A Novel Washout Filter Design for a Six Degree-of-Freedom Motion Simulator*

Chung-Shu LIAO**, Chih-Fang HUANG*** and Wei-Hua CHIENG**

The motion cue performed by a motion simulator is restrained by the workspace of the simulator structure. A typical reasoning is then to build a large motion simulator unless it is for entertainment, which involves in small simulators. This study proposes a novel approach to designing the washout filter of the motion control of a six degree-of-freedom motion simulator for entertainment purposes. Using information obtained from the inverse kinematics of the simulator, the workspace boundary, detected in real-time, is fed into the washout filter as a reference for the motion planning. The main focus of this approach is to make the motion cue feasible for use in a simulator with a restricted workspace, while ensuring the robustness of the driving system. In this paper, different indices are established to specify the performance of the motion cue. A classical linear washout filter was implemented and compared with the proposed washout filter using the performance indices to demonstrate the benefits of the latter.

***Key Words*:** Motion Cue, Motion Simulator, Washout Filter, Motion Planning, Motion Control

1. Introduction

The purpose of the washout filter is to transform trajectories generated by a dynamic model of virtual reality (VR), which incorporates very large displacements, into driving system commands that can give a pilot realistic motion cues while remaining within the simulator's limited workspace.

Designing an efficient washout filter is a complex problem. These filters are first of all complex control systems whose robustness and stability must be ensured to prevent mechanical damage to the simulator. Furthermore, designs of washout filters must take into account the spatial-disorientation of the pilot making a "realistic" simulation hard to define. The problem's complexity derives from human factors and the human-machine interaction. Many schemes have been presented in the last 20 years. Classical washout filters (Fig. 1) were developed first(1)–(3), followed by adaptive algorithms(4),(5), optimal control filters(6),(7), hybrid classical-adaptive filters(8) and robust filters(9),(10). Importantly, even though these studies extensively address applications in simulators with relatively large workspaces, the performance of various techniques for simulating specific VR motion in a motion simulator with a restricted workspace has rarely been discussed.

This study presents a novel washout filter design as shown in Fig. 2, that consists of a classical linear washout filter (CLWF), an adjustable scaling filter (ASF), a yawing washout filter (YWF), a dead zone washout filter (DZWF) and an adaptive washout filter (AWF). The CLWF sepa-

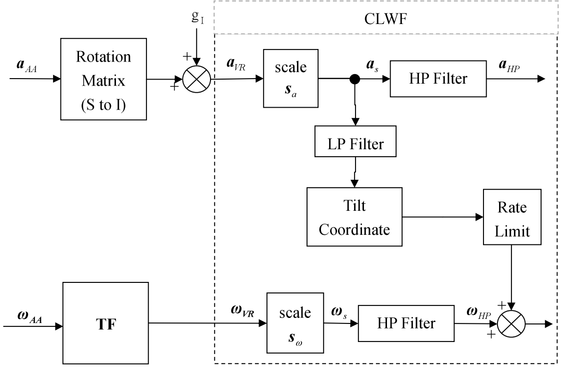


Fig. 1 Classical linear washout filter (referred to Nahon and Reid, 1990)

* Received 24th February, 2003 (No. 03-5020)

** Department of Mechanical Engineering, National Chiao Tung University, 1001 TA Hsueh Road, Hsinchu, Taiwan 30010, R.O.C. E-mail: whc@cc.nctu.edu.tw

*** Chung Shan Institute of Science and Technology, Taiwan, R.O.C. E-mail: jeffh.me83g@nctu.edu.tw

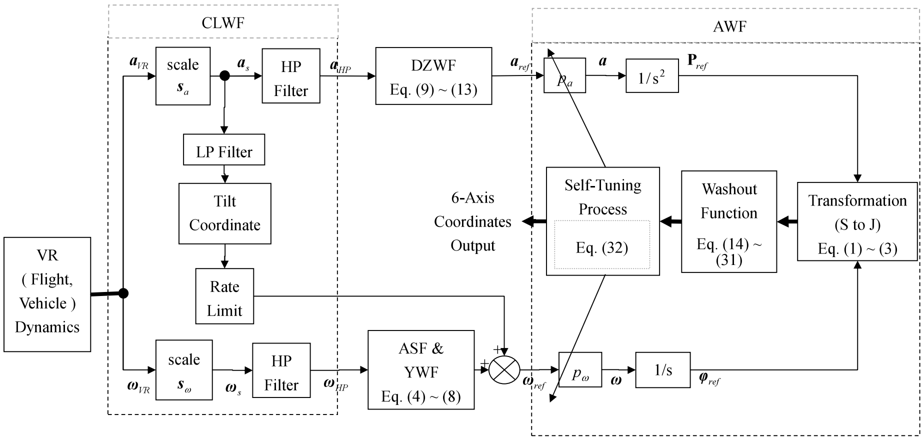


Fig. 2 Block diagram of motion-cueing system, using the proposed control strategy

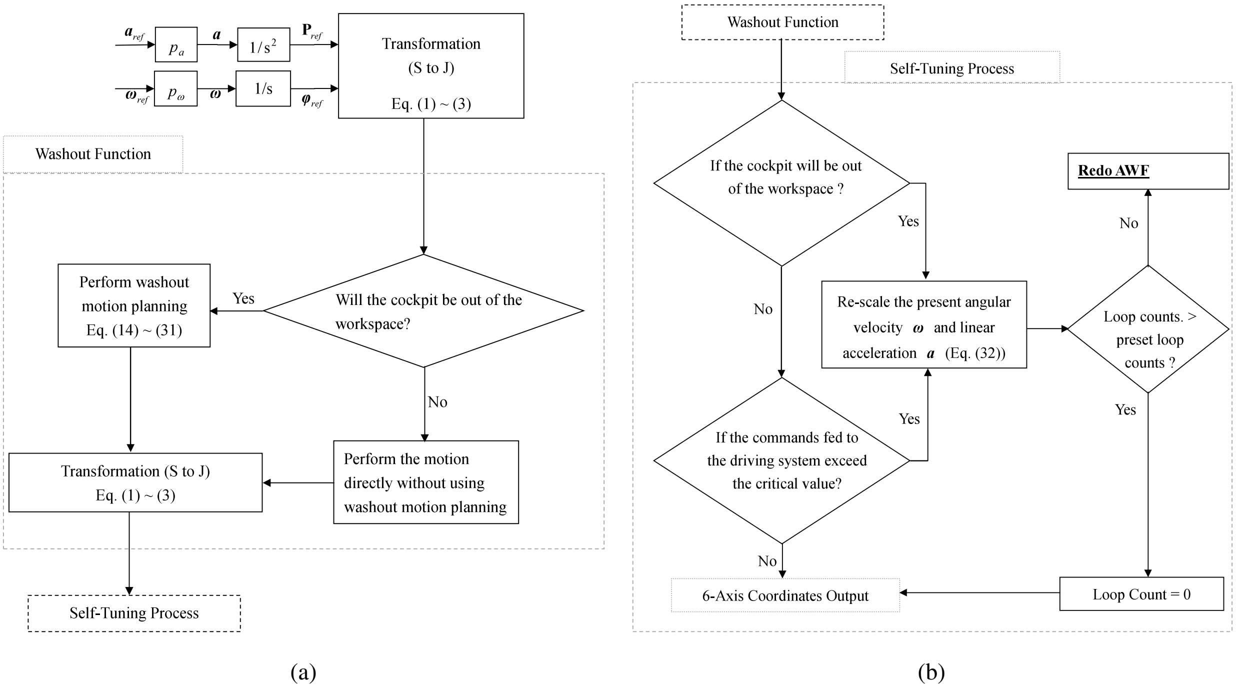

a_{ref}
p_a
a
$1/s^2$
P_{ref}
ω_{ref}
p_ω
ω
1/s
φ_{ref}
Transformation (S to J) Eq. (1) ~ (3)
Washout Function
Perform washout motion planning Eq. (14) ~ (31)
Yes
Will the cockpit be out of the workspace?
No
Transformation (S to J) Eq. (1) ~ (3)
Perform the motion directly without using washout motion planning
Self-Tuning Process
(a)
Washout Function
Self-Tuning Process
If the cockpit will be out of the workspace ?
Yes
No
Redo AWF
No
Re-scale the present angular velocity ω and linear acceleration a (Eq. (32))
Loop counts. > preset loop counts ?
Yes
If the commands fed to the driving system exceed the critical value?
Yes
No
6-Axis Coordinates Output
Loop Count = 0
(b)

Fig. 3 The AWF structure

rates the motion cues into high ("onset") and low ("sustained") frequency components so that cues can be managed and displayed within the physical confines of a given platform system. However, for motion simulators with severely restricted workspaces, even if the cutoff frequencies are properly selected$^{(11)}$, the position of cockpit may still exceed the platform's workspace during a given motion because the linear accelerations and the angular velocities are mutually independent in the Cartesian coordinate system, but are coupled after applying inverse kinematics to every independent joint of the motion simulator. Moreover, the constraints on the driving system limit the performance of the motion simulator, such as the saturation for the driving current. Accordingly, the CLWF with appropriate cutoff frequencies is not always suitable for many practical platforms, especially those simulators with smaller workspace, but it remains useful in reducing the probability of leaving the limited workspace. The ASF dynamically tunes the cockpit's angular velocities instead of the purely static scaling. Sometimes, the magnitude of linear acceleration is lower than the human sensible threshold$^{(12)-(14)}$, and the DZWF utilizes this moment to drive the cab stealthily back to its home position by accelerating it under the indifference threshold$^{(12)-(14)}$. As shown in Fig. 3 (a) and (b), an AWF that includes the (S to J) transformation, washout function and self-tuning process, greatly improves the motive performance for strictly confined simulators.

Two cost functions are defined to determine the performance index (*PI*) of VR motion. The *PI* quantifies the realism of the motion and is improved by inducing an empirical rule while adaptive scaling factors in the self-tuning

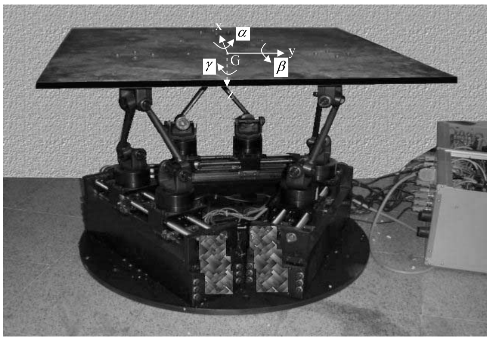


Fig. 4 Prototype SP-120

process are tuned offline using numerical induction. Additionally, real-time software has been developed to implement the proposed criteria online in an automotive simulation for a six degree-of-freedom (DOF) motion simulator, such as SP-120 shown in Fig. 4. Comparing the proposed washout filter with the CLWF shows that the experimental results indicate that the former is much more adaptive and realistic, especially for the motion simulator with a small workspace.

2. Nomenclature

$a_{threshold}$: indifference threshold for acceleration

$\boldsymbol{a}$: acceleration received from the output of self-tuning process

$\boldsymbol{a}_i$: the i th component of acceleration vector $\boldsymbol{a}$

$\boldsymbol{a}_{AA}$: specific acceleration of aircraft

$\boldsymbol{a}_{HP}$: acceleration received from the output of high-pass filter

$\boldsymbol{a}_{HP,i}$: the i th component of acceleration vector $\boldsymbol{a}_{HP}$

$\boldsymbol{a}_s$: acceleration obtained from scaling $\boldsymbol{a}_{VR}$

$\boldsymbol{a}_{ref}$: acceleration received from the output of dead zone washout filter

$\boldsymbol{a}_{ref,i}$: the i th component of acceleration vector $\boldsymbol{a}_{ref}$

$\boldsymbol{a}_{res}$: restoring acceleration in the dead zone washout filter

$\boldsymbol{a}_{res,i}$: the i th component of acceleration vector $\boldsymbol{a}_{res}$

$\boldsymbol{a}_{VR}$: acceleration obtained by summing gravity ($\boldsymbol{g}_I$) and applied force

$\boldsymbol{a}_{VR,i}$: the i th component of acceleration vector $\boldsymbol{a}_{VR}$

$\boldsymbol{a}_1(\bullet)$, $\boldsymbol{a}_2(\bullet)$: acceleration function in adaptive washout filter

$\boldsymbol{a}_{2,i}(\bullet)$: the i th component of acceleration vector $\boldsymbol{a}_2(\bullet)$

G : mass center of the cockpit

g : acceleration due to gravity

$\boldsymbol{g}_I$: gravity vector in the inertia reference frame

HP filter : high-pass filter

LP filter : low-pass filter

L : length of linkage

$\boldsymbol{p}_i$: position of i th slider ball joint ($i = 1$ to 6)

$\boldsymbol{q}_i$: position of i th ball joint ($i = 1$ to 6)

${}^{O}\boldsymbol{q}_i$: coordinates of $\boldsymbol{q}_i$ in the frame X-Y-Z-O

${}^{G}\boldsymbol{q}_i$: coordinates of $\boldsymbol{q}_i$ in the frame x-y-z-G

${}^{S_i}\boldsymbol{q}_i$: coordinates of $\boldsymbol{q}_i$ in the frame x_i-y_i-z_i-S$_i$

${}^{S_i}\boldsymbol{p}_i = {}^{S_i}[p_{x_i}\ p_{y_i}\ p_{z_i}]^T$: coordinates of $\boldsymbol{p}_i$ in the frame x_i-y_i-z_i-S$_i$

${}^{O}_{G}R(\alpha,\beta,\gamma)$: rotational transformation matrix from x-y-z-G to X-Y-Z-O

${}^{S_i}_{O}R$: constant rotational transformation matrix from X-Y-Z-O to x_i-y_i-z_i-S$_i$ ($i = 1, 3, 5$)

$RMS(\bullet)$: root mean square of $\bullet$

s : Laplace operator

s_a : static scaling factor used to scale the linear accelerations

s_ω : static scaling factor used to scale the Euler angular velocities

(S to I) : transform from simulator to inertia reference frame

(S to J) : transformation of coordinates from x-y-z-G to x_i-y_i-z_i-S$_i$

TF : transform from angular velocity to Euler angle rates

T : system sampling time

X-Y-Z-O : inertial coordinate system

^{O}G : coordinates of G relative to the X-Y-Z-O frame

x-y-z-G : cockpit coordinate system

x_i-y_i-z_i-S$_i$: joint coordinate system (i = 1, 3, 5) used to represent the positions of slider ball joints $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$

ω_{indiff} : indifference threshold for angular speed

$\boldsymbol{\omega}$: angular velocity received from the output of self-tuning process

$\boldsymbol{\omega}_{AA}$: specific angular velocity of aircraft

$\boldsymbol{\omega}_{HP}$: angular velocity received from the output of high-pass filter

$\boldsymbol{\omega}_s$: scaled Euler's angular velocity $\boldsymbol{\omega}_{VR}$

$\boldsymbol{\omega}_{ref}$: angular velocity received from the output of adjustable scaling filter and yawing washout filter

$\boldsymbol{\omega}_{VR}$: Euler's angular velocity

α : roll angle of the platform (upper plate)

β : pitch angle of the platform (upper plate)

γ : yaw angle of the platform (upper plate)

$\boldsymbol{\varphi} = [\alpha\ \beta\ \gamma] = [\phi_x\ \phi_y\ \phi_z]$: Euler angle in the x-y-z-G frame

3. Inverse Kinematics

The motion cue control may be also called the cockpit position control, because the position of the cockpit, including both translation and rotation, must be transformed into the coordinates of the six sliders' ball joints (S to J) using inverse kinematics. The inverse kinematics of the motion simulator SP-120 is presented as follows.

As shown in Fig. 5, the coordinate of ${}^{S_i}\boldsymbol{p}_i = {}^{S_i}[p_{x_i}\ p_{y_i}\ p_{z_i}]^T$ is determined by

$$\|{}^{S_i}\boldsymbol{q}_i - {}^{S_i}\boldsymbol{p}_i\| = L^2 \qquad (1)$$

in which all parameters are fixed in the S$_i$ coordinate system, where

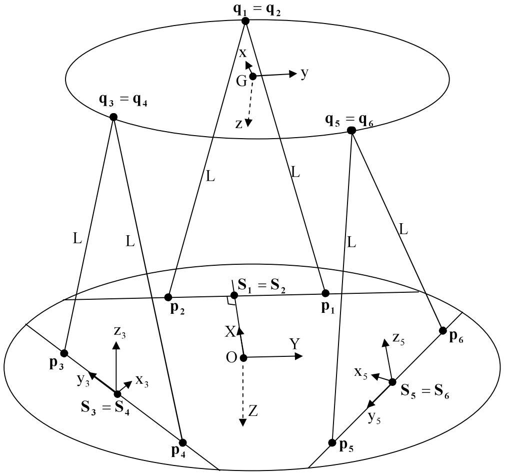


Fig. 5 Kinematical skeleton of simulator platform SP-120

$$ {}^{S_i}\boldsymbol{q}_i = {}^{S_i}O + {}^{S_i}_{O}R \cdot [{}^{O}G + {}^{O}_{G}R(\alpha,\beta,\gamma) \cdot {}^{G}\boldsymbol{q}_i] \qquad (2) $$

and ${}^{O}_{G}R(\alpha,\beta,\gamma)$ is the transformation matrix of the Euler angle and can be expressed as,

$$ {}^{O}_{G}R(\alpha,\beta,\gamma) = \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix} \qquad (3) $$

and $c\beta = \cos\beta$, $s\alpha = \sin\alpha$, ..., and so on.

4. Preview of Classical Linear Washout Filter (CLWF)

The most widely used CLWF drive rules today are derived from the design of Schmidt and Conrad(12). Figure 1 shows a block diagram of the typical implementation(3). Modern implementations tend to drive the simulation with angular rates rather than angular accelerations, since this method has generally been found to produce a more realistic cue. As shown in Fig. 1, specific acceleration ($\boldsymbol{a}_{AA}$) is transformed to the inertial reference frame (S to I) and converted into acceleration ($\boldsymbol{a}_{VR}$), obtained by summing gravity ($\boldsymbol{g}_I$) and applied forces before the high-pass filter operation is performed. This approach uses a more convenient frame of reference for generating the commands of the simulator's driving system. Similarly, high-pass onset filtering is applied to the scaled Euler's angular velocity ($\boldsymbol{\omega}_s$). The low-frequency specific acceleration components are low-pass filtered, and operated upon by a "tilt coordinate" block, much like that in Schmidt and Conrad's residue-tilt design; the tilt coordinate cross-feed is rate-limited to ensure that the commanded rates do not exceed the pilot's indifference threshold, which is set to 3 deg/s(12). As stated above, the CLWF technique is combined with the following auxiliary washout filters, yielding a novel washout filter, presented in Fig. 2.

5. Adjustable Scaling Filter (ASF) and Yawing Washout Filter (YWF)

The limited workspace of the simulator constrains the Euler angles, including roll, pitch and yaw. Therefore, this paper proposes that the angular velocity of the cockpit must be adjusted by a dynamic tuning process called adjustable scaling filtering which involves a nonlinear filter, rather than by purely static scaling down, which would also reduce the active intensity even if the angular rates are originally lower. Applying this nonlinear filter can guarantee that the signals of angular velocities, fed to the simulator are more realistic than those associated with traditional static scaling down, unless the limited workspace is sufficiently large that the magnitude of the static scaling factor is approximately unity. Restated, the degree of scaling down is traded off with the limited size of the workspace, but can be greatly reduced after the ASF is used. The algorithm is as follows.

$$ \begin{cases} \phi_{i,k+1} = \phi_{HP,i,k+1}, \\ \quad \text{if } \|(\phi_{HP,i,k+1})\|_2 \le \phi_{critical} \\ \phi_{i,k+1} = \phi_{i,k+1} \cdot \phi_{critical,k+1} / \|\phi_{HP,k+1}\|_2, \\ \quad \text{if } \|(\phi_{HP,k+1})\|_2 > \phi_{critical} \end{cases} \qquad (4) $$

and,

$$ \omega_{i,k} = (\phi_{i,k+1} - \phi_{i,k})/t_{VR} \qquad (5) $$

where $\phi_{i,k}$ and $\phi_{HP,i,k}$ both represent the present Euler angles, the latter of which is received from the high-pass filter at time kt_{VR}; $\phi_{critical}$ is the magnitude of a given critical Euler angle; $\|\bullet\|_2$ represents the 2-norm of $\bullet$; t_{VR} is the sampling period of VR; both $\omega_{i,k}$ and $\omega_{HP,i,k}$ are present Euler angular velocities, the latter of which is received from the output of the high-pass filter at time kt_{VR}. The subscript indicates the i-axis ($i = x$, y or z).

Applying the above algorithm greatly improves the rotational performance of the motion simulator, not only to prevent the cockpit from moving outside the limited workspace during pure rotation but also to obtain more realistic angular velocities or attitudes of the platform during real-time VR motion.

Importantly, the platform's posture in terms of roll and pitch involves an actual tilt coordination that enables the pilot to feel the component of gravity; thus, the roll or pitch cannot be arbitrarily changed during the restoration unless the attitude is obtained by low-pass filtering of the acceleration along the y- or x-axis, as in residue tilt. Contrarily, the yaw angle is not important: the only concern is the yawing velocity. Therefore, a yawing washout algorithm is proposed as follows.

$$ \begin{aligned} &\text{if } (\phi_{z,k+1} < \phi_{z,critical} \text{ and } \omega_{HP,z,k} > \omega_{indiff}) \\ &\quad \text{or } \operatorname{sign}(\omega_{HP,z,k}) = -\operatorname{sign}(\phi_{z,k}) \\ &\Rightarrow \ \omega_{z,k} = \omega_{HP,z,k}, \end{aligned} \qquad (6) $$

$$\begin{aligned}&\text{if } (\phi_{z,k+1} \geq \phi_{z,critical} \text{ or } |\omega_{HP,z,k}| < \omega_{indiff})\\&\quad \text{and } \mathrm{sign}(\omega_{HP,z,k}) = \mathrm{sign}(\phi_{z,k}) \text{ and } t \leq t_{res,yaw}\\&\Rightarrow \ \omega_{z,k} = -\mathrm{sign}(\phi_{z,k}) \cdot \omega_{indiff} \qquad (7)\end{aligned}$$

where $\phi_{z,critical}$ is the given critical yaw about the simulator; ω_{indiff} is the indifference threshold for angular speed; t and $t_{res,yaw}$ are the present restoring time and the total periodic restoring time, respectively, and

$$t_{res,yaw} = |\phi_{z,k}|/\omega_{indiff}, \qquad (8)$$

where,

$$\begin{cases}\mathrm{sign}(\bullet) = 1, \text{ if } (\bullet) > 0\\ \mathrm{sign}(\bullet) = -1, \text{ if } (\bullet) < 0\\ \mathrm{sign}(\bullet) = 0, \text{ if } (\bullet) = 0\end{cases}$$

Applying the above yawing washout algorithm, the zero-crossing phenomenon does not occur during the yawing washout period, facilitating the other actions including roll, pitch and translation.

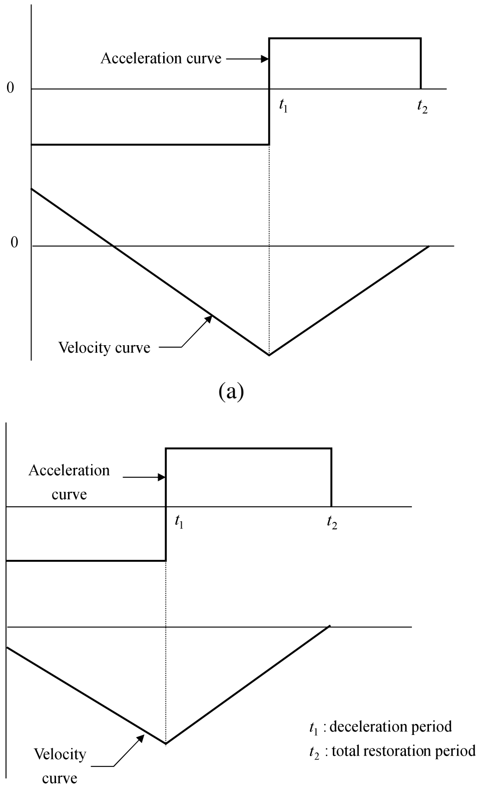
6. Dead Zone Washout Filter (DZWF)

During the restoration period, the limitations on linear acceleration and angular velocity(13),(14) almost prohibit the restoration of the cockpit to its home position except by extending the restoration period, or when the original motion in VR are at sufficiently low frequencies. Clearly, extending the restoration time may cause some significant motion to be lost, so this strategy is not favored. Translation with lower frequencies may give enough time to carry the cockpit back stealthily after proper high-pass filtering is performed, but generally, such a motion implies unexcited motion and poorly represents most normal actions. Consequently, a proposed strategy, called dead zone washout filtering, is adept at utilizing time. Dead time is defined as the period during which the linear acceleration is lower than the pilot's sensible threshold(12)-(14). Restated, the acceleration enters the dead area during this dead time; otherwise, it is in the scaled area.

During the dead time, the cockpit is translated to its home position rather than being scaled to zero. Every component of the restoring acceleration $\boldsymbol{a}_{res} \in R^3$ must be lower than the indifference threshold $a_{threshold}$ (0.17 – 0.28 m/s²(14), here set to 0.017 g). Even if the restoring acceleration slightly exceeds the indifference threshold, an adaptive restoring acceleration must be modified as follows, to prevent the workspace boundary from being touched.

$$\begin{cases}|a_{res,i}| = v_{0,i}^2/2S_{\max,i}, \text{ if } |v_{0,i}| \geq \sqrt{2a_{threshold}S_{\max,i}}\\[2ex] |a_{res,i}| = a_{threshold}, \text{ if } |v_{0,i}| < \sqrt{2a_{threshold}S_{\max,i}}\end{cases} \qquad (9)$$

where $S_{\max,i}$ is the maximum distance from the present position to the nominal workspace boundary in the direction of the present velocity; subscript i indicates the i-axis ($i = x$, y or z).



(a)

(b)

Fig. 6 Restoration process; (a) present velocity has the same direction as the present displacement, (b) present velocity has the opposite orientation to the present displacement, with respect to the home position

The next urgent task is to determine the maximum restoration period. Figure 6 (a) indicates a situation in which the present velocity has the same direction as the present displacement, with reference to the home position. Figure 6 (b) presents a situation in which the present velocity is in the opposite direction from the displacement. These two cases are both treated by the basic law of kinematics, yielding,

$$\begin{aligned}t_1 &= \mathrm{sign}(P_{cur,i}) \cdot v_{0,i}/|a_{res,i}|\\&\quad + (1/2)\sqrt{2v_{0,i}^2/a_{res,i}^2 + 4d_i/|a_{res,i}|} \qquad (10)\\ t_2 &= \mathrm{sign}(P_{cur,i}) \cdot v_{0,i}/|a_{res,i}|\\&\quad + \sqrt{2v_{0,i}^2/a_{res,i}^2 + 4d_i/|a_{res,i}|} \qquad (11)\end{aligned}$$

and,

$$d_i = |P_{\mathrm{home},i} - P_{cur,i}| \qquad (12)$$

where d_i is the distance from the present position to the home position along the i-axis; $P_{\mathrm{home},i}$ and $P_{cur,i}$ are the home position and the present position along the i-axis, respectively; t_1 is the period of acceleration and $t_2 - t_1$ is the period of deceleration during the restoration. The velocity is importantly guaranteed to be zero at the restoring time t_2.

During the restoration period, the restoring action will continue unless the direction of acceleration along the i-axis is opposite neither that of the present velocity nor the present position. Then, the active acceleration along the i-axis can be expressed as,

$$\begin{cases} a_i = a_{HP,i}, \\ \quad \text{if } a_{HP,i} > a_{threshold} \text{ and} \\ \quad\quad \text{sign}(a_{HP,i}) = -\text{sign}(v_{0,i}) = -\text{sign}(P_{cur,y}) \\ a_i = a_{res,i}, \text{ others} \end{cases} \tag{13}$$

where $a_{HP,i}$ represents the linear acceleration along the i-axis, received from the output of high-pass filter.

Like the yawing washout, the DZWF procedure involves no zero-crossing and improves the rotational performance. Restated, it greatly reduces the cross coupling of rotation and translation.

7. Structure of the Adaptive Washout Filter (AWF)

This DZWF algorithm cannot always guarantee that the cockpit of simulator does not leave the actual workspace because not all of the workspace boundaries are very explicit. Therefore, adding an adaptive washout filter is proposed to compensate for the insufficiency of the prior proposed filters and thus accommodate the more severe restrictions, such as the smaller workspace and the limited driving current. The AWF involves the transformation (S to J), the washout function (Fig. 3 (a)) and the self-tuning process (Fig. 3 (b)).

7.1 Transformation (S to J)

The transformation (S to J) is presented using inverse kinematics, as described in the preceding section.

7.2 Washout function

The purpose of the washout function (Fig. 3 (a)) is to prevent the cockpit from exiting its limited workspace. Figure 7 depicts a trajectory along the i-axis proposed to plan the washout motion, initially making the pilot feel an instantaneous linear acceleration and later carrying the cockpit to its starting position stealthily in the period $t_d \le t \le t_f$. The planned trajectory is as follows.

The continuous trajectory $\boldsymbol{P}(t) \in R^3$ of the translation of the cockpit consists of two cubic polynomial segments.

$$\boldsymbol{P}(t) = \boldsymbol{P}_1(t) + \boldsymbol{P}_2(t) \quad \text{for} \quad 0 \le t \le t_f \tag{14}$$

where the vectors $\boldsymbol{P}_1(t)$ and $\boldsymbol{P}_2(t)$ are in R^3;

$$\boldsymbol{P}_1(t) = 0 \quad \text{for} \quad t_d \le t \quad \text{and}$$
$$\boldsymbol{P}_2(t) = 0 \quad \text{for} \quad t < t_d$$

The $\boldsymbol{P}(t)$ at t_f is the desired target position and that at t_d is the transition position. At least eight constraints on $\boldsymbol{P}(t)$ are evident. The initial and final values of the function are constrained.

$$\boldsymbol{P}(0) = 0, \quad \boldsymbol{P}(t_f) = 0.$$

Continuity at the transition position yields,

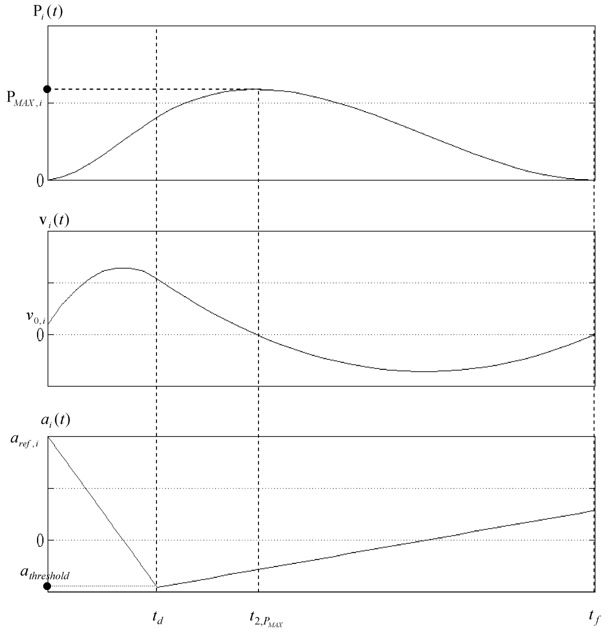


Fig. 7 Washout trajectory along the i-axis, where the subscript i represents the three mutually orthogonal axes (x-, y-, and z-axis)

$$\boldsymbol{P}_1(t_d^-) = \boldsymbol{P}_2(t_d^+).$$

The function yields continuous velocities, implying that the initial and transitional velocities' vectors (R^3) are both continuous and the final velocity's vector is zero in the procedure of washout motion planning, and can be expressed as,

$$\boldsymbol{v}(0) = \boldsymbol{v}_0, \quad \boldsymbol{v}(t_f) = 0, \quad \boldsymbol{v}_1(t_d^-) = \boldsymbol{v}_2(t_d^+ x).$$

A further constraint is that the transition acceleration must be continuous:

$$\boldsymbol{a}_1(t_d^-) = \boldsymbol{a}_2(t_d^+) \in R^3.$$

The following equation implies that the initial acceleration must fulfill the demands of VR — to ensure that the pilot feels an instantaneous linear acceleration.

$$\boldsymbol{a}_1(0) = \boldsymbol{a}_{ref} \in R^3$$

where $\boldsymbol{a}_{ref}$ is the linear acceleration received from the output of the DZWF. These eight constraints can be used to determine two consecutive cubic polynomial segments, since two such segments have exactly eight coefficient vectors.

$$\boldsymbol{P}_1(t) = \boldsymbol{b}_0 + \boldsymbol{b}_1 t + \boldsymbol{b}_2 t^2 + \boldsymbol{b}_3 t^3 \tag{15}$$

$$\boldsymbol{P}_2(t) = \boldsymbol{c}_0 + \boldsymbol{c}_1 t + \boldsymbol{c}_2 t^2 + \boldsymbol{c}_3 t^3 \tag{16}$$

where the eight vectors, $\boldsymbol{b}_0$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\boldsymbol{b}_3$, $\boldsymbol{c}_0$, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, and $\boldsymbol{c}_3$ are all in R^3. The vectors of velocities and accelerations along the path are derived as follows.

$$\boldsymbol{v}_1(t) = \dot{\boldsymbol{P}}_1(t) = \boldsymbol{b}_1 + 2\boldsymbol{b}_2 t + 3\boldsymbol{b}_3 t^2 \tag{17}$$

$$\boldsymbol{v}_2(t) = \dot{\boldsymbol{P}}_2(t) = \boldsymbol{c}_1 + 2\boldsymbol{c}_2 t + 3\boldsymbol{c}_3 t^2 \tag{18}$$

$$\boldsymbol{a}_1(t) = \ddot{\boldsymbol{P}}_1(t) = 2\boldsymbol{b}_2 + 6\boldsymbol{b}_3 t \tag{19}$$

$$\boldsymbol{a}_2(t) = \ddot{\boldsymbol{P}}_2(t) = 2\boldsymbol{c}_2 + 6\boldsymbol{c}_3 t \tag{20}$$

Combining Eqs. (15) to (20) with the eight constraints yields eight-by-three constrained equations in eight-by-three unknowns. Let $t_d = \kappa t_f$, where $0 < \kappa < 1$. Now,

$$\boldsymbol{b}_0 = \boldsymbol{0} \tag{21}$$

$$\boldsymbol{b}_1 = \boldsymbol{v}_0 \tag{22}$$

$$\boldsymbol{b}_2 = \boldsymbol{a}_{ref}/2 \tag{23}$$

$$\boldsymbol{b}_3 = (2\boldsymbol{a}_{ref}\kappa t_f + 2\boldsymbol{v}_0\kappa + \boldsymbol{a}_{ref}t_f + 4\boldsymbol{v}_0)/(6\kappa t_f^2) \tag{24}$$

$$\boldsymbol{c}_0 = -\kappa^2 t_f(\boldsymbol{a}_{ref}t_f + 4\boldsymbol{v}_0)/(6(1-\kappa)^2) \tag{25}$$

$$\boldsymbol{c}_1 = (2\boldsymbol{v}_0(1+\kappa^2) + \boldsymbol{a}_{ref}\kappa t_f)/(2(1-\kappa)^2) \tag{26}$$

$$\boldsymbol{c}_2 = (\boldsymbol{a}_{ref}\kappa^2 t_f - 4\boldsymbol{v}_0 - 2\boldsymbol{a}_{ref}\kappa t_f)/(2t_f(1-\kappa)^2) \tag{27}$$

$$\boldsymbol{c}_3 = (-\boldsymbol{a}_{ref}\kappa t_f(3-2\kappa) - 2\boldsymbol{v}_0(3-\kappa^2))/(6(1-\kappa)^2 t_f^2) \tag{28}$$

From Eq. (20), the maximum linear deceleration is $\boldsymbol{a}_2(t_d)$, and,

$$a_{2,i}(t_d) \le a_{threshold} \tag{29}$$

where the subscript i represents the three mutually orthogonal axes (x-, y-, and z-axis). Equation (28) implies that κ can be treated as a ratio to constrain deceleration during restoration. The magnitude of deceleration must be constrained below an indifference threshold $a_{threshold}$ to prevent the pilot from becoming aware of this restoration(14).

The maximum displacement is at a stationary value when the velocity is zero. For the second segment of the polynomial,

$$\boldsymbol{v}_2(t) = \boldsymbol{c}_1 + 2\boldsymbol{c}_2 t + 3\boldsymbol{c}_3 t^2 = \boldsymbol{0}$$

which yields,

$$\begin{aligned} t_{2,P_{MAX}} &= t_f(2v_{0,i}(1+\kappa^2) + a_{ref,i}\kappa t_f) \\ &\quad /(2v_{0,i}(3-\kappa^2) + a_{ref,i}\kappa t_f(3-2\kappa)) > t_d \end{aligned} \tag{30}$$

The maximum displacement along the i-axis is obtained by substituting $t_{2,P_{MAX}}$ into the i-axis washout function $\boldsymbol{P}_i(t)$, such that,

$$\boldsymbol{P}_{MAX,i} = \boldsymbol{P}_i(t_{2,P_{MAX}}) \tag{31}$$

which will be used to determine whether the washout planning is executed.

7.3 Self-tuning process (Fig. 3 (b))

The saturation of the driving current also constrains the performance of the simulator, and may cause the angular speed of the servo motor to exceed its critical value and, causing a problem related to the robustness of the driving system. Additionally, motion may sometimes still violate the workspace after filtering, because the indifference threshold of deceleration always limits the washout efficiency. Thus, a final check on the self-tuning process is proposed to guarantee that the system of motion simulator is robust. The following steps determine the rules.

1. Calculate whether the commanded velocity fed to the driving system exceed the critical value.

2. Calculate whether the cockpit will be outside the limited workspace.

3. If at least one of the answers to the preceding questions is positive, let the linear acceleration and the angular velocity be multiplied by two appropriately predetermined scaling functions, $p_a(\lambda_a)$ and $p_\omega(\lambda_\omega)$, respectively. Then, redo steps 1 and 2 until the answers are both negative or the iterative loop is performed more than n times, where n is the limit preset by considering whether the total calculation time will meet the demands of real-time programming.

The following two simple equations represent the above strategies.

$$\boldsymbol{a} = \boldsymbol{a}_{res} \cdot p_a^n(\lambda_a) \quad \text{and} \quad \boldsymbol{\omega} = \boldsymbol{\omega}_{ref} \cdot p_\omega^n(\lambda_\omega) \tag{32}$$

where $\boldsymbol{a} \in R^3$ and $\boldsymbol{\omega} \in R^3$ represent the cockpit's present linear acceleration and angular velocity, respectively; $\boldsymbol{a}_{ref}$ and $\boldsymbol{\omega}_{ref}$ (Fig. 2) are the linear acceleration received from the output of DZWF and the angular velocity that combines the output of ASF with the tilt coordination. The functions $p_a(\lambda_a)$ and $p_\omega(\lambda_\omega)$ are the adaptive scaling functions of linear acceleration and angular velocity, respectively. The adaptive scaling factors λ_a and λ_ω are properly predicted before the first simulation test and are later tuned offline by inducing an empirical rule to obtain a heuristically selected pair of adaptive scaling factors, as described in the next section.

8. Performance Index

In this paper, the VR motion fed to the specific motion simulator SP-120 is specified by a performance index (PI) combined with two cost functions, $E_{a,k}$ and $E_{\omega,k}$.

$$\begin{aligned} E_{a,k} &= [\|\boldsymbol{a}(kt_{VR}) - \boldsymbol{a}_{ref}(kt_{VR})\|_2/RMS(\boldsymbol{a}_{ref})] \\ &\quad \cdot RMS(\boldsymbol{a}_{VR})/RMS(\boldsymbol{a}_{ref}) \\ &\quad = RMS(\boldsymbol{a}_{VR}) \cdot (1 - p_a^{n_k}(\lambda_a))\|\boldsymbol{a}_{ref}(kt_{VR})\|_2 \\ &\quad /[RMS(\boldsymbol{a}_{ref})]^2 \end{aligned} \tag{33}$$

$$\begin{aligned} E_{\omega,k} &= [\|\boldsymbol{\omega}^2(kt_{VR}) - \boldsymbol{\omega}_{ref}^2(kt_{VR})\|_2/RMS(\boldsymbol{\omega}_{ref}^2)] \\ &\quad \cdot RMS(\boldsymbol{\omega}_{VR})/RMS(\boldsymbol{\omega}_{ref}) \\ &\quad = RMS(\boldsymbol{\omega}_{VR}) \cdot (1 - p_\omega^{2n_k}(\lambda_\omega))\|\boldsymbol{\omega}_{ref}^2(kt_{VR})\|_2 \\ &\quad /[RMS(\boldsymbol{\omega}_{ref}^2) \cdot RMS(\boldsymbol{\omega}_{ref})] \end{aligned} \tag{34}$$

and,

$$PI = W_a \cdot RMS(E_a) + W_\omega \cdot RMS(E_\omega) \tag{35}$$

$$\begin{aligned} RMS(E_a) &= \sqrt{\left(\sum_{k=0}^{N} E_{a,k}^2\right)\Big/N}, \\ RMS(E_\omega) &= \sqrt{\left(\sum_{k=0}^{N} E_{\omega,k}^2\right)\Big/N} \end{aligned} \tag{36}$$

where W_a and W_b are the weighting parameters of $RMS(E_a)$ and $RMS(E_\omega)$; $RMS(\bullet)$ means the root mean square of $\bullet$; N is the total number of samples of VR motion; n_k is the total number of self-tuning iterations at time

kt_{VR} and is determined by the self-tuning process online. By considering the dimensions of Eqs. (33) and (34), we can properly define the adaptive scaling functions $p_a(\lambda_a)$ and $p_\omega(\lambda_\omega)$,

$$\begin{cases} p_a(\lambda_a)=\lambda_a^2, & 0\le\lambda_a\le 1 \\ p_\omega(\lambda_\omega)=\lambda_\omega, & 0\le\lambda_\omega\le 1 \end{cases} \tag{37}$$

where λ_a and λ_ω are both set to constant values during one test, provided that the total number of iterations exceeds zero, such that $n_k \ge 1$. Otherwise, $\lambda_a = \lambda_\omega = 1$, provided that the answers in both steps 1 and 2 in the first loop of the self-tuning process are negative, such that $n_k = 0$.

The magnitudes of the two weighting parameters W_a and W_ω represent the relative significances of linear acceleration and angular velocity, respectively, determined by the pilot's response. The proper values are set to (0.5, 0.5), after consultation with ten pilots. A smaller *PI* implies more realistic motion. To yield a smaller *PI*, the effects of using many different pairs of adaptive scaling factors are observed and an empirical rule induced off-line to obtain an heuristically selected pair $(\lambda_a^*, \lambda_\omega^*)$, which is in future tests to be substituted in the self-tuning process and used instead of the old adaptive scaling factors.

Equations (33) – (37) state that the *PI* is function of λ_a, λ_ω and n_k, where n_k is also coupled with the adaptive scaling factors and varies irregularly with the sampling number k; that is, the *PI* is not an explicit function of λ_a and λ_ω, so obtaining an optimal pair of adaptive scaling factors by directly minimizing the *PI* is difficult. Furthermore, the *PI* is determined instantaneously after online testing, which in turn is performed after the adaptive scaling factors are determined off-line. Thus, these two factors must be correctly predetermined. The values of these two cost functions after many test runs using different adaptive scaling factors, indicate that a tradeoff exists between λ_a and λ_ω to reduce the magnitude of *PI*. One of the rules of thumb is that the extreme values of Eq. (35) may be at the boundaries of λ_a and λ_ω. Accordingly, the corresponding set of adaptive scaling factors may be,

$$(\lambda_a^*, \lambda_\omega^*) = (1,0) \text{ or } (0,1) \tag{38}$$

where the set (0, 0) is irrational. This result implies that the heuristically selected set of adaptive scaling factors always tends to one direction (toward (1, 0) or (0, 1)), determined by comparing the magnitudes of $RMS(E_a)$ and $RMS(E_\omega)$.

For example, if $RMS(E_\omega)$ exceeds $RMS(E_a)$ in the preceding test, then the degree of self-tuning of the angular velocities may be too small to perform the more difficult specific actions. Therefore, in the following test, the adaptive scaling factors are adjusted using above heuristically selected results ((1, 0)) to reduce the magnitude of $RMS(E_\omega)$. If $RMS(E_\omega)$ is reduced normally but still exceeds $RMS(E_a)$ after the second test, then the probably optimal set of adaptive scaling factors is (1, 0); otherwise, performing the third test by substituting the set (0, 1) into the self-tuning process, and then comparing the magnitudes of *PI* obtained in these two tests, enables the other set of adaptive scaling factors to be heuristically selected. Restated, the set of adaptive scaling factors must be sought at least twice, implying that one of the magnitudes of *PI* in the second and third tests is heuristically selected.

9. Experimental Results and Comparison

A specific VR motion of an automotive system and its dynamics are considered to apply the proposed washout filter to the motion simulator SP-120. Moreover, real-time software was developed to demonstrate the advantages of the proposed control strategy and this technique is compared with the classical one. The following figures represent Euler's angular velocities ω_x and ω_y as pure rotational velocities. That is, the residue-tilting effect was omitted during the processing of data.

Figures 8 and 10 show the data concerning linear accelerations along the x-axis and Euler's angular velocities (ω_x) for the three segments. These data were obtained from the scaled VR dynamic output ($\boldsymbol{a}_s$, $\boldsymbol{\omega}_s$) and the simulator's two outputs using the control strategies of CLWF

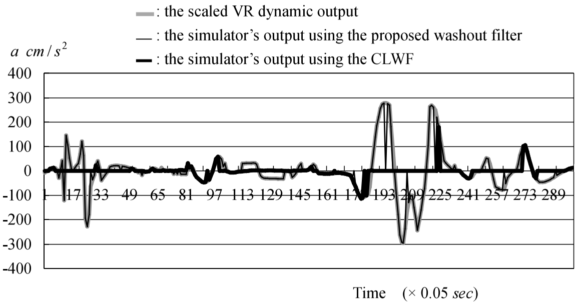
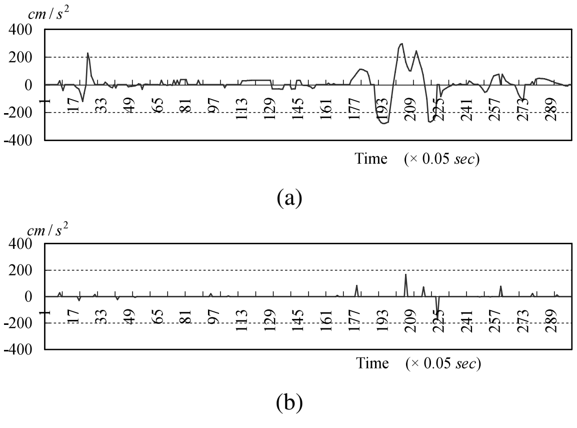


Fig. 8 Segmental data concerning linear accelerations along the x-axis, collected from the static scaled VR dynamic output ($a_{s,x}$) and the simulator's two outputs (a_x) using the CLWF and the proposed strategies, for the three segments



(a)

(b)

Fig. 9 Segmental errors of linear accelerations along the x-axis, between the static scaled VR dynamic output ($a_{s,x}$) and the simulator's output (a_x) using the (a) CLWF and (b) the proposed strategies

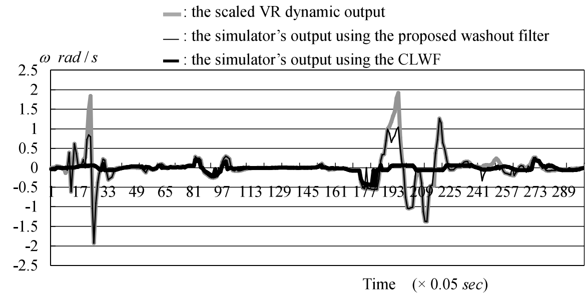
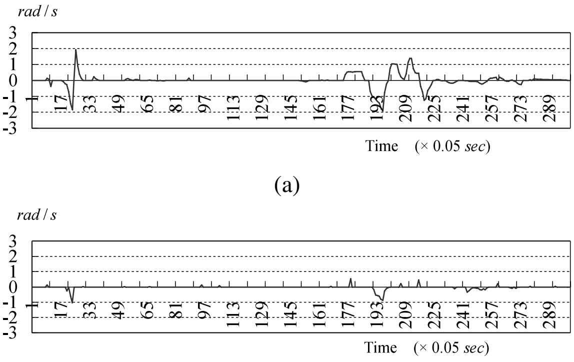


Fig. 10 Three segmental data concerning Euler's angular velocities collected from the static scaled VR dynamic output ($\omega_{s,x}$) and the simulator's two outputs (ω_x) using the CLWF and the proposed strategies



(a)

(b)

Fig. 11 Segmental errors of Euler's angular velocities between the static scaled VR dynamic output ($\omega_{s,x}$) and the simulator's output (ω_x) using (a) the CLWF and (b) the proposed strategies

and the proposed washout filter. Figures 9 and 11 present the segmental errors of linear accelerations along the x-axis and Euler's angular velocities (ω_x), respectively. The two sets of errors are both between the scaled VR dynamic output and the simulator's outputs obtained using the control strategies of CLWF and the proposed washout filter. As shown in these figures, the maximum acceleration along the x-axis is around 0.3 g and most of the scaled linear accelerations ($\boldsymbol{a}_s$) and scaled angular velocities ($\boldsymbol{\omega}_s$) can be simulated by applying the novel washout filtering to the simulator, which works in a relatively small workspace, as stated in Table 1(11). In contrast, the CLWF technique performs very poorly with such a simulator. Sometimes, the acceleration may drop back to zero because the current of the driving system is saturated. Linear accelerations or angular velocity may have been maintained in a particular direction for so long that the current of the driving system may exceed the critical value. The proposed strategy maximally suppresses cases in which the linear acceleration drops back to zero.

As shown in Figs. 8 to 11, the proposed washout filter outperforms the CLWF in the SP-120 motion simulator. Table 2 compares the CLWF with the proposed washout filter in terms of the magnitudes of $RMS(E_a)$, $RMS(E_\omega)$ and PI, using various static scaling factors ($\boldsymbol{s}_a$, $\boldsymbol{s}_\omega$) successively, to scale the linear accelerations and the angular velocities of the dynamic output of VR. In Table 2, the performance obtained when CLWF is applied to the simulator SP-120 shows that the efficiency associated with the simulated scaled data ($\boldsymbol{a}_s$, $\boldsymbol{\omega}_s$) is better when the static

Table 1 Capabilities of several motion simulators

Maximum Displacement / Simulator Platform	Heave (*feet*)	Sway (*feet*)	Surge (*feet*)	Roll (*degree*)	Pitch (*degree*)	Yaw (*degree*)
*FSAA 6-DOF	10	100	8	90	44	60
*VMS 6-DOF	50	35	8	36	36	48
*NADS	4	90	30	80	80	continuous
*LAMARS 5-DOF	20	20	–	50	50	50
*MIL-STD 1558 6-DOF	5.6	5.6	5.6	40	50	40
SP-120	0.3	0.3	0.3	8	8.6	11.5

* Edward A. Martin, "Motion and Force Cuing, Part I: Whole Body Motion," Flight & Ground Simulation Update, State University of New York, Binghamton NY, January 2000, pp. 5-18.

Table 2 Comparison between the CLWF and the proposed washout filter in terms of the magnitudes of $RMS(E_a)$, $RMS(E_\omega)$ and PI using various static scaling factors (s_a, s_ω), used successively to scale the linear accelerations and the angular velocities of VR dynamic output

Method	Performance items / (s_a, s_ω)	The magnitude of $RMS(E_{a,k})$	The magnitude of $RMS(E_{w,k})$	Performance Index (PI)
C L W F	(0.8, 1.0)	0.9313	0.8762	0.90375
	(0.8, 0.7)	0.9052	0.6984	0.95145
	(0.8, 0.5)	0.9394	0.7671	0.96970
	(0.8, 0.3)	0.8968	0.7254	0.94840
	(0.8, 0.1)	0.5569	0.5916	0.77845
The proposed washout filter	(0.8, 1.0)	0.2790	0.4831	0.38105
	(0.8, 0.7)	0.2790	0.3536	0.39205
	(0.8, 0.5)	0.2797	0.2356	0.37545
	(0.8, 0.3)	0.3087	0.1263	0.36485
	(0.8, 0.1)	0.2733	0.2630	0.63665

Note: The results were obtained after the first test using an arbitrary pair of adaptive scaling factors set to (1, 0.5).

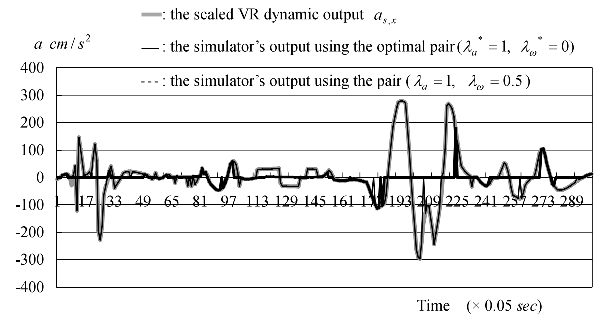


Fig. 12 Three segmental data concerning linear accelerations along the x-axis, collected from the static scaled VR dynamic output ($a_{s,x}$) and the simulator's two outputs (a_x) using the heuristically selected pair of weighting parameters (1,0) and the pair of weighting parameters (1,0.5)

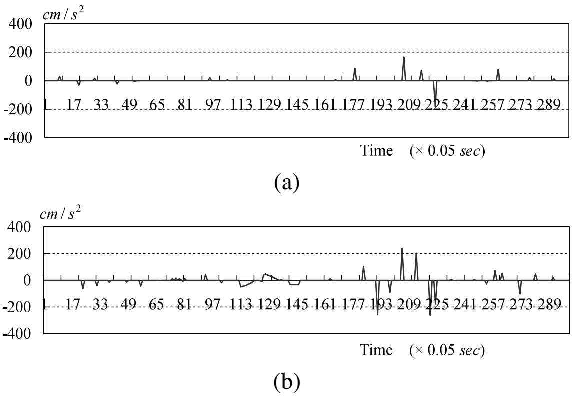


Fig. 13 Segmental errors of linear accelerations along the x-axis between the static scaled VR dynamic output ($a_{s,x}$) and the simulator's output (a_x) using (a) the optimal pair of weighting parameters (1,0) and (b) the pair of weighting parameters (1,0.5)

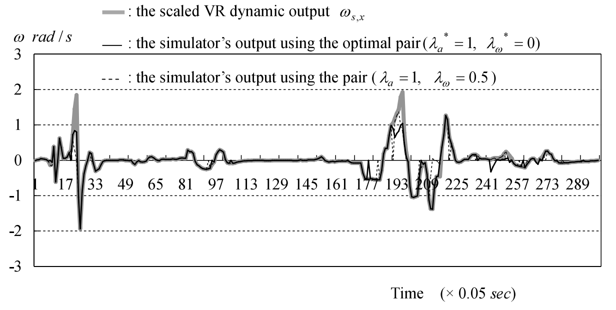


Fig. 14 Three segmental data concerning Euler's angular velocities collected from the static scaled VR dynamic output ($\omega_{s,x}$) and the simulator's two outputs (ω_x) using the optimal pair of weighting parameters (1,0) and the pair of weighting parameters (1,0.5)

scaling factor is smaller. This finding implies that the CLWF technique may be suited to a simulator with a large workspace but not one that operates in a more restricted workspace, such as the SP-120. Furthermore, using the proposed washout filter, even for this restricted simulator, the performance in terms of reality, strength and practicability remains excellent in many repeated tests in real time.

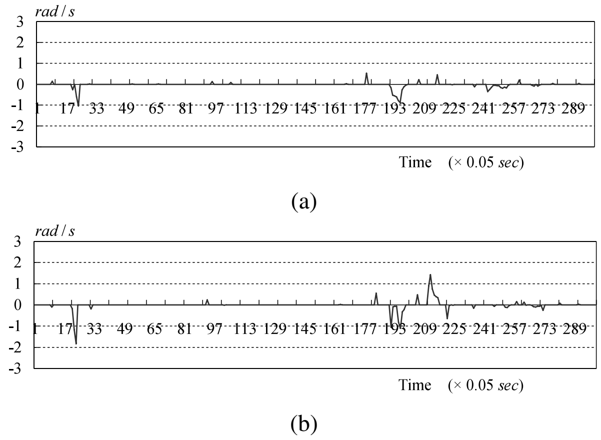


Fig. 15 Segmental errors of Euler's angular velocities between the static scaled VR dynamic output ($\omega_{s,x}$) and the simulator's output (ω_x) using (a) the optimal pair of weighting parameters (1,0) and (b) the pair of weighting parameters (1,0.5)

Table 3 Magnitudes of $RMS(E_a)$, $RMS(E_\omega)$ and PI using various pairs of adaptive scaling factors (λ_a, λ_ω); the static scaling factors (s_a, s_ω) are set to (0.8,0.7) (which will be later used to scale the linear accelerations and Euler's angular velocities of VR dynamic output)

Test Sequence	Performance items λ_a, λ_ω	$RMS(E_{a,k})$	$RMS(E_{\omega,k})$	PI
1st	1.0, 1.0	0.3854	0.3178	0.3516
2nd	1.0, 0.5	0.3113	0.1904	0.2509
3rd	0.5, 1.0	0.3769	0.2793	0.3281
4th	1.0, 0.3	0.3093	0.1376	0.2235
5th	1.0, 0.1	0.2655	0.1569	0.2112
6th	1.0, 0.0	0.2095	0.1834	0.1965

Figures 12 – 15 compare the use of the heuristically selected pair of adaptive scaling factors (1,0) with the arbitrary pair (1,0.5). The results reveal the advantages of using the former pair. Table 3 presents the magnitudes of $RMS(E_a)$, $RMS(E_\omega)$ and PI obtained using various pairs of adaptive scaling factors (λ_a, λ_ω). The PI is a heuristically selected value when the pair of adaptive scaling factors is (1,0).

10. Conclusion

The proposed washout strategy is a general method and can be used in many simulators with different workspaces and driving systems. However, it is particular effective in a simulator with a small workspace. This approach is practical and efficient, especially for use in motion simulators used for entertainment, with restricted workspaces. This paper establishes the performance index to quantify conveniently the efficiency of motion as the reference for realism. Repeated tests were performed online; they demonstrated that the proposed washout filter yields much more realistic motion than the classical technique for a motion simulator with a restricted workspace and an inexpensive driving system.

Acknowledgement

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC91-2212-E009-050.

References

(1) Sinacori, J.B., A Practical Approach to Motion Simulation, Visual and Motion Simulation Conference, AIAA Paper 73-931, Palo Alto, CA, (1973).

(2) Schmidt, S.F. and Bjorn, C., Motion Drive Signals for Piloted Flight Simulators, Analytical Mechanics Associated, Technical Report Contract NAS2-4869, (1970).

(3) Nahon, M.A. and Reid, L.D., Simulator Motion-Drive Algorithms: A Designer's Perspective, Journal of Guidance, Control, and Dynamics, Vol.13 (1990), pp.356–362.

(4) Ariel, D. and Sivan, R., False Cue Reduction in Moving Flight Simulators, IEEE Transactions on Systems, Man and Cybernetics, SMC-14, No.4 (1984), pp.665–671.

(5) Bowles, R.L., Parrish, R.V. and Dieudonne, J.E., Coordinated Adaptive Washout for Motion Simulators, Journal of Aircraft, Vol.12, No.1 (1975), pp.44–50.

(6) Sivan, R., Ish-shalom, J. and Huang, J.K., An Optimal Control Approach to the Design of Moving Flight Simulators, IEEE Transactions on Systems, Man and Cybernetics, SMC-12, No.6 (1982), pp.818–827.

(7) Wu, W. and Cardullo, F.M., Is There an Optimum Cueing Algorithm?, AIAA Modeling and Simulation Technologies Conference, New Orleans, LA, (1997), pp.23–29.

(8) Reid, L.D., Nahon, M.A. and Kirdeikis, J., Adaptive Simulator Motion Software with Supervisory Control, Journal of Guidance, Control, and Dynamics, Vol.15, No.2 (1992), pp.376–383.

(9) Moshe, I. and David, S., Robust Controller for a Dynamic Six Degree of Freedom Flight Simulator, AIAA Proceeding of Conference on Flight Simulator Technologies, (1996), pp.53–60.

(10) Moshe, I. and Nahon, M.A., Offline Comparison of Classical and Robust Flight Simulator Motion Control, Journal of Guidance, Control, and Dynamics, Vol.22, No.5 (1999), pp.702–709.

(11) Martin, E.A., Motion and Force Cuing, Part I: Whole Body Motion, Flight & Ground Simulation Update, State University of New York, Binghamton, NY, (2000), pp.5–18.

(12) Conrad, B. and Schmidt, S.F., A Study of Techniques for Calculating Motion Drive Signals for Flight Simulators, NASA CR-114345, (1971).

(13) Reid, L.D. and Nahon, M.A., Flight Simulator Motion-Based Drive Algorithm: Part 3 — Pilot Evaluations, Technical Report UTIAS Report 319, Univ. of Toronto — Canada, (1986).

(14) Pouliot, N.A., Gosselin, C.M. and Nahon, M.A., Motion Simulation Capabilities of Three-Degree-of-Freedom Flight Simulators, Journal of Aircraft, Vol.35, No.1 (1998), pp.9–17.