Application of Task-Pilot-Vehicle (TPV) Models in Flight Simulation

Robert K. Heffley
Owner
Robert Heffley Engineering
Cupertino, CA
rhef@aol.com

ABSTRACT

The Task-Pilot-Vehicle (TPV) model structure is able to represent the combined pilot-vehicle system in order to produce realistic performance of many real-world flight tasks and maneuvers. The scheme employs a set of graphical user interfaces for running simulations, setting pilot control parameters, and analyzing results. It includes a concise method for defining flight tasks and maneuvers and pilot control strategy and technique. The TPV model has run with several vehicle math models, including CASTLE, FlightLab©, RotorGen2©, as well as linear state-space models. The current TPV model is implemented in Simulink® and uses FlightGear open-source software to provide a visual 3D display of the simulation. The TPV model scheme is useful for rapid prototyping system design and as a simulation or flight planning tool. This paper describes recent developments in the TPV model scheme and its implementation, describes several applications, and suggests some potential uses and benefits.

INTRODUCTION

The TPV model structure is shown in Figure 1 as a feedback control system with three main component modules: Task, pilot, and vehicle.

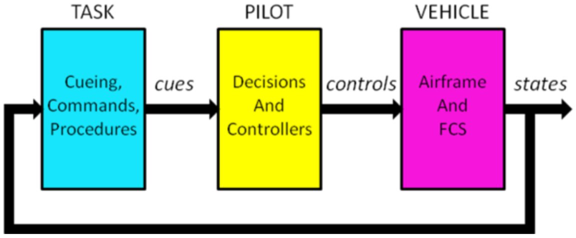


Figure 1. Basic TPV Model Structure.

This form permits a clear partitioning of each function and allows the user to choose from several selections in each module. For example, for a given pilot and vehicle, several flight tasks or maneuvers can be run in rapid succession. The resulting performance may be the basis for altering the pilot model or for adjusting the vehicle or flight control system design.

This scheme is useful in examining many manned aircraft flight tasks and maneuvers without the complications associated with using a human pilot in a manned simulator. But it is not intended as a substitute for measuring and understanding the behavior of the human pilot. Human pilot behavior is essential to devising the task and pilot functions in the TPV scheme.

We begin by discussing each of the TPV model components, including the means available for presenting plotted and visual results. This is followed by showing examples of how the TPV model has been applied and offer some potential uses. Finally we show the graphical user interfaces and analysis tools that have been developed to support the current TPV model design.

OVERVIEW OF THE TASK-PILOT-VEHICLE SYSTEM MODEL

Figure 2 shows the above TPV model structure as it is implemented in a *Simulink*® block diagram that represents the overall task-pilot-vehicle system. Each of the main blocks in the feedback loop, from left to right, are represented pictorially and include the task, the pilot, and the vehicle, respectively. Various output forms include an array of time-history plots and a *FlightGear* 3D visualization of the aircraft from a choice of viewpoints.

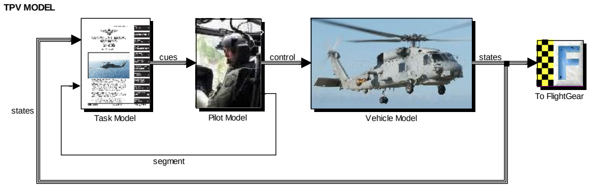


Figure 2. Simulink® TPV Modeling Environment.

[1] Presented at the American Helicopter Society 66th Annual Forum, May 11-13, 2010, Phoenix, AZ.. Copyright © 2010 by the American Helicopter Society International, Inc. All rights reserved.

The purpose of this math model form is to provide a facsimile of the human pilot performing a realistic flight task or maneuver using a given vehicle math model. The vehicle can be linear, non-linear, or even a vehicle math model under development. The task model can consist of a simple continuous tracking task or it can be a complex series of segments that mimics a realistic sequence of events or segments. The pilot is represented as having two main functions, decision-making and controlling.

The modularity of the model allows for substitution of alternative model forms so long as the basic module interfaces are consistent with the input/output relationships listed in Table 1. Note that the term "cues" includes commands per the task definition.

Table 1. I/O for TPV Modules.

Module	*Input*	*Output*
Task	states from Vehicle	cues to Pilot
Pilot	cues from Task	controls to Vehicle
Vehicle	controls from Pilot	states to Task

In general, the task model can be defined independent of the pilot and vehicle models. The pilot model is dependent on the specific vehicle but may not vary much with flight condition. The arrays of state, cue, and control variables depend upon the specific aircraft type and flight task being performed.

Notwithstanding the variety of flight tasks and vehicles that may be of interest, the TPV model form described here has been capable of simulating a wide range of task and vehicle cases. These range from helicopter maneuvers and shipboard terminal operations, to STOVL and tilt-rotor takeoff and landing, to the complex fixed-wing carrier landing task. These are all able to be defined using the concise task and pilot model functions that are part of the current TPV model scheme.

HISTORY

The TPV model concept is an extension of several "discrete-maneuver" models developed and applied to analysis of several airplane and helicopter flight tasks such as landing, deceleration, quick-stop, etc. (References 1-7). This class of pilot models differs from "tracking tasks" in not being continuous. There is a well defined start and end.

The TPV model consists of a series of discrete maneuvers or task elements that are connected by pilot decisions for when to transition from one segment to the next and by a shift in pilot control strategy or technique appropriate to a given segment.

The original TPV model was devised to represent a helicopter pilot performing some of the ADS-33E demonstration maneuvers (Reference 8) in a SBIR sponsored by the AAFD[2] at Ames Research Center (References 9 and 10). This entailed the characterization of specific tasks (e.g., ADS-33 precision hover, depart/abort, and pirouette maneuvers) using the ADS-33E task descriptions, consideration of pilot commentary, and analysis of actual manned-simulation data for validation.

The TPV scheme was further developed under a NAVAIR SBIR (Reference 11) in order to examine the ship-aircraft interface for a series of fixed- and rotary-wing aircraft and ship types, particularly with respect to the effects of ship-generated air wakes. This work included applying the TPV model to the carrier landing task, helicopter approach to, landing on, and departure from a DDG deck, and VSTOL and tilt-rotor operations from an LHA flattop deck.

TASK MODEL SCHEME

The task model receives state variable inputs from the vehicle model and outputs cues to the pilot model. The two subsystems comprising the task model in Figure 3 are the task-segment-dependent *command generator* in series with the pilot *cue generator*.

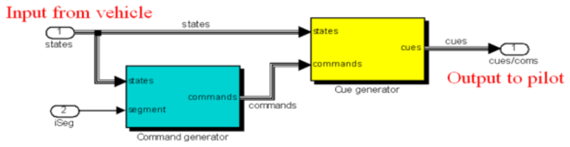


Figure 3. Task Model Components

The command generator supplies the pilot model with appropriate pilot controller commands for each of the four controller axes during each task segment.

The role of the cue generator is to transform vehicle state variables into cues that the pilot observes from cockpit instruments or senses visually, proprioceptively, or through motion. In the current

[2] U.S. Army Aeroflightdynamics Directorate.

model random noise can be added to provide a level of uncertainty or error in sensed information. This is one means of modeling a degraded visual environment.

The task model can be configured with varying levels of complexity depending upon the application. In general, the most direct formulation is to define a set of serial segments based on a standard task description such as might be found in aircrew training manuals or operations manuals. For example, the US Navy NATOPS manuals for each carrier aircraft contain detailed descriptions of the sequence of events for fixed-wing recovery (e.g., Reference 12) or for helicopter recovery and launch from the decks of air-capable ships (Reference 13). Figure 4 shows an example of task segments for an F/A-18 carrier approach starting from a racetrack pattern. About twelve segments are required to perform the approach starting at a downwind position and ending with arrestment on the deck.

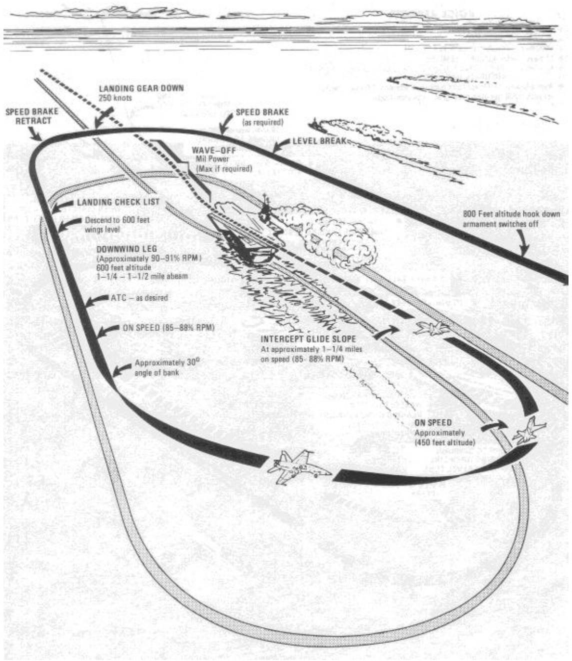


Figure 4. F/A-18 CV Recovery Task.

Figures 5 and 6 show the task layouts for the ADS-33E Precision Hover and Depart/Abort maneuvers, respectively. Using these layouts along with the detailed task descriptions in Section 3.11 of ADS-33E, both tasks are expressible directly using the serial segment method in the current TPV modeling system.

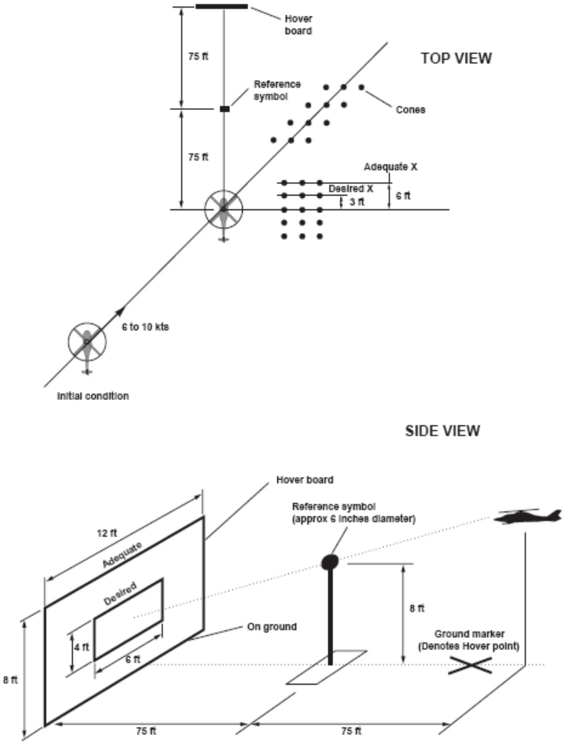


Figure 5. ADS-33E Precision Hover Maneuver.

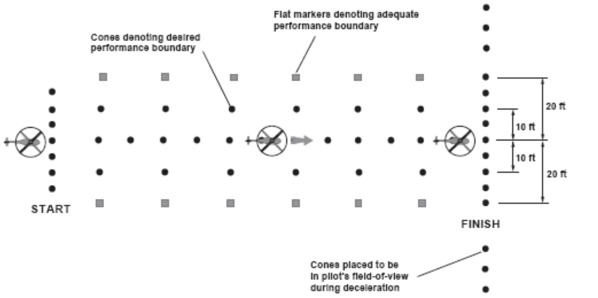


Figure 6. ADS-33E Depart/Abort Demonstration Maneuver.

The task model consists of a set of commands and cues implied by the above description for each segment.

We illustrate next a simple task model for a helicopter flight maneuver consisting of a hover with a series of position changes in each of four axes. Table 2 describes the six segments that comprise this task.

Table 2. Stationkeeping Task Over a DDG Deck—Segment Array.

segment	*description*
1	start at steady hover at 35 ft for 5 sec
2	move aft 25 ft, pause 5 sec
3	move to port 25 ft, pause 5 sec
4	move upward 15 ft, pause 5 sec
5	yaw 30 deg clockwise, pause 5 sec
6	return to original position and heading

This task is represented by a two-dimensional array consisting of six rows and ten columns. This array contains the command value and control structure index for each of the four primary control axes plus the decision variable and test value for transition to the next segment.

The task transition array for this task consists of six rows containing a column of indices corresponding to the transition variable and a column containing values for the test value. For example, in this case each of the tests is on elapsed time from the previous transition and the value for each segment is simply 5 sec. Transition tests might also be based on arriving at a given airspeed, waypoint position, altitude, or level of control activity.

The task control mode array is specified for each control axis. For the above example, each axis remains the same throughout the task and consists of x-, y-, h-, and ψ-hold controllers. The respective commands for each axis are simply the values already defined in the above segment description (Table 2).

Available controller modes for the x- and y-axes include position command, velocity command, attitude command, and cyclic control position command. For the height control axis the modes include position, vertical velocity, and collective control command. Finally, for the yaw axis the commands include heading command, heading-rate command, and pedal position command. Each is denoted by an integer index. The available modes are easily expanded by the TPV model user.

A collection of task models is assembled in a single Matlab file, ‘taskSetup.m’. Individual task models are created either for an entire set of task segments or for subtask maneuvers that may be used to compose a complete task description. The task definition is composed of a concise Matlab cell array (integers, doubles, and characters) that give for each segment the transition data, controller modes, commands, and text descriptions.

PILOT MODEL SCHEME

The pilot model consists of two subsystems, the decision-making functions and the controller functions as shown in Figure 7.

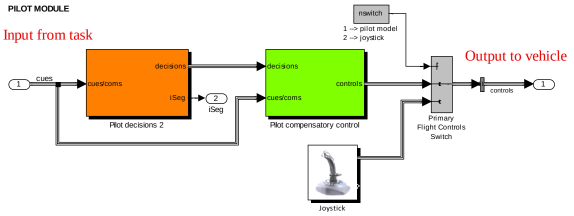


Figure 7. Pilot Model Components

Pilot decisions are based on the task-defined tests of the specified transition variable against the respective pilot cue. Upon segment transition, the task-defined control mode is adjusted and the pilot controller handles the next segment.

The pilot controller function is loosely based on the Hess structural pilot model form (Reference 14) shown here in Figure 8.

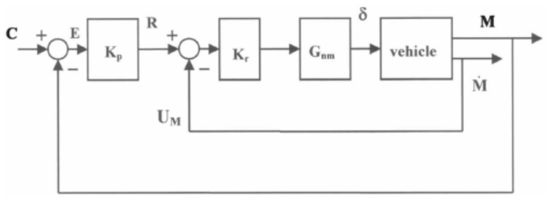


Figure 8. Hess Structural Pilot Model Form.

The actual TPV model implementation consists of a nested series of loops beginning with inner rate and attitude loops and extending to outer velocity and position loops. Figure 9 shows a typical inner loop controller.

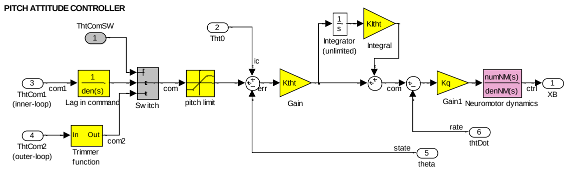


Figure 9. Typical TPV Inner-Loop Pilot Model Structure (pitch axis).

This controller structure contains a neuromuscular lag or delay function, an inner-loop on pitch rate, an integrator in the pitch command loop, a pitch loop gain (equal to the desired crossover frequency), a pitch command limiter, and a switch between either a pitch-attitude command or an outer-loop command of x-velocity or x-position. While this scheme provides broad flexibility in approximating human pilot behavior, it can normally be configured using only the pitch-rate and pitch-attitude gains and a first- or second-order neuromuscular lag.

Table 3 shows a typical set of pilot model parameters used for the longitudinal axis of control and applicable for pitch attitude command, side-velocity command, and lateral position command control modes. Similar sets would be used for longitudinal, vertical, and directional control axes.

Table 3. Typical Pilot Model Parameters for Helicopters (Longitudinal Control Axis).

Parameter	*Value*	*Description*
Kq	1.2 in/deg/s	inner-loop pitch rate gain
Ktht	2 deg/deg/s	inner-loop pitch attitude gain
ThtMax	15 deg	pitch angle command limit
KxDot	-1.2 deg/ft/s	outer-loop velocity gain
Kx	0.23 ft/s/ft	outer-loop position gain

The pilot model is affected by noise in the cueing in order to represent a degraded visual environment (DVE). The effects of pilot skill or pilot background can be adjusted in the controller structure (e.g., use of control crossfeeds, compensation, and delay). Normally the crossover frequencies of each successive outer loop are separated by a factor of 2.5 or 3.

A collection of pilot models is assembled in a single Matlab file accessed by the TPV model, ‘pilotSetup.m’. Individual pilot models are created for specific aircraft and possibly a variety of skill levels, flight tasks, visual conditions, cue availability, etc. As with the task model, pilot model parameters are defined in a Matlab cell array containing both numerical values and text descriptions.

Quantification of pilot control parameters (gains, compensation, delay, etc.) can be based on measurements of human pilot behavior or on estimates. One technique for deriving a human pilot’s crossover frequency is shown in Figure 10. This shows an intentional longitudinal position change for a skilled Navy pilot hovering above DDG deck in a manned-simulation of an SH-60 helicopter (i.e., the Stationkeeping maneuver described earlier in Table 2). The upper two plots show x-velocity and x-position over several seconds. The lowest plot shows a phase-plane trajectory for the rearward position change occurring at about 0 sec. Using the ratio of peak velocity to magnitude of position change (about 7ft/s/68ft = 0.10) and applying the factor of 2.4[3] to estimate the crossover frequency yields a value of about 0.25rad/s. This value can be used directly to set the pilot’s position gain Kx (also the value of crossover frequency for regulation of x-position).

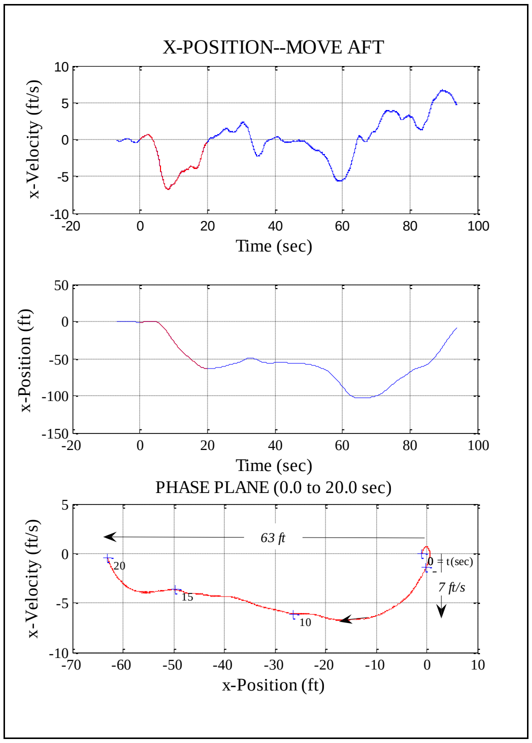


Figure 10. Summary of x-Position Response as a Human Pilot Moves Leftward in the StationKeeping Task (Reference 11).

Using a variety of techniques such as shown above, we can develop reasonable values for pilot model gains, compensation, and command limits for each of the primary control axes. Further, based on pilot commentary, it is possible to infer changes in control strategy or technique that can be implemented directly in the TPV controller model.

[3] The “2.4” factor gives an estimate of crossover frequency based on peak rate for a unit position change in an equivalent second-order system with damping ratio 0.8 (Reference 15).

VEHICLE MODEL

The vehicle model may be represented in any form that uses the primary control inputs from the pilot model and produces output states sufficient to generate the necessary cues for the task model.

Generally, it is convenient to configure the vehicle model as shown in Figure 10. This arrangement consists of flight control system, force and moment calculation, and equations of motion.

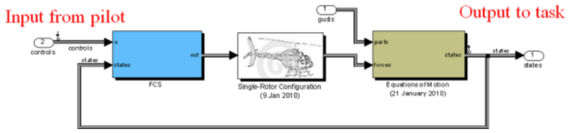


Figure 10. Vehicle Model Components

Several vehicle models have been implemented into the TPV model, including both linear and nonlinear forms. The nonlinear models have included several CASTLE aircraft models (Reference 16), the ART FlightLab© model (Reference 17, and RotorGen2© (Reference 18).

CASTLE models were run by a Simulink TPV model using a Simulink S-function, “simcas,” described in Reference 19. This requires UNIX and Windows connectivity supplied by Hummingbird software. In the case of FlightLab©, Advanced Rotorcraft Technology, Inc. created a similar S-function to link FlightLab© software with Simulink while both ran on a Linux platform. RotorGen2© runs directly on Microsoft Windows and is configured as a set of S-functions that provide either a single-rotor or tandem-rotor configuration.

Currently the TPV model is part of an ongoing internal development effort to produce a multi-purpose simulation tool. The specific flight tasks modeled for the US Army and Navy have been incorporated into a general task model framework. This permits rapid task model synthesis having an unlimited series of segments, control in all axes, and segment switching based on any set of available cues. The pilot model controllers are now based on the Hess structural pilot model than accommodates nearly any form of vehicle dynamics, including unaugmented rotary-wing dynamics. Finally, the vehicle module can accommodate any linear or non-linear model form, and has been run with CASTLE, FlightLab©, and RotorGen2© math model examples.

The TPV model software runs in Simulink and enjoys the advantages of Matlab and its numerous arrays of toolboxes and block sets. The auxiliary functions include:

- A fast, robust trim function (separate from Simulink)
- Linearization tools (separate from Simulink)
- Frequency- and time-domain analysis tools
- Fast non-real-time solutions
- Cockpit and outside 3D views using FlightGear visualization
- Custom development of 3D vehicle/ship models and terrain/seascape features

3D-VISUALIZATION

The capability to observe directly flight task performance either from the cockpit or from the point of view of an outside observer is particularly useful. This capability is provided in the TPV model by FlightGear, an open-source software package (Reference 20).

The TPV model employs FlightGear using the Matlab/Simulink Aerospace Blockset (Reference 21). FlightGear portrays TPV model performance using the position and orientation variables output from the vehicle module.

Many realistic air-vehicle models are available as downloads from the FlightGear website (http://flightgear.org/Downloads/). Also, FlightGear provides a broad array of terrain and seascape models. These models can be augmented by the many other 3D models available on the internet. Some models can be used without modification. In other cases the user may wish to enhance the models in various ways, including cosmetic appearance, alteration of cockpit instruments, or details in control surface or landing gear articulation. One example, creation of a ship environment, is shown below.

The array of ships and aircraft illustrated in Figure 12 was created using a combination of FlightGear models (H-60 and CVN) and a DDG model downloaded from a no-cost source (http://www.3dmodelworks.com/). Each of these models was modified for use with the TPV model. The FlightGear H-60 was repainted gray and given transparent main and tail rotor disks. The DDG was color detailed and a ship wake added. A second DDG was placed adjacent to the FlightGear CVN model.

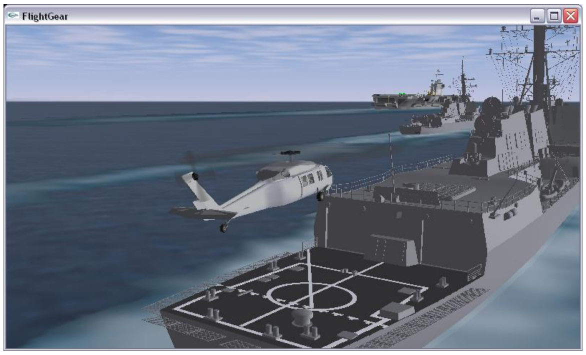


Figure 12. Navy Near-Ship Environment Using FlightGear Software—Chase View.

The runway environment used for several ADS-33E demonstration maneuvers is shown in Figure 13. Here the pilot's view from a CH-53E cockpit is shown with Moffett Field's Hangar One on the left.

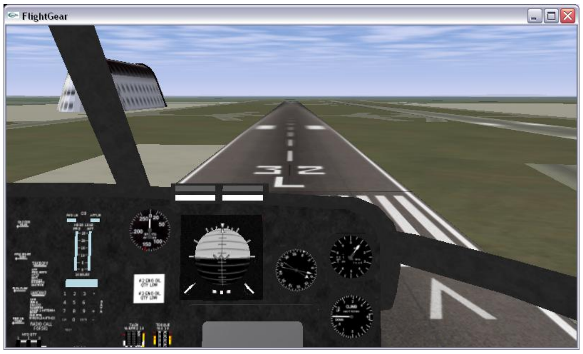


Figure 13. Moffett Field Runway 32L Environment Using FlightGear Software—Cockpit View.

Similar views are obtainable using the NAVAIR CasView software available for use with CASTLE math models. Figure 14 shows an F/A-18 on final approach to a CVN.

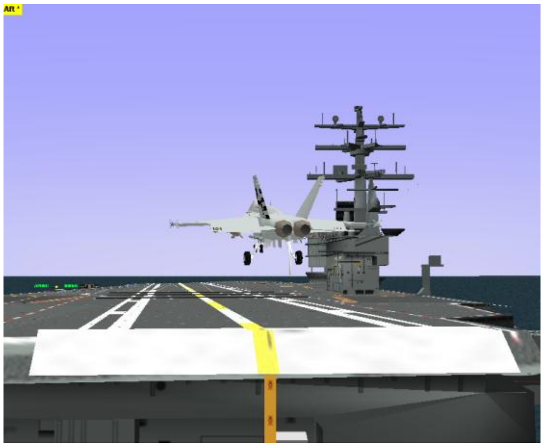


Figure 14. CASTLE F/A-18 At the Ramp Using CasView 3D Software.

TPV MODEL APPLICATIONS

The TPV model is capable of performing a variety of roles involving simulating and viewing realistic human pilot task performance. Four applications are cited here, and the last will be described in further detail shortly.

The original TPV development effort reported in Reference 9 was intended to produce a tool for examining ADS-33 maneuvers that may be performed in either a simulator or flight experiment. It enabled task and vehicle parameter variations in the context of a pilot-controlled system after creating a pilot model based on measured human pilot behavior. Only simple linear vehicle model was used.

A subsequent development program (Reference 11) sponsored by NAVAIR applied the TPV model concept to providing a tool for observing piloted flight in the presence of ship-generated air wake. Here several disparate types of aircraft were used among the several nonlinear NAVAIR CASTLE math models. These included F/A-18 fighter/attack, SH-60 helicopter, AV-8 STOVL, and MV-22 tilt rotor.

Next, a TPV model capability was added to FlightLab© nonlinear helicopter math models of CH-53 and SH-60 for examining handling qualities in the context of operations in a near-ship environment. In this case the Matlab Simulink TPV model was run on a LINUX platform that drove FlightLab© from a Simulink S-function in the TPV vehicle module.

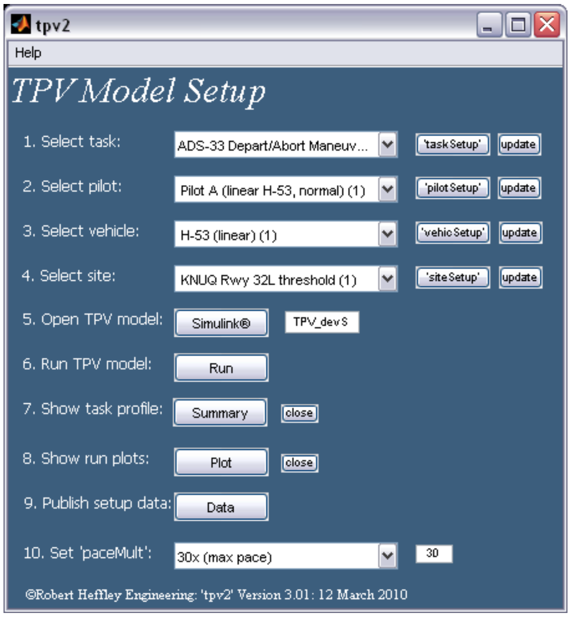
A current application of the TPV model is to aid in the development of RotorGen2© helicopter math models. RotorGen2© is a minimal-complexity nonlinear helicopter math model form that has been used for both manned and unmanned simulation, for vehicle design, and for handling qualities evaluations.

In connection with RotorGen2©, the TPV model is applied interactively to determining the vehicle model parameters. For a set of candidate model parameters the TPV software generates trims, performance characteristics (torque required, airspeed and rate of climb limits, control margins, etc.), stability and control derivatives, and frequency and time response. Then the TPV model flies the candidate vehicle model through a set of specified maneuvers in order to assess required pilot gains and compensation. All these steps are completed within a few minutes. Depending upon the results of these steps, the vehicle model parameters are readjusted and the process repeated until an acceptable model is obtained.

GRAPHICAL USER INTERFACES

Several graphical user interfaces (GUIs) support use of the TPV simulation math model.

Figure 15 shows the GUI that enables selection of the task, pilot, and vehicle from a pre-determined collection of functions. Following selection of conditions, the user can run a task execution. In addition, the user can open the Simulink model for inspection or modification, open task, pilot, or vehicle setup files, plot results, publish a run summary, and set the Simulink model pace.



tpv2

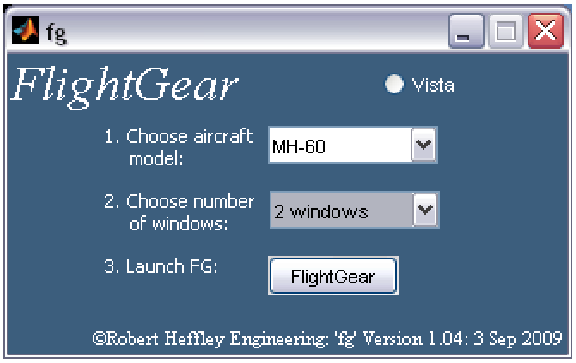
TPV Model Setup

1. Select task: ADS-33 Depart/Abort Maneuv...
2. Select pilot: Pilot A (linear H-53, normal) (1)
3. Select vehicle: H-53 (linear) (1)
4. Select site: KNUQ Rwy 32L threshold (1)
5. Open TPV model: Simulink® TPV_devS
6. Run TPV model: Run
7. Show task profile: Summary
8. Show run plots: Plot
9. Publish setup data: Data
10. Set 'paceMult': 30x (max pace) 30

©Robert Heffley Engineering: 'tpv2' Version 3.01: 12 March 2010

Figure 15. GUI for Selection of TPV Model Conditions.

Figure 16 loads a FlightGear model to display the actions of the TPV model. The user has a wide selection of aircraft, including R44, MD500, Bo105, AH-1, UH-1, MH-60, CH-47, CH-53, and several others. Also, up to three windows can be opened to permit views from the cockpit (say, forward and side views) and a view from an outside observer.



fg

FlightGear

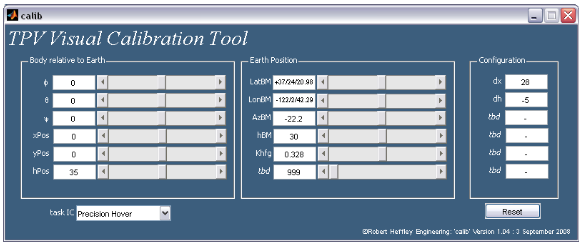
○ Vista

1. Choose aircraft model: MH-60
2. Choose number of windows: 2 windows
3. Launch FG: FlightGear

©Robert Heffley Engineering: 'fg' Version 1.04: 3 Sep 2009

Figure 16. GUI for Selection of FlightGear Model.

The use of FlightGear in conjunction with the TPV model often requires the ability to manipulate position and orientation in order to rescale models or evaluate positions on the terrain model. Figure 17 shows the utility GUI that enables direct control of position and orientation relative to any terrain benchmark (latitude, longitude, and altitude). In addition the vehicle center of rotation can be set relative to its reference datum point.



calib

TPV Visual Calibration Tool

Body relative to Earth

φ 0
θ 0
ψ 0
xPos 0
yPos 0
hPos 35

Earth Position

LatBM +37/24/20.98
LonBM -122/2/42.29
AzBM -22.2
hBM 30
Khfg 0.328
tbd 999

Configuration

dx 28
dh -5
tbd -
tbd -
tbd -
tbd -

task IC Precision Hover

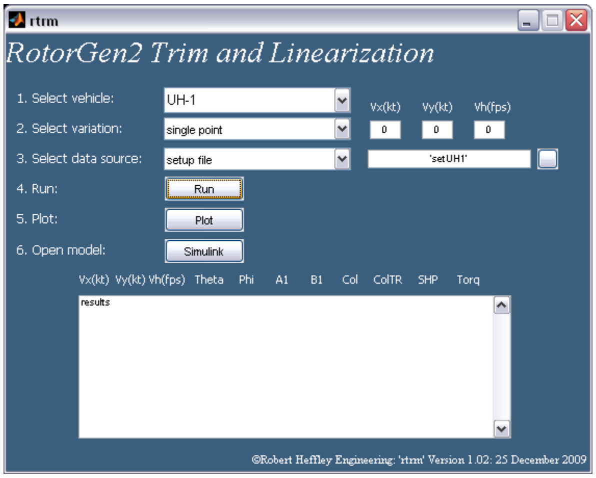
©Robert Heffley Engineering: 'calib' Version 1.04 : 3 September 2008

Figure 17. GUI for Calibration of FlightGear Model Size and Position.

ANALYSIS TOOLS

Several analysis tools have been designed into the TPV model system in order to enhance its usefulness for simulation. Three of these tools are shown below: (i) Trim and linearization, (ii) pilot model controller parameter adjustment, and (iii) task model comparison with manned-simulator or flight data.

Figure 18 shows a GUI for the *RotorGen2* vehicle model that obtains linearization solutions, computes trims, and presents results in various tabular and graphical forms.



rtrm

RotorGen2 Trim and Linearization

1. Select vehicle: UH-1
2. Select variation: single point
3. Select data source: setup file 'setUH1'
4. Run: Run
5. Plot: Plot
6. Open model: Simulink

Vx(kt) 0 Vy(kt) 0 Vh(fps) 0

Vx(kt) Vy(kt) Vh(fps) Theta Phi A1 B1 Col ColTR SHP Torq

results

©Robert Heffley Engineering: 'rtrm' Version 1.02: 25 December 2009

Figure 18. GUI for Selection of Conditions to Generate *RotorGen2* Vehicle Model Trim and Linearization.

The results of the trim and linearization routine are compared with validation data in several ways. Figure 19 shows a direct time-domain comparison with validation data for step inputs in each of the primary control axes.

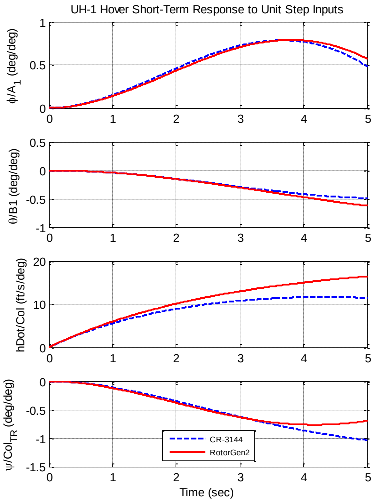
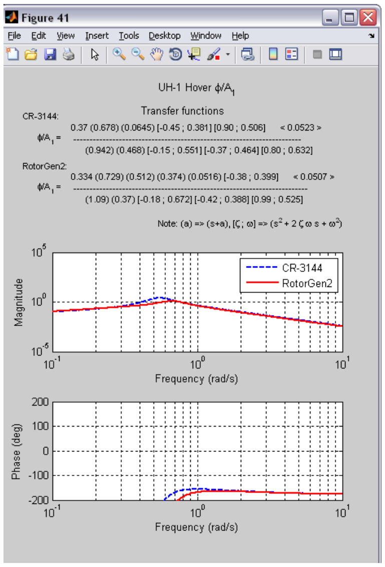


Figure 19. Trim and Linearization Time Response Comparisons.

Similarly, Figure 20 shows a direct comparison of frequency response for the roll axis. Factored transfer functions are given in the upper portion and amplitude and phase plots in the lower.



Transfer functions

CR-3144:

```
         0.37 (0.678) (0.0645) [-0.45 ; 0.381] [0.90 ; 0.506]       < 0.0523 >
φ/A1 =  ------------------------------------------------------------------------
           (0.942) (0.468) [-0.15 ; 0.551] [-0.37 ; 0.464] [0.80 ; 0.632]
```

RotorGen2:

```
         0.334 (0.729) (0.512) (0.374) (0.0516) [-0.38 ; 0.399]       < 0.0507 >
φ/A1 =  --------------------------------------------------------------------------
           (1.09) (0.37) [-0.18 ; 0.672] [-0.42 ; 0.388] [0.99 ; 0.525]
```

Note: (a) => (s+a), [ζ ; ω] => $(s^2 + 2\zeta\omega s + \omega^2)$

Figure 20. Trim and Linearization Frequency Response Comparisons.

In addition, force and moment stability derivatives are produced for the model under development and tabulated with the comparison validation data.

Next, the TPV model produces a set of trim sweeps with airspeed, vertical velocity, and side velocity. Examples of airspeed sweeps are shown in Figures 21 and 22.

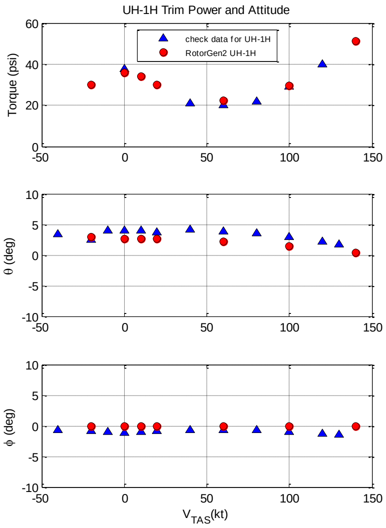


Figure 21. RotorGen2 Torque and Attitude Trim Comparisons for an Airspeed Sweep.

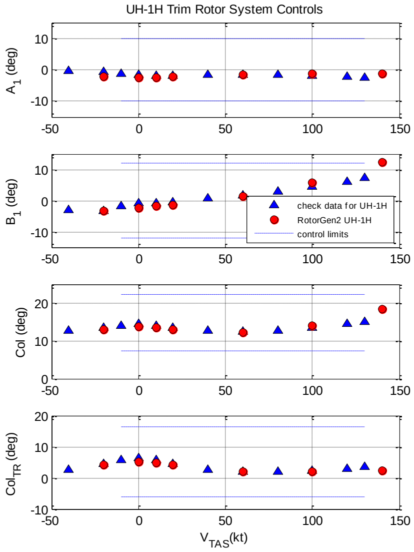
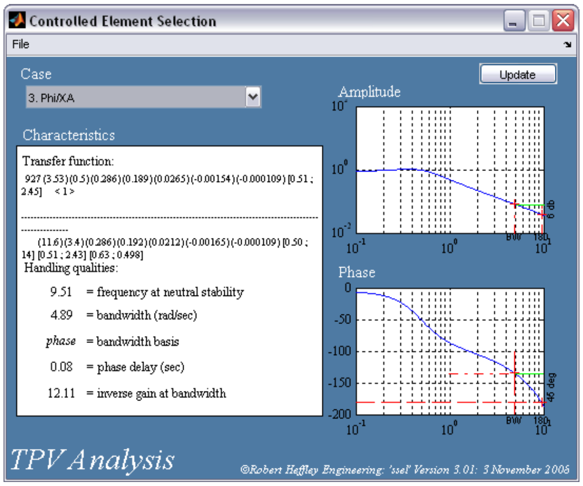


Figure 22. RotorGen2 Control Trim Comparisons for an Airspeed Sweep.

The rapid generation of these math model characteristics permits effective interactive adjustment of vehicle models, in this case, RotorGen2©.

Next, pilot model gains and compensation parameters can be set precisely using the GUIs shown in Figures 23 and 24. First, the user selects a control axis in the GUI of Figure 23 and obtains a concise view of control characteristics, including bandwidth, phase delay, and a frequency response plot.


Controlled Element Selection

Case

3. Phi/XA

Characteristics

Transfer function:

927 (3.53)(0.5)(0.286)(0.189)(0.0265)(-0.00154)(-0.000109) [0.51 ; 2.45]   < 1 >

(11.6)(3.4)(0.286)(0.192)(0.0212)(-0.00165)(-0.000109) [0.50 ; 14] [0.51 ; 2.43] [0.63 ; 0.498]

Handling qualities:

9.51 = frequency at neutral stability

4.89 = bandwidth (rad/sec)

phase = bandwidth basis

0.08 = phase delay (sec)

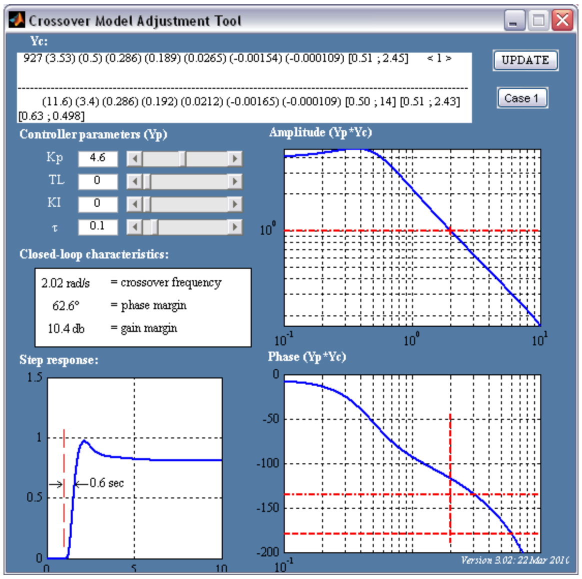
12.11 = inverse gain at bandwidth

TPV Analysis

©Robert Heffley Engineering: 'ssel' Version 3.01: 3 November 2006

Figure 23. GUI for Selection of a Primary Control Axis and Display of Characteristics.

Following selection of a control axis (roll cyclic in this case), the GUI shown in Figure 24 permits the determination of a set of pilot gain and compensation using crossover frequency and phase margin as a guide. For example, a typical value of crossover frequency for altitude-rate is about 2 rad/s. For the case shown here with a neuromuscular delay of 0.10 sec a gain, Kϕ, of 4.6 in/deg gives a 63 deg phase margin at 2.0 rad/s. This would be expected to yield good outer-loop control of lateral velocity and lateral position when used in the pilot model for this aircraft.


Crossover Model Adjustment Tool

Yc:

927 (3.53) (0.5) (0.286) (0.189) (0.0265) (-0.00154) (-0.000109) [0.51 ; 2.45]   < 1 >

(11.6) (3.4) (0.286) (0.192) (0.0212) (-0.00165) (-0.000109) [0.50 ; 14] [0.51 ; 2.43] [0.63 ; 0.498]

Controller parameters (Yp)

Kp 4.6

TL 0

KI 0

τ 0.1

Closed-loop characteristics:

2.02 rad/s = crossover frequency

62.6° = phase margin

10.4 db = gain margin

Version 3.02: 22 Mar 2010

Figure 24. GUI for Pilot Model Adjustment of Closed-Loop Control of a Primary Control Axis.

As an example of a third analysis tool, the TPV model software is designed to enable direct comparison to piloted flight or simulator data. This is useful either to aid in setting task or pilot model parameters, or to permit additional analysis of flight task performance using the TPV model as an extension to manned simulation.

Figure 25 shows an example of the TPV model results for ADS-33E Depart/Abort maneuver plotted with several manned simulator runs (three pilots). In this case it was useful to examine how the TPV model parameters varied within the range of the human pilot performance.

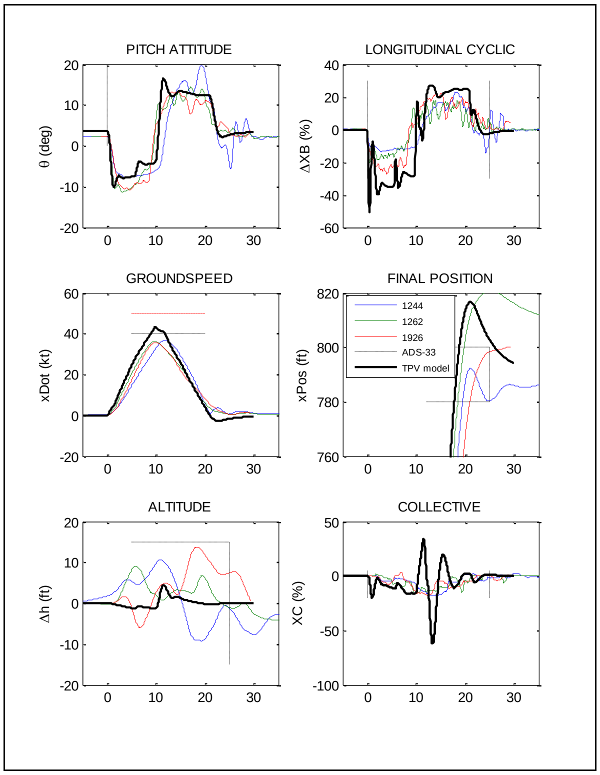


Figure 25. Initial TPV model overplotted on Several Ames VMS Depart/Abort Piloted-Simulator Runs.

CONCLUSIONS

The TPV simulation provides a full-context environment for exploring manned (or unmanned) flight tasks and maneuvers within the confines of a desktop computer. It transcends the limits of an open-loop vehicle simulation math model or of a pilot-in-the-loop simulation of a simple tracking task. The TPV model permits realistic simulation of complex multi-segment tasks.

The TPV model Simulink environment described in this paper includes 3D visualization as well as conventional time-history information normally available with Matlab and Simulink. The visual modality is a powerful asset not only for viewing simulation solutions but also for troubleshooting of system modification during development.

Advantages of the current TPV model include concise task and pilot definition, acceptance of a range of vehicle model forms, and an open-source image generation application, i.e., FlightGear. Task and pilot functions are defined by general structural forms that are set up using concise arrays. Vehicle models used to date include CASTLE, FlightLab©, and RotorGen2© plus low-order linear perturbation models. Several air-vehicle types have been demonstrated, including fixed-wing, helicopter, STOVL, and tilt-rotor

The TPV math model software is presently being used as a tool for rapid prototyping of helicopter simulator math models now under development. It enables a pilot model to begin flying the helicopter math model as it undergoes refinement of aerodynamic and flight control system models. The result of such testing by a pilot model provides an immediate indication of handling qualities, flight control deficiencies, and likely observations by a real pilot when finally run in a manned simulator environment.

REFERENCES

1. Heffley, R. K., “A Model for Manual Decelerating Approaches to Hover,” *Proceedings of the Fifteenth Annual Conference on Manual Control,* AFFDL-TR-79-3134, November 1979, pp. 545-554.

2. Heffley, R. K., Pilot Models for Discrete Maneuvers, AIAA Paper 82-1519CP, August 1982.

3. Heffley, R. K., “A Pilot-in-the-Loop Analysis of Several Kinds of Helicopter Acceleration/Deceleration Maneuvers,” *Helicopter Handling Qualities,* NASA CP 2219, April 1982.

4. Heffley, R. K., T. M. Schulman, R. J. Randle, Jr., and W. F. Clement, *An Analysis of Airline Landing Data Based on Flight and Training Simulator Measurements,* NASA CR 166404 (STI TR 1172-1R), August 1982.

5. Heffley, R. K., “Pilot Workload Factors in the Total Pilot-Vehicle-Task System,” *Proceedings of the Human Factors Society 27th Annual Meeting*, October 1983.

6. Heffley, R. K., Pilot Workload Modeling for Aircraft Flying Qualities Analysis, NADC-82094-60, May 1984.

7. Heffley, R. K., S. M. Bourne, and W. S. Hindson, “Helicopter Pilot Performance for Discrete-Maneuver Flight Tasks,” *Proceedings of the Twentieth Annual Conference on Manual Control,* NASA Conference Publication 2341, June 1984, pp. 223-231.

8. Anon.; “Aeronautical Design Standard: Handling Qualities Requirements for Military Rotorcraft,” ADS-33E-PRF, U. S. Army Aviation and Missile Command, 21 March 2000.

9. Heffley, Robert K., Ronald A. Hess, and Yasser Zeyada; “Computer Modeling and Simulation for Helicopter Task Analysis,” USAAMCOM TR 02-D-16, June 2002.

10. Hess, R. A., Zeyada, Y., and Heffley, R. K., "Modeling and Simulation for Helicopter Task Analysis,” *Journal of the American Helicopter Society,* Vol. 47, No. 4, 2002, pp. 243-252.

11. Heffley, Robert K., Simon M. Bourne, David Mitchell, and Ronald A. Hess; “Pilot Behavioral Modeling for Flight Operations Near Ships,” RHE-NAV-TR 2007-1, 1 May 2007.

12. Anon., *NATOPS Flight Manual Navy Model F/A-18A/B/C/D,* A1-F18AC-NFM-000, 1 Dec 1985, Change 3 – 15 June 1987.

13. Anon., *NATOPS Flight Manual Navy Model SH-60B Aircraft*, A1-H60BB-NFM-000, 1 May 2000.

14. Hess, R. A., and Marchesi, F., “Modeling the Human Pilot Controlling a Rotorcraft with Time-Varying Dynamics,” *Proceedings of the American Helicopter Society 65th Annual Forum*, Gaylord Texan Resort, Grapevine, Texas, May 27-29, 2009.

15. Heffley, Robert K., Simon M. Bourne, Howard C. Curtiss, Jr., William S. Hindson, and Ronald A Hess; Study of Helicopter Roll Control Effectiveness Criteria, NASA CR 177404 (USAAVSCOM TR 85-A-5), April 1986.

16. Anon., CASTLE Controls Analysis and Simulation Test Loop Environment Version 5.5, User’s Manual, SAIC Report No. 01-1393-1548-B003, August 2000.

17. The MathWorks; "Aerospace Blockset 3 Data Sheet," http://www.mathworks.com/products/aeroblks/, September 2007.

18. Anon.; "FLIGHTLAB Development Software," http://www.flightlab.com/includes/flightlabdatasheet.html .

19. Heffley, R. K., Synopsis of RotorGen2©, http://rhef.net/docs/Sim_modeling/Synopsis%20of%20RotorGenR3.pdf, 12 March 2010.

20. Olson, Curtis L.; "Introduction to FlightGear," http://flightgear.org/introduction.html, March 2010.

21. Magyar, Thomas J. and Anthony B. Page, *Integration of the CASTLE Simulation Executive with Simulink*, AIAA-2001-4121, August 2001.