Robotica (2012) volume 30, pp. 709–719. © Cambridge University Press 2011
doi:10.1017/S0263574711000944

Design and control of a cable-controlled haptic motion simulator

M. Karkoub†,*, M-G. Her‡, C-C. Peng‡, C-C. Huang‡ and M-I. Ho§

†*Mechanical Engineering Program, Texas A & M University at Qatar, Doha, Qatar*
‡*Department of Mechanical Engineering, Tatung University, Taipei, Taiwan*
§*China University of Science and Technology, Taipei, Taiwan*

(Accepted July 7, 2011. First published online: August 16, 2011)

SUMMARY

In this work, we discuss the design, construction, and testing of a cable-controlled motion simulator for a Virtual Reality (VR) hang gliding environment. The system comprises a cable-controlled motion simulator, a joystick, a Lego™ direction sensor, and a VR environment. The motion simulator and joystick are built out of motors, pulleys, cables, and aluminum beams. The VR environment and motion simulator interact haptically with each other to give a realistic feel to the operator. A dynamic analysis is performed on the simulator to show the effect of gravity and the directional motion on the operator. A series of experiments are then performed to test the effectiveness of the cable-controlled simulator, and the results were very encouraging despite minor glitches with high-speed maneuvers.

KEYWORDS: Cable-controlled motion simulator; Visual helmet; Haptic interaction.

1. Introduction

For decades, cable-controlled systems have been used to perform special tasks efficiently. For example, cable-driven parallel mechanisms are used by many researchers due to their large motion ranges, low inertia, low cost, and robustness in case of interference.[1–4] They have the potential to overcome several of the drawbacks of conventional mechanisms, such as large accelerations over a large workspace. Cable-driven mechanisms have been used and studied in VR sports machine based on a wire-driven system,[5] cable-suspended haptic interface,[6] a high-speed manipulator using parallel wire-driven robots,[7] and an air vehicle simulator.[8] Viscomi *et al.*[9] described the construction of a cable-controlled Stewart platform crane for use in integrated building systems. The challenge for such constructions lies in that the cables have to be maintained in tension. Other systems that made use of cable-controlled mechanisms are the so-called cable-array robots.[10,11] Cable-array robots possess several advantages over traditional serial or parallel robot mechanisms because they can operate over much larger workspaces and provide higher performance in terms of relative stiffness and speed. Moreover, they have fewer moving parts and can handle large loads relative to the total robot weight. However, a lot of work remains for cable-array robots to overcome some of the known problems such as cable interference, inaccuracies at the end effector due to cable stretch, limited force application in the downward direction, and the requirement for an overhead space.

Many of these cable-controlled devices have been used to actuate VR applications. Virtual reality applications have mushroomed over the past few decades in terms of variety and usage. Many researchers developed VR applications that can be applied to the industrial sector, gaming industry, robotics, military, and medical fields.[12–17] For example, in the gaming industry, Wii™ is one of the most popular VR systems available in the market that allows the operator to control the characters' motion on a TV screen through a joystick (http://us.wii.com/). In military applications, the operator may simulate missile launch and target practice through VR environment. For example, Her *et al.*[18] developed a low cost image-tracking system to match a VR system simulating the launch of a shell from a naval vessel. In the medical field, researchers from Body Viz (http://www.bodyviz.com/) developed a 3D magnetic resonance imaging (MRI) VR system. The system shows human organs in a 3D VR environment, which help students and physicians alike learn about certain aspects of human body without need of a human subject.

It is often required to integrate haptic feedback with the VR environment to make simulation more realistic and enjoyable. Therefore, a haptic mechanism is needed to interface between the VR environment and the operator. Her *et al.*[19] used load cells to measure the force exerted by an operator on a joystick and used the results to simulate a hit on a ball. The VR environment, in turn, applies a reaction force resulting from hitting the ball on the operator hand through a force feedback joystick. Adamovich *et al.*[20] designed a device to help rehabilitate the arms of a stroke patient. The patient is able to see the path of the arm movement and feel the reaction forces through joystick. Tseng and Fong[21] used the Stewart platform along with VR environment to simulate driving a car. The operator (driver) receives vision feedback from the VR environment and acceleration feel from the Stewart platform.

In this work, we present a cable-controlled motion simulator used in conjunction with a VR environment to simulate hang gliding action or flight of an airship. The VR system includes motion information, force feedback, and visual and audio feedback. The motion information is transmitted from the cable-controlled motion simulator (the platform); however, the force feedback is from joystick. The visual and audio feedbacks are produced from a screen

* Corresponding author. E-mail: mansour.karkoub@qatar.tamu.edu

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

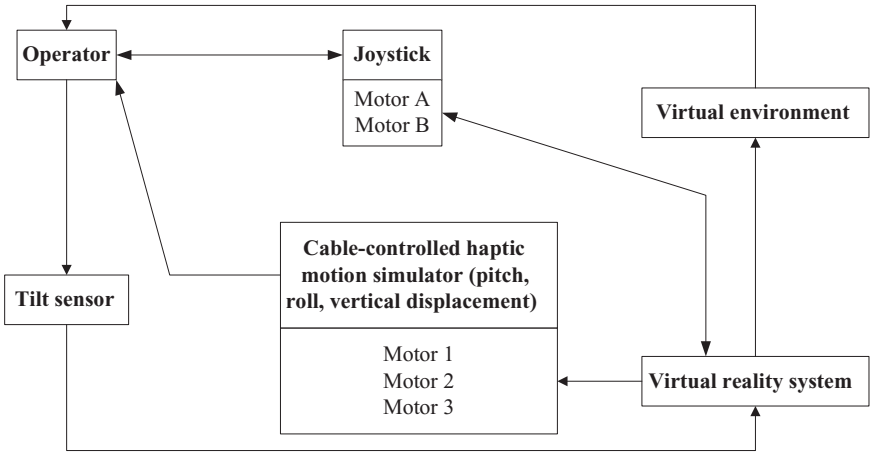


Fig. 1. Interconnection diagram of the cable-controlled motion simulator.

and a visual helmet or glasses, which provide some sense of reality (telepresence) of the actual environment. The haptic interaction between simulator and VR environment is created through force feedback from joystick. This feedback is essential in making the experience of the operator more realistic and more interesting.

The rest of the paper is organized as follows: The design of the cable-controlled motion simulator is presented in Section 2. Modeling of the dynamics of the VR airship and the simulator is done in Section 3. Displacement, velocity, and acceleration analyses of the airship and simulator are performed in Section 4. Some experimental results are presented in Section 5. Finally, some concluding remarks are given in Section 6.

2. Design of the Cable-Controlled Motion Simulator

The concept of the motion simulator system, including the VR environment, is shown in Fig. 1. An operator controls the joystick and tracks the scene with a visual helmet and the resulting motion data is relayed for processing to a computer. The joystick data is transformed into speed commands for VR environment and the order of the velocities is sent to the servomotors of the joystick and cable-controlled motion simulator. A tilt sensor is used to detect the rotation angle of operator's head and directly relays the data to VR environment. In order to make the environment impressive, a force feedback system is put in place between the simulator and the operator.

2.1. The joystick

The joystick is made of two servomotors, an aluminum rod, and two cables (see Figs. 2 and 3). The joystick has three operating modes: (1) position control, (2) speed control, and (3) torque control. In our experiment, position control is used for motion simulator. The joystick is used to control pitch motion in VR environment and also transmits the VR environment disturbances to the user, thus creating a haptic feedback. Figure 4 explains the process of information exchange between the joystick and VR environment.

Table I describes all the variables used in the derivation of dynamics and control equations of the simulator system, including the joystick. Figure 5 shows a free-body diagram of joystick, which is mainly an aluminum rod controlled by the operator through two motors, cables, and pulleys. When the operator moves the joystick, motor A rotates an angle ϕ_L and motor B rotates an angle ϕ_R. Assuming the gain between the motor and the joystick to be K_j, the rotation of the left side of the joystick is given by $K_j\phi_L$, and the rotation of the right side of the joystick is given by $K_j\phi_R$. The center rotation of the joystick ϕ_{CM} is

$$\phi_{CM} = K_j\left(\frac{\phi_R - \phi_L}{2}\right). \tag{1}$$

Now, assuming the cables between motors A and B and the joystick remain vertical at all times, the rotation of the joystick equals the difference in rotation between motors A and B, i.e., if the gain between the joystick and the motors is equal to 1, then the rotation of the center of the joystick could be written as follows:

$$\phi_{CM} = \frac{\phi_R - \phi_L}{2}. \tag{2}$$

A similar reasoning could be used to determine the operator's angle, θ. Therefore, assuming that the cables between motors 1, 2, and 3 and the operator remain vertical at all times, it could be easily shown that the operator's angle of rotation is

$$\theta = \frac{\phi_R - \phi_L}{2} = \phi_{CM}. \tag{3}$$

This means the input that operator gives to airship, u_p, is same as the joystick angle. In other words,

$$u_p = \frac{\phi_R - \phi_L}{2} = \phi_{CM}. \tag{4}$$

The input force to the airship is proportional to the input command and is given by

$$f_p = ku_p, \tag{5}$$

where k is a gain.

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

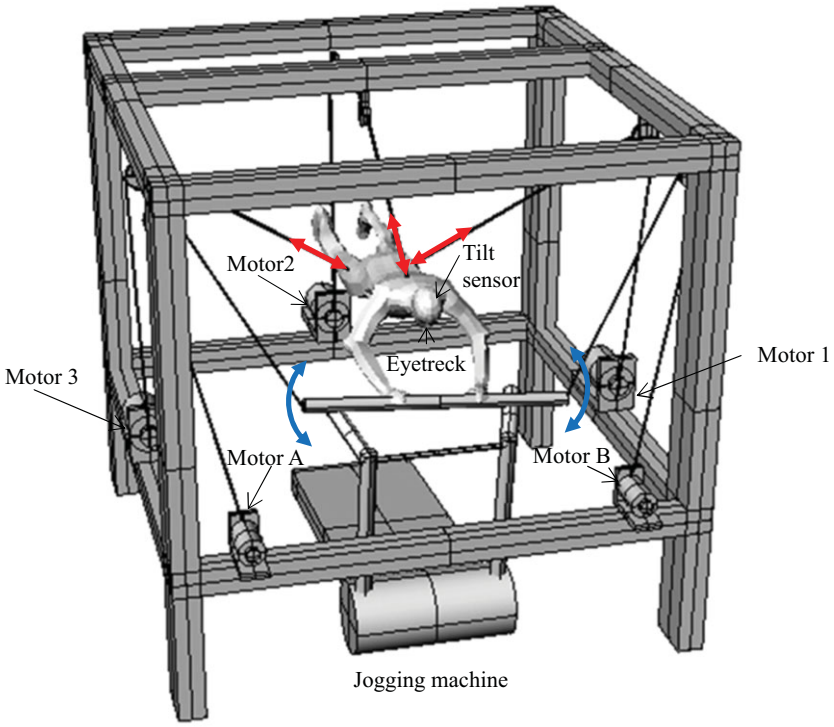


Fig. 2. (Colour online) Solid model of the cable-controlled motion simulator.

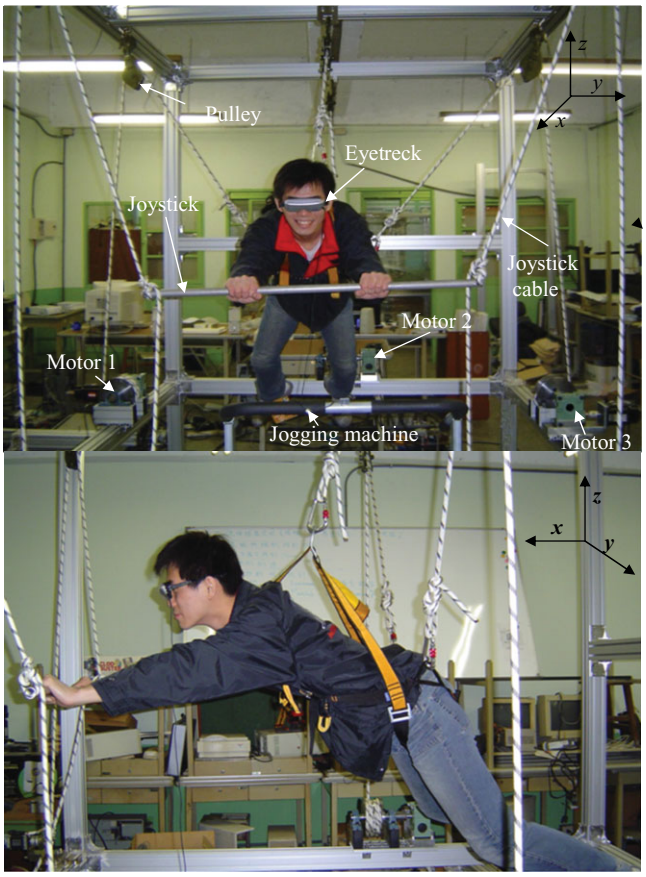


Fig. 3. (Colour online) Front view (top); side view of the cable-controlled motion simulator (bottom).

2.2. *Head motion detection*

The orientation of the operator's head, which indicates the direction of sight for VR environment, is measured via a Lego™ tilt sensor Model ADX311. The sensor detects motion in two directions only and provides direct input to the VR environment. This direct interaction between the tilt sensor and the VR environment eliminates the effect of simulator frame and screen size on the head motion. A video showing the interaction between the tilt sensor and VR environment can be found at http://www.youtube.com/watch?v=QddegBX8YzU This sensor is used to control roll and yaw angles of the airship.

2.3. *Cable-controlled motion simulator*

The cable-controlled motion simulator consists of aluminum beams, pulleys, and servomotors (see Figs. 2 and 3). The aluminum beams make up the frames, whereas the cables and pulleys and servomotors are used for motion detection and creation. The gliding motion of the simulator is achieved through pull forces of the cables created by the rotation of motors 1, 2, and 3 in clockwise and counterclockwise directions. The final component of the simulator is the jogging machine. This latter simulates the takeoff part of hang gliding action: The operator starts jogging until the speed is high enough for the glider to lift off. The operator then uses the joystick to maneuver simulated airship in all possible directions. The commands from the joystick and the Lego sensor are relayed to servomotors of the simulator to pull on the operator to make him/her feel the pitch, yaw, rolling, and vertical motions of the simulated airship.

2.4. *System integration*

The main goal of virtual reality system is to simulate various environments. More realistic the environment, greater effect the simulator has on the operator. In this work, only two VR environments are considered. The first VR environment deals

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

Table I. Nomenclature.

Var.	Definition	Var.	Definition
ϕ_{CM}	Rotation angle of the joystick's central point	V_x	Velocity of the VR environment in x-direction
ϕ_L	Rotation angle of motor A	V_z	Velocity of the VR environment in z-direction
ϕ_R	Rotation angle of motor B	ω	Angular velocity of the VR environment
K_j	Gain between motor and joystick	K	Gain between VR environment and motion simulator
m	Mass of virtual airship	a	Value of washout filter
I	Moment of inertia of the virtual airship in x–z plane	A_x^*	Acceleration of the motion simulator in x-direction
l	Half length of the airship	A_z^*	Acceleration of the motion simulator in z-direction
θ	Pitch angle of the airship	α^*	Angular acceleration of the motion simulator
f_d	Drag force received by the airship	V_x^*	Velocity of the motion simulator in x-direction
f_e	External force received by the airship	V_z^*	Velocity of the motion simulator in z-direction
f_p	Force given by the accelerator of the airship	ω^*	Angular velocity of the motion simulator
f_d^s	Drag force of airship in simulator coordinate system	x^*	Position of the motion simulator in x-direction
k_p	Gain of accelerator of airship	z^*	Position of the motion simulator in z-direction
u_p	Accelerator input	θ^*	Pitch angle of the motion simulator
ϕ	Angle of between the accelerator and the centerline of the airship	A_{z^s}	Acceleration of the VR environment in z^s-direction of operator
b_h	Drag force coefficient of wind in x-direction	α_s	Angular acceleration of VR environment in pitch direction of operator
b_s	Drag force coefficient of wind in z-direction	g_{x^s}	Gravity acceleration of VR environment in x^s-direction
b_θ	Drag force coefficient of wind in pitch direction	g_{z^s}	Gravity acceleration of VR environment in z^s-direction
b	Drag force coefficient of wind	A_s^*	Acceleration of motion simulator in operator's coordinate
y_s	Position of motion simulator	$A_{x^s}^*$	Acceleration of motion simulator in x^s-direction of operator
A_x	Acceleration of VR environment in x-direction	$A_{z^s}^*$	Acceleration of motion simulator in z^s-direction of operator
A_z	Acceleration of VR environment in z-direction	α_s^*	Angular acceleration of motion simulator in pitch direction of operator
α	Angular acceleration of VR environment	$g_{x^s}^*$	Gravity acceleration of motion simulator in x^s-direction
A_s	Acceleration of VR environment in operator's coordinate	$g_{z^s}^*$	Gravity acceleration of motion simulator in z^s-direction
A_{x^s}	Acceleration of VR environment inx^s-direction of operator		

with fast accelerating game systems, such as racecars and speedboats. These types of games require the environment to speed up instantly and steering wheels to be turned quickly. The simulation of this type of environment requires an acceleration-based analysis; consequently, the results are converted to instant forces and displacements in the vertical, longitudinal, and lateral directions on the operator's body. The second VR environment that will be considered in this work deals with slow moving types of games, such as hang gliding and underwater exploration. In this type of VR environment there is little acceleration and no sudden turning; therefore, the focus will be on gravity forces that the

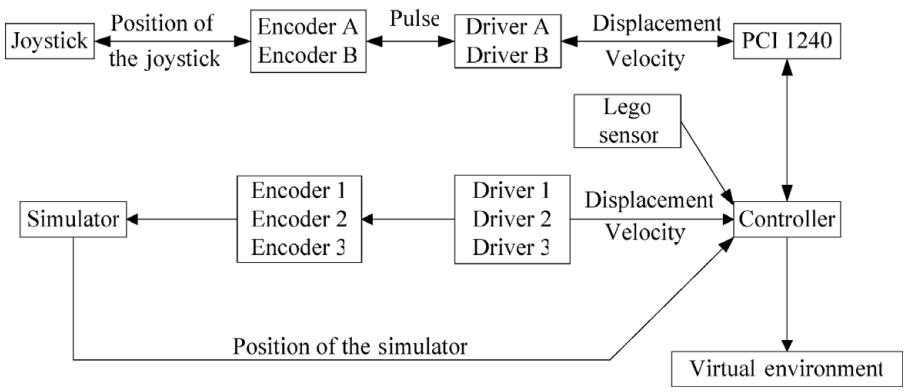


Fig. 4. Interaction flow diagram between joystick and simulator.

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

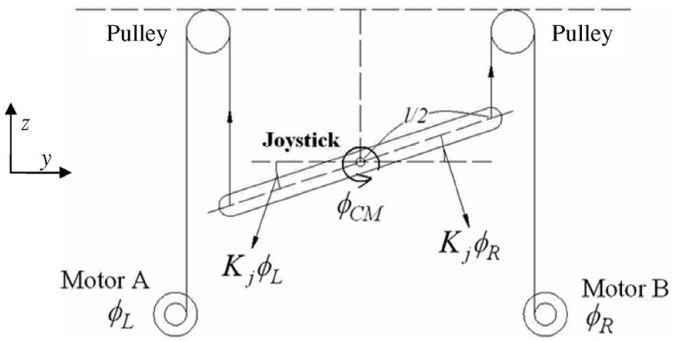


Fig. 5. Schematic diagram of joystick.

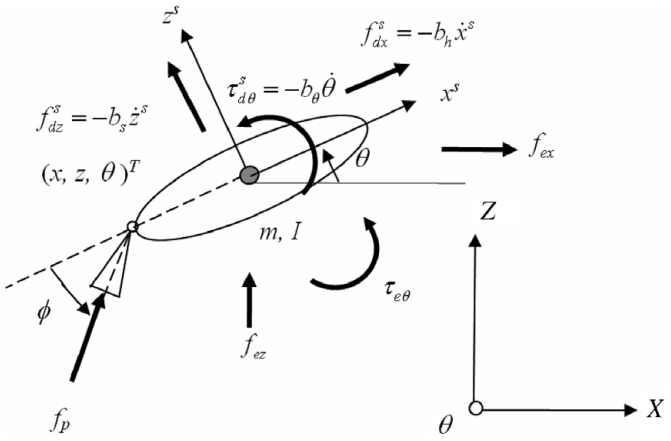


Fig. 6. Free-body diagram of airship/VR environment.

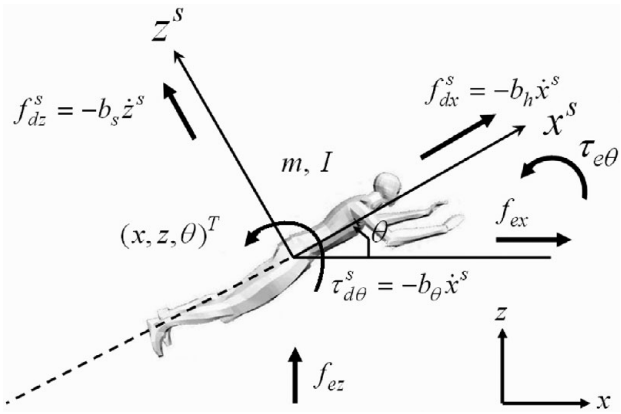


Fig. 7. Free-body diagram of operator/simulator.

operator feels. The gravity forces increase or decrease due to the operator's pitch, roll, and yaw angles. For economical reason, the same motors are used here for both slow and fast VR environments.

3. Dynamic Analysis of Airship for VR Environment

The variables used in the dynamic analysis of the simulator are listed in Table I. Figures 6 and 7 show the free-body diagram of airship and operator, respectively. The two systems would be analyzed simultaneously due to their interdependency. The equations of motion of the airship are obtained by summing forces in x- and z-directions, and the moment around y-axis (see Fig. 9); where f_{pi}, f_{di}, and f_{ei} are the i-direction components of accelerator, drag, and external forces. τ_{pi}, τ_{di}, and τ_{ei} are accelerator, drag, and external moments, respectively, and I is the mass moment of inertia of the airship.

From Fig. 6, the accelerator force f_p creates one force in x-direction f_{px}, one force f_{pz} in z-direction, and a torque $\tau_{p\theta}$ around y-axis, whose values are given by the following equation:

$$\{F_p\} = \{f_{px} \quad f_{pz} \quad \tau_{p\theta}\}^T = \{k_p u_p \cos(\theta+\phi)\ k_p u_p \sin(\theta+\phi)\ -k_p u_p l \sin(\phi)\}^T, \tag{6}$$

where ϕ is an angle between accelerator and centerline of the airship. Similarly, one can determine the components of the drag force f_d. The viscous damping forces in the simulator coordinate system are given by

$$\left\{F_d^s\right\} = \left\{b_x\dot{x}^s \quad b_z\dot{z}^s \quad b_\theta\dot{\theta}^s\right\}^T,$$

where $\dot{x}^s, \dot{z}^s, \dot{\theta}^s$ and b_x, b_z, b_θ denote the speeds and the coefficients of viscous damping in corresponding directions. Therefore, the representation of these forces in the Cartesian coordinate system (o, x, z) are given by

$$\{F_d\} = \{f_{dx} \quad f_{dz} \quad \tau_{d\theta}\}^T = -[R]\left\{F_d^s\right\}, \tag{7}$$

where

$$[R] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In order to obtain drag force components in the Cartesian coordinate system as a function of the same coordinate speeds, a transformation of coordinates from the absolute coordinate system (o, x, z) to the simulator coordinate system (o^s, x^s, z^s) is required. The transformation is given by

$$\{\dot{X}^s\} = \{\dot{x}^s \quad \dot{z}^s \quad \dot{\theta}^s\} = [R]^T\{\dot{X}\}, \text{ where } \{\dot{X}\} = \{\dot{x} \quad \dot{z} \quad \dot{\theta}\}^T. \tag{8}$$

Therefore, Eq. (8) can be rewritten as follows:

$$\{F_d\} = \{f_{dx} \quad f_{dz} \quad \tau_{d\theta}\}^T = -[R][B][R]^T\{\dot{X}\},$$
$$\text{where} \quad [B] = \begin{bmatrix} b_h & 0 & 0 \\ 0 & b_s & 0 \\ 0 & 0 & b_\theta \end{bmatrix}. \tag{9}$$

Equation (6) can be rewritten as follows:

$$[M]\{\ddot{X}\} + [R][B][R]^T\{\dot{X}\} = \{F_p\} + \{F_e\}, \tag{10}$$

where the inertia matrix [M] of the airship is given as $[M] = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}$ and the external force vector $\{F_e\}$ is given by $\{F_e\} = \{f_{ex} \quad f_{ex} \quad \tau_{e\theta}\}^T$.

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

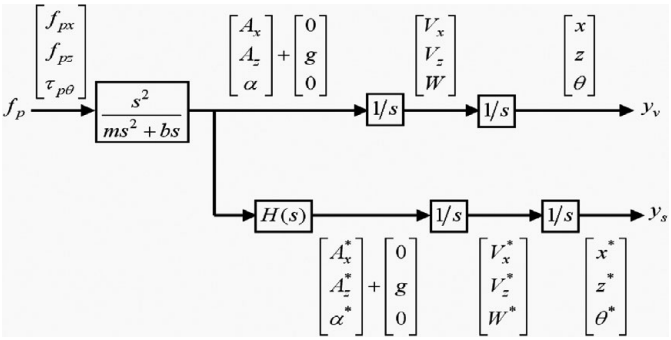


Fig. 8. Block diagram for the force to displacement conversion between motion simulator and VR environment.

For simplicity and no loss of generality, let us assume that the wind drag coefficients in x- and z-directions to be the same and that $b_h = b_s = b$,[22] and that no external force is applied on the airship, i.e.,

$$\{f_e\} = \{0 \quad 0 \quad 0\}^T.$$

Then

$$[M]\,\{\ddot{X}\} + [R]\,[B]\,[R]^T\{\dot{X}\} = \{F_p\}. \tag{11}$$

Remark 1:
Equation (11) represents three decoupled equations of motion in x-, z-, and $\theta-$directions. From here onwards, only one direction will be discussed. Motions in other directions are analyzed and synthesized in the same manner.

3.1. Transfer functions of the motion simulator system
Equation (11) represents three decoupled equations and each equation can be written as follows:

$$m\ddot{\eta} + b\dot{\eta} = f_p. \tag{12}$$

Taking Laplace Transform of Eq. (12) leads to

$$G(s) = \frac{\eta(s)}{F_p(s)} = \frac{1}{ms^2 + bs}. \tag{13}$$

The acceleration transfer function is given by $A(s) = s^2G(s)$.

The conversion diagram for the VR environment and the simulator is shown in Fig. 8. The accelerator force f_p is fed to block $A(s)$ whose outputs are the accelerations $\{A_x, A_z, \alpha\}$ in x-, z-, and θ-directions, respectively, in addition to the gravitational acceleration g. Integrating the output of $A(s)$ leads to velocities $\{V_x, V_z, \omega\}$; integrating once more leads to displacements $\{x, z, \theta\}$ of VR environment. In order to obtain the displacements $\{x^*, z^*, \theta^*\}$ of simulator, a similar procedure is followed. That is, the outputs of the acceleration transfer function $A(s)$ are processed through the washout filter transfer function $H(s)$ and the outputs of $H(s)$[23] are integrated twice to obtain $\{x^*, z^*, \theta^*\}$,

$$H(s) = \frac{Ks}{s + a}. \tag{14}$$

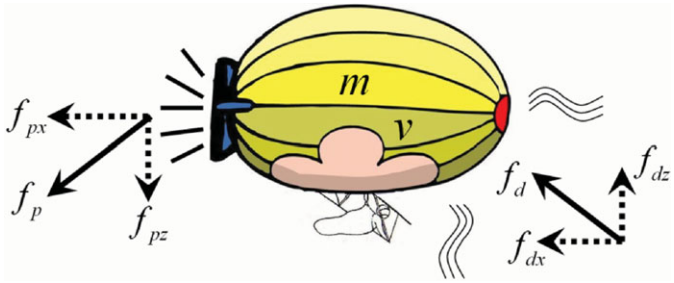


Fig. 9. (Colour online) Free-body diagram of VR environment airship.

The parameters of the washout filter are chosen such that the motion of the user in the simulator space is within a safe working area.

3.2. Controllability and observability of motion simulator system
In this section, we show that the simulator system is controllable as well as observable. The system's state space representation can be obtained using Eq. (12).

Let $z_1 = \eta$ and $z_2 = \dot{\eta}$, one can rewrite Eq. (12) as follows:

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -b/m\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}0\\ 1/m\end{bmatrix} f_p, \tag{15}$$

$$\eta = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix}. \tag{16}$$

Using Eqs. (15) and (16), the state space matrices A, B, and C are given by

$$A = \begin{bmatrix}0 & 1\\ 0 & -b/m\end{bmatrix}, \quad B = \begin{bmatrix}0\\ 1/m\end{bmatrix}, \quad C = \begin{bmatrix}1 & 0\end{bmatrix}.$$

The controllability matrix is given by

$$Q_c = [B \quad AB] = \begin{bmatrix}0 & 1/m\\ 1/m & -b/m^2\end{bmatrix}. \tag{17}$$

Since the controllability matrix has a rank of 2 (rank $(Q_c) = 2$), the system is controllable. Similarly, one can show that the simulator system is observable.

4. Displacement, Velocity, and Acceleration Analyses of Airship and Simulator

4.1. Displacement analysis of VR environment and motion simulator
The relationship between the displacements of the VR environment and motion simulator is given by

$$Z^* = \left(\frac{Ks}{s + a}\right) Z. \tag{18}$$

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

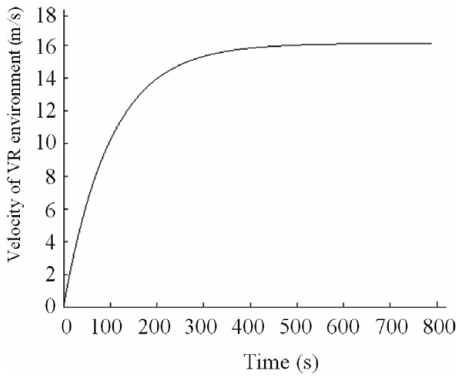


Fig. 10. Velocity of VR environment.

Using Eqs. (11) and (18) and assuming as step input u_p, leads to

$$Z^* = \frac{Ks}{s+a} \cdot \frac{k_p \sin(\theta+\phi)}{ms^2+bs} \cdot U_p(s) = \frac{k'_p K}{(s+a)(ms+b)} \frac{u_p}{s} = \frac{k'_p u_p K}{s(s+a)(ms+b)}. \quad (19)$$

Therefore, using the final value theorem, the maximum displacement of the motion simulator is given by

$$z^*_{\max} = \lim_{s\to 0} s \cdot \left(\frac{k'_p u_{p\max} K}{s(s+a)(ms+b)} \right) = \frac{u_{p\max} k'_p K}{ab}. \quad (20)$$

Knowing the space limitations of the motion simulator (see Fig. 10), the maximum allowed displacement should not exceed the space available, z_{limit}. Therefore, one can obtain the washout filter constant a as follows:

$$z^*_{\max} < z_{\text{limit}}$$

$$\frac{u_{p\max} k'_p K}{ab} < z_{\text{limit}} \quad \Rightarrow \quad K < \frac{ab z_{\text{limit}}}{k'_p u_{\text{pmax}}}, \quad (21)$$

$$\Rightarrow a > \frac{K k'_p u_{\text{pmax}}}{b z_{\text{limit}}}. \quad (22)$$

4.2. Velocity analysis of VR environment and motion simulator

Referring back to Eq. (11), and taking Laplace Transform of the equation of motion in z-direction leads to

$$\frac{Z(s)}{F(s)} = \frac{1}{ms^2+bs} \Rightarrow \frac{Z(s)}{U_p(s)} = \frac{k_p \sin(\theta+\phi)}{ms^2+bs} = \frac{k'_p}{ms^2+bs}, \quad (23)$$

or,

$$Z(s) = \frac{k'_p}{ms^2+bs} U_p(s), \quad (24)$$

$$V_z(s) = \frac{k'_p}{ms+b} \frac{u_p}{s}. \quad (25)$$

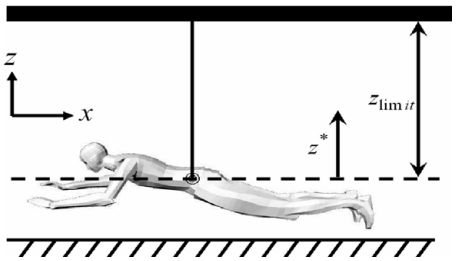


Fig. 11. Space limits of motion simulator.

The time domain response of the airship $z(t)$ can be obtained by taking inverse Laplace Transform of Eq. (25):

$$z(t) = L^{-1}\left(\frac{k_p}{ms^2+bs} \frac{u_p}{s} \right), \quad (26)$$

$$v_z(t) = L^{-1}\left(\frac{k_p}{ms+b} \frac{u_p}{s} \right). \quad (27)$$

When $t \to \infty$, the velocity of the airship is given by

$$v_z(t) = \lim_{s\to 0}(s) \left(\frac{k_p}{ms+b} \right) \left(\frac{u_p}{s} \right) = \frac{k'_p u_p}{b}. \quad (28)$$

Similarly, one can obtain the dynamics of the airship in other directions. A typical mass of the airship is about 6000 kg and the flying speed is about 16.67 m/s. (www.goodyearblimp.com). Using Eq. (28) and assuming $k'_p = 100$, $u_p = 10$, one can get the drag coefficient in z-direction as follows:

$$v_z(t) = \frac{k'_p u_p}{b} = \frac{(100)\,(10)}{b} = 16.67 \text{ m/s} \Rightarrow \quad b = 60.$$

The z-direction velocity of the VR environment as a function of time is shown in Fig. 11.

4.3. Acceleration analysis of VR environment and motion simulator

A washout filter with a transfer function equal to $Ks/(s+a)$ is used as an interface between the motion simulator and the VR environment. As a result, when operator moves the joystick slowly, which is a low-frequency action, $Ks/(s+a)$ approaches zero. Therefore, the motion simulator will remain fixed and the operator will sense speed visually from VR environment. However, if the operator moves the joystick very fast, which is considered a high-frequency action, $Ks/(s+a)$ approaches K. That means there is a K-fold relationship between the motion simulator and the VR environment.

Let V_z and A_z denote the velocity and the acceleration of the VR environment, respectively. Assuming zero initial conditions and the movement of the joystick representing velocity command to VR environment, one can write,

$$A_z = sV_z. \quad (29)$$

Now, let A^*_z and Z^* denote the acceleration and the displacement of the motion simulator, respectively. One can

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

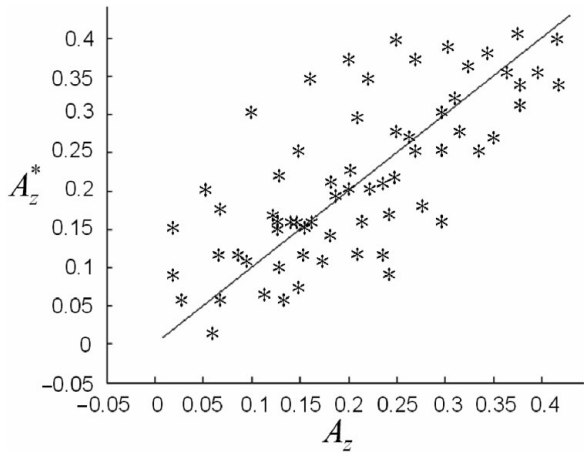


Fig. 12. Experimental accelerations of VR environment and motion simulator.

write,

$$Z^* = \frac{K}{s+a} V_z. \tag{30}$$

The relationship between the displacement and the acceleration of the motion simulator is given by

$$A_z^* = s^2 Z^* = \frac{K s^2}{s+a} V_z. \tag{31}$$

The ratio of displacements between the motion simulator and the VR environment is given by

$$\frac{A_z^*}{A_z} = \frac{K s}{s+a}. \tag{32}$$

Therefore, when the operator moves the joystick rapidly, he feels the acceleration of both VR environment and motion simulator when A_z^*/A_z approaches K. However, when the joystick moves slowly, A_z^*/A_z approaches zero, which means that the operator only feels the acceleration of the VR environment. The accelerations of the simulator and VR environment were measured and plotted against each other for different command inputs and the results are shown in Fig. 12. The trend of the ratio A_z^*/A_z shows a proportional distribution, which supports our analysis, i.e., when the input of the system is of high frequency, the magnitude of A_z^*/A_z approaches one and the phase approaches zero. This means, the simulator and VR environment will be in phase no matter how much the operator accelerates or decelerates.

From Fig. 13, the acceleration of the airship felt by the operator inside the airship is given by

$$[A_s + g_s] = \begin{bmatrix} A_{x^s} + g_{x^s} \\ A_{z^s} + g_{z^s} \\ \alpha_s \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} A_x \\ A_z + g \\ \alpha \end{bmatrix}. \tag{33}$$

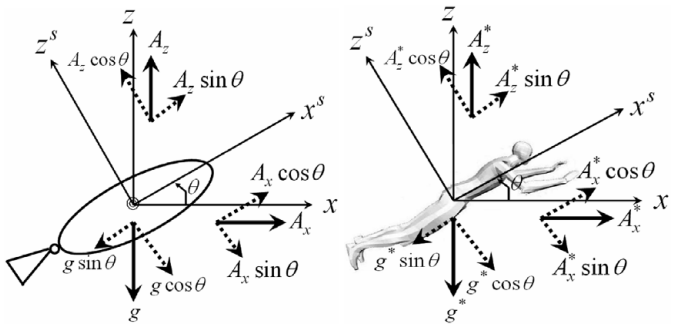


Fig. 13. Acceleration-based free-body diagrams of VR environment and motion simulator.

The acceleration of the motion simulator felt by the operator is given by

$$[A_s^* + g_s^*] = \begin{bmatrix} A_{x^s}^* + g_{x^s}^* \\ A_{z^s}^* + g_{z^s}^* \\ \alpha_s^* \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} A_x^* \\ A_z^* + g \\ \alpha^* \end{bmatrix}. \tag{34}$$

The maximum acceleration that the operator feels is 9.81m/s^2 in the z^s-direction of the VR environment (see Fig. 14). The acceleration will go through the washout filter and the maximum acceleration felt by the operator will be 9.755m/s^2 in the z^s-direction of the motion simulator.

Now, consider the case where the directional accelerations are set to zero. Therefore, from Eq. (40) only gravity will have an effect on the operator and the resultant acceleration will be

$$[g_s^*] = \begin{bmatrix} g_{x^s}^* \\ g_{z^s}^* \\ \alpha_s^* \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}. \tag{35}$$

The maximum acceleration felt by the operator in z^s-direction of the motion simulator is $g_{z^s}^* = 9.8\ \text{m/s}^2$. Therefore, the error resulting from neglecting the directional accelerations can be calculated as follows:

$$\text{Error} = \left| \frac{g_{z^s}^* - \left(A_{z^s}^* + g_{z^s}^*\right)}{A_{z^s}^* + g_{z^s}^*} \right| \times 100\% = \left| \frac{9.8 - 9.7555}{9.7555} \right| \times 100\% = 0.46\%.$$

This is a very small error and it is safe to assume that the operator mainly feels the effect of the acceleration of gravity. Based on the above analysis, motion in x- and z-directions could be neglected, thus reducing the cost of the motion simulator.

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

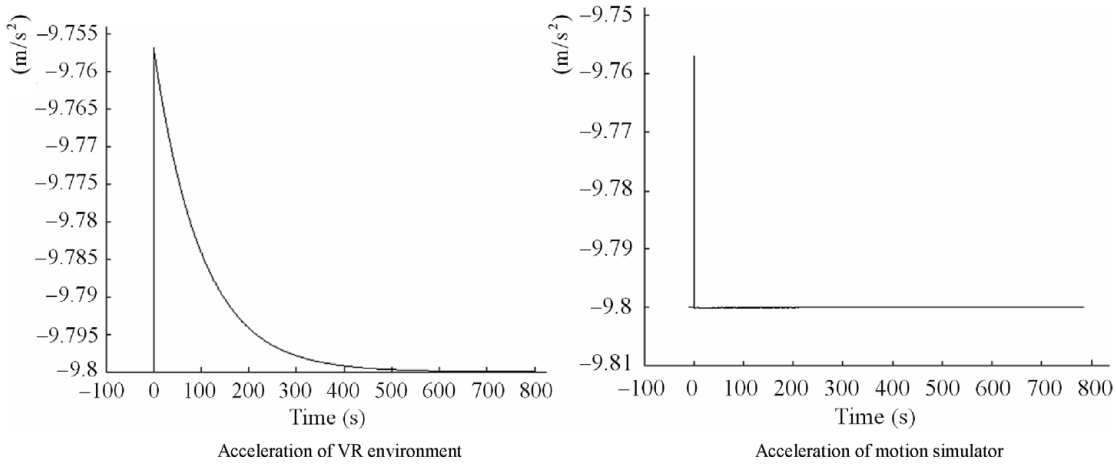


Fig. 14. Acceleration response of both VR environment and motion simulator.

4.4. Haptic feedback and closed-loop control of the simulator

A closed-loop controller is designed based on our earlier work published in ref. [24] and the block diagram is shown in Fig. 15. The user is able to feel the force feedback from the joystick and the motion simulator through transfer functions T_{hj} and T_{hs}, respectively. The transfer functions C_s and C_j represent the closed-loop controllers for simulator and joystick, respectively. The environment dynamics is represented by the transfer functions T_{ej} and T_{es} whereas K_{ej} and K_{es} represent the compensator transfer functions for the joystick and simulator, respectively. F_j and F_s are the joystick and simulator forces, whereas the transfer functions T_{ej}, T_{es}, K_{ej}, and K_{es} provide vision feedback to the operator.

The closed-loop control system includes two levels of haptic feedback. One is the joystick force feedback, and the other is the cable force feedback. When the airship is flying in a windy VR environment, the computer sends a command to motors A and B to apply a torque on the joystick in the direction opposite to that intended by the user, therefore creating a situation where maneuvering the airship takes more effort from the user than under normal conditions. Motors A and B work in tandem to create the necessary effect on the joystick. Similarly, motors 1 to 3 are used to transmit the VR flight conditions (force feedback) onto the user through the cables attached to the user and the motors. The washout filter $K_{hs} = Ks/s + a$ guarantees that the user's motion remains in the safe working space.

Experimental results

In this section, we discuss some experimental results from the designed motion simulator and the VR environment simulating an airship flight. The simulator movement is based on command inputs from joystick and helmet and interlinked with the VR environment motion through various transfer functions described in the previous sections. A set of experiments were conducted on the motion simulator and

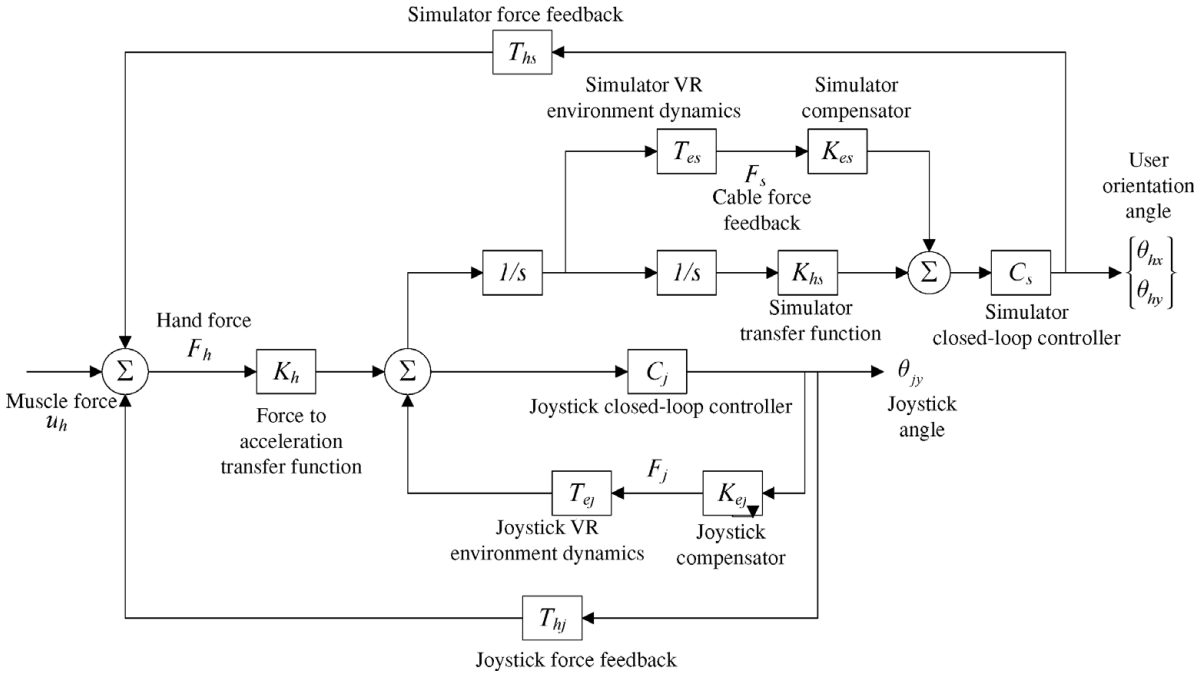


Fig. 15. Closed-loop control diagram between joystick, simulator, and VR environment.

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

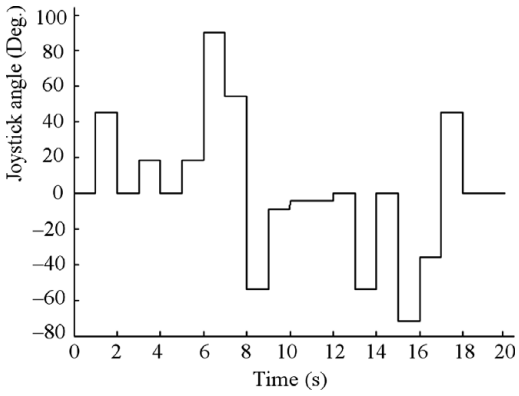


Fig. 16. Joystick command input angle.

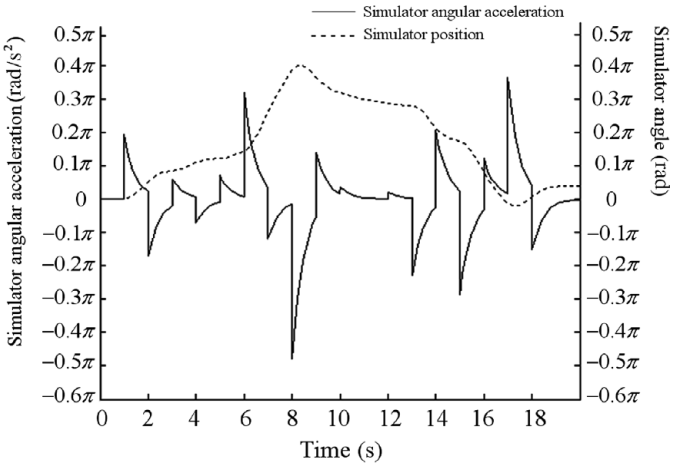


Fig. 17. Position and angular acceleration of motion simulator in response to the joystick command shown in Fig. 16.

the results are shown in Figs. 15 and 16 using a gain of one for the washout filter, i.e., $K = 1$. The joystick angular input by the operator is shown in Fig. 15, and the resulting angular acceleration and displacement of the motion simulator are shown in Fig. 16. Although the VR environment and motion simulator are in different spaces, the operator feels the displacements of both 1:1 as indicated in Eq. (32). It is clear from Fig. 17 that the simulator response to a command input is of the first-order type in accordance with Eq. (25). Moreover, the input command, which is of same type as the VR environment, is in phase with the simulator response.

Another objective of this work is modeling the limited movement of the simulator by an unlimited one in the VR environment. Therefore, an investigation of the effect of the washout filter constant a could shed light on the effectiveness of this type of filter as an interface between the VR environment and the motion simulator. When $K = 1$, the maximum displacement of the simulator is set by adjusting the value of a (refer to Eqs. (20) and (22)). When a is small, the simulator working area becomes large; therefore, a minimum value of a is needed to set the maximum range of the simulator's movement. For example, let us assume that the maximum displacement of the simulator is 2 m ($z_{\text{limit}} = 2$), the input signal of the airship is a unit step with final value of 10 ($u_{p\max} = 10$), and the initial angle to be $\theta_o = \pi/4$. Using Eq. (22), one can determine the minimum

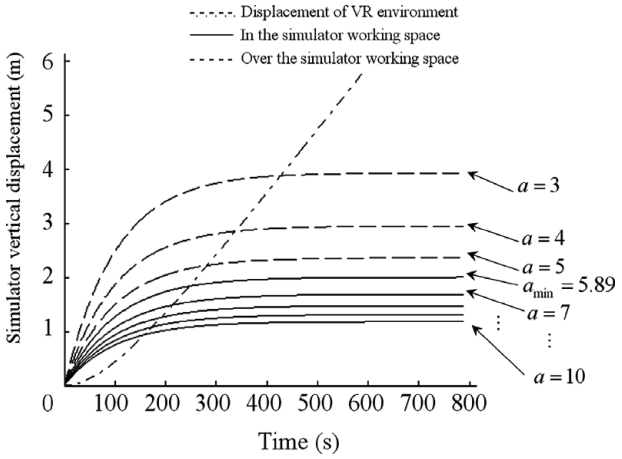


Fig. 18. Time response of motion simulator and VR environment as a function of the washout filter parameter a.

value of a as follows:

$$a_{\min} = \frac{K k'_p u_{p\max} \sin\theta}{b z_{\text{limit}}} = \frac{(1)(100)(10)\sin(\pi/4)}{(60)(2)} = 5.89.$$

The simulator will move less than its maximum displacement of 2 m if $a > 5.89$; however, if $a < 5.89$, the displacement of the simulator will be larger than 2 m, which could lead to unsafe scenarios for operator. Figure 18 shows the effect of the values of constant a on simulator and the VR environment displacements.

Remark 2
It was noticed during the experiment that the response of the motors is slow, which delays the reaction of the operator. Moreover, it was noticed at times that the joystick does not deliver the correct force to the motors due to slack in cables. This was usually the case when the operator makes a fast turn; however, when simulating the airship flight, these aforementioned problems did not occur due to slow required maneuvers.

5. Concluding Remarks

A cable-controlled motion simulator is designed, built, and tested in our laboratory. The simulator is made up of three major subsystems: VR environment, cable-controlled motion simulator, and joystick. The motion simulator and the joystick are made of cables, pulleys, motors, and aluminum frames. A closed-loop control system is designed taking advantage of the decoupled dynamics of the system and the interaction between the VR environment and the motion simulator. Haptic feedback is also introduced in the system to give realistic feel to the operator. It was possible to simulate the unlimited motion of VR environment with limited movement of the cable-controlled simulator, thanks to an interfacing washout filter. Also, although the tilt sensor can sense motion in two directions only, it was enough for the VR environment to simulate motion in all three directions (x, y, θ). Efforts are underway to improve the current simulator design. These include integration of all accelerations, including the lateral ones, omni-directional

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press

projection instead of the eyetreck, and better motor–cable interaction. Videos of our experiment are posted at http://www.youtube.com/watch?v=QddegBX8YzU and http://www.youtube.com/watch?v=SG9Ly4cZ2yc

References

1. S. E. Landsberger and T. B. Sheridan "A Minimal, Minimal linkage: the Tension-Compression Parallel Link Manipulator," **In**: *Proceedings of the IMACS/SICE International Symposium on Robotics Mechatronics and Manufacturing Systems*, Kobe-Japan (1992) pp. 81–88.
2. S. E. Landsberger, "Design and Construction of a Cable-Controlled Parallel-Link Manipulator" *Masters Thesis* (Mechanical Engineering Department, Massachusetts Institute of Technology, USA, 1984).
3. C. Gosselin and S. Bouchard, "A gravity-powered mechanism for extending the workspace of a cable-driven parallel mechanism: Application to the appearance modeling of objects", *Int. J. Autom. Technol.* **4**(4), 372–379 (2010, July).
4. E. Ottaviano, M. Ceccarelli, A. Paone and G. Carbone, "A Low-Cost Operation 4-Cable Driven Parallel Manipulator," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation ICRA05*, Barcelona, Spain (2005) pp. 4019–4024.
5. S. Kawamura, M. Ida, T. Wada and J.-L. Wu, "Development of A Virtual Sports Machine using A Wire Drive System: A Trial of Virtual Tennis", **In**: *IEEE Proceedings of the International Conference on Intelligent Robots and Systems* (1995) pp. 111–116.
6. R. L. Williams II, "Planar Cable-Suspended Haptic Interface: Design for Wrench Exertion," *Proceedings of the ASME Design Technical Conferences, 25th Design Automation Conference*, DETC99/DAC-8639, Las Vegas, NV, USA (1999).
7. S. Kawamura, H. Kino and C. Won, "High-speed manipulation by using parallel wire-driven robots," *Robotica* **18**(1), 13–21 (2000).
8. K. Usher, G. Winstanley, P. Corke and D. Stauffacher, "Air Vehicle Simulator: An Application for a Cable Array Robot," **In**: *IEEE: Proceedings of the International Conference on Robotics and Automation (ICRA05)*, Barcelona, Spain (Apr. 2005) pp. 2253–2258.
9. B. V. Viscomi, W. D. Michalerya and L.-W. Lu, "Automated construction in the ATLSS integrated building systems," *Autom. Constr.* **3**(1), 35–43 (1994).
10. A. Reichel and I. Ebert-Uphoff, "Force-Feasible Workspace Analysis for Underconstrained Point-Mass Cable Robots," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA (2004) pp. 4956–4962.
11. P. Bosscher and I. Ebert-Uphoff, "Wrench-Based Analysis of Cable-Driven Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA (2004) pp. 4950–4955.
12. A. R. W. Huang and C. Chen, "A low-cost driving simulator for full vehicle dynamics simulation," *Trans. Veh. Technol. IEEE* **52**(1), 162–172 (2003).
13. R. G. MenCndez and J. E. Bernard, "Flight simulation in synthetic environments." *Trans. Aerosp. Electron. Syst. Mag. IEEE* **16**(9), 19–23 (2001).
14. M. S. Kadavasal and J. H. Oliver, "Sensor augmented virtual reality based teleoperation using mixed autonomy," *J. Comput. Inf. Sci. Eng.* **9**(1), 014502 (2009).
15. L. K. Chen and A.G. Ulsoy, "Identification of a driver steering model, and model uncertainty, from driving simulator data," *J. Dyn. Syst. Meas. Contr.* **123**(4), 623–629 (2001).
16. K. Watanuki, "Development of virtual reality-based universal design review system," *J. Mech. Sci. Technol.* **24**(1), 257–262 (2010).
17. P. Lamb and D. Owen, "Human Performance in Space Telerobotic Manipulation," **In**: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, Monterey, CA, USA (Nov. 7–9, 2005) pp. 31–37.
18. M. G. Her, M. Karkoub and J. M. Chen, "Design and application of a low cost visual tracking system," *Aust. J. Electr. Electron. Eng.* **4**(3), 1–10 (2008).
19. M-G. Her, K-H. Hsu, and T-S. Lan, "Virtual reality application for direct drive robot with force feedback," *Int. J. Adv. Manuf. Technol.* **21**(1), 66–71 (2003).
20. S. V. Adamovich, G. G. Fluet, A. S. Merians, A. Mathai and Q. Qiu, "Incorporating haptic effects into three-dimensional virtual environments to train the hemiparetic upper extremity," *Trans. Neural Syst. Rehabil. Eng. IEEE* **17**(5), 512–520 (2009).
21. H-L. Tseng and I-K. Fong, "Implementation of a driving simulator based on a stewart platform and computer graphics technologies," *Asian J. Control* **2**(2), 88–100 (2000).
22. L. Konstantinov, "The Basics of Gas and Heat Airship Theory," *Montgolfier* # 1 (AEROPLAST Inc., Kyiv, Ukraine, 2003) pp. 1–21.
23. F. Barbagli, D. Ferrazzin, C. A. Avizzano and M. Bergamasco, "Washout Filter Design for a Motorcycle Simulator," *Proceeding of the IEEE Virtual Reality* (2001) pp. 225–232.
24. M. Karkoub, M-G. Her and J-M. Chen, "Design and analysis of a haptic interactive motion simulator for virtual entertainment systems," *Robotica* **28**(1), 47–56 (2009).

https://doi.org/10.1017/S0263574711000944 Published online by Cambridge University Press