Simulator Motion-Drive Algorithms: A Designer's Perspective

Meyer A. Nahon* and Lloyd D. Reid†
The University of Toronto, Toronto, Ontario, Canada

Changes in the design of software algorithms for generating physical motion in flight simulators have typically been put forward on the grounds of improved motion cueing. Little attention has been paid to more practical criteria such as computational cost, ease of adjustment, or evaluation by experienced pilots in a realistic simulation environment. A comparison of three of the algorithms most commonly found in the literature has been performed: classical washout, optimal control, and coordinated adaptive. This consisted of pilot evaluations of these algorithms implemented on a six-degree-of-freedom flight simulator simulating a large transport aircraft during low-altitude flight and ground maneuvering. This paper presents the results of that study from the designer's viewpoint. In it, we contend that, with enough effort, most algorithms can be massaged to perform reasonably well, and that a more important consideration is the ease with which a given algorithm can be brought to high performance levels. If this criterion is used, it appears that the classical algorithm is a good starting point, and that the benefits of an adaptive algorithm can be added gradually to obtain the advantages conferred by nonlinear filtering and "intelligent" cost functions.

Nomenclature

a_{AA} = body-axis components of the aircraft acceleration at the cockpit reference point
aC = intermediate acceleration variable in the adaptive filters
a_{SI}, S_I = inertial components of the simulator reference-point acceleration and position
f_{AA} = body-axis components of the aircraft specific force at the cockpit reference point
g, g_A, g_I = gravitational acceleration, scalar and as components in the body and intertial frames
L_{IS} = rotation matrix that transforms vector components from the simulator reference frame to the inertial frame
p' = adaptive coefficients in the coordinated adaptive algorithm
s = Laplace operator
T_S = transformation matrix from angular velocity to Euler angle rates
W_{11}, W_{12}, W_{21}, W_{22} = filters in the optimal control algorithm
β_A, β_S = aircraft and simulator Euler angles
ω_{AA} = body-axis components of the aircraft angular velocity
ω_n, ζ = filter break frequency and damping ratio

Introduction

THE past 20 years have seen numerous advances in the field of flight simulation that have resulted in an increasing degree of simulation fidelity and realism. Many of these advances have been prompted by the increased sophistication of digital hardware that has enabled, for example, more realistic visual displays and smoother, more accurate control of the motion and control-loading subsystems. Also, advances in mechanical hardware, such as the use of hydrostatic actuators and the highly developed synergistic motion-base, have yielded motion systems with excellent performance characteristics in the frequency range of interest. The more recent aircraft models have allowed accurate representation of flight characteristics throughout the flight envelope, whereas computing power has become available to allow all relevant complexity in a real-time simulation to be included at a reasonable cost.

At present, one of the primary obstacles to the generation of realistic motion is the software that transforms aircraft motions into commanded simulator motion, commonly called the washout algorithm. The purpose of this software is to present the pilot with the best possible motion cues within the limited motion capability of the simulator, throughout the flight envelope. As examples of the complexity of this problem, consider that what is "best" will lead the designer into the field of human factors, whereas the criterion "throughout the flight envelope" usually leads to some investigation of adaptive software. The problem is further complicated by the nondeterministic nature of pilots.

Numerous schemes have been proposed for motion generation in simulators, beginning with variations on the classical algorithm,[1-3] followed by variations on the adaptive algorithm,[4-7] the optimal control approach,[8-10] the quasioptimum control technique,[11] and the subliminal scheme[12] among others. We have attempted to perform an unbiased comparison[13-15] of the more common techniques in order to choose a best algorithm for the University of Toronto Institute of Aerospace Studies (UTIAS) flight research simulator. The simulator was configured to represent a large transport aircraft during low-altitude and ground maneuvers and was evaluated by a group of experienced airline pilots.[16] Results of this work were previously summarized from the human factors' viewpoint[16] by presenting the pilots' evaluations of the various algorithms tested. The present work complements that summary by presenting our experience from the designer's viewpoint. Together, the two papers are meant to provide guidance and practical advice on motion software implementation in order to allow other people involved in this field to converge on a best solution more quickly. The results are expected to apply to the motion simulation of transport aircraft, particu-

Received April 4, 1988; revision received Oct. 20, 1988.

*Project Engineer, Institute for Aerospace Studies.
†Associate Director, Institute for Aerospace Studies. Associate Fellow AIAA.

larly when using a six-degree-of-freedom motion-base of the type used for commercial airline training.

Previous Work

Before embarking on this motion algorithm comparison, the literature was reviewed and papers describing numerous techniques and their variants were located. Difficulties were identified with a number of these, sometimes only with the benefit of hindsight.

Some of the schemes presented are purely conjectural in that they present an idea (sometimes intriguing) that was neither implemented nor evaluated by experienced pilots.[6,11] Without these features, the completeness of the study and relevance of the professed improvements are questionable. This is closely related to the fact that certain studies concentrate on improving the shape of a motion or sensation time history. In fact, "it is possible to scrutinize an unending number of time histories without knowing for certain whether one design is better than another because the characteristics that are being sought in the time history have not been pinpointed."[11] This statement has been amply verified in our work and is a tribute to the complexity of pilots' perception of motion.

Some works consider a limited number of degrees of freedom[6,9] (often only one or two) or unrealistic motion-base constraints. For example, although an available lateral travel of 10 m can permit a realistic coordinated turn simulation,[6] most commercial simulators are limited to much less than this. Thus, for an algorithm to be useful for application to commercial airline simulators capable of very limited six-degree-of-freedom motion, it should be adapted to those capabilities and constraints.

Perhaps the most serious limitation of previous comparisons is the question of how much effort has been invested in "comparing apples with apples." As will be discussed later, the utility of a given scheme can be vastly improved or degraded by the choice of parameters used. This choice is largely a trial-and-error process, so that a statement such as "scheme A is better than scheme B" implicitly assumes that the researcher has made the effort to bring each scheme to its highest level of performance. In fact, all that can be claimed is that with the parameter set chosen, scheme A is better than scheme B, a statement of greatly reduced significance. For example, although Refs. 3-5 constitute a remarkably complete investigation of the adaptive algorithm, it is not clear how well the classical algorithm was adjusted for comparison.

The ease of adjustment of a given scheme was not considered in any of the articles surveyed. Whereas some algorithms have the advantage that a given pilot complaint can be remedied by a simple adjustment to one or two parameters, other schemes are relatively opaque and rectification of a problem can require many complex iterations of parameter adjustment. This is important when one considers that the usual process of a typical training simulator acceptance involves "flight testing" by the airline's acceptance pilot in conjunction with the simulator manufacturer's motion expert. In this process, there is usually a limited amount of time available to adjust the washout parameters to satisfy the pilot; and a limited but easy-to-tune algorithm could yield better results than one with greater potential that is difficult to adjust.

Finally, the computational cost of more sophisticated schemes is rarely reported. Since this software must run in real time, this criterion should also be relevant in the evaluation of an algorithm.

Description of the Algorithms Tested

The present work investigated the three following techniques: 1) classical washout, 2) optimal control, and 3) coordinated adaptive. A detailed treatment of these was previously reported,[13-15] but some of their salient characteristics are given here for completeness. A comparison of the three techniques follows the individual descriptions.

Classical Washout

The classical algorithm is the most widely used in commercial simulators. It is characterized by the empirically determined combination of linear high- and low-pass filters whose break frequencies and damping ratios can be adjusted off-line by trial and error. A flowchart of the classical algorithm implemented in the present work is shown in Fig. 1, and the filter parameters used in the version with the most favorable pilot comments (denoted as CW2[16]) are given in Table 1. Attention is drawn to the following points in Fig. 1:

1) The inputs to the algorithm are the three aircraft cockpit specific forces in body axes, defined as $f_{AA} = a_{AA} - g_A$, and the three aircraft angular rates.

2) The aircraft specific forces are high-pass filtered to yield the simulator translational accelerations under the assumption that low-frequency specific forces would tend to drive the motion base to its limits. The longitudinal and lateral specific forces are also passed through low-pass filters, scaled, and rate-limited to produce pitch and roll tilt angles, respectively. The purpose of this "tilt-coordination" mechanism is to orient the gravity vector in the simulator in the same way relative to the pilot as the low-frequency specific force in the aircraft, thus allowing sustained aircraft accelerations to be simulated. Since this trick is not available in the vertical direction, pilots frequently complain of the lack of sustained changes in vertical loading (this applies to all the algorithms studied).

3) The aircraft angular motion is high-pass filtered to yield the high-frequency component of simulator angular motion, whereas the tilt mechanism described in point 2 supplies the low-frequency component. An analysis of the frequency response of the washout algorithm was performed.[14] In this analysis, all scaling and limiting were neglected and small angles assumed. It was found that the overall transfer function between uncoordinated aircraft pitch or roll motion and the

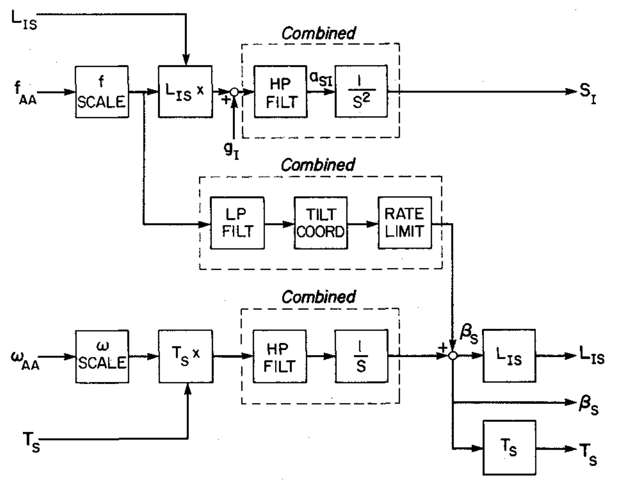

Fig. 1 Classical algorithm.

Table 1 CW2 filter characteristics

Filter	Order	ω_n	ζ
x translational high-pass	2	2.5	1.0
y and z translational high-pass	2	4.0	1.0
All rotational high-pass	1	1.0	—
x translational low-pass	2	5.0	1.0
y translational low-pass	2	8.0	1.0

Second-order high-pass filters of the form $s^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$
First-order high-pass filters of the form $s/(s + \omega_n)$
Second-order low-pass filters of the form $\omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$

Scale factor on all inputs: 0.5
f_{AA} input limit (all components): 10 m/s^2
ω_{AA} input limit (all components): 34.4 deg/s
Tilt-rate limit: 3 deg/s

Table 2 Algorithm comparison

Algorithm	Total number of differential equations	CPU time/ iteration, ms[a]	Number of free parameters	Transparency
Classical washout	13	1.0	21	Very good
Optimal control	24	1.3	38	Fair
Coordinated adaptive	38	1.8	64	Fair

[a]Using a Perkin Elmer 3250 digital computer.

Table 3 OC2 filter characteristics[a]

Weight	Pitch/surge	Roll/sway	Yaw	Heave
R_{11}	0.9	1.0	0.577	0.5
R_{22}	0.158	0.1	—	—
Rd_{11}	0.158	0.1	0.577	0.5
Rd_{22}	0.158	6.0	0.577	0.5
Rd_{33}	0.158	6.0	—	0.5
Q_{11}	0.447	5.0	1.0	1.0
Q_{22}	0.894	0.2	—	—
An_{11}	−0.2	−0.2	−1.0	−1.0
An_{22}	−1.0	−0.0001	—	—
ρ	8.0	1.0	0.8	10.0

Filter	W_{22} surge	W_{22} sway	W_{22} heave	W_{12} pitch/ surge	W_{12} roll/ sway	W_{11} yaw
Order	4	4	3	5	5	3

[a]Symbols as defined in Ref. 13.
Scale factor on all inputs: 0.5
a_{AA} input limit (all components): 10 m/s^2

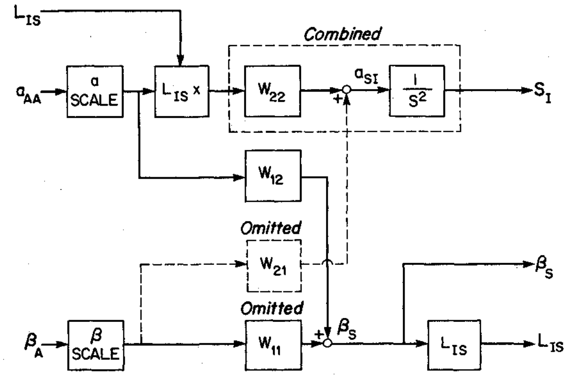

Fig. 2 Optimal controller algorithm.

corresponding simulator motion was close to unity at all frequencies. Thus, an uncoordinated aircraft motion such as a sustained nose-up attitude will manifest itself in the simulator as a corresponding nose-up attitude through the tilt algorithm.

4) During aircraft yaw and coordinated rotational motions (such as a coordinated turn), the orientation of the specific force vector relative to the pilot's head does not change. Thus, during these maneuvers, the cross-feed channels remain inactive and only high-frequency rotational motions are simulated.

5) Tilt-rate limiting is a good example of the nonlinearities that can be introduced to cope with special situations. Since the angular velocity associated with tilt-coordination is an artifact generated to trick the pilot's senses, it is important that the pilot not sense its presence. Thus, an effort is made to limit the tilt rate to a level below that which the pilot is likely to sense.

6) When we took the worst-case aircraft motion to be a steady translational acceleration or rotational velocity, third-order translational and second-order rotational high-pass filters were shown to be sufficient to return the motion base to its neutral position.[13] If the effects of scaling and limiting are neglected and small angles are assumed, the longitudinal and lateral high-pass filters need only be of second order to achieve this effect.[14] Furthermore, since transport aircraft motions are rarely this severe, lower-order filters are usually sufficient. Table 1 shows the order of the filters used in the CW2 classical filter case.[16]

Table 2 shows the number of differential equations that must be solved numerically in real time to implement this technique. This number bears a close relation to the CPU time necessary to complete one iteration through this algorithm, as shown in Table 2. Furthermore, the number of independent free parameters to be adjusted in this algorithm is shown in Table 2, as is the authors' evaluation of its relative "transparency." This latter criterion is an indication of how easily the designer can predict changes in the simulator motion that would result from a change in one of the free parameters. The primary advantages of the classical algorithm can be summarized as follows:

1) It is mathematically and computationally simple, and hence computationally cheap.

2) It is relatively transparent to the designer, and therefore pilot complaints can often be easily rectified.

The principal disadvantage of this scheme is that it uses linear elements (except for the input limiting and tilt-rate limiting blocks) and so does not fully exploit the simulator capabilities or take into account the nonlinear characteristics of human motion perception. Because of their fixed parameters, classical filters must be designed for worst-case maneuvers and often yield minimal motion under more gentle maneuvering. This disadvantage can be partially offset by including logic to change the filter characteristics under certain conditions or to remedy specific complaints.

Optimal Control

The optimal control algorithm used in the present work is an extension of the time-invariant solution to the optimal control problem.[10,13-15] It is characterized by a systematic combination of linear filters that is determined by a separate off-line design algorithm. The design algorithm solves the linear quadratic optimization problem that results in the optimal form, order, and characteristics for the filters given the underlying assumptions. A flowchart of the optimal control algorithm is shown in Fig. 2, and the weights used in the best version studied (denoted as OC2[16]) are given in the first part of Table 3. The following points should be noted in Fig. 2:

1) The inputs to the algorithm are the aircraft body-axis cockpit accelerations and Euler angles.

2) The aircraft translational accelerations are passed through W_{22} to yield simulator translational accelerations. W_{22} always takes on the form of a high-pass filter.[13,14] As well, the longitudinal and lateral aircraft accelerations are fed through W_{12} to yield a component of the simulator Euler angles. In general, W_{12} has the form of a low-pass filter[13,14] so that this channel fulfills a similar function to tilt-coordination in the classical algorithm.

3) The aircraft Euler angles are passed through W_{11} to produce the simulator Euler angles. W_{11} tends to have a unity transfer function in pitch and roll[13,14] and acts as a high-pass

filter in yaw. It is interesting to note that if the effects of scaling and limiting are neglected, this results in a one-to-one correspondence between aircraft and simulator pitch and roll motion in uncoordinated maneuvers. Furthermore, only the high-frequency component of coordinated pitch or roll maneuvers will be simulated as in the classical algorithm, but through a different mechanism. For example, a coordinated turn will produce a lateral acceleration in the aircraft that will feed through the low-pass W_{12} filter to eventually cancel the angular motion feeding through W_{11}.

4) The equations coded in the design program are such that the filters produced are numerous and of high order. However, examination of these filters almost always reveals that extensive simplifications can be made. Thus, the cross-feed filters from angular to translational channels (W_{21}) have extremely small gains over the whole frequency range and so can be omitted from the reduced version. As well, filters W_{11} in the pitch and roll channels are found to have unity transfer functions over the frequency range of interest and so are not included. The order of the remaining filters can usually be reduced by cancelling some roots from the numerators and denominators of their transfer functions. The second part of Table 3 shows the order of the reduced filters for the OC2 case.[16]

Table 2 shows the number of differential equations that must be solved numerically to implement this technique, the execution time for one iteration, the number of free parameters, and the transparency. The advantages of this technique can be summarized as follows:

1) The design algorithm includes a vestibular model that enables it to minimize the pilot's motion sensation error between aircraft and simulator, rather than minimizing the motion error itself. Thus, this method inherently exploits the features of the modeled human motion-sensing apparatus.

2) The algorithm produces an optimal filter, assuming the underlying assumptions are reasonable.

3) Adjustment of the algorithm is performed by changing weights on physically meaningful quantities, rather than by changing filter coefficients. This would seem to make the algorithm more transparent by having the design algorithm optimize coefficients while the designer only deals with more understandable quantities.

The primary disadvantages of this technique are the following:

1) Certain assumptions made in the derivation of the design algorithm are not applicable to a realistic simulator environment. Among these are the stochastic aircraft motion and the decoupled degree-of-freedom assumptions.

2) In practice, adjustment was rarely as straightforward as anticipated and often required modification of parameters that seemed unrelated to the problem at hand. Thus, this scheme was rated as being relatively opaque.

3) Tilt-rate limiting was not included in this algorithm because it had negative effects on its behavior.[14] Because of this and the fact that the high-frequency gain of the W_{12} filters was appreciable, significant amounts of high-frequency translational motion were fed through to the angular channels. This resulted in simulator motion that was judged much too "responsive" by the pilots.

4) As with the classical scheme, the optimal control scheme yields fixed-parameter filters that do not exploit the motion capabilities of the motion base and must be adjusted for worst-case maneuvers.

Coordinated Adaptive

Adaptive algorithms have received much attention in the past decade because of their flexibility and "intelligent" behavior. Although various schemes have been proposed, the present one is based on that developed at NASA Langley.[4,5] Like classical washout, it is made up of an empirically determined combination of filters. However, unlike the classical scheme, some of the filter coefficients are systematically varied in real time to minimize a cost function. Figure 3 is a flowchart of the coordinated adaptive algorithm, and Table 4 gives the values of the coefficients used in the version that obtained the most favorable pilot ratings (denoted as AW2[16]). It should be noted that this version of the algorithm received the best overall rating of all the algorithms tested.[16] The following points are to be noted in Fig. 3:

1) The inputs to the algorithm are the aircraft cockpit body-axis specific forces and angular rates, as in the classical algorithm.

2) The aircraft specific forces are passed through high-pass filters of adaptive gain to yield the simulator translational accelerations. The longitudinal and lateral specific forces are also adaptively scaled and fed to the pitch and roll channels, respectively. Once again, this achieves the same effect as the tilt-coordination channel of the classical algorithm.

3) The aircraft angular motion is adaptively scaled and added to the cross-feed components to yield the simulator angular motion. A frequency-response analysis of these filters was performed.[14] The effects of scaling, limiting, and adaptive gains were neglected and small angles were assumed. As with the classical and optimal cases, this algorithm was found to have an overall unity transfer function between aircraft and simulator angular motion during uncoordinated pitch or roll maneuvers.

Table 4 AW2 filter characteristics[a]

	$i = x$	$i = y$	$i = \psi$	$i = z$
γ	1.0	1.0	—	—
ρ_i	1.0	1.0	1.0	1.0
W_{i1}	10.0	1500	0.0	2.5
W_{i2}	10.0	15.0	50.0	15.0
W_{i3}	2.5	7.5	1.0	1.5
W_{i4}	1.0	1.0	—	—
W_{i5}	500	500	—	—
W_{i6}	0.40	1.5	—	—
W_{i7}	17.26	17.26	—	—
W_{i8}	46.45	20.0	—	—
k_{i1}	0.36	1.0	0.0	0.625
k_{i2}	2.4	4.0	0.1	7.0
k_{i3}	—	—	—	7.6
G_{i1}	3.5	7.5	20.0	1.0
G_{i2}	0.0058	0.0005	—	—
G_{i3}	0.10	0.15	—	—
p_{i10}	1.0	1.0	1.0	1.0
p_{i20}	0.12	−0.032	—	—
p_{i30}	1.0	1.0	—	—
p_{i1lim}	−0.1	−0.1	—	−0.1

[a]Symbols as defined in Ref. 13.
Scale factor on all inputs: 0.5
f_{AA} input limit (all components): 10 m/s²
ω_{AA} input limit (all components): 34.4 deg/s
Tilt-rate limits—(pitch) : 3 deg/s
(roll) : 2.3 deg/s

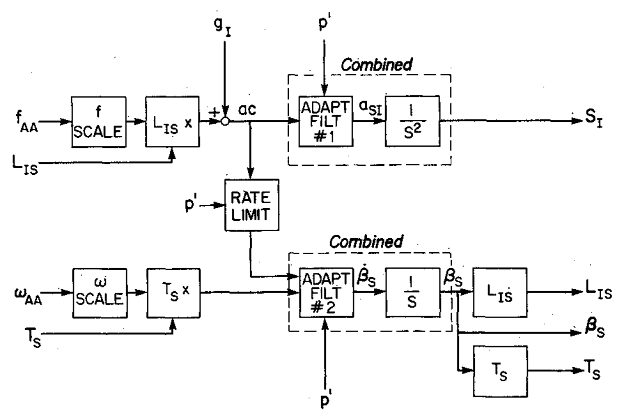

Fig. 3 Coordinated adaptive algorithm.

4) The pitch and roll channels do not contain explicit high-pass filters in this algorithm, although their overall behavior makes it appear as if they do. This is because, unlike the classical algorithm, the cross-feed channels are active even during coordinated rotational motions. Consider, for example, a coordinated turn where $f_{AA} = [0\ 0\ \ -kg]^T$. The L_{IS} transformation will cause a nonzero cross-feed term aC^y to appear. This term will feed into the roll channel and act to return the simulator roll angle to zero. In the frequency-response analysis,[14] an explicit equation is derived for this implicit high-pass filter behavior.

5) Tilt-rate limiting is included in the algorithm to try to keep the cross-feed rates below values that the pilot is likely to sense. Interestingly, this rate limiting also acts to enhance the low-pass filter behavior of the cross-feed channels and the high-pass filter behavior of the pitch and roll channels in coordinated maneuvers.

Table 2 shows the number of differential equations to be solved numerically in the implementation of this technique, the CPU time per iteration, the number of free parameters, and the transparency. The potential advantages of this technique can be summarized as follows:

1) The adaptive characteristics serve to give more realistic motion cues when the simulator is near its neutral position and only reduce the motion fidelity when the simulator approaches its limits. In this way, better use is made of the motion system's capabilities.

2) The cost function to be minimized is extremely flexible. Although the one used in the present study was relatively simple, vestibular models could be included,[6] and nonquadratic functions could be introduced to vary penalties in more imaginative ways. It could be argued that the algorithm can be imbued with unlimited intelligence through its cost function, thereby responding in the best possible way throughout the flight envelope.

The primary disadvantages of the coordinated adaptive scheme are the following:

1) Its behavior is difficult to adjust and so the algorithm is judged relatively opaque. The structure of the cross-feed channels is the biggest contributor to these difficulties.

2) Execution time is relatively high due to the large number of differential equations to be solved and will increase as more sophisticated cost functions are introduced.

It should be noted, as well, that the potential exists to make parameters other than the filter gain adaptive. For example, adaptive filter break frequencies have been used,[7] and other possibilities exist and should be investigated.

Algorithm Comparison

Before comparing the algorithms, it is useful to summarize the features that we consider desirable in a motion-drive algorithm.

1) It should have the capability to achieve good pilot ratings and the potential to achieve excellent ones. The latter would involve the following: a) the ability to exploit the nonlinearities of human motion sensation. b) The ability to severely limit motion cueing only when the motion base nears its limits. c) The ability to be adjustable for different pilots. This feature would be valuable if the intent is to provide the pilot with the motion cues that he or she feels to be the most realistic, rather than ones that are closest to those in the aircraft. d) The ability to be adjustable for different flight segments. For example, since motion requirements tend to be different on the ground and in the air, the algorithm should change its characteristics accordingly.

2) It should be easy to adjust, which would involve the following qualities: a) There should be a minimum of free parameters; b) the effect of a change in a parameter should be easy to determine a priori; c) parameters should be applied to physically meaningful quantities; and d) in the ultimate case, it should be adjustable with no requirement of expertise on the adjuster's part to know how the algorithm works.

3) Fewer differential equations and high execution speed.

The foregoing requirements often conflict with each other and compromises must be made.

The results of the pilot evaluations[16] indicate that the pilot rating of an algorithm depends greatly on how well its parameters have been adjusted. From these results and our experience in adjusting the algorithms, we believe that given enough effort, all of the algorithms we studied have the capability to achieve good ratings. However, the potential to behave nonlinearly in order to exploit the motion base's capabilities and human motion-sensation characteristics would be more easily included in an adaptive algorithm since the other algorithms are made up mostly of linear elements. Requirements 1c and 1d could be achieved with any algorithm, although additional "supervisory" software would be required to orchestrate the parameter changes.

Features 2 and 3 are more clearly a property of the algorithm used, and in this respect, Table 2 indicates how each algorithm rates in terms of complexity, speed, and transparency. In these areas, the classical algorithm is definitely the most attractive since it contains the fewest parameters and their effect is relatively clear. Although the optimal control algorithm appears to fulfill requirement 2c, it does not clearly do so. Thus, in terms of convenience to the designer, the classical algorithm would be rated highest, whereas the optimal control and coordinated adaptive algorithms would rate equally and somewhat lower.

An important point to note in all three algorithms, but particularly in the optimal control and coordinated adaptive ones, is that the cross-feed channels and the placement of the L_{IS} transformation are responsible for a great deal of realism in the motion simulation as well as complexity. Thus, it has been pointed out that these features operate quite differently and at different locations in each of the three algorithms. It also appears that the architecture of the classical algorithm in this respect is the clearest, without being less effective than that of the other algorithms.

In the ideal case of feature 2d, an expert system could be implemented in which the evaluation pilot or instructor could enter his or her comments and the algorithm would adjust its parameters accordingly. Once again, this would entail additional higher-level software to calculate and implement these adjustments.

Numerical Implementation

The motion-base drive techniques investigated in the present work must be implemented numerically in the simulator environment. Certain additional problems and considerations are introduced in this process and the numerical implementation should be discussed as it affects the accuracy of the results and the speed with which they are obtained. The present motion-base drive techniques were coded in FORTRAN on the Perkin Elmer 3250 32-bit minicomputer that drives the UTIAS flight research simulator. The 20 Hz simulation update rate results in a time frame of 50 ms available for all necessary computations, of which the washout algorithm is only a small part.

The most important numerical consideration in the washout algorithm is the numerical integration of the differential equations involved. Various techniques for doing this were evaluated[13] with the following conclusions:

1) The Tustin method was best for linear differential equations of order three or less and was therefore used in the classical algorithm.

2) The improved Euler method was best suited for higher-order equations and was chosen for the optimal control algorithm.

3) The second-order Adams-Bashforth method should have been used for the low-order nonlinear differential equations in the coordinated adaptive algorithm. However, the 40-50% increase in CPU time over the Euler method was considered prohibitive in view of the large number of differential equa-

tions that needed to be solved. Therefore, the Euler method was chosen for the coordinated adaptive algorithm.

Use of the coordinated adaptive algorithm results in appreciably larger time lags between control inputs and motion-base response when step control inputs are used.[16] It was found that this is because of the use of an explicit integration method (Euler) to integrate the coordinated adaptive washout differential equations. It is therefore suggested that future implementations use an implicit integration method such as second-order Runge-Kutta to eliminate this problem.

Other steps were taken to speed the execution times of the algorithms. Certain of these would be performed by a good optimizing compiler, but were coded explicitly to guarantee transportability. These were as follows:

1) The number of operations was minimized by combining certain blocks in the block diagrams. For example, the HP FILT and double integration were combined into a single set of equations.[14] This was done in all three algorithms where a high-pass filter was followed by an integrator, thus resulting in more efficient and more accurate results. It was found that when filter equations and integrators were coded separately, the numerical errors introduced slow drifts in the simulator position.

2) Minimal use was made of subroutines and in-line expansion was performed. When subroutines were used, most data were transformed through COMMON blocks to avoid the overhead associated with transferring subroutine argument memory locations. This resulted in faster, albeit less readable code.

3) Fast operations were used in preference to slower ones. Integer exponentials were expanded and multiplications by constants were used rather than divisions by constants.

4) Expressions were factored when possible.

5) Intermediate variables were stored to avoid computing expressions more than once.

6) Trigonometric functions were approximated by the first three terms of their series expansions when it was known that they would not exceed a limited range.

Alternative Algorithms

As noted in the comparison section, the layout of the classical algorithm was found to be the most advantageous, whereas the adaptive algorithm appears to be more amenable to exploiting the motion-base capabilities. Our work is now directed toward a "hybrid" algorithm with the same layout as Fig. 1, but with some or all of the filter blocks replaced by adaptive filters. A similar approach has been proposed[17] with encouraging results. It is envisaged that this algorithm will combine the best features of the classical and adaptive schemes. When all the steepest descent slopes are set to zero, it becomes identical to the classical algorithm and is therefore easy to adjust to obtain good performance. With further effort, the adaptiveness of each block can be introduced separately and gradually to improve performance.

There is considerable challenge in deciding which of the filter blocks should be made adaptive. Since adding adaptiveness increases the complexity of adjustment, the objective is to build an algorithm that has a few adaptive filters in the right places.

The adaptiveness can also be made more intelligent by using more sophisticated nonlinear, nonquadratic cost functions. The form of the equations are such that any differentiable cost function can be used. We are investigating high-order displacement penalties in order to give less attenuation to low-amplitude motions and more to high-amplitude ones. If this shows enough improvement, it could do away with the need for actuator soft-limiting software.

The coordinated adaptive algorithm studied in this report featured an adaptive gain that varied in real time. It may be beneficial to replace this or augment it with adaptive break frequencies[7] and/or damping ratios.

Conclusions

Based on the pilot evaluations and our experience in adjusting these algorithms, it is concluded that all three algorithms studied could be adjusted to give good performance in terms of pilot ratings. This is so even though in the tests performed, the optimal control algorithm fared the worst, whereas a coordinated adaptive algorithm gave the best results. In the authors' opinion, the coordinated adaptive algorithm shows the most potential for further improvement because of its nonlinear nature that allows greater flexibility. The classical algorithm was found to be best in terms of simplicity, ease of adjustment, and execution speed and is recommended when quick results are needed. Finally, although the optimal control algorithm could eventually be adjusted to give good results, it does not seem to warrant the effort required. That is, if a linear filter is desired, the classical algorithm is a better compromise.

The flowchart structure of the classical algorithm is the most desirable of the three studied. That is, a structure in which aircraft cockpit specific forces and angular rates are used as input, and with translational and rotational channels mostly decoupled. The cross-feed channels should only be active when it is desired to simulate a translational aircraft motion with a rotational simulator motion. Tilt-rate limiting is an essential component of any motion-base drive algorithm. However, since tilt-coordination angular velocity is an artifact, it must be kept at levels that the pilot is not likely to sense. The order of the filters in the classical algorithm shown in Table 1 are the lowest that should be used for transport aircraft. Lower-order filters would introduce unacceptable offsets in simulator position during certain maneuvers. All numerical integration of the washout differential equations should be performed with implicit methods in order to avoid lags in motion-base response to step-control inputs.

Certain problems are inherent to flight simulator motion systems and cannot be resolved with better software. The inability to generate sustained changes in vertical loading and the difficulty in representing any large-amplitude medium-frequency motion are examples of these. Although better software may help in these situations, it cannot replace the benefits of increased physical motion capability.

Acknowledgments

The work reported in this paper was made possible by a three-year contract held with the Canadian Transportation Development Centre, Contract OSD83-00101, Project 6069. The development of the simulator used in this study was funded by the Canadian Natural Sciences and Engineering Research Council and the Ontario Government.

References

[1]Schmidt, S. F., and Conrad, B., "Motion Drive Signals for Piloted Flight Simulators," NASA CR-1601, May 1970.

[2]Baarspul, M., "The Generation of Motion Cues on a Six-Degrees-of-Freedom Motion System," Dept. of Aerospace Engineering, Delft Univ. of Technology, Delft, the Netherlands, Rept. LR-248, June 1977.

[3]Parrish, R. V., Dieudonne, J. E., and Martin, D. J., Jr., "Motion Software for a Synergistic Six-Degree-of-Freedom Motion Base," NASA TN D-7350, Dec. 1973.

[4]Parrish, R. V., Dieudonne, J. E., Bowles, R. L., and Martin, D. J., Jr., "Coordinated Adaptive Washout for Motion Simulators," *Journal of Aircraft*, Vol. 12, No. 1, 1975, pp. 44–50.

[5]Parrish, R. V. and Martin, D. J. Jr., "Comparison of a Linear and Nonlinear Washout for Motion Simulators Utilizing Objective and Subjective Data from CTOL Transport Landing Approaches," NASA TN D-8157, June 1976.

[6]Ariel, D. and Sivan, R., "False Cue Reduction in Moving Flight Simulators," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-14, No. 4, 1984, pp. 665–671.

[7]Jex, H. R., Magdaleno, R. E., and Jewell, W. F., "Effects on Target Tracking of Motion Simulator Drive-Logic Filters," Air Force Aerospace Medical Research Lab., Wright-Patterson AFB, OH, AFAMRL-TR-80-134, Aug. 1981.

[8]Sivan, R., Ish-Shalom, J., and Huang, J.-K., "An Optimal Con-

trol Approach to the Design of Moving Flight Simulators," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-12, No. 6, 1982, pp. 818–827.

[9]Sullivan, R. B., "The Use of Vestibular Models in Flight Simulator Motion Washout Systems: An Experimental Evaluation," M. S. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, May 1985.

[10]Ish-Shalom, J., "The Design of Optimal Control Motion for Flight Simulators," Ph.D. Dissertation, M.I.T. Center for Space Research, Cambridge, MA, Dec. 1982.

[11]Friedland, B., Ling, C.-K., and Hutton, M.F., "Quasi-Optimum Design of a Six Degree of Freedom Moving Base Simulator Control System," NASA CR-2312, Oct. 1973.

[12]Hofman, L. G., and Riedel, S. A., "Manned Engineering Flight Simulation Validation Part I: Simulation Requirements and Simulator Motion System Performance,"Air Force Flight Dynamics Lab, Wright-Patterson AFB, OH, AFFDL-TR-78-192, Feb. 1979.

[13]Reid, L. D., and Nahon, M. A., "Flight Simulator Motion-Base Drive Algorithms: Part 1—Developing and Testing the Equations," Univ. of Toronto, Ontario, Canada, UTIAS Rept. 296, Dec. 1985.

[14]Reid, L. D., and Nahon, M. A., "Flight Simulator Motion-Base Drive Algorithms: Part 2—Selecting the System Parameters," Univ. of Toronto, Ontario, Canada, UTIAS Rept. 307, May 1986.

[15]Reid, L. D., and Nahon, M. A., "Flight Simulator Motion-Base Drive Algorithms: Part 3—Pilot Evaluations," Univ. of Toronto, Ontario, Canada, UTIAS Rept. 319, Dec. 1986.

[16]Reid, L. D. and Nahon, M. A., "The Response of Airline Pilots to Variations in Flight Simulator Motion Algorithms," *Proceedings of the AIAA Flight Simulation Technologies Conference*, AIAA, New York, Aug. 1987, pp. 77–85.

[17]Dorbolo, G. and Van Sliedregt, J. M., "Improvements in Motion Drive Algorithms," *Proceedings of the 1987 Summer Computer Simulation Conference,* Society for Computer Simulation, San Diego, CA, July 1987, pp. 721–723.