JOURNAL OF AIRCRAFT
Vol. 36, No. 5, September–October 1999

Optimization of Six-Degrees-of-Freedom Motion Systems for Flight Simulators

S. K. Advani,* M. A. Nahon,† N. Haeck,‡ and J. Albronda‡
Delft University of Technology, 2629 HS Delft, The Netherlands

The cueing capabilities of a synergistic flight-simulator motion system are limited primarily by the maximum translational and rotational travel allowed by the motion-base. This travel capability, also known as the workspace, is dictated by the kinematic layout of the motion system. Furthermore, the Jacobian matrix, which maps velocities from platform space to joint space, indicates the dexterity of the mechanism, or the mechanical effort needed by the actuators to move the platform. To systematically design unconventional motion-bases, a methodology has been developed to analyze arbitrary six-degrees-of-freedom motion systems. The approach is based on an optimization program to determine the optimal layout of the motion system, given the workspace performance objectives and the design constraints. This allows the investigation of unconventional platform geometries and actuator attachment points, thus allowing the designer to tailor the workspace as required by the simulation task, to ensure that a satisfactory dexterity is maintained, and to guarantee that the actuator legs do not interfere mechanically. This paper describes the proposed methodology, and shows examples of its applications, first to generic workspaces, and then to the workspace required for the simulation of a large transport aircraft.

Introduction

THE motion bases of modern flight simulators are generally based on a mechanism known as the Stewart platform[1] (originally proposed in 1938 for the testing of tires), which is composed of a base-frame, six prismatic actuator legs (the jacks), and an upper moving platform that carries the payload. The legs are attached in pairs, via gimbal joints, to the upper and lower platforms near the vertices of their triangular frames. Figure 1 shows the general arrangement of a modern[2] Stewart platform.

In the conventional arrangement, the locations of the six upper and six lower gimbal joints can be mapped on circles, and the gimbal pairs are separated by a fixed distance (Fig. 2). The motion of the legs, which are all identical, is constrained by their minimum and maximum lengths. These constraints impose on the mechanism a kinematic envelope that the upper platform can achieve with respect to the inertial reference frame. The envelope of the total kinematic excursion (or travel), known as the workspace, determines the cueing ability of the simulator. A larger workspace in a given direction generally allows longer cue durations. Increasing proportionally the size of all of the members of the Stewart platform simultaneously extends the maximum translational capabilities of the upper platform, but will not affect the rotational limits.

Freely changing the layout itself, thereby deviating from the standard circular arrangement, is possible. However, the designer must prevent the platform from achieving a pose that is at, or close to singularities. In this situation, the ratio of actuator displacements to the resulting platform displacements is very low, meaning that the positioning accuracy may suffer, the mechanical loads can become very high, or the control difficult.

Currently, the design of motion-based flight simulators is carried out by specifying the performance required of the motion cueing mechanism, to generate translational and angular motions. These motions are intended to approximate the specific forces and angular accelerations encountered by the pilot in the simulated aircraft, and work in conjunction with the visual cues to generate the perception of self-motion. The Stewart platform motion-base is sized as a function of these requirements, and other design- and manufacturing-based constraints. Once the simulator is available, a motion drive algorithm is applied to generate the motion cues that would most closely represent those that the aircraft pilot would experience. This motion drive algorithm reads the outputs of the simulated aircraft model, and generates jack length commands for the motion cueing mechanism.[3–5] The motion drive algorithm can be heuristically adjusted such that the motion platform will never exceed the limits of the actuators, and will simultaneously provide reasonable inertial motion cues to the pilot.[6] If the kinematic limits are reached, however, the pilot may detect a bump in the motion, known as a false cue. On the other hand, if the lower-amplitude motion cues are attenuated, there is a risk that they will not be detected by the pilot. Clearly, choosing appropriate motion is an optimization problem in itself.

If, however, the spatial requirements of the motion platform in its motion-cueing role were better known prior to its construction, then the architecture of that mechanism could be specified on the basis of these workspace requirements. Rather than trying to tune a particular motion base, the process could be reversed by tailoring the mechanism to provide the required motion cueing workspace.

The design of the cueing mechanism is by no means restricted to the typically symmetric shape of the Stewart platform. In fact, the designer can choose in three-dimensional space the locations of each upper and lower leg attachment point, as well as the properties of each leg, all of which influence the resulting workspace, and the usefulness of the motion base. Although there has been some limited investigation of unconventional six-degrees-of-freedom geometries,[7] little prior work exists on the optimal design of motion cueing mechanisms. A limited amount of work on the optimal design of six-degrees-of-freedom parallel mechanisms can be found in the field of robotics.[8,9]

This paper describes an approach to the optimal design of asymmetric Stewart-type motion cueing mechanisms. This approach is tested using aircraft responses generated from a flight-simulation model of a Boeing 747-400, which are fed through a unity-gain classical washout filter[3] to predict the simulator trajectories. Based on these trajectories, a workspace is specified, and a mechanism is designed to best fit that workspace.

Received 25 January 1998; revision received 12 October 1998; accepted for publication 26 October 1998. Copyright © 1999 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Director of SIMONA and Assistant Professor, International Centre for Research in Simulation, Motion and Navigation Technologies (SIMONA), Anthony Fokkerweg 1; s.advani@lr.tudelft.nl. Member AIAA.

†Associate Professor, International Centre for Research in Simulation, Motion and Navigation Technologies (SIMONA), Anthony Fokkerweg 1; on sabbatical leave from the University of Victoria, Canada. Senior Member AIAA.

‡Undergraduate Student, International Centre for Research in Simulation, Motion and Navigation Technologies (SIMONA), Anthony Fokkerweg 1.

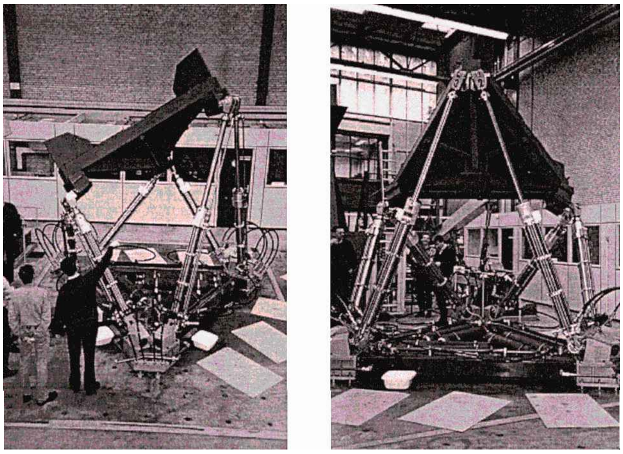

Fig. 1 Modern Stewart platform,[2] composed of six identical actuators attached to a triangular base-frame and moving platform.

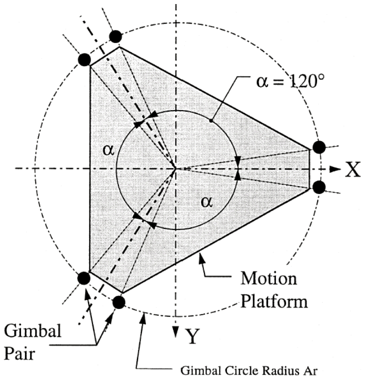

Fig. 2 Typical layout of base-frame or moving platform of a conventional Stewart platform mechanism.

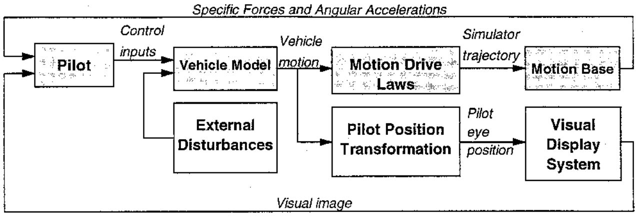

Fig. 3 Process of motion cueing in flight simulators inherently involves calculating the desired motion-base trajectories. Simulator trajectories command the actuators.

Simulator Workspace Requirements

The factors that effectively determine the simulator workspace requirements are given by the shaded boxes in Fig. 3. The workspace required for a particular simulation application depends on the vehicle properties, as well as the maneuvers that must be simulated. Although the motion requirements could be generated using offline simulations, a more realistic approach was adopted here. A full-flight simulator, with a mathematical model representing the Boeing 747-400, was flown by a qualified pilot. Thirty-one training-critical maneuvers were recorded. The time histories of the aircraft specific forces and angular rates were then passed through a classical motion-drive algorithm.[10] This particular motion-drive law was selected because it does not distort the simulator trajectory through the use of adaptive filters. The scaling factors of the washout algorithm, which had previously been tuned to a similar aircraft type, were made unity, and the input signal limiting was removed.

The ellipses in Fig. 4 circumscribe the maximum excursions in each degree of freedom, and will serve as the weighting factors for the design synthesis phase. Because of the six required degrees of freedom, 15 two-dimensional cross-sectional mappings are required. Finally, all of these are combined to create a hyperellipsoid in six-space.

Design Freedom

In this work, deviations from the conventional Stewart platform are introduced in the mechanism design process and the geometry is allowed increasing freedom. In its fully general form, the geometric design of a six-degrees-of-freedom synergistic platform consists of determining the six upper gimbal attachment points and the six lower gimbal attachment point in three-dimensional space, as well as the minimum and the maximum actuator lengths for each of the six actuators. This leaves the designer with $6 \times 3 + 6 \times 3 + 6 \times 2 = 48$ free design variables to select. Varying any one of these variables will influence the final performance of the design. Furthermore, the influence of each individual variable on the final performance is nonlinear and dependent on the values of all other variables. Clearly, considering the large number of design requirements as well, this design problem must be addressed in a systematic way if there is to be any hope of solving it.

For the purposes of the present work, the solution of the fully general design was considered excessively complex, and was found to be beyond the capacity of our available computing platforms. Thus, the number of design variables had to be reduced to a more realistic subset. After some testing and iteration, the subset chosen was one that resulted in nine design variables. The mechanisms investigated were symmetrical about the X–Z plane. The location of the upper and lower attachment points on their respective platforms was generalized from a circle (in the conventional Stewart platform) to an ellipse, as shown in Fig. 5. The size and aspect ratio of these ellipses were made variable, thus resulting in two design variables per platform. The separation between each pair of attachment points was fixed at the minimum that was physically achievable ($2d$ for the upper platform, and $2p$ for the lower platform). One pair of attachment points was fixed to lie symmetrically on the x axis. The other two pairs of attachment points were located symmetrically an angle α from the x axis (and correspondingly an angle β on the lower platform). The angles α and β constituted two additional design variables. Finally, the actuator minimum lengths $q_{\min}$ were also used as design variables, but the symmetry of the platforms reduced these to only three additional design variables. The actuator maximum lengths $q_{\max}$ were then based on the minimum lengths according to

$$q_{\max} = 2q_{\min} - 0.981 \tag{1}$$

where $q_{\max}$ and $q_{\min}$ are in meters. This represents a family of actuators with identical mechanical hardware, except that the cylinders and pistons are cut to differing lengths. Equation (1) is determined empirically and is based on existing motion-base hardware.

The geometry specified by these nine design variables is substantially more general than that allowed by the conventional Stewart platform, and resulted in problems that were solvable in a few hours on our available computing platforms. It was also felt that this reduced set of variables would constitute a good first attempt at the optimal design of a more general class of flight-simulator motion-base. It should be noted that there is no fundamental reason preventing the generalization of this problem to the fully general one of 48 design variables, though it would be expected that each optimization would then require days or weeks of computing time. In addition, it could be questioned how practically viable a fully general configuration may be as a result of the manufacturing complexities it may entail.

Optimization Method

To be amenable to an optimization approach, the problem of optimal platform design must first be cast in the standard form of an optimization problem, namely:

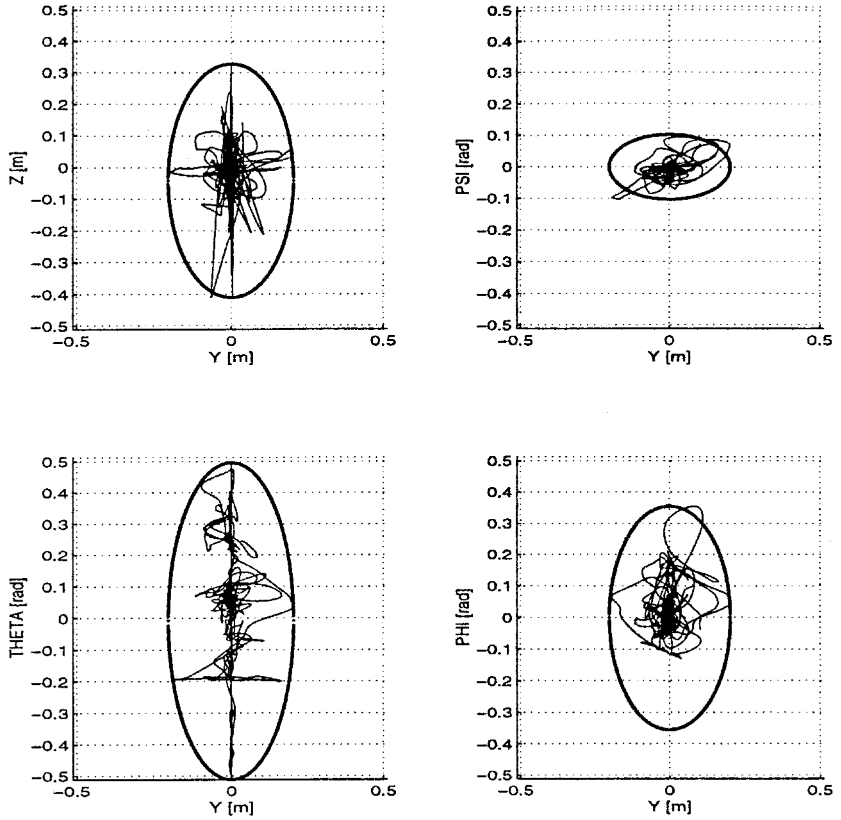

Fig. 4 Trajectory maps of the predicted simulator motions plotted for each pair of degrees of freedom (15 combinations in total), represent the measured vehicle model responses to 31 maneuvers, passed through the motion-drive laws. Weighting ellipses indicate mechanism minimum workspace criteria.

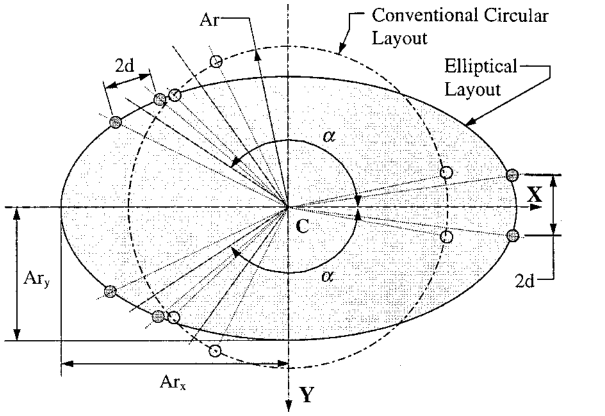

Fig. 5 Layout of standard (circular) and elliptical (shaded) upper platform. Gimbals are mapped along the boundary of the ellipse (rather than circle), and angle α can range between 90 deg and 170 deg. Distance 2d is held constant. Similar variations are allowed to the base-frame layout during the optimization.

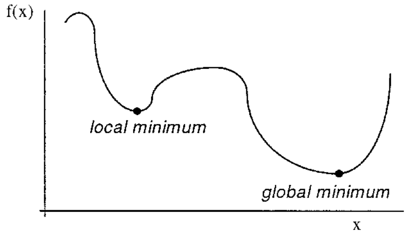

Fig. 6 Distinction between a local minimum and the desired global minimum.

$$
\begin{aligned}
&\underset{x}{\text{minimize}} && f(\boldsymbol{x}) && \\
&\text{subject to} && g_i(\boldsymbol{x}) = 0, && i = 1, n_e \\
& && g_j(\boldsymbol{x}) \le 0, && j = n_e + 1, n_e + n_i
\end{aligned}
\tag{2}
$$

where $\boldsymbol{x}$ is a vector of design variables that, in the present application, includes the major and minor semi-axes of the upper and lower platform (Ar_x, Ar_y, Br_x, Br_y); the gimbal attachment angles (α and β); and the three minimum leg lengths ($q_{\min 1}$, $q_{\min 2}$, $q_{\min 3}$). The objective function $f(\boldsymbol{x})$ defines the scalar quantity that the designer is attempting to optimize. As will be detailed in the next section, the objective function was chosen to maximize the workspace in the various degrees of freedom. The n inequality constraints, $g_i(\boldsymbol{x}) \le 0$, are used to place bounds on the design variables and functions of them. In the present work, these included bounds on the minimum dexterity of the motion-base, as well as on the minimum leg lengths. In a fully general nonlinear programming problem, the functions $f(\boldsymbol{x})$, $g_i(\boldsymbol{x})$, and $g_j(\boldsymbol{x})$ are completely general and have no particular form, e.g., linear, quadratic, convex, etc. When this is the case, as in the present work, the optimization problem can be quite difficult to solve consistently and reliably.

Many techniques exist to solve general optimization problems.[11,12] All of these techniques are iterative in nature, i.e., they start from an initial guess for the solution, which is supplied by the user, and take steps toward a local optimum that may or may not be the global optimum to the optimization problem. Thus, an important consideration in any minimization problem is the number of minima that the problem can be expected to have. In general, although there can only be one true global minimum, there may also be a number of local minima. This is illustrated in Fig. 6 for the unidimensional case of an objective function $f(\boldsymbol{x})$ which varies with a single design variable $\boldsymbol{x}$. A result of this situation is that the solution algorithm may converge onto one of the local minima without detecting the global minimum. Our results indicated that our problem, as formulated, does indeed have multiple local minima. To circumvent this problem, a technique was implemented in which 70 different randomly chosen initial starting points were used for each optimization problem solved. Each of these would converge onto a particular minimum, and these were then sorted. Typically, 40–50% of the starting points would converge onto the same best minimum found, and this gave some assurance that this should be the global minimum. However, it cannot be proven that this minimum was truly global because of the complexity of the functions involved. The search for techniques that can consistently guarantee convergence to the global optimum for fully general optimization problems is still an active area of research.

The size and direction of the steps taken toward the optimum are determined by the principle of operation of the particular technique used. Each solution iterate is checked against certain optimality conditions,[13] which must be satisfied by the true solution to the problem. If these conditions are not satisfied within a user-supplied tolerance, another step is required. The computational cost of each iteration, as well as the convergence characteristics of the technique, will determine the CPU time required to find the solution, as well as the variety of problems for which the algorithm will successfully converge. A comprehensive survey of evaluations of these techniques[12] concludes that overall, the sequential quadratic programming and the generalized reduced gradient techniques seem to work best for a variety of constrained nonlinear optimization problems. In the present work, the sequential quadratic programming method[13] was chosen. The principle of operation of the sequential programming technique is as follows: at each iteration, based on the current solution iterate, the algorithm formulates and solves a simplified problem whose solution is used as a step toward an improved solution iterate. This simplified problem consists of a quadratic objective function and linear constraints, i.e., a linearly constrained quadratic programming subproblem. The new solution is then checked against the optimality conditions for the true problem to determine whether another step must be taken.

Specification of the Objective Function

A hyperellipsoid, created by merging the two-dimensional mappings shown in Fig. 4 into a six-dimensional mathematical function, describes the required motion volume. Its mathematical form is defined as

$$\left(\frac{X-X_0}{\rho_X}\right)^2+\left(\frac{Y-Y_0}{\rho_Y}\right)^2+\left(\frac{Z-Z_0}{\rho_Z}\right)^2+\left(\frac{\psi-\psi_0}{\rho_\psi}\right)^2$$
$$+\left(\frac{\theta-\theta_0}{\rho_\theta}\right)^2+\left(\frac{\phi-\phi_0}{\rho_\phi}\right)^2\le 1 \tag{3}$$

where $X_0,\ldots,\phi_0$ denote the neutral position of the motion base (when all actuators are at their midstroke). The variables X, Y, and Z represent the translational position of the centroid of motion base, whereas ϕ, θ, and ψ are the Euler angles that describe the platform's roll, pitch, and yaw. The neutral position also corresponds to the midpoint of the hyperellipsoid. The weighting factors $\rho_X,\ldots,\rho_\phi$ denote the ellipsoid semi-axis length in each direction. They are determined by drawing ellipses that nearly circumscribe the foreseen aircraft trajectories, as shown in Fig. 4 for 4 of the 15 two-dimensional mappings, and determining the major semi-axis lengths of these ellipses in each of the six degrees of freedom. A few assumptions are made in the current solution. When the trajectories are projected in a particular pair of degrees of freedom, only the maximum excursion or rotation along a particular axis is considered. This assumes that the trajectories will then fall within the resulting ellipse, which in most cases, it does. For this particular case, this method results in the following $\boldsymbol{\rho}$ vector: $[\rho_x, \rho_y, \rho_z, \rho_\psi, \rho_\theta, \rho_\phi]^T = [0.8206, 0.2035, 0.3695, 0.1030, 0.5038, 0.3553]^T$.

Note that one could also describe the ellipsoid around the total trajectories in six degrees of freedom, and then map this larger ellipsoid onto the respective two-dimensional surfaces. This would slightly change the preceding weighting vector.

The objective function to be maximized is then specified as

$$\bar{X}^2+\bar{Y}^2+\bar{Z}^2+\bar{\psi}^2+\bar{\theta}^2+\bar{\phi}^2=R \tag{4}$$

where $\bar{X}$, $\bar{Y},\ldots$ are generalized coordinates, denoting a scaled, nondimensional distance from the platform's neutral position, e.g.,

$$\bar{X}=(X-X_0)/\rho_X \tag{5}$$

The factor R, which is called the weighted radius of the hyperellipsoid, scales the ellipsoid proportionally without changing its shape. The objective function given by Eq. (4) attempts to find the maximum weighted radius $R_{\max}$, describing the largest hyperellipsoid that just fits into the workspace of the motion platform. If $R_{\max}\ge 1$, the largest scaled ellipsoid that fits in the workspace, described by Eq. (4), will be larger than the design ellipsoid given by Eq. (3). Because the latter is known to circumscribe the design trajectories, this effectively guarantees that the workspace will contain the design trajectories. Alternatively stated, when $R_{\max}=1$, we know that the design trajectories fit inside the design ellipsoid of Eq. (3), which in turn, fits inside the largest scaled ellipsoid, which in turn, fits inside the workspace.

Because the motion-base must be symmetric about the X–Z plane, it follows that

$$Y_0=\psi_0=\phi_0=0 \tag{6}$$

The X_0–Z_0 location of the platform reference point indicates the neutral position of the platform with respect to the base, whereas θ_0 is the initial platform pitch angle. All of these values are determined during the optimization process.

Estimation of $R_{\max}$

Because there is no analytical method available to calculate $R_{\max}$, a numerical approximation must be used. However, the computational effort of this approximation should be kept within reasonable limits because each optimization loop typically requires a few hundred iterations to locate an optimum.

The approach used here is to restrict the analysis of the workspace to cross-sectional planes of the hyperellipsoid. For instance, in a three-dimensional case, an ellipsoid could be evaluated by restricting the analysis to its three cross-sectional planes, namely the X–Y, X–Z, and Y–Z plane. In the six-dimensional case, there are 15 cross-sectional planes in the volume, including combinations of both translational and rotational directions: X–Y, X–Z, X–ψ, X–θ, X–ϕ, Y–Z, Y–ψ, Y–θ, Y–ϕ, Z–ψ, Z–θ, Z–ϕ, ψ–θ, ψ–ϕ, and finally, θ–ϕ.

This strategy reduces the estimation of $R_{\max}$ to the estimation of 15 inscribed ellipses, i.e., determining the local R_{XY}, $R_{XZ},\ldots,R_{\theta\phi}$, in the cross-sectional planes of the workspace. The ellipse that yields the smallest value for $R_{\max}$ is considered critical, thus implying

$$R_{\max}\approx\min(R_{XY}, R_{XZ},\ldots,R_{\theta\phi}) \tag{7}$$

Figure 7 depicts this concept for three cross sections of the translational workspace, namely X–Y, X–Z, and Y–Z. Here, the weighting

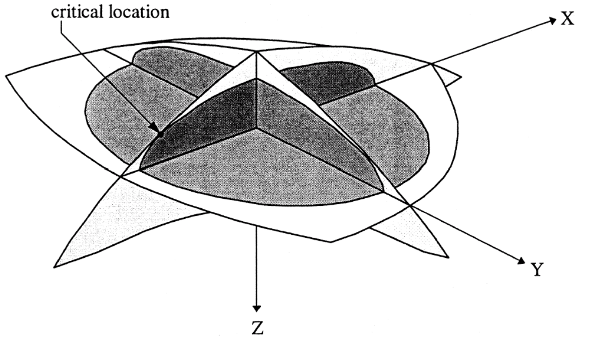

Fig. 7 Three inscribed ellipses in cross-sectional planes of the workspace.

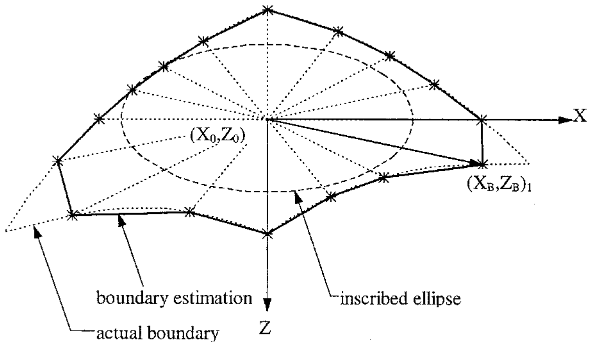

Fig. 8 Estimate for X–Z motion boundary based on 16 discrete boundary searches.

factors are $\rho_X = \rho_Y = 1$ and $\rho_Z = 0.5$, and the critical ellipse is located in the X–Z plane.

Determining the Weighted Radii of the Inscribed Ellipses

To define the inscribed ellipse for each cross section, the boundary of the workspace in that cross section must first be determined. Although an analytical determination of motion boundaries is possible,[14] it is restricted to translational motions, and is computationally intensive.

The numerical approach used here to find the boundary is, however, straightforward, as there is a direct relation between the platform pose and the leg lengths. By moving the platform from its neutral position in a certain direction, the leg lengths are changed. At some point, one of these leg lengths will reach its specified length limit, thus restraining the platform motion.

The workspace boundary can be estimated by repeatedly searching for the boundary, in a different direction each time (the offset direction), while always starting from the same initial position. Figure 8 shows how a boundary estimate for the X–Z plane is found with 16 discrete offset directions.

The weighted radius of the maximum inscribed ellipse can be found by an appropriate choice of offset directions. This will now be described mathematically for the X–Z plane, using N_{bp} discrete boundary searches. First, equally spaced angles α_i are defined:

$$\alpha_i = (i/N_{bp}) \times 2\pi, \qquad i = 1, \ldots, N_{bp} \tag{8}$$

Next, an offset vector is defined for each direction:

$$\begin{bmatrix} \Delta X \\ \Delta Z \end{bmatrix}_i = \begin{bmatrix} \rho_X \times \cos(\alpha_i) \\ \rho_Z \times \sin(\alpha_i) \end{bmatrix} \tag{9}$$

Then, the boundary points are detected by an iterative process, using the Jacobian matrix, as well as the actuator lengths and their limits. The process converges quadratically, normally yielding the boundary points in 2–3 iterations. One parameter k_i per direction defines the boundary points:

$$\begin{bmatrix} X_B \\ Z_B \end{bmatrix}_i = \begin{bmatrix} X_0 \\ Z_0 \end{bmatrix} + k_i \begin{bmatrix} \Delta X \\ \Delta Z \end{bmatrix}_i \tag{10}$$

The accuracy of the workspace determination depends on the number of discrete boundary points that are analyzed per cross section (N_{bp}), as well as the accuracy in the boundary-point estimation itself (ε). In this work, we used $N_{bp} = 48$ and $\varepsilon = 10^{-5}$.

Finally, the weighted radius of the inscribed ellipse is defined as the minimum of all values k_i:

$$R_{XZ} = \min\left(k_1, k_2, \ldots, k_{N_{bp}}\right) \tag{11}$$

Applying Eq. (7) then represents an approximation to the maximum weighted radius of the inscribed hyperellipsoid, $R_{\max}$, because only the cross-sectional planes of the workspace are considered.

Dexterity

In the six degrees-of-freedom synergistic mechanism, six actuated links determine the position and rotation of the platform. By contracting and extending the links, the final platform pose (position and rotation) is altered. This implies that there should be a one-to-one relation between the actuator link positions (the length of the actuators) and the platform pose. Problems can arise with the controllability of the platform when the platform pose varies substantially for very small motions of the actuators.

This situation is called a singularity of the platform and should be avoided throughout the workspace, as it can lead to malfunctions and failure of the platform. Furthermore, designs that yield nearness to singularities should be avoided, as they can result in excessively high actuator loads.

Singularities can be detected by analyzing the Jacobian matrix, which relates rates of change in platform pose to rates of change in actuator lengths:

$$\dot{q} = J\dot{x} \tag{12}$$

where $\boldsymbol{q}$ denotes the actuator lengths, and x denotes the platform pose, including both translations and rotations. The 6×6 matrix J is defined as

$$J = \begin{bmatrix} \dfrac{\partial q_1}{\partial x} & \cdots & \dfrac{\partial q_1}{\partial \phi} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial q_6}{\partial x} & \cdots & \dfrac{\partial q_6}{\partial \phi} \end{bmatrix} \tag{13}$$

Note that J is not constant, but varies throughout the workspace. The dexterity D is the reciprocal of the condition number of J, and can be defined as

$$D = \sigma_{\min}/\sigma_{\max} \tag{14}$$

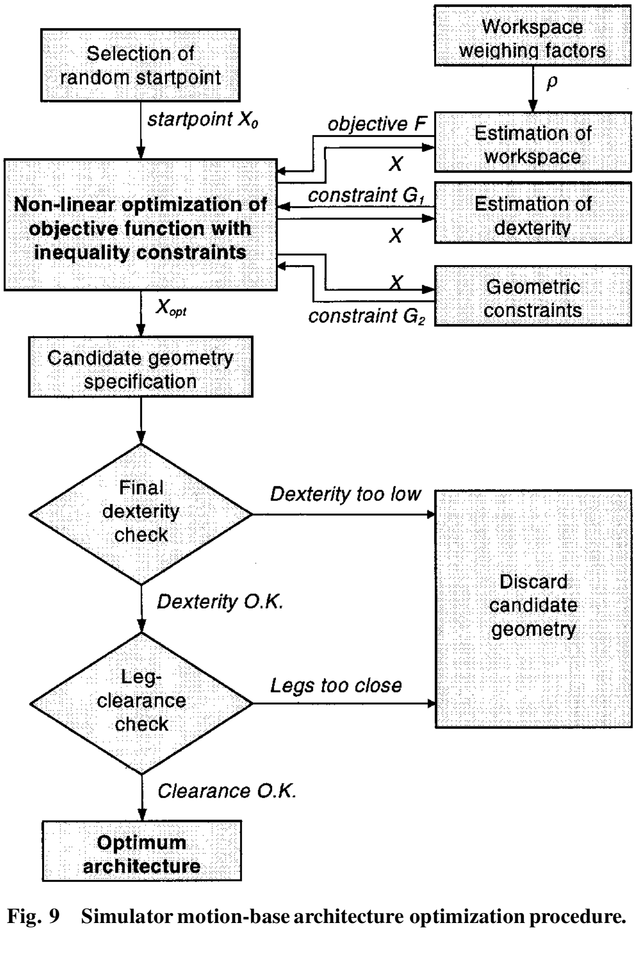

Fig. 9 Simulator motion-base architecture optimization procedure.

in which $\sigma_{\min}$ and $\sigma_{\max}$ denote the minimum and maximum singular value of J, obtained by a process called singular value decomposition.

The dexterity indicates the controllability of the platform. A value of $D = 1$ indicates an isotropic condition, in which equal actuator effort is needed to obtain motion in every direction. When approaching a singularity of the platform, $\sigma_{\min}$ will tend to zero, and thus, a value of $D = 0$ will indicate a singular configuration.

Optimizing the Motion-Base Architecture

The optimization process attempts to define a motion system architecture that achieves the desired workspace criteria, while also maintaining a dexterity no less than 0.2, a value obtained through design experience. Because of the computational intensity, only an estimation of the dexterity is calculated during the optimization (at the 64 minimum/maximum actuator length combinations and at the boundary points of the workspace evaluation). The workspace and dexterity are thus used to specify the optimal motion system architecture, as shown in Fig. 9.

Bounds are placed on the design variables to prevent them from reaching unreasonable values. The major and minor semi-axes of the ellipses defining the upper and lower platforms are constrained to lie within the range [1.5 3.0] m, the angles α and β are constrained to the range [90 170] deg, and the minimum leg lengths are constrained to the range [1.381 2.981] m. And finally, to prevent all of the minimum leg lengths from going to their maximum limit (in an effort to maximize the platform workspace), an upper bound of 12.786 m is placed on the sum of all the minimum leg lengths.

Once the optimum has been found, a more thorough check of the minimum dexterity is performed, by dividing each of the actuator lengths into 10 sections, and evaluating the dexterity at each of the 10^6 combinations. During these calculations, the minimum absolute distance between all of the legs is also computed. If this minimum ever falls below a specified level (in this case 15 cm), the program indicates that there is a potential leg-crossing situation and the candidate geometry is rejected.

Discussion of Results

Three test cases are now shown to illustrate the use of the design software. In the first test case, the weighting factors in all degrees of freedom are equal, and the relationship between the angular and rectilinear motions specified such that 1 rad of rotation is weighted equally to 1 m of displacement. In the second test case, because the use of pitch motions in flight simulators recreates both the aircraft pitch attitude as well as long-duration specific forces through tilt-coordination, the pitch motion is weighted by a factor of 2, whereas all others are unity. Finally, in the third test case, the aircraft- and washout-specific weighting factors of $[\rho_X, \rho_Y, \rho_Z, \rho_\psi, \rho_\theta, \rho_\phi]^T = [0.821, 0.204, 0.369, 0.103, 0.504, 0.355]^T$, described in an earlier section, were used.

Table 1 gives the final values of the design variables, the constraint functions, and $R_{\max}$ at the global optimum for each of the three test cases. The salient points in this table are as follows:

1) The upper platform is circular or near-circular in all test cases, and the semi-axes lie against their minimum constraint of 1.5 m.

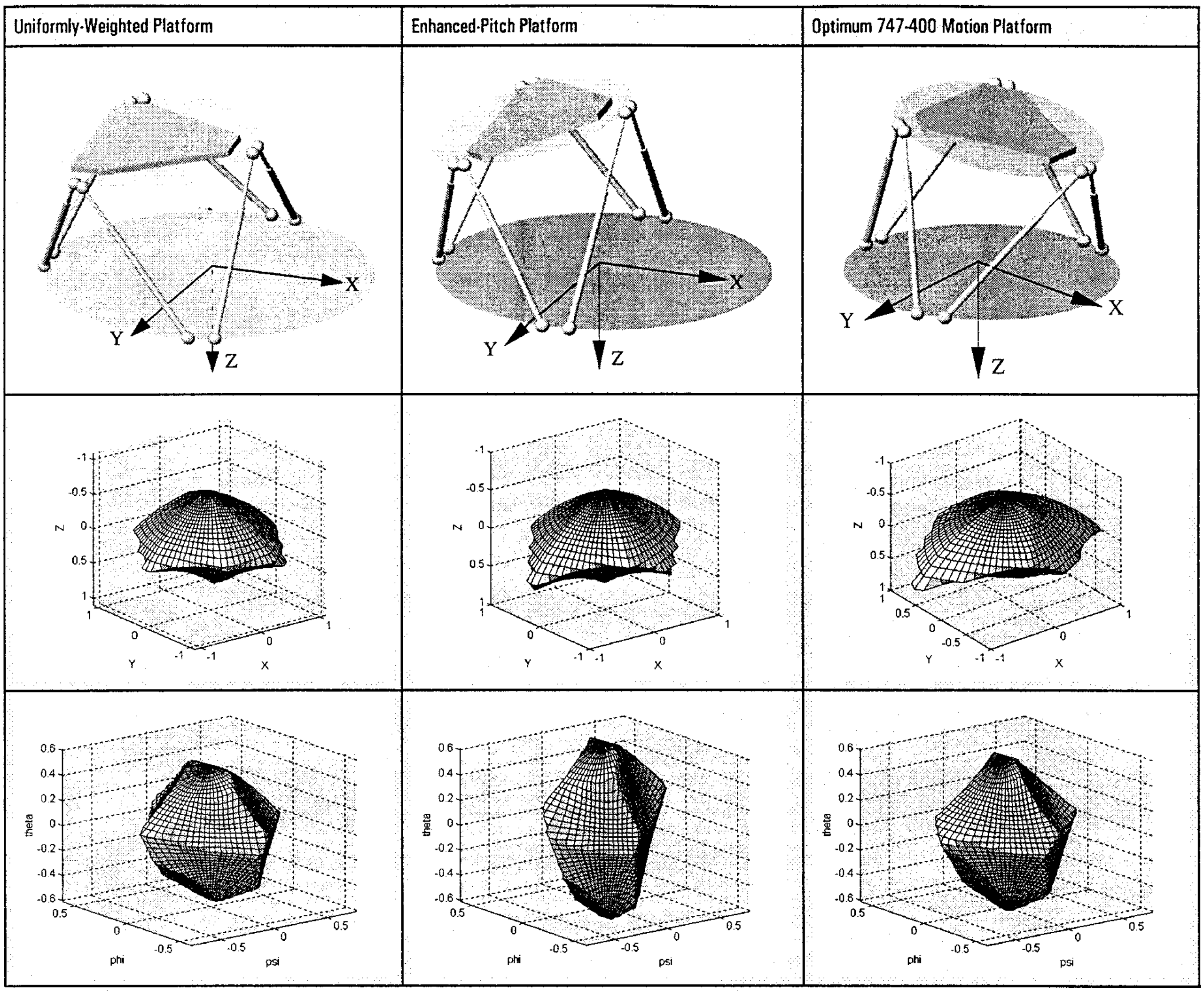

Fig. 10 Graphical depiction of the geometry of the motion-base in the neutral position, and a three-dimensional representation of translational and rotational workspaces.

Table 1 Values of the design variables, constraint functions, and objective function for the optimum geometries

	Unity weighting	Enhanced pitch	Boeing 747-400
Ar_X, m	1.500	1.500	1.500
Ar_Y, m	1.500	1.581	1.500
Br_X, m	2.544	2.415	1.837
Br_Y, m	1.808	2.055	2.130
α, deg	123.5	112.5	123.5
β, deg	101.4	90	106.2
$q_{\min 1}$, m	2.203	2.437	2.213
$q_{\min 2}$, m	2.139	1.987	2.042
$q_{\min 3}$, m	2.051	1.970	2.138
D	0.2	0.2	0.2
$\Sigma q_{\min}$	12.786	12.786	12.786
$R_{\max}$	0.414	0.304	1.014

2) The lower platform is always elliptical, though its major semiaxis lies in the x direction in the first two test cases, but in the y direction in the third test case.

3) The angle β lies against its lower bound of 90 deg in the second test case.

4) The dexterity constraint is active in all three test cases, indicating that the dexterity attains a minimum value of 0.2 somewhere in each platform's workspace

5) The constraint on the sum of all minimum leg lengths is active in all three test cases. In the first and third test cases, the individual minimum leg lengths do not vary substantially, whereas in the second test case, actuators 1 and 6 are considerably longer than actuators 2, 3, 4, and 5.

6) $R_{\max}$ is less than unity in the first two test cases, indicating that only a scaled-down hyperellipsoid was achievable.

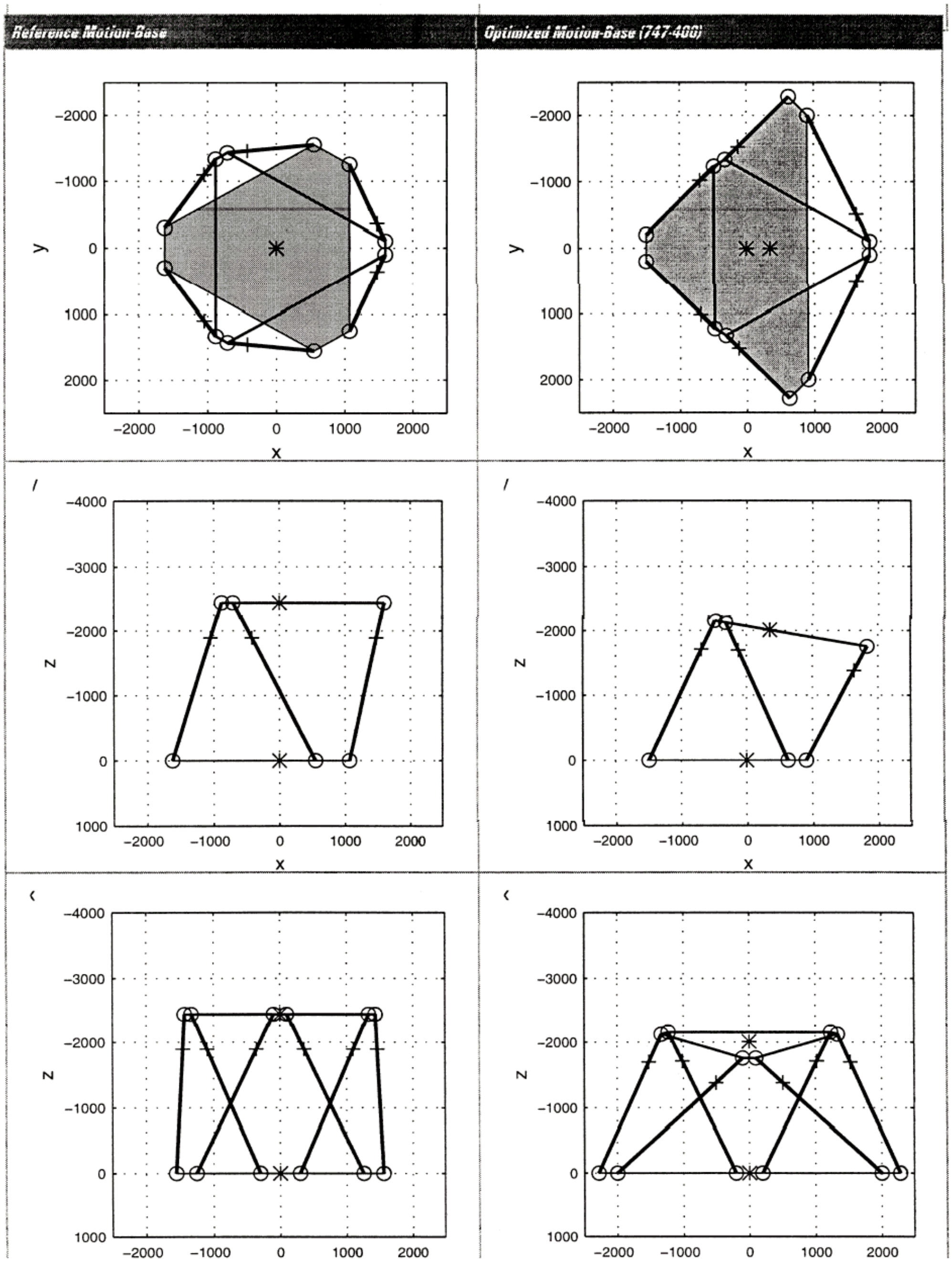

Fig. 11 Graphical depiction of the geometry of the reference (left column) and the optimized (right column) motion bases, showing top, side, and front views, respectively.

7) $R_{\max}$ is greater than unity in the third test case, indicating that the Boeing 747-400 trajectories are achievable with this motion-base geometry.

Figure 10 shows the resulting mechanisms graphically as well as their translational and rotational workspaces. The first column, showing the results of the uniformly weighted ellipsoid, has the most uniform workspace of the three geometries. However, the layout of the mechanism deviates considerably from the standard Stewart platform. When the pitch weighting is doubled, the mechanism in the second column is obtained. The rotational workspace shows a distinct elongation along the pitch axis, achieved with a rather unconventional form of the mechanism. Finally, using the weighting function derived from the predicted trajectory of the 747-400 aircraft, the mechanism in column 3 is obtained. Its workspace shows that the x-direction motions are the most emphasized, relative to all others. While this mechanism does not deviate drastically from conventional designs, it offers optimum cueing capabilities, based on the washout filter used. Moreover, the forward gimbals are lower than the aft pairs, which would allow easier placement of the visual display system with respect to the pilot eye position. In fact, this architecture lends itself well to a full-flight simulator.

Figure 11 demonstrates the effectiveness of the optimization, showing both the Stewart platform and the optimized architectures.

The goal set forth in the beginning of this research was to tailor the workspace available by manipulating the design variables, while remaining within the specified constraints. The freedom to allow the upper and lower attachment points to fall on elliptical, rather than on circular lines, was found to yield a significant advantage. This does, however, require unequal actuator lengths. The costs related to this nonuniformity can be reduced by developing actuators with similar hydromechanical component geometries, while allowing only the piston–cylinder lengths to vary.

Furthermore, this approach allows the realization of mechanisms as functions of the cueing requirements, rather than trying to work around the workspace constraints of standard Stewart platforms through excessive attenuation of the aircraft motions, or the use of nonlinear adaptive filters.

Conclusions

The design of a flight simulator is a highly integrated multidisciplinary activity with the eventual goal of providing realistic cues to the simulator pilot. This paper reconsiders the conventional motion system design procedure, by addressing the cueing requirements prior to specifying the motion system, and then defining an appropriate motion system geometry. The flexibility offered by this technique allows tailoring the motion system to particular needs. The optimization approach adopted here emphasizes that the specification of a motion platform geometry need not be restricted to the conventional Stewart platform.

While this technique thoroughly examines the kinematic aspects of motion-bases, further analysis should follow. The remaining task involves designing the crew module, a visual display system, and the structure to integrate all onboard systems with the moving platform. Then, with knowledge of the mass and stiffness properties of the entire moving load, detailed dynamic analyses can be performed.

It should be highlighted, however, that the platforms of multipurpose facilities, such as the SIMONA Research Simulator, often require a wide range of workspace capabilities, as well as a high bandwidth. In these instances, a generic Stewart platform may be the optimal choice, given the wider range of applications required than with a training simulator.

Acknowledgments

This paper overviews part of the first author's doctoral research. The Flight Training Centre of KLM Royal Dutch Airlines is thankfully acknowledged for their input in this research, particularly with regard to the use of their facilities for the generation of the motion envelope of the 747-400 aircraft. The sabbatical stay of M. A. Nahon was partially funded by The Netherlands Scientific Agency NWO. Herold Cremer and Bert Thalen are thanked for the creation of the motion-base CAD drawings.

References

[1] Stewart, D., "A Platform with Six-Degrees-of-Freedom," *Proceedings of the Institute of Mechanical Engineers*, Vol. 180, Pt. 1, No. 5, 1965, pp. 371–386.

[2] Advani, S. K., et al., "SIMONA—A Reconfigurable and Versatile Research Facility," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, AIAA, Reston, VA, 1997, pp. 458–466.

[3] Nahon, M. A., Reid, L. D., "Simulator Motion-Drive Algorithms: A Designer's Perspective," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1990, pp. 356–362.

[4] Ish-Shalom, J., "Design of Optimal Motion for Flight Simulators," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, 1982.

[5] Martin, E. A., "Force and Motion Cueing, In Flight Simulation Update," *Aerospace America*, Jan. 1996.

[6] Grant, P. R., and Reid, L. D., "Motion Washout Filter Tuning: Rules and Requirements," *Journal of Aircraft*, Vol. 34, No. 2, 1997, pp. 145–151.

[7] Nahon, M. A., Ricard, R., and Gosselin, C. M., "A Comparison of Flight Simulator Motion-Base Architectures," *Proceedings of Making it REAL, CEAS Symposium on Simulation Technologies*, Delft Univ. of Technology, Delft, The Netherlands, 1995, pp. 1MSY02–16MSY02.

[8] Pittens, K. H., and Podhorodeski, R. P., "A Family of Stewart Platforms with Optimal Dexterity," *Journal of Robotic Systems*, Vol. 10, No. 4, 1993, pp. 463–479.

[9] Stoughton, R. S., and Arai, T., "A Modified Stewart Platform Manipulator with Improved Dexterity," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 2, 1993, pp. 166–173.

[10] Reid, L. D., and Nahon, M., "Response of Airline Pilots to Variations in Flight Simulator Motion Algorithms," *Journal of Aircraft*, Vol. 25, No. 7, 1988, pp. 639–646.

[11] Gill, P., Murray, W., and Wright, M. H., *Practical Optimization*, Academic, San Diego, CA, 1981.

[12] Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M., *Engineering Optimization: Methods and Applications*, Wiley, New York, 1985.

[13] Schittkowski, K., "NLPQL: A Fortran Subroutine Solving Constrained Nonlinear Programming Problems," *Annals of Operations Research*, Vol. 5, No. 1–4, 1986, pp. 485–500.

[14] Gosselin, C., "Determination of the Workspace of 6-DOF Parallel Manipulators," *Journal of Mechanical Design*, Vol. 112, No. 3, 1990, pp. 331–336.