JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 3, March 2006

Implementation of a Free-Vortex Wake Model in Real-Time Simulation of Rotorcraft

Joseph F. Horn* , Derek O. Bridges†
The Pennsylvania State University, University Park, PA 16802

Daniel A. Wachspress‡ , Sarma L. Rani§
Continuum Dynamics, Inc., Ewing, NJ

Free-vortex wake models are capable of providing an accurate and physically detailed representation of the main rotor wake for flight dynamics simulation. Recent advances in computing power and efficient algorithms have made it feasible to use free wakes for real-time simulation. The CHARM free-vortex wake model was integrated with the GENHEL flight dynamics simulation of the UH-60A helicopter. A high fidelity wake model was defined by increasing the spatial and temporal resolution of the wake until a converged response was observed, but this baseline model could not execute in real-time. A parametric study was performed to find the best combination of wake parameters to achieve real-time execution with minimal deviation from the baseline model in terms of the frequency and time responses in the pitch and roll axes. A real-time model was found and showed reasonable agreement with the baseline model as compared to a simple finite-state inflow model. A parallel implementation of the free wake model was also investigated. An increase in computational efficiency could be achieved using a distributed processing approach with asynchronous communications.

Introduction

A critical issue in the development of high-fidelity simulations of rotorcraft is the modeling of the rotor wake.[1] The rotor wake influences the local air velocities in the plane of the rotor, which in turn affect the air loads on the main rotor blades. Furthermore, the rotor wake is responsible for a number of important aerodynamic interactions with the fuselage and empennage.[2,3] A major challenge in wake modeling for flight dynamics simulation is the need for real-time performance. While off-line calculations of flight dynamics are still valuable, real-time calculations can be used in pilot-in-the-loop simulation in order to analyze handling qualities. While high-fidelity free-vortex wake models have been around for many years, their applications have mainly been restricted to off-line analysis of performance and rotor dynamics in trimmed and quasi-steady flight. However, advances in computing power

Received 17 June 2005; revision received 27 October 2005; accepted for publication 28 October 2005.

* Assistant Professor, Department of Aerospace Engineering, 233 Hammond Building; joehorn@psu.edu. Senior Member AIAA.
† Graduate Research Assistant, Department of Aerospace Engineering, 233 Hammond Building; dbridges@psu.edu. Student Member AIAA.
‡ Associate, Continuum Dynamics, Inc., 34 Lexington Avenue; dan@continuum-dynamics.com
§ Associate, Continuum Dynamics Inc., 34 Lexington Avenue; sarma@continuum-dynamics.com

and efficient algorithms are now making free-vortex wake models a viable option for real-time flight dynamics analysis.

It has long been recognized that quasi-static models of the rotor inflow, while sufficient for steady-state analysis or low order dynamic models, fail to capture important characteristics. For example, a quasi-static inflow model fails to capture the dynamic overshoot of thrust and torque following a collective input.[4] The development of dynamic models for the rotor wake has been a challenging problem because the air mass effectively has an infinite number of degrees of freedom. Over the years a number of inflow models have been applied that model the inflow with a small number of dynamic states.[5] Many rotorcraft simulation codes first addressed dynamic inflow effects using simple lag filters.[6] A major advance occurred with the advent of the finite-state dynamic inflow models developed by Peters et al.[7] The third order Pitt-Peters model and the higher order Peters-He model represent the rotor downwash in the plane of the rotor with a set of radial and azimuthal basis functions. The distribution of downwash is then governed by a finite set of state equations driven by the loading on the rotor blades. These models provide a reasonable prediction of induced velocities in the rotor disk for moderate amplitude maneuvers and they are readily incorporated into real-time simulation models. The number of states used in the Peters-He model is scalable in order to achieve the desired tradeoff in terms of fidelity versus real-time performance. The calculations required for a 15-state Peters-He model, running at a time step typical for a full-scale helicopter, can easily be achieved in real-time with modern computers. As a result, the Peters models have found wide use in real-time simulations of rotorcraft, and updates and improvements to these models are ongoing.[8–12] Recent advances to the Peters-He theory have added the capability to model effects such as wake curvature, wake contraction, and calculation of tangential and radial velocity components.

Although the dynamic inflow of Peters et al. has found wide usage there are known limitations. There are certain flight conditions in which the assumptions of the finite state inflow break down, for example in certain descending flight conditions or vortex ring state. Using the Peters-He model to model wake interactions with the empennage may produce inaccurate results since the model does not account for all wake distortion effects downstream of the rotor disk. Finally, the Peters-He model is inherently a small disturbance formulation, with the wake dynamics modeled as a set of first order, quasi-linear ordinary differential equations. Correlation studies have shown it to be accurate for predicting aircraft response in small to moderate amplitude maneuvers, but the inherent assumptions of the model break down for large amplitude control inputs and generalized maneuvering flight.

On the other hand, free-vortex wake models are inherently capable of modeling all important wake distortion effects and are well suited for modeling wake-empennage interactions. The basic assumptions of the free-vortex models are perfectly valid for all generalized flight maneuvers, including descending flight conditions and large amplitude maneuvers. Thus, researchers have begun to use maneuvering free wake codes to calculate more accurate predictions of vehicle dynamics[13] and external acoustics[14] in generalized maneuvering flight. Free wake models had not been considered for use in real-time simulation due to the excessive computation required. However, with advances in computing power, efficient free-vortex algorithms, and parallel processing, it may now be feasible to use free wakes in real-time simulation. This might have several advantages:

1. Increased fidelity for real-time simulation.
2. Increased use of first-principles/physics-based models in real-time flight simulation. Thus, the flight dynamics models could be used more as a predictive tool and not rely as much on empirical models.
3. The expansion of the use of real-time simulation to disciplines outside of flight dynamics and handling qualities, such as acoustics and vibration analysis. For example, recent studies have shown the potential for real-time simulation of coupled flight dynamics and acoustics.[15]

The objective of this paper is to investigate methods for incorporating a free-vortex wake model in real-time flight dynamics simulation of rotorcraft. Although advances in computing power appear to make this quite feasible, there are still limitations in the level of modeling detail that can be achieved in running real-time free wake models. The quantity of wake and the spatial and temporal resolution with which it is modeled is limited. A key objective of this study is to establish a method for determining the best combination of modeling parameters to use in a real-time implementation of a free wake. The initial part of this paper presents the development of such a method as applied to the implementation of the GENHEL flight dynamics model coupled with the CHARM free vortex wake module on a single microprocessor. The second part of the paper discusses a parallel or distributed implementation in which the flight dynamics and the free wake calculations are run on separate microprocessors.

Simulation Implementation

Penn State University has developed a low-cost, real-time rotorcraft simulation facility.[16] This facility is designed to make use of distributed computing to provide efficient real-time performance. Several graphical nodes are used to provide both out-the-window and cockpit displays, while separate computing nodes are used for the flight dynamic model, modeling of automatic flight control systems, acoustics prediction, and other calculations. The simulation uses the NASA/Army GENHEL flight dynamics model of the UH-60A helicopter.[6] Several updates and improvements have been made to the original GENHEL software. A MATLAB interface was developed in order to provide pre- and post-processing and to provide a graphical user interface (GUI) that allows the simulation operator to easily adjust aircraft properties and initial conditions. The simulation can also be launched via a system of scripts for generating large trim and time history data sets. The GENHEL software has also been modified to include the capability to model advanced control laws, generate high order linear models, and interface with acoustics codes.[15] The modified software is referred to as GENHEL-PSU.

Recently, the GENHEL-PSU software was modified to interface with the CHARM free-vortex wake module.[17] The CHARM module is a full-span free-vortex wake model designed to interface with existing blade element rotor models. It propagates the free wake using a 2^{nd} order accurate, backward difference time marching scheme. CHARM uses a Constant Vorticity Contour (CVC) wake model consisting of a set of equal and constant strength vortex filaments emanating from the full span of each rotor blade. The release points and number of vortex filaments released adjust based on the continually changing circulation distribution along the blade. In this way, the model accounts for both trailed and shed vorticity in transient flight with maximum efficiency. Vortex elements automatically concentrate in regions of the wake with the most vorticity. The near wake is typically modeled with several vortices emanating from both the primary circulation zone in the middle of the blade and a negative circulation zone near the tip (if one exists). In the far wake, the vortices are typically rolled into a single tip and single root vortex.

The CHARM wake model is ideally suited for use in flight simulation applications, both in terms of speed and adaptability to varying flight conditions. The CVC wake model directly computes the roll-up and characteristics of the concentrated tip vortices instead of relying on user-provided settings. This is critical for flight simulation applications where these properties will continually change. Curved vortex elements are used with an analytical solution for their self-induced velocity effect. This is critical for maintaining fidelity in real-time operations where long wake elements are necessary. A fast multipole method is used to accelerate the Biot-Savart integration for wake-induced velocities from $O(N^2)$ to O(N). This method automatically adapts to any transient maneuver, including an aircraft flying into its "old" wake. Furthermore, an extensive prior research project was undertaken to accelerate all areas of the CHARM solution procedure to support real-time operation with maximum fidelity and to provide efficient coupling between the wake and flight simulation solutions.[18]

The integration of the CHARM module with the GENHEL flight dynamics code was relatively straightforward and has already been achieved in previous studies.[17,19] The GENHEL model provides the circulation at each blade element to the CHARM module. CHARM in turn performs two calculations; it calculates the induced velocities at the rotor blades (and at any other desired location needed to model wake interaction effects) and it updates the structure of the wake. In our initial analysis, the CHARM free-wake model is linked directly with the GENHEL flight dynamics model on a single processor. In the latter part of this paper, a parallel implementation of the CHARM module is presented.

Parametric Study of Rotor Wake Models

A free-vortex wake model will typically have a number of parameters that can be adjusted in order to achieve a wide range of levels of fidelity. The parameters are related to the following properties of the free wake model:

1. The number of vortex filaments emanating along the span of each rotor blade.
2. The number of vortex elements along each filament modeled as the rotor wake propagates downstream. This is affected by both the length of time that a vortex filament is modeled after it emanates from the rotor blade and the resolution with which the downstream wake is modeled (vortex elements can be rolled into a fewer number of vortices).
3. The length of each vortex element, which in this case is tied to the size of the time step or temporal resolution used for updating the structure of the wake and for calculating the velocities induced by the wake.

Clearly, modeling a larger volume of wake with higher temporal and spatial resolution is expected to produce a more accurate physical model of the wake aerodynamics. Likewise, using higher spatial and temporal resolution in the flight dynamics model (in terms of the number of blade elements and simulation time step) is assumed to produce a more accurate physical model of the vehicle dynamics. Increased detail and resolution in the simulation inevitably results in more computation and slower execution speeds. A key objective of flight simulation is to achieve real-time performance such that the speed of execution is faster than the time simulated to allow realistic pilot-in-the-loop simulation. Some degree of fidelity must be sacrificed in order to achieve this. As microprocessors continue to improve, one can expect to be able to use higher fidelity models in real-time simulation, but due to the large number of parameters that can be varied, the selection of the proper free wake model for real-time execution on a given computer platform is not a straightforward task. A formal procedure for selecting the best combination of input parameters would be of value.

There are several parameters in both the GENHEL flight dynamics code and the CHARM free wake module that can be adjusted to trade fidelity for shorter execution time. In order to find the set of parameter values that achieve the best fidelity within the constraint of real-time operation, a parametric study is performed in which individual model parameters are changed, and simulation results are compared to a high fidelity baseline case. Each case is evaluated in terms of the average execution speed and relative error with respect to the baseline case.

Definition of the Baseline Case

The first step was to establish a high fidelity baseline model, which might have relatively slow execution time, but produces an aircraft response that is insensitive to further increases in resolution and model detail. This model is then considered a "truth model," or the best representation of the aircraft that we can achieve. To do this several parameters were varied until the aircraft response showed convergence. The number of blade segments, the flight dynamics and free wake model time steps, and the total amount of free wake modeled were varied until a converged time history response was observed. This analysis was performed for a 45 knot level flight condition using simple doublet inputs. Figures 1 and 2 show time history responses for several different variations in the time step and total amount of free wake modeled respectively. The solid line indicates the response chosen for the baseline case.

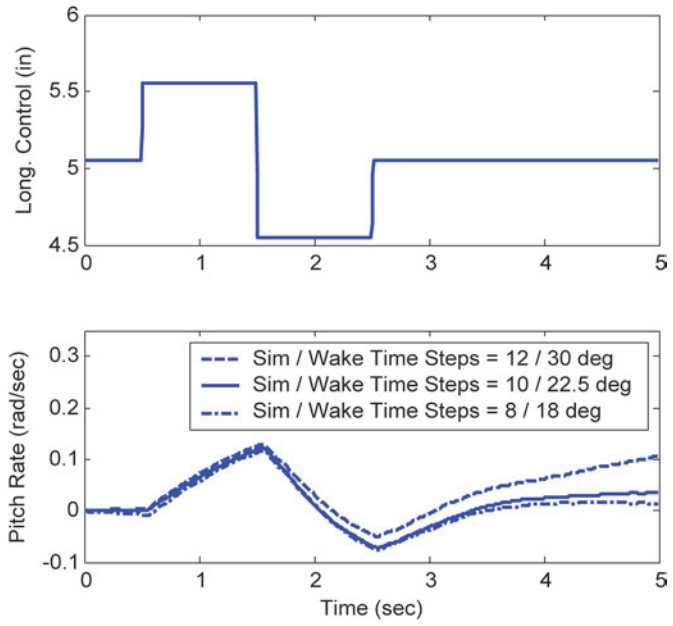

Fig. 1 Convergence Test, Longitudinal Doublet Response, 45 knots.

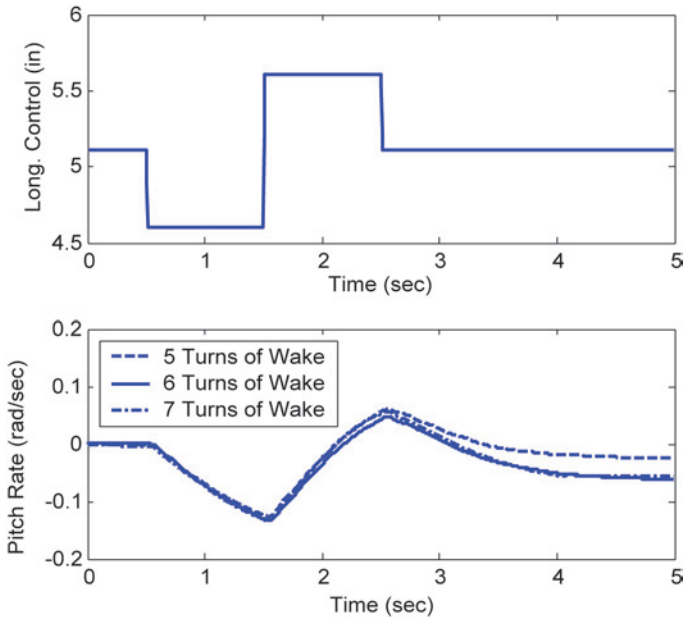

Fig. 2 Convergence Test, Longitudinal Doublet Response, 45 knots.

Parametric Variation in the Wake Model

The next step was to perform a detailed parametric study of the different input parameters for the coupled GENHEL/CHARM simulation. For the initial parametric study, individual parameters were changed one at a time to study their relative effect on execution time and on the vehicle dynamic response. The parameters adjusted were the following properties of the flight dynamics and free wake models:

1. The time step of the flight dynamics model.
2. The number of blade elements in the flight dynamics model.
3. The time step of the free wake model.
4. The maximum number of vortex filaments in the full-span wake.
5. The amount of full-span wake modeled as opposed to far-field wake (which is modeled with a single root vortex and single tip vortex). The total wake modeled in all cases is six rotor turns. In the baseline case this is split as 4 turns of full-span and 2 turns of root and tip vortices in the far wake.
6. The use of a prescribed wake as opposed to a free wake. CHARM has the added option of using a prescribed wake for all elements outside of the rotor disk and free wake for elements within the rotor disk.
7. The frequency at which the wake model is updated.

The resulting cases are summarized in Table 1. Case 1 is the high fidelity baseline case selected based on the convergence analysis discussed above. Cases 2–4 investigate the use of fewer blade elements in the GENHEL simulation code. This is strictly a modification to the GENHEL model and does not change the CHARM input parameters. In cases 5–7 the GENHEL time step is increased (lower temporal resolution). Note that the CHARM time step must also be reduced in these cases, as the CHARM module time step should be at least twice that of the flight dynamics simulation. In cases 8 and 9, only the CHARM time step is increased with the GENHEL time step held constant. In cases 10 and 11, the maximum number of vortices emanating per rotor blade is decreased. In cases 12–14, the amount of full span wake modeled is reduced. In all cases, the total number of turns of wake modeled is held at 6, but in cases 12–14 the wake is more quickly rolled into a pair of tip and root vortices. In cases 15 and 16, the different versions of prescribed wakes are investigated. Finally, in cases 17 and 18, the frequency at which induced velocities are updated is decreased. In addition to the free wake model cases, a basic Pitt-Peters three state inflow model was also evaluated for comparison to the free wake results (case 19 in the table). It should be, noted that this Pitt-Peters inflow model is a very basic version of finite-state inflow and it is not representative of

Table 1 CHARM Model Cases.

		GENHEL Parameters		CHARM Wake Model Parameters					
	Case #	Time Step ($\Delta\psi$)	# of Blade Elements	Time Step ($\Delta\psi$)[a]	Max. # of Vortex Filaments in Near Wake[b]	# of Rotor Turns of Near Wake[c]	# of Rotor Turns of Far Wake[d]	Free Wake Model[e]	Freq. of Induced Velocity Update[f]
Baseline	1	10°	30	22.5°	14	4	2	Yes	16
Blade Elements	2	10°	20	22.5°	14	4	2	Yes	16
	3	10°	10	22.5°	14	4	2	Yes	16
	4	10°	5	22.5°	14	4	2	Yes	16
FD Time Step	5	12°	30	30°	14	4	2	Yes	12
	6	12°	30	45°	14	4	2	Yes	8
	7	15°	30	45°	14	4	2	Yes	8
Wake Time	8	10°	30	30°	14	4	2	Yes	12
Step	9	10°	30	45°	14	4	2	Yes	8
# of	10	10°	30	22.5°	10	4	2	Yes	16
Filaments	11	10°	30	22.5°	6	4	2	Yes	16
Amount of	12	10°	30	22.5°	14	3	3	Yes	16
Full-Span	13	10°	30	22.5°	14	2	4	Yes	16
Wake	14	10°	30	22.5°	14	1	5	Yes	16
Convected Wake	15	10°	30	22.5°	14	4	2	Free wake elements in rotor disk	16
	16	10°	30	22.5°	14	4	2	Prescribed Wake	16
Induced Velocity	17	10°	30	22.5°	14	4	2	Yes	4
Update	18	10°	30	22.5°	14	4	2	Yes	2
Pitt-Peters Model	19	10°	30	Pitt-Peters Inflow Model					

Notes:
a. Defined using CHARM parameter NPSI, $\Delta\psi = 360°/\text{NPSI}$.
b. Defined using CHARM parameters NVORT(1) and NVORT(2).
c. Defined using CHARM parameters NPTFW(1) and NPTFW(2), # of turns = NPTFW(1)/NPSI.
d. Defined using CHARM parameter NPTFW(3), # of turns = NPTFW(3)/NPSI.
e. Defined using CHARM parameter ICNVCT.
f. Defined using CHARM parameter NUPDATE. Induced velocities updated 4/NUPDATE times per rotor revolution or, if NUPDATE = 0, NPSI per rotor revolution.

the state-of-the-art in Peters-He inflow theory. The Pitt-Peters inflow results are intended to provide a comparison to a simple low order inflow model.

Simulation Testing

Each of the nineteen cases was tested using a set of fourteen control test inputs in the lateral and longitudinal axes (listed in Table 2). The test inputs included frequency sweeps (chirps) and doublet inputs of varying magnitude. All cases were run in batch non-real-time simulations. The frequency sweep cases used a 10% sinusoidal input with frequency varying from 0.2 to 12 rad/sec over a 120 second period. Simulations were performed using chirp signals in both the lateral and longitudinal cyclic control axes. These cases are used for frequency domain comparisons against the baseline case. In order to keep the aircraft near the trim point throughout the maneuver, these cases were run with all of the UH-60 automatic flight control systems engaged. This includes the stability augmentation system (SAS) and the autopilot or flight path stabilization system (FPS). The frequency domain analysis is then performed using the mixer inputs measured as equivalent pilot stick inputs (which include the SAS inputs) which then provide an

Table 2 Summary of Test Inputs.

Input Number	Type	Axis	Initial Direction	Magnitude (%)
1	Chirp	Lateral	N/A	10
2		Long.		
3	Doublet	Lateral	Positive	5
4		Long.		
5		Lateral	Negative	
6		Long.		
7		Lateral	Positive	10
8		Long.		
9		Lateral	Negative	
10		Long.		
11		Lateral	Positive	20
12		Long.		
13		Lateral	Negative	
14		Long.		

estimate of the open-loop dynamics of the aircraft. Small amplitude noise was also injected into the other cyclic input and the directional control so that the effect of cross-correlation could later be removed in the frequency response calculations discussed below.

The doublet inputs were run with SAS and FPS off to observe the short-term response of the simulation over a five second period. These simulations are used for time domain comparisons with the baseline case. The doublets are initiated after 0.5 seconds with the controls held at their trim values, with each positive and negative input lasting 1 second, ending with another 2.5 seconds of the controls at their trim values, for a total simulation time of 5 seconds. The responses shown in Figs. 1 and 2 are representative of the doublet inputs. Doublets were performed in both the positive and negative directions (as measured by the initial direction of stick movement) for three magnitudes: 5%, 10%, and 20% of maximum stick. The purpose of using a range of magnitudes was to investigate the importance of non-linear effects in the simulation. The frequency response analysis and the small amplitude doublets are expected to capture the small amplitude linear properties of the aircraft, while the 20% doublets (with SAS off) result in large angular rates and are expected to include significant non-linear effects.

All 19 cases were simulated with each of the 14 test inputs for a hover flight condition and at forward airspeeds of 45 knots and 90 knots. This resulted in a total of 798 simulation runs. The use of scripts for generating large batches of simulation data and a file numbering scheme for keeping track of all of the simulation time history data were both critical for performing an efficient analysis.

In order to analyze the relative accuracy of each of the model configurations presented in Table 1, the aircraft response in pitch and roll for the test cases (cases 2 to 19) is compared to the response of the baseline case (case 1) for the same flight condition and control inputs. Methods were established for analyzing the error between each of the test cases and the baseline case in order to quantify the fidelity of each of the cases (this is discussed below). In addition, the execution time of all cases was recorded, and the average framerate (the average number of time step updates that can be calculated per second) was calculated in order to quantify the real-time performance of the different cases.

All cases were run a desktop PC with a Pentium 4 3.06 GHz CPU. The simulation code was compiled using COMPAQ Visual FORTRAN 6.1 with the compiler options set to optimize speed of execution. The simulation process was executed under the Windows XP operating system with "High" priority. The cases were run overnight using batch processes such that the measured execution times were not affected by other processes running on the computer.

In the case of the frequency sweep or chirp inputs, the resulting time history data are processed to get the frequency responses for the roll and pitch rates due to the lateral and longitudinal cyclic control inputs. The process uses chirp-Z transforms along with multiple window composite averaging methods (essentially the same methods developed for frequency domain identification by Tischler et al.[20]). This resulted in four different frequency responses for each case: the on-axis responses—roll rate due to lateral cyclic and pitch rate due to longitudinal cyclic, as well as the

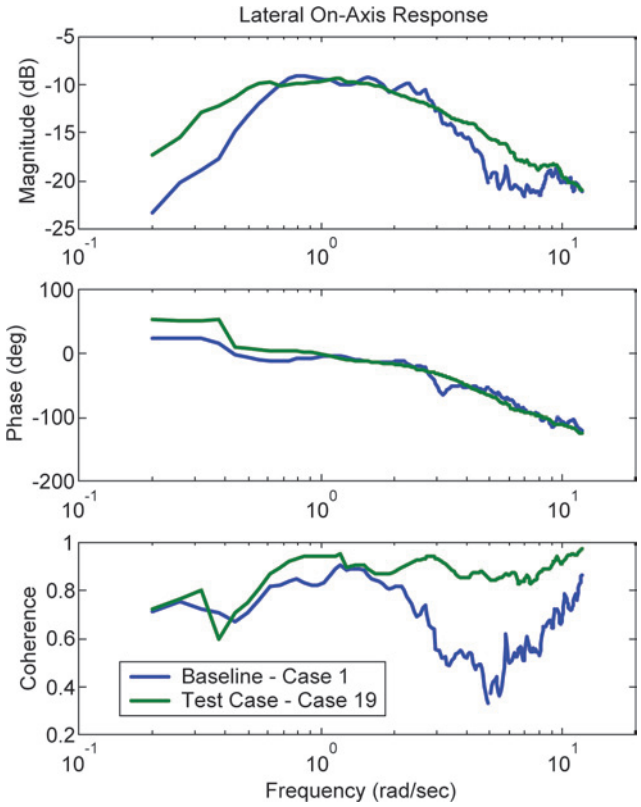

Fig. 3 Sample Frequency Responses—Lateral On-Axis Response, Hover.

off-axis responses—roll rate due to longitudinal cyclic and pitch rate due to lateral cyclic. The frequency responses include the magnitude (in dB) and phase (in deg). Coherence is also calculated and provides an estimate of the quality of the data. Figure 3 shows a sample of the lateral on-axis frequency response at 45 knots for the baseline case and case 19 (the Pitt-Peters inflow model). Note that the coherence for the free wake model is inherently lower than that achieved for the finite-state inflow model. The complex interactions between the rotor and the free wake result in unsteady air loads that produce small amplitude disturbances in the vehicle response. Thus, the free wake model results in "noisy" angular rate response, whereas the rate responses for the finite-state inflow model are very clean. The disturbances in the free wake have no apparent correlation to the control inputs, and therefore result in low coherence.

Error Analysis

The relative error of the frequency response of each of the test cases was measured using a weighted error cost function based on the formula:

$$e_{f_i} = \frac{\sum_{j=1}^{N} W_j \left[\left(M_b(\omega_j) - M_i(\omega_j) \right)^2 + \left(\Phi_b(\omega_j) - \Phi_i(\omega_j)/7.57 \right)^2 \right]}{\sum_{j=1}^{N} W_j}$$

where $W_j = 1.58\left(1 - e^{-1/2(\gamma_b(\omega_j)+\gamma_i(\omega_j))}\right)$

This cost function is similar to that used by Tischler et al for frequency domain identification of rotorcraft.[20] The symbols M, Φ, and γ represent the magnitude in dB, phase in degrees, and coherence for a given frequency response. These are evaluated at N different frequencies for both the baseline case (subscript b) and a given test case (subscript i). The error cost function was calculated for each of the four different frequency responses for a given case. A total of 200 sample frequencies between 0.2 and 12 rad/sec was used in the error analysis, as that

range of frequencies is generally considered the most important for piloted simulation. The sample frequencies were logarithmically spaced in this study so as not to put too much emphasis on the high frequency response. Note that the weighting is designed such that 7.57 deg in phase error is roughly equivalent to 1 dB in magnitude error. The error at each frequency is weighted based on the average coherence for the baseline and test case at that frequency.

To evaluate error for the doublet simulations (inputs 3–14 in Table 2), the time histories of roll rate and pitch rate for each test case were compared to those of the baseline case. An integrated squared error parameter roll and pitch response were defined to evaluate relative accuracy:

$$e_{q_i} = \int_0^5 (q_b(t) - q_i(t))^2 dt$$

$$e_{p_i} = \int_0^5 (p_b(t) - p_i(t))^2 dt$$

where the subscript b stands for the baseline case, and the subscript i indicates one of the lower fidelity test cases.

The basic test inputs and error analysis discussed above are intended to measure the linear dynamics and some basic non-linear flight characteristics of the different wake model configurations. Obviously, the true advantage of the free-wake model occurs when varying away from these types of standard maneuvers into arbitrary maneuvering flight where more complex interactions with the rotor wake cannot be captured by more primitive wake models. The free wake model also provides a physics-based tool for predicting complex interactions between the rotor wake and empennage, which were not considered in this study but have been achieved in other analyses.[17] The objective of this study was not to establish the advantage of the free-wake model in this regard. The objective was to develop methods for defining the best real-time version of the free-wake model, for which these standard maneuvers provide an appropriate test bed.

Results

The simulation model configurations were evaluated using trend charts as shown in Figs. 4 to 11. The vertical axis shows error relative to the baseline case—a measure of fidelity or accuracy of that case as defined using the equations given above. The horizontal axis shows the average framerate of the calculations relative to the framerate required for real-time operation. The quantity needs to exceed 1 for real-time simulation to be feasible. It is not expected that one of the cases listed in Table 1 will produce the best real-time version of the model. The purpose of the parametric

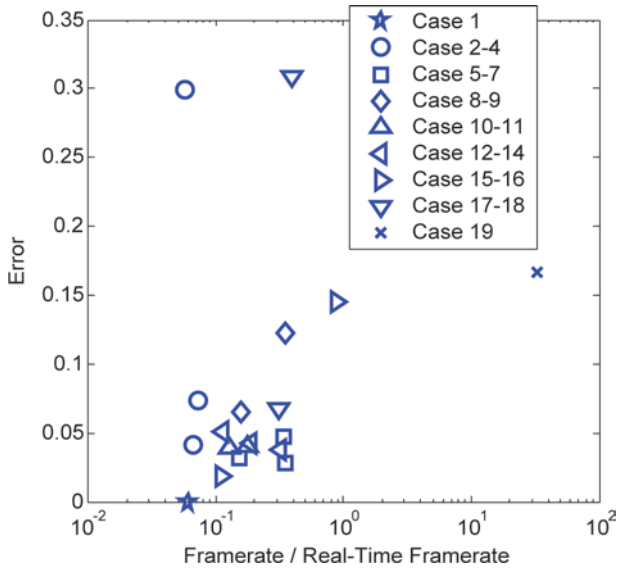

Fig. 4 Error Trend: Longitudinal On-Axis Frequency Response, Hover.

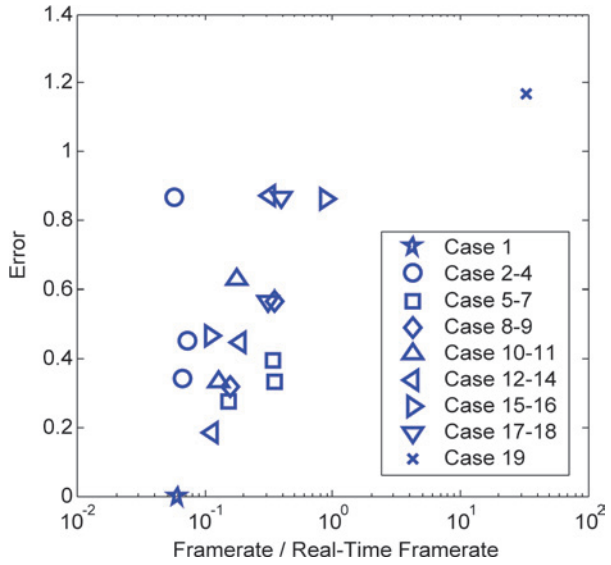

Fig. 5 Error Trend: Longitudinal Off-Axis Frequency Response, Hover.

study is to observe the impact of various parameters on the accuracy and speed of the model one at a time, and then to select the best combination of parameters that will result in an accurate real-time model. There were many different permutations of results generated in the parametric study:

- Results include frequency responses as well as small, medium, and large amplitude doublet responses.
- Error was calculated for lateral on-axis response, longitudinal on-axis response, lateral off-axis response, and longitudinal off-axis response.
- Results were generated at hover, 45 knots, and 90 knots.

This results in 48 different trend charts. Figures 4 to 11 represent a small sampling of these results.

In each plot, the star symbol represents the baseline case, which typically results in the slowest framerate, and is represented as having zero error (since error is measured relative to the baseline case). The Pitt-Peters model is

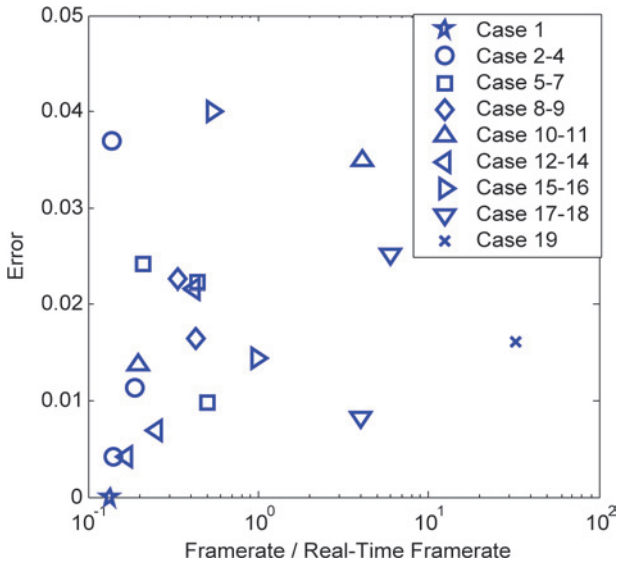

Fig. 6 Error Trend: Longitudinal On-Axis Frequency Response, 90 knots.

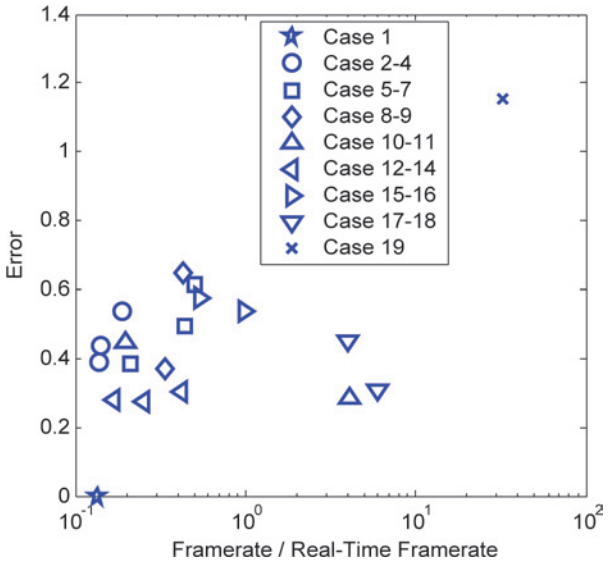

Fig. 7 Error Trend: Longitudinal Off-Axis Frequency Response, 90 knot.

indicated with an X symbol, and typically results in a very high framerate, but a relatively large error compared to the other free wake cases. Variations in the various wake model parameters are represented with like symbols (circles for different number of blade segments, squares for different time steps, etc). Normally, reduction in spatial and temporal resolution results in higher framerate and error, although there are exceptions to this trend as noted below.

Some key observations from the parametric study are listed below:

1. In all simulations, the framerate for the Pitt-Peters model is very high. On modern computers, there is no problem in achieving real-time execution of the GENHEL model with this simple inflow model.
2. The baseline model with the CHARM wake runs much slower than real-time. This was expected as the configuration was chosen for maximum fidelity and is not expected to run in real-time. The results indicate

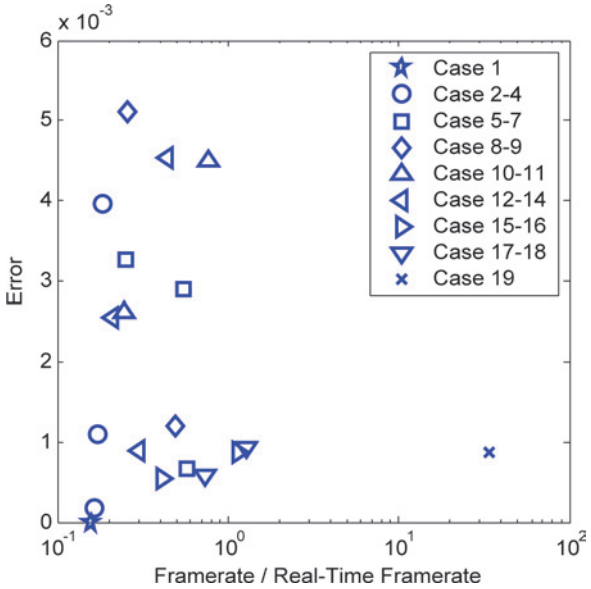

Fig. 8 Error Trend: Longitudinal On-Axis, Small Amplitude Doublet Response, 45 knots.

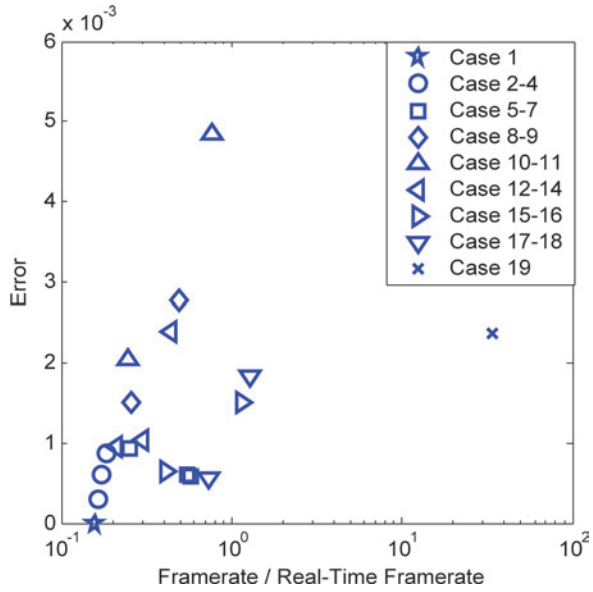

Fig. 9 Error Trend: Longitudinal Off-Axis, Small Amplitude Doublet Response, 45 knots.

that the execution was significantly slower in hover, the framerate was less than half of that achieved in simulations run at 45 knots and 90 knots.

3. As expected, the difference between the Pitt-Peters inflow and free wake model is more noticeable in the off-axis response. Figures 4 and 5 show the error in the frequency response in hover for the longitudinal on-axis and off-axis responses respectively. Figures 6 and 7 show similar results for 90 knots. Note that the vertical scale (for error) is significantly larger for the off-axis results. In the on-axis results, some of the free wake model cases show little or no improvement over the Pitt-Peters inflow. This is mainly because the Pitt-Peters inflow does not show a large difference from the baseline free wake model, at least for small amplitude motion. On the other hand, there is a distinct difference in the off-axis results. It is known

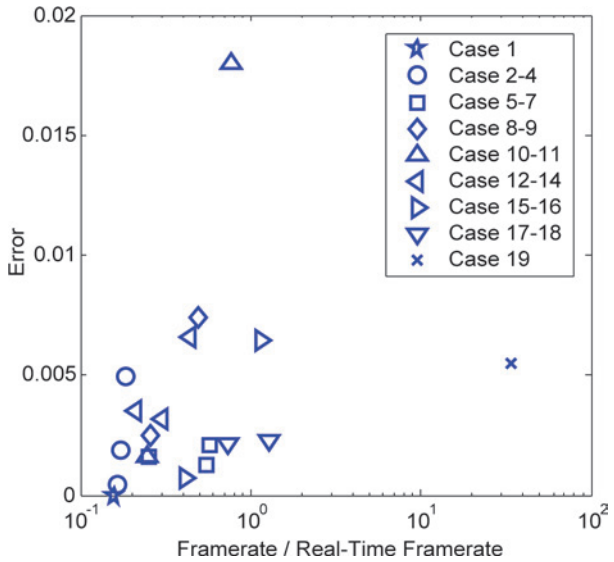

Fig. 10 Error Trend: Longitudinal On-Axis, Large Amplitude Doublet Response, 45 knots.

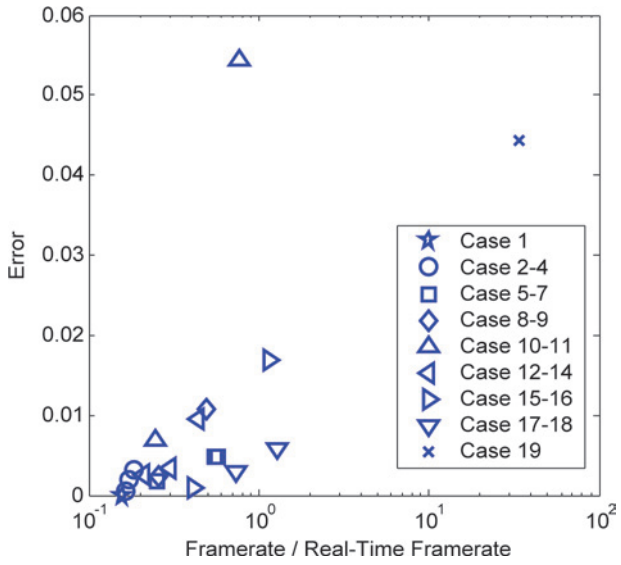

Fig. 11 Error Trend: Longitudinal Off-Axis, Large Amplitude Doublet Response, 45 knots.

that simple finite state inflow models with no wake distortion effects are inadequate for predicting off-axis response, whereas free wake models inherently capture the wake curvature effects that can have strong influence on off-axis response. More advanced Peters-He inflow models include wake curvature and other distortion effects to more accurately model off-axis behavior.

4. The difference between the Pitt-Peters inflow model and the free wake models is more significant for large amplitude inputs. Figures 8 and 9 show the error in on-axis and off-axis response for small amplitude (5%) longitudinal doublets at 45 knots. In many cases the Pitt-Peters inflow model performs as well as some of the free wake models. Figures 10 and 11 show results for large amplitude (20%) longitudinal doublets. There is more clear distinction between the behavior of the free wake models and the Pitt-Peters inflow model for the large amplitude case. This indicates that the free wake model captures some significant non-linear effects not represented by a simple finite state inflow model that is based on small disturbance theory.
5. The use of a prescribed wake (ICNVCT = 2) results in a large increase in framerate, but in many cases it results in error comparable to that of the finite-state inflow.
6. The use of a partially prescribed/partially free wake (ICNVCT = 1) results in only minor improvements in framerate for the hover flight condition. This is not surprising since most wake elements will reside within the cylinder below the rotor disk, and would be influenced by the free wake.
7. Reducing the number of filaments in the full span wake (cases 10 and 11) produced inconsistent results; in some cases it resulted in very high errors.
8. When using a free wake model, reducing the number of blade segments in the GENHEL model provides little to no benefit in terms of execution speed (GENHEL accounts for a relatively small portion of the calculation time), but it had a significant effect on accuracy. Reducing to five blade segments in some cases caused significant error *and* slowed down the calculation of the free wake. On the other hand, using only five blade segments has relatively little effect on accuracy when using a simple finite state inflow model. The CHARM free wake seems to perform better when higher resolution blade loading data is provided by the simulation model.
9. Increasing the time step of the free wake and the simulation produced significant speed up without much loss in accuracy.
10. Increasing the update rate for induced velocities to 4 per rotor revolution (NUPDATE = 1) resulted in significant speed up with tolerable reductions in accuracy. Updating induced velocities at 2 per revolution resulted in excessive error in several cases.

11. In most cases, reducing the amount of full span wake from 4 rotor turns to 3 rotor turns resulted in significant speed up with tolerable increase in error.

Real-Time Free Wake Simulation

Based on observations from the parametric study discussed above, a set of wake parameters for real-time simulation were selected and tested. The difference between the real-time free wake and the baseline free wake model are as follows:

1. The wake time step is based on a 45° azimuth sweep of the blades per time step (NPSI = 8).
2. Induced velocities are updated every 90° of blade sweep (NUPDATE = 1).
3. Use 3 turns of full-span near wake and 3 turns of root/tip vortex far wake. (NPTFW(1) = 24 and NPTFW(3) = 24).

The average framerate of the real-time free wake model was almost twice the framerate needed for real-time execution in the hover flight condition. The framerate exceeds the real-time framerate by a wider margin in forward flight. Subsequent testing of the model in the real-time flight simulator showed that the model could consistently operate in real-time, and that the performance and handling characteristics of the aircraft appeared to be reasonable. Note that the testing in the piloted simulator included additional computing overhead associated with network communications with the graphics nodes.

Figure 12 shows the framerate and error in frequency response in hover of the real-time model as compared to the baseline model and the Pitt-Peters inflow model. Figure 13 shows similar results for the large amplitude doublet response at 45 knots. Figure 12 shows that the real-time free wake model is significantly closer to the off-axis frequency response predicted by the baseline free wake model than that of the simple finite-state inflow model. There is little difference between the on-axis frequency responses for each case. Figure 13 shows that the real-time wake model compares favorably to the baseline model for large amplitude inputs in both roll and pitch. Figure 14 shows a sample frequency response (longitudinal off-axis response in hover) comparing the baseline, real-time free wake

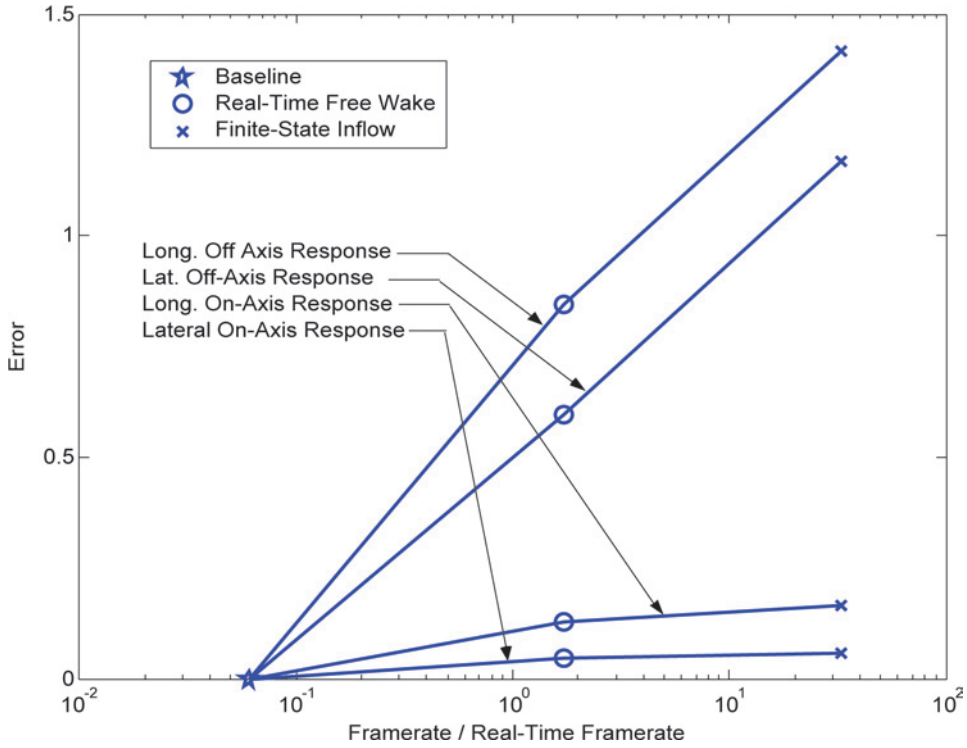

Fig. 12 Error Trend with Real-Time Model, Frequency Response, Hover.

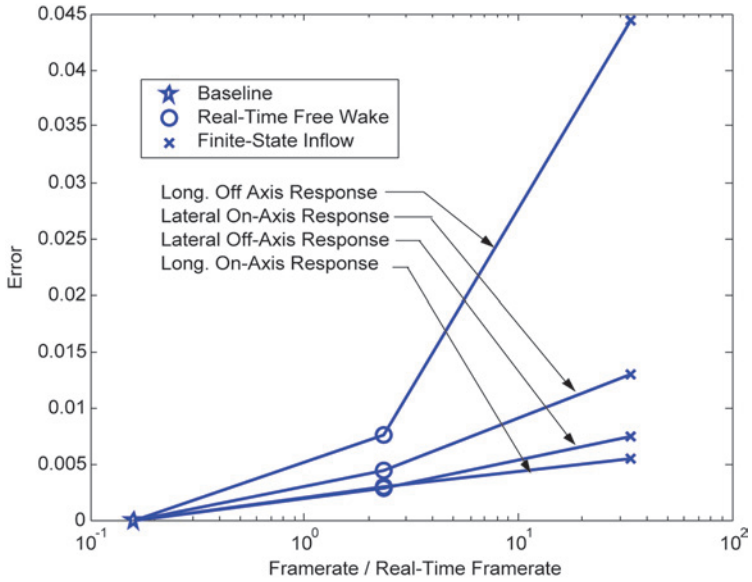

Fig. 13 Error Trend with Real-Time Model, Large Amplitude Doublet Response, 45 knots.

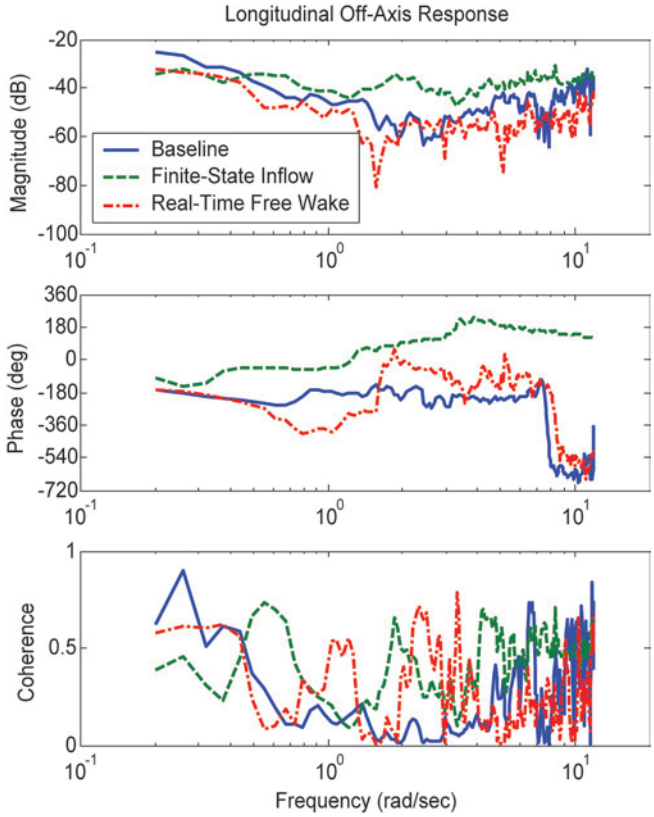

Fig. 14 Longitudinal Off-Axis Frequency Response of Real-Time Wake Model, Hover.

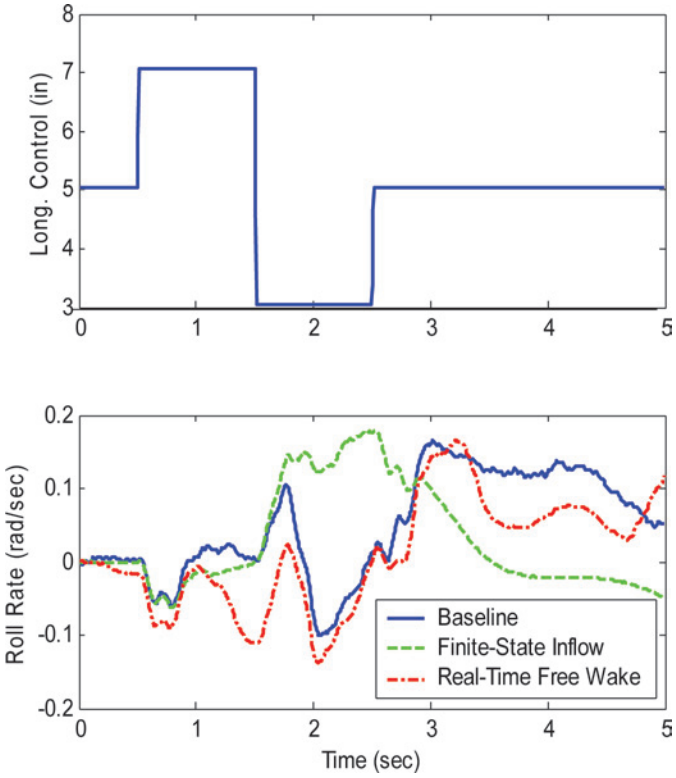

Fig. 15 Large Amplitude Doublet Response of Real-Time Wake Model, 45 knots.

model, and Pitt-Peters inflow model. Figure 15 compares the same three cases using a sample time history of the off-axis response to a longitudinal doublet at 45 knots. The frequency responses in Fig. 14 show that the coherence in the off-axis response is not particularly good, especially for the free wake model. However, the results of the frequency domain analysis were consistent for the various cases and with the doublet responses, and are therefore assumed to be valid. Overall, the real-time free wake model appears to offer a reasonable compromise in model fidelity while achieving real-time performance.

Animations of the rotor wake for the 20% longitudinal doublet at 45 knots have been created; follow these links to see the baseline wake model and the real-time wake model. The animations show the tip vortices emanating from each of the rotor blades during the 5 second doublet maneuver. The blue lines indicate the part of the wake where the full span wake is modeled (although inboard filaments are not shown), while the red lines indicate where the wake is modeled with root and tip vortices only. This is a large amplitude maneuver, as the pitch attitude of the aircraft varies by as much as 30°. The animations show that the real-time wake, although much coarser than the high-fidelity baseline wake, captures the general structure of the wake throughout the maneuver. The induced inflow ratio distribution over the rotor disk for this same maneuver was also calculated for each rotor revolution during the maneuver. Follow these links for animations that compare the baseline free wake versus the Pitt-Peters model and also the baseline free wake versus the real-time free wake model. The pause function in the media player can be used to observe side-by-side comparisons of inflow distribution for each rotor revolution. The results show that the real-time wake generates a similar inflow distribution as the high-fidelity baseline case during the course of the maneuver. Note that the three state Pitt-Peters model generates a linear distribution of inflow over the rotor disk, but there is some distortion in the animations due to blade lag angle effects.

In this study, real-time performance was achieved by running both the flight dynamics model and the free wake model on a single processor. The results showed that bulk of the required computation is in the free wake model (although this would not necessarily be the case if one were to use a more complex dynamics model with flexible

blades for example). This would seem to indicate that running the flight dynamics model on a separate processor would not provide much improvement. However, the free wake calculations run at a slower time frame than the flight dynamics model. Thus, an asynchronous parallel implementation of the model might yield significant improvements.

Parallel Implementation

In the following discussion, the term *parallel* operation is being used to describe what is typically referred to in the parallel computing world as *distributed* operation. That is, each component of the module still runs serially, but components can reside on separate processors and pass information through network communications. A key advantage, besides more processor power, is that in this mode one is assured that the CHARM module will not preclude real-time performance of the simulation. This is achieved through what is called *asynchronous* communication. With asynchronous communication, the simulation never *waits* for the free-wake module. When the simulation requests wake-induced velocities from CHARM, it does not perform the computation at that point in time, instead it provides the current best predictions available. Meanwhile, the second processor is running full-time to keep the free wake predictions as current as possible. The faster that second processor is, the more current the predictions.

It is important that the CHARM processor does not fall behind the simulation processor. Instead, if it cannot keep up during real-time operation, the framerate of the CHARM processor decreases. Thus, the flight dynamics model may be updating at 5 ms (a 200 Hz framerate), and the CHARM module may update every 20 ms (a 50 Hz framerate). In the 5 ms intervals, the CHARM module returns values to the simulation by extrapolating from previous answers, or, for the rotor wake-induced velocities, interpolating between current predictions and those obtained during previous blade passages. We describe this as *reduced frequency* updating. It should be noted that control system design requires information at a framerate of roughly 20 Hz, and that most pilots response times are at best in the 3 to 5 Hz range, so the CHARM module need only update at a framerate of 20 Hz or above for flight dynamics applications before accuracy noticeably drops.

The interaction between a host helicopter simulation and the CHARM module is shown in Fig. 16. It can be seen from this figure that in serial operation, the speed of the host analysis is limited by that of the CHARM module. The helicopter simulation passes the blade positions at time T to CHARM and then waits for it to pass back the wake/surface-induced velocities at the desired on-blade and off-blade locations. Subsequently, the CHARM module updates the wake to time $T + \Delta T$ after receiving the blade forces/circulation from the host.

Computational speed-up can sometimes be obtained by performing the two tasks, host and CHARM, in parallel. The two most commonly used paradigms of parallel programming are, Message Passing Interface (MPI) and OpenMP. The latter follows a Single Program Single Data (SPSD) approach and primarily involves parallelizing the DO loops, or *loop parallelization*. On the other hand, MPI achieves true *task parallelization* using the Single Program Multiple Data (SPMD) approach. As such, OpenMP is unsuitable for the task at hand. MPI allows individual processes to

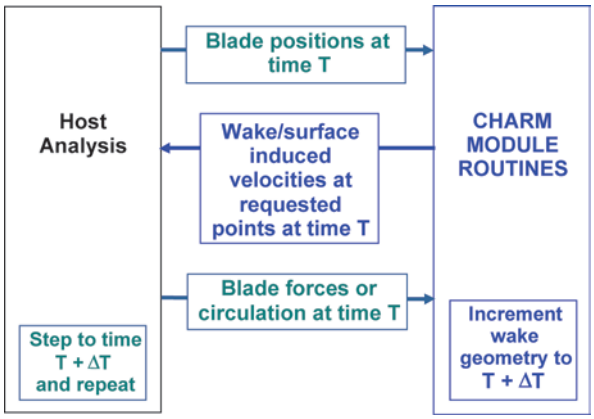

Fig. 16 Schematic of Host/CHARM Coupling Procedure.

work on different sets of data, with no implicit sharing of data amongst the processes whatsoever. Hence, the need for explicit *sends* and *receives* of data between the processors.

The difference between *loop parallelization* and *task parallelization* is illustrated in Fig. 17. The calculation is comprised of a series of individual tasks. In loop parallelization, each task is performed simultaneously on each processor. However, tasks that consist of DO loops with repetitive operations can be divided between the processors, each writing results to shared memory. In task parallelization, the paradigm is still to perform the calculation on each processor, but now individual tasks can be divided among processors. Information from tasks performed on individual processors is communicated back to the other processors through the network upon completion of the task.

In the MPI paradigm, two processors run *the same calculation* in parallel in a synchronous fashion. That is, the code "synchronizes" after tasks are performed on different processors before continuing on with the same calculation on each processor. In the asynchronous implementation performed here, the tasks of the *master* (flight simulation) and *slave* (CHARM module) are completely different.

To perform the parallelization, a simple flight simulation *template* code, CDIHEL, was used that mimics the operation and functionality of a generic flight simulation. CDIHEL has been extensively validated using a serial coupling to the CHARM module. Here, CDIHEL was restructured to support parallel, asynchronous operation. The coupled CDIHEL/CHARM analysis used MPI coding such that the host and the module run on individual processors, termed *master* and *slave* respectively. Data communication occurs between the processors whenever needed. In particular, the CHARM module routine which provides wake-induced velocities to the simulation now is called by only the *master* processor while the rest of the CHARM module routines are called by the *slave* processor. The information required by this routine is now explicitly communicated between the two processors using MPI coding.

In the new structure, while the CHARM module is evolving the wake solution on the *slave* processor, the *master* processor continues with the flight simulation extrapolating from the most recent wake-induced velocity information. When new wake-induced velocities are available, a flag is set that tells the *master* that the *slave* is ready to exchange

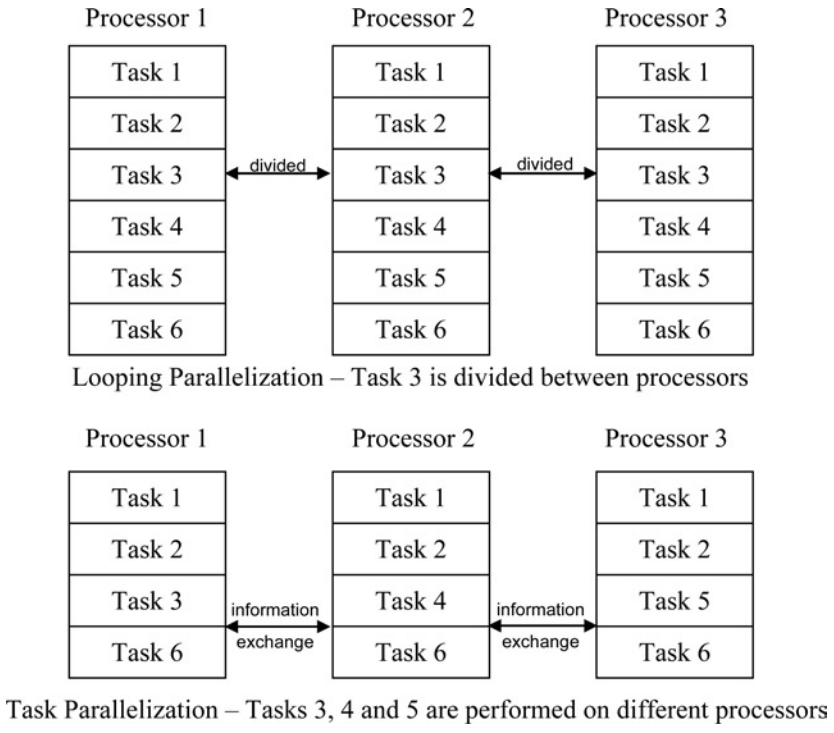

Fig. 17 Schematic of loop parallelization and task parallelization.

information. After the information exchange, the *slave* then updates the wake solution once again as the *master* continues onward with the simulation.

An important issue was to ensure that the data communication was completed properly after having posted the *sends* and *receives*. One way to do this is to use *blocking* sends and receives. This means that the master and slave will not carry on with their respective tasks until each has sent/received the data completely, even if the data was not used in the subsequent part of the program. This is safe, but does not allow asynchronous operation. For asynchronous operation, the coding must use *non-blocking* sends and receives, which require more care to maintain proper flow control of the calculation. In the work performed here, both types of sends and receives were used.

A *blocking* send and receive is used to transfer data from the *master* (flight simulation) to the *slave* (CHARM module routines). This ensures that both the *master* and the *slave* process stop computing until the communication is complete. Each time this *blocking* send occurs, it essentially synchronizes the *slave* processor to the simulation time of the *master* processor. "Simulation time" is among the data sent to the *slave*. The physical location of the aircraft and rotor blades in the flight simulation is also communicated to the *slave*.

After this *blocked* communication is done, a *non-blocking* send and receive of the information sent from the *slave* to the *master* is posted. This is the information required for the CHARM routine that provides the simulation with wake-induced velocities. Because it is non-blocking, the *master* can continue unimpeded as the *slave* updates the wake solution. In the meantime, the CHARM routine that provides wake-induced velocities to the simulation extrapolates from data obtained in the previous call. When the *slave* (CHARM) process completes the wake solution a flag is set telling the *master* that data is now ready to be received, and the non-blocking receive is executed, updating the data available to this routine.

A study was performed to demonstrate the real-time performance of the new CDIHEL/CHARM code operating in parallel. In this study, real-time operation was mimicked using a pause function in the flight simulation (*master* process) to synchronize the flight simulation time with the wall clock time. The CHARM module provided information to the flight simulation every 4 ms, (250 Hz framerate). Initial implementation revealed that real-time performance could not be attained due to large overhead associated with MPI communications. To minimize overhead, all data communicated between *master* and *slave* were placed into two arrays, one real and one integer. In this way, both the number of arrays and the number of elements being sent between the two processes was minimized. This cut down the communication time dramatically, and real-time performance at the level expected was obtained.

Real-time calculation results of the parallel MPI code are shown in Fig. 18 for a $\mu = 0.15$ case with the CHARM module providing information to the flight simulation every 4 ms (250 Hz framerate). Three settings of the CHARM module wake solution were applied:

1. Full free-wake solution, (ICNVCT = 0).
2. Prescribed wake solution wherein the wake convects at the free stream velocity plus a uniform downwash velocity based on a linear inflow model, (ICNVCT = 2).
3. Partial free-wake model consisting of a full free-wake solution for elements within the rotor disk and a prescribed wake solution for elements downstream from the rotor disk. (ICNVCT = 1).

The x-axis in Fig. 18 shows the number of curved vortex elements used in the wake solution. Nominally a value of 2000 curved elements, (equivalent to 4000 straight line elements) should be adequate for all handling qualities applications. The y-axis shows the framerate of the wake solution update achieved by the CHARM module during real-time operation. As previously described, 20 Hz should be adequate for any flight dynamics application.

With ICNVCT = 2, nearly 10,000 curved vortex elements could be processed at a 20 Hz framerate during real-time operation. With ICNVCT = 1, roughly 7500 curved vortex elements could be processed at 20 Hz, and with ICNVCT = 0, 1600 elements. Roughly speaking, at advance ratios $\mu = 0.2$ or above, ICNVCT = 2 should be more than adequate for any flight dynamics applications. At advance ratio $\mu = 0.15$ or above, ICNVCT = 1 should be more than adequate. Below advance ratio $\mu = 0.1$, the advantage of ICNVCT = 1 over ICNVCT = 0 in CPU time will most likely not be worth the drop in fidelity.

It is important to note that the "framerate" in Fig. 18 *includes* the additional communication time required between the processors. If one reduces the rate at which the simulation requests wake-induced velocities, the communication time drops and thus the process time drops as well, effectively increasing the wake solution update rate. For example, having CDIHEL request wake-induced velocity information at a framerate of 100 Hz instead of 250 Hz allowed the

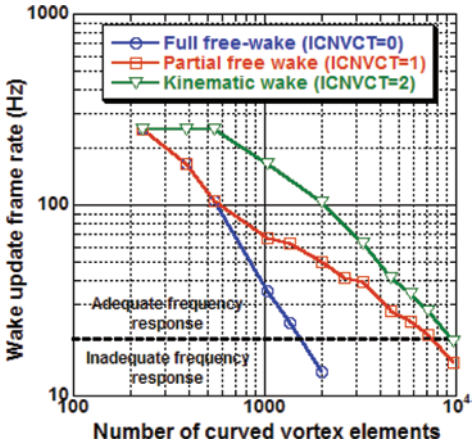

Fig. 18 CHARM module wake solution update rate in real-time parallel operation.

CHARM module to process 2000 elements instead of 1600 elements of a full free-wake solution with a wake solution update rate of 20 Hz.

It is important to note that the gains obtained with parallel operation over serial operation will strongly depend on the amount of CPU time required by the "host" or "simulation" that is coupled to the CHARM module. If the simulation code is able to operate in only a small fraction of real time, then parallel operation will not be that beneficial. If the host code requires a large fraction of real-time, then parallel operation becomes essential. The importance of what is demonstrated is that, through asynchronous parallel operation, flight simulations can now benefit from a high fidelity, full-span, free-vortex, wake solution with a guarantee of real-time performance.

Conclusions

This paper addressed the issue of implementing a free-vortex wake model in real-time flight simulation of rotorcraft. Clearly, the feasibility of using free wakes in real-time flight simulation is already a reality with modern computing power. A real-time configuration was developed in this study and tested in piloted simulation with the wake and flight dynamics model running on a single processor. Further improvements could be obtained using a parallel implementation on multiple processors, if the time required by the flight dynamics model is a significant fraction of real-time.

The number of vortex elements that can be modeled and the spatial and temporal resolution at which they are modeled will continue to grow as computing power improves. This paper presented a method for selecting free wake modeling parameters for a specific combination of flight dynamics and free wake codes (GENHEL and CHARM), although a similar approach could certainly be applied to other flight simulation codes. Some specific conclusions and recommendations that can be drawn from the results of this study are as follows:

1. Not surprisingly, the differences between the free wake models and the simple finite state inflow model were most prominent for large amplitude response and the off-axis responses. When observing small amplitude on-axis response, the difference between a high fidelity wake model and the simple finite state inflow were negligible in many cases, and thus a real-time free wake model shows no real advantage over a simple finite-state inflow. However, the use of a real-time free wake is desirable for capturing off-axis and large amplitude response.
2. The CHARM free wake module executes more slowly in the hover flight condition, making this the critical flight condition for developing a real-time model.
3. The use of a relatively coarse time increment for the free wake update (45° of blade sweep) and a 4 per rev induced velocity update seemed to produce a reasonably accurate response (as compared to the baseline case) with dramatic increase in execution speed. The use of curved vortex elements in the CHARM module would appear to be a key feature for achieving real-time performance, as it would allow larger vortex elements and a larger time step.

4. The CHARM free wake model seems to perform better when higher resolution blade loading data is provided by the flight simulation. Increasing the number of blade segments did not significantly decrease framerate since the GENHEL model accounted for a relatively small amount of the computation time. Furthermore, the increased resolution appeared to have significant impact on the response of the vehicle for the free wake model (whereas it did not impact vehicle response with the finite-state inflow model).
5. Preliminary studies indicate that that parallel implementation of the free wake and flight dynamics model on separate processors might be used to improve computational efficiency. In this study, the flight dynamics code accounted for a relatively small portion of the computational load, and thus the improvements in framerate would not be that large. On the other hand, significant improvements could be made when using a more complex simulation model (with flexible rotor blades for example). Furthermore, using asynchronous communications, a parallel implementation can guarantee real-time performance.
6. Previous studies have investigated the use of the CHARM wake model for predicting interactions between the rotor wake and the empennage.[12, 17] This was not considered in the present study (only rotor wake on rotor effects were considered and aerodynamic interference on the tail was represented with existing empirical models in GENHEL). Future studies might consider the impact of the wake empennage interactions for real-time simulation. It is critical to model a sufficient amount of wake downstream of the rotor in order to accurately capture these effects, which in turn could affect framerate.
7. Future studies might investigate the feasibility of automatically changing the wake properties as the aircraft transitions to different flight conditions. For example, in hover it may be less important to model the wake with a large number of filaments. In transition flight, spanwise resolution may not be as important as having enough rotor wake downstream to capture tail interaction effects. In high speed forward flight, it may be feasible to use a prescribed wake model. A method that could smoothly transition to different wake models as flight condition changes could be of interest.

References

[1]Curtiss, Jr., H. C., and Quackenbush, T., "Influence of the Rotor Wake on Rotorcraft Stability and Control," 14th European Rotorcraft Forum, Milan, Italy, September 1988.

[2]Prouty, R. W., "Development of the Empennage Configuration of the YAH-64 Advanced Attack Helicopter," USAAVRADCOM-TR-82-D-22, 1983.

[3]Shanthakumaran, P., Cerchie, D., Jackson, B., Thompson, T., and Dockter, G., "MD900 Handling Qualities Empennage Configuration Development," American Helicopter Society 52nd Annual Forum, Washington, DC, June 1996.

[4]Chen, R. T. N., and Hindson, W. S., "Influence of Dynamic Inflow on the Helicopter Vertical Response," *Vertica*, Vol. 11, No. 1, 1989, pp. 77–91.

[5]Chen, R. T. N., "A Survey of Nonuniform Inflow Models for Rotorcraft Flight Dynamics and Control Applications," *Vertica*, Vol. 14, No. 2, 1990, pp. 147–184.

[6]Howlett, J., "UH-60A BLACK HAWK Engineering Simulation Program: Volume I—Mathematical Model", NASA CR-177542, USAAVSCOM TR 89-A-001, September 1989.

[7]Peters, D. A., and HaQuang, N., "Dynamic Inflow for Practical Applications," *Journal of the American Helicopter Society*, Vol. 33, No. 4, October 1988.

[8]Peters, D. A., Morillo, J. A., and Nelson, A. M., "New Developments in Dynamic Wake Modeling for Dynamics Applications," American Helicopter Society 57th Annual Forum, Washington, DC, May 2001.

[9]Morillo, J. A., and Peters, D. A., "Velocity Field Above a Rotor Disk by a New Dynamic Inflow Model," *Journal of Aircraft*, Vol. 39, No. 5, Sept.–Oct. 2002, pp. 731–738.

[10]Zhao, J., Prasad, J. V. R., and Peters, D. A., "Investigation of Wake Curvature Dynamics for Helicopter Maneuvering Flight Simulation," American Helicopter Society 59th Annual Forum, Phoenix, AZ, May 2003.

[11]Zhao, Prasad, and Peters, "Validation of a Rotor Dynamic Wake Distortion Model for Helicopter Maneuvering Flight Simulation," American Helicopter Society 60th Annual Forum, Baltimore, MD, June 7–10, 2004.

[12]Zhao, Prasad, and Peters, "Effect of Rotor Wake Distortion on the Stability of Flapping Dynamics," American Helicopter Society 61st Annual Forum, Grapevine, TX, June, 2005.

[13]Theodore, C. and Celi, R., "Helicopter Flight Dynamic Simulation with Refined Aerodynamics and Flexible Blade Modeling," *AIAA Journal of Aircraft*, Vol. 39, No. 4, July–August 2002, pp. 577–586.

[14]Chen, H., Brentner, K. S., Ananthan, S., and Leishman, J. G., "A Computational Study of Helicopter Rotor Wakes and Noise Generated During Transient Maneuvers," American Helicopter Society 61st Annual Forum, Grapevine, TX, June 1–3 2005.

[15]Brentner, K. S., Lopes, L., Chen, H., and Horn, J. F., "Near Real-Time Simulation of Rotorcraft Acoustics and Flight Dynamics," American Helicopter Society 59th Annual Forum, Phoenix, AZ, May 2003.

[16]Horn, J. F., Bridges, D. O., Sharma, C., Lopes, L., and Brentner, K. S., "A Multi-Disciplinary Rotorcraft Simulation Facility Composed of Commodity Components and Open Source Software," American Helicopter Society 60th Annual Forum, Baltimore, MD, June 7–10, 2004.

[17]Kothmann, B. D., Lu., Y., DeBrun, E., and Horn, J. F., "Prospective on Rotorcraft Aerodynamic Modeling for Flight Dynamics Applications," American Helicopter Society 4th Decennial Specialist's Conference on Aeromechanics, San Francisco, CA, January 21–23, 2004.

[18]Quackenbush, T. R., Wachspress, D. A., Boschitsch, A. H., Keller, J. D., "Real Time Free Wake Methods for Rotorcraft Simulation and Design," CDI Report No. 01–11, September 2001.

[19]Spoldi, S. and Ruckel, P., "High Fidelity Helicopter Simulation Using Free Wake, Lifting Line Tail, and Blade Element Tail Rotor Models," American Helicopter Society 59th Annual Forum, Phoenix, AZ, May 2003.

[20]Tischler, M. B., and Cauffman, M. G., "Frequency-Response Methods for Rotorcraft System Identification: Flight Applications to BO-105 Coupled Rotor/Fuselage Dynamics," *Journal of the American Helicopter Society*, Vol. 37, No. 3, July 1992, pp. 3–17.