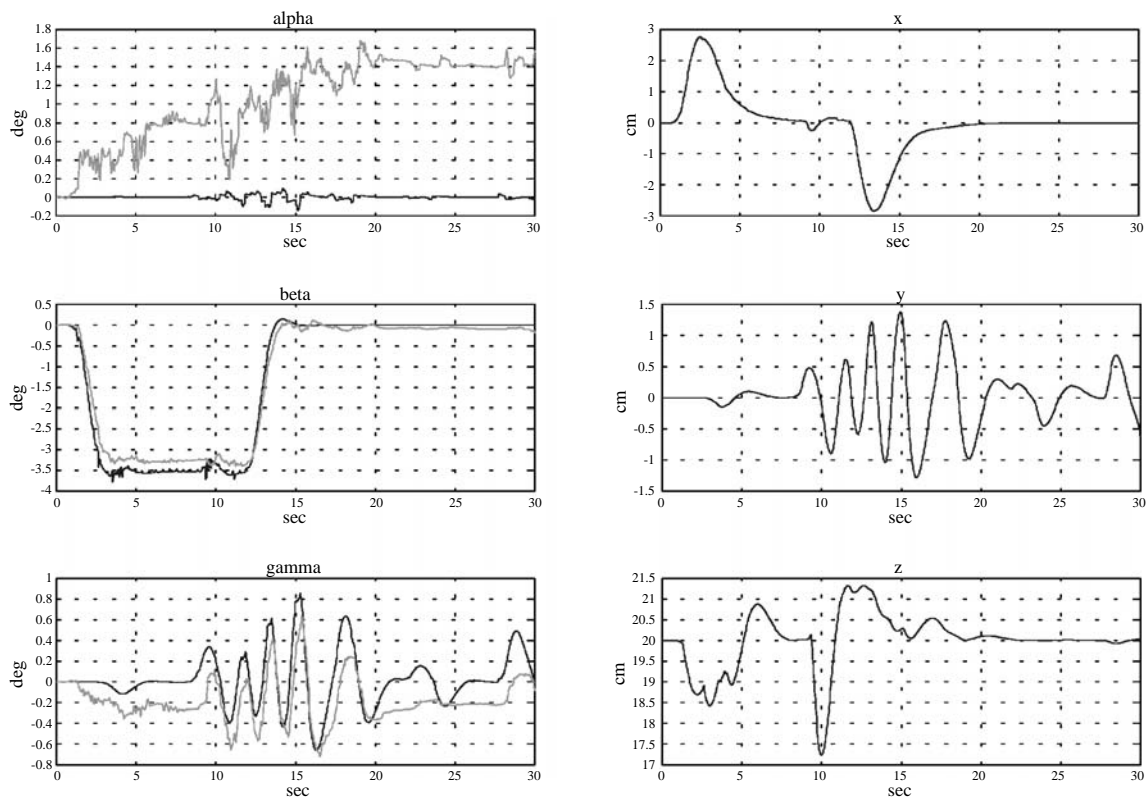


Fig. 23. Platform orientation/position commands and orientation responses.



**Hung-Lung Tseng** was born in 1971 in Taipei and received his M. Sc. from National Taiwan University in electrical engineering in 1999. Before receiving his M.S. degree, he held a research position in the industry. In the Institute of Electrical Engineering at NTU, he was a member

of the control group, and his main research topics included motion/visual cue technologies for virtual reality systems as well as the construction of a vehicle driving simulator using a Stewart platform in the Advanced Control Laboratory of the Institute. Currently, he works for the Nanya Technology Corporation as a Computer Integrated Manufacturing engineer.



**I-Kong Fong** received his B. Sc. and Ph.D. degrees, both in electrical engineering, from National Taiwan University, Taipei, in 1981 and 1986, respectively. From 1984 to 1986 and from 1987 to 1993, respectively, he was an instructor and Associate Professor in the Department of Electrical Engineering, National Taiwan University.

During 1986, he conducted research at the University of California, Davis, as a Research Associate. Since 1993, he has been a Professor in the Department of Electrical Engineering, National Taiwan University. Currently, his research interests include robust control theory, flight control systems, dynamic optimization methods, and decentralized control systems.