ARMY RESEARCH LABORATORY

U.S. Army Research Laboratory (ARL) DESCENT Model Roadmap: Current Scope and Near-Term Extensions

by Andrew W. Drysdale

ARL-TR-6554 August 2013

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5068

ARL-TR-6554 **August 2013**

U.S. Army Research Laboratory (ARL) DESCENT Model Roadmap: Current Scope and Near-Term Extensions

Andrew W. Drysdale
Survivability/Lethality Analysis Directorate, ARL

REPORT DOCUMENTATION PAGE						*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**						
1. REPORT DATE *(DD-MM-YYYY)* August 2013		**2. REPORT TYPE** Final				**3. DATES COVERED (From - To)** October 2008–September 2013
4. TITLE AND SUBTITLE U.S. Army Research Laboratory (ARL) DESCENT Model Roadmap: Current Scope and Near-Term Extensions						**5a. CONTRACT NUMBER**
						5b. GRANT NUMBER
						5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Andrew W. Drysdale						**5d. PROJECT NUMBER**
						5e. TASK NUMBER
						5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-SLB-D Aberdeen Proving Ground, MD 21005-5068						**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-6554
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)						**10. SPONSOR/MONITOR'S ACRONYM(S)**
						11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT Recent enhancements of the DESCENT autorotation model are discussed in the context of near-term plans for the code's further development and future applications. Specific examples of near-term plans include flight simulator test data generation, occupant injury analysis, and design space (trade space) evaluation. The technical report concludes with an overview of how data is processed within the expanded model and where opportunities exist for interaction with other portions of the U.S. Army Research Laboratory (ARL) analysis process.						
15. SUBJECT TERMS helicopter, autorotation, DESCENT, simulator						
16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**		**19a. NAME OF RESPONSIBLE PERSON** Andrew W. Drysdale
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	50		**19b. TELEPHONE NUMBER** (*Include area code*) (410) 278-4762

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Purpose

In response to new emphases in survivability/vulnerability (S/V) analysis requirements and the continued need for rigorous verification and validation (V&V) of U.S. Army Research Laboratory, Survivability/Lethality Analysis Directorate (ARL/SLAD) models, several extensions to DESCENT's core processes have been completed, are being implemented, or are envisioned for the near-term. This paper will summarize the core processes as they currently exist and briefly discuss the intended scope and implementation strategy for each of three proposed extensions: simulator data collection, occupant outcome prediction, and trade-space/ design space analysis. It will conclude with a description of how the extensions are intended to work together in an integrated DESCENT application.

2. Development Schedule

Current and future development timing for proposed DESCENT extensions is displayed below in figure 1. The green squares indicate a task year. Refer to the relevant sections for detailed descriptions of the modeling extensions, specific tasks, and progress updates.

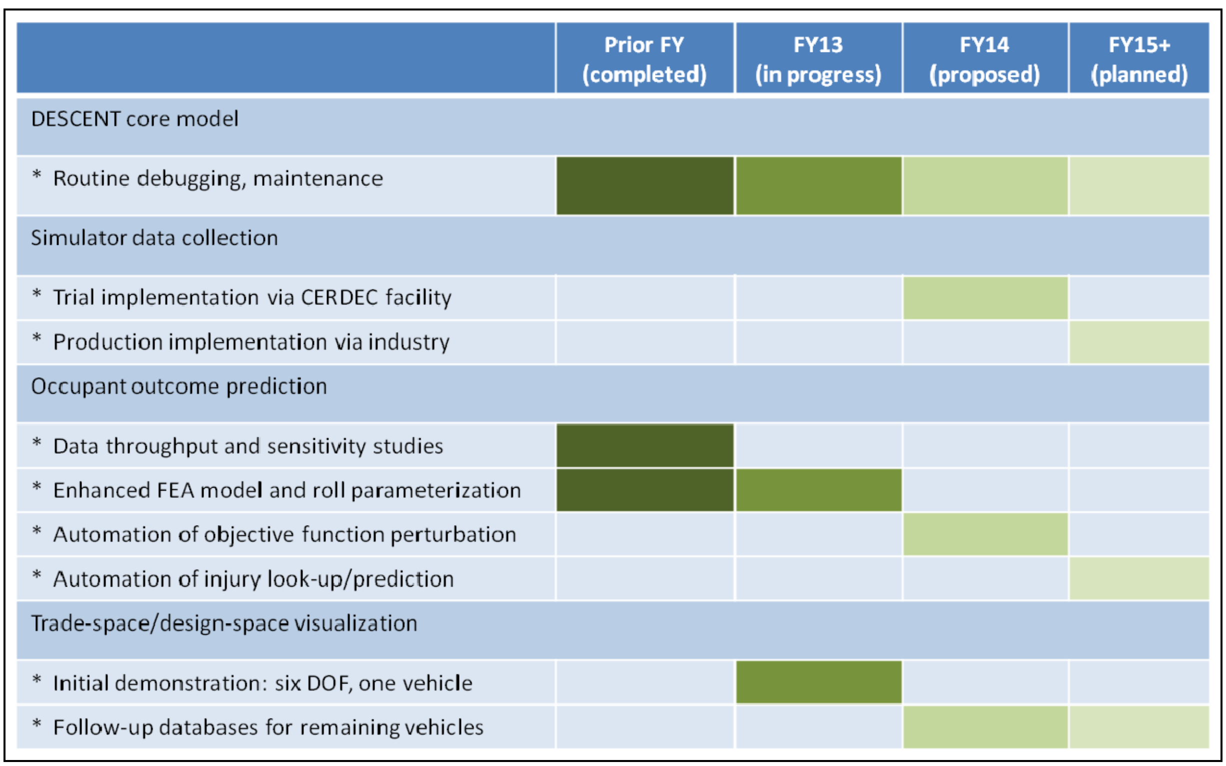

Figure 1. Scheduling of DESCENT model maintenance and extension.

3. DESCENT Background

DESCENT (not an acronym) is ARL/SLAD's model for analyzing single-main-rotor helicopter autorotation, typically in the context of a loss of engine power due to a ballistic event. This model predicts the ability of a helicopter to maintain flight after suffering a reduction in power, or predicts the impact velocity in instances when maintaining flight is not possible. DESCENT uses two-dimensional rigid-body dynamics and an actuator-disk aerodynamic model to optimize the flight path of a helicopter when given vehicle characteristics, mission, environmental, initial flight conditions, and internal parameters such as pilot response time. There are two control variables that govern the magnitude and orientation of the main rotor's lift vector; DESCENT iteratively perturbs the time-histories of these variables to optimize the control schedule and corresponding flight path. The resulting solution then represents the best-case impact conditions for the rotorcraft. "Best-case" is flexibly defined, but typically refers to a minimization of one or both components (horizontal and vertical) of the impact velocity vector.

Currently, DESCENT's most common application is within the context of an S/V analysis on a rotorcraft platform. The code is executed at regular intervals within the height above ground level (HAGL) and forward velocity flight envelope under consideration. Then the optimized outcome—either the fact that flying away under reduced power is possible, or the best-case impact conditions—at each height/velocity (H/V) case is compared to threshold criteria, and a "kill category" is assigned. The kill category represents the severity of the resulting vehicle damage. The percentage of the overall flight envelope assigned to each kill category is called the kill probability given damage ($P_{k|d}$), for that platform and for all the attendant parameter values, such as the extent of power loss. DESCENT-produced $P_{k|d}$ lists are an important input for S/V analyses that contemplate main rotor power loss, or otherwise necessitate an autorotation.

4. Current Core S/V Process

DESCENT's core process is the production of $P_{k|d}$'s, and related intermediate output data, from necessary input information. The program initializes by reading all input data from a text file, as described in section 4.1 and cited in appendix A. For each analysis case in the defined H/V envelope, it uses these data and its equations of motion to construct a "hands free" flight path that assumes no change in the control settings after the power loss event (time zero). This flight path serves as the initial estimate for the iterative optimization process that perturbs the control settings until an optimal outcome is identified (section 4.2). Once optimized flight paths have been identified throughout the H/V envelope, the solutions are postprocessed to produce $P_{k|d}$'s. The scope of available output and the method of $P_{k|d}$ production is detailed in section 4.3.

4.1 Input Requirements

DESCENT is written in Fortran 90 and compiled into a single executable. In its current form, the code requires only a single input file in the Fortran "NameList" text format. An annotated example is provided in appendix A. Although NameList provides considerable flexibility in arranging information, DESCENT input files are generally broken into the following sections:

- Domain and analysis definition. This sets the H/V flight envelope under consideration and the width of spacing in the domain grid. Some important analysis parameters are also included.
- Helicopter aerodynamic and dynamic characteristics. Size of the main rotor, inertia, lift, and stall qualities. Fuselage drag area and gross weight.
- Mission conditions. Air density, engine power available, and other parameters.
- Objective function weightings. Allows the user to emphasize certain state variables (such as vertical velocity, or rotor speed) more or less in relation to others during the autorotation maneuver and/or at landing.
- Optional parameters. Allows the user to set more detailed initial conditions or prescribe targets for certain aspects of the maneuver.

4.2 Flight Path Calculation and Optimization

DESCENT's optimization process is driven in recent versions by SNOPT,[1] a commercially available algorithm for finding global extrema in large-dimensioned problems with multiple constraints. In this case, constraints are either equations of motion, limits on the capabilities of the vehicle (i.e., maximum and minimum rotor speed), or "hard-coded" limitations on the nature of the autorotation maneuver (i.e., negative HAGL is disallowed).[2] The cost function to be minimized in the optimization is referred to as the objective function; it values the current autorotation solution according to a set of flexible, user-defined criteria (common criteria quantities and relative valuations for the function are laid out in appendix B). In each iteration of the code; first, the current solution is adjusted to ensure that constraints are not violated; second, control variables are perturbed according to objective function derivatives calculated by SNOPT; third, the new solution is evaluated against the objective function to ensure an improvement has occurred. The loop ends when no further improvement is possible or an exit criterion (such as a sufficiently gentle landing) has been satisfied.

---

[1] Gill, P. E., Murray, W., Saunders, M. A. SNOPT: An SQP Algorithm for Large Scale Constrained Optimization; *Society for Industrial and Applied Mathematics Review*, **2005**, *47* (1), 99–131.

[2] A detailed discussion of DESCENT's Equations of Motion, Additional Constraints, and Objective Function can be found in the code's user manual; ARL-TR-5906; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, February 2012.

4.3 Output Data and Visualization

Code execution produces a text-file output detailing the optimized solutions found by DESCENT for each H/V case in the analysis domain (flight envelope). The heading of the output file contains a listing of the inputs read from the NameList input file and some intermediate quantities created from them. For each case within an analysis, DESCENT outputs the time-history of the most important state variables, control settings, and some instantaneous derivatives. It also produces a summary section that lists each case's location in the H/V domain and vertical impact velocity for $P_{k|d}$ creation. Sample output is attached as appendix C.

Visual postprocessing is done via a MATLAB* Mfile script attached as appendix D. Four plots are created (figures 2–6) that display different aspects of the output for the analyst. Discussion of each plot type is included in the figure captions. Typically, a small number (less than 10%) of the cases in an analysis fail to converge in an intuitive manner, or in a manner consistent with the surrounding cases. These cases will often be excluded from the output plots in order to increase the contrast in the colormap of the remaining data (in the case of an outlying result) or avoid erroneous appearances in the plots. Removal of outlying data is demonstrated in figure 3. In any event, all of the underlying information is retained in the text-file output, so the data is still available if required.

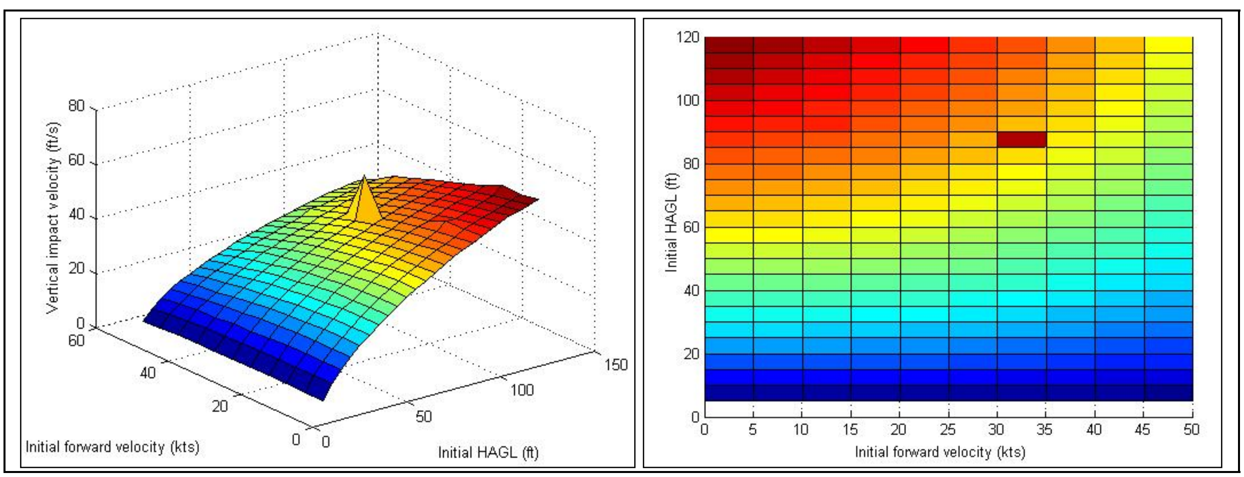

Figure 2. Surface plot of vertical impact velocity over the analysis domain for a total-power-loss scenario. Note: Although any combination of state variables can determine kill category, vertical impact velocity is the typical quantity, displayed here. Each grid vertex (data point) represents a separate optimization done by the code. Note that one H/V location displays particularly poor convergence.

* MATLAB is a registered trademark of The MathWorks, Inc.

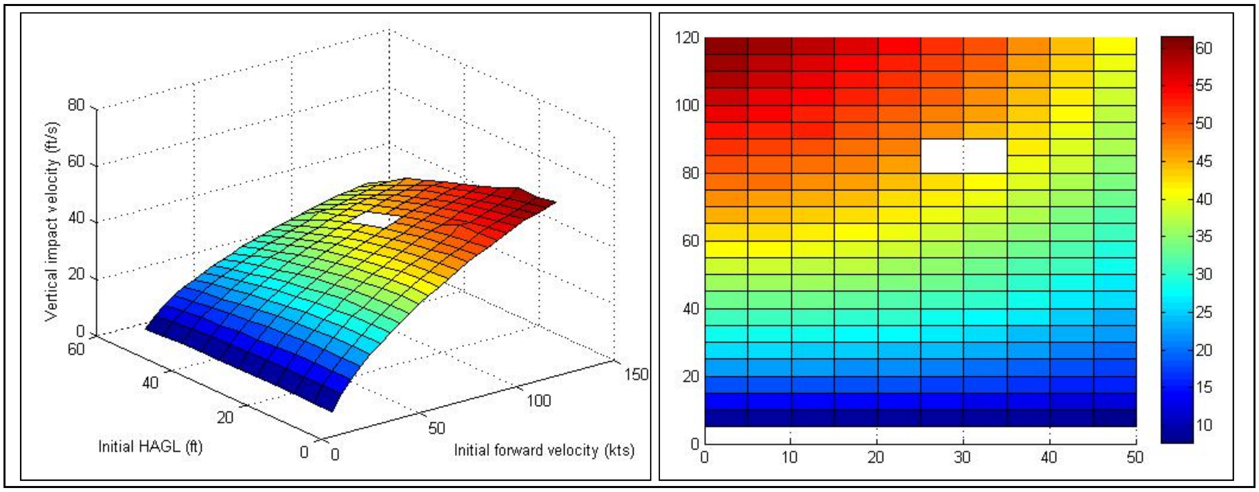

Figure 3. Surface plot with the poorly converged data point removed.
Note: This improves the contrast evident in the colormap (right) when the range of impact velocities is narrowed.

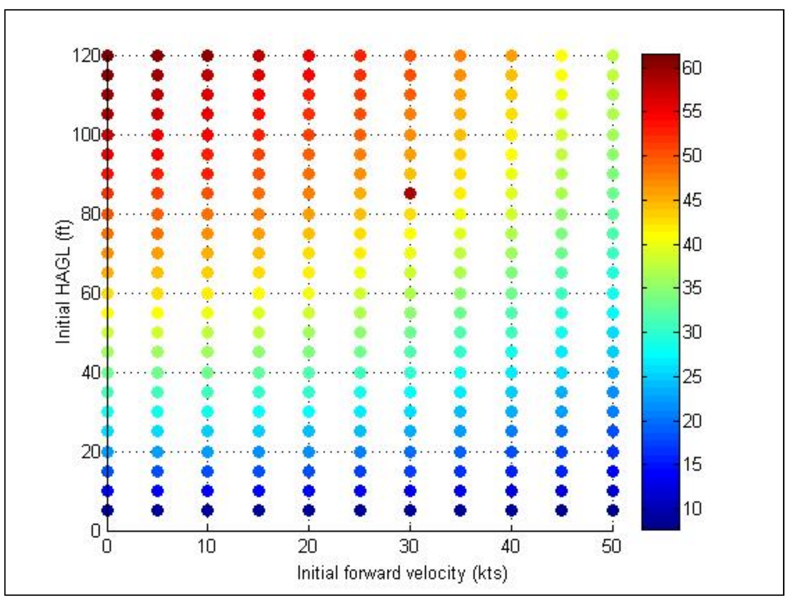

Figure 4. 3-D scatterplot of the same output.
Note: This example is rotated to mimic a typical H/V diagram. The colormap legend, as before, represents vertical impact velocity in ft/s.

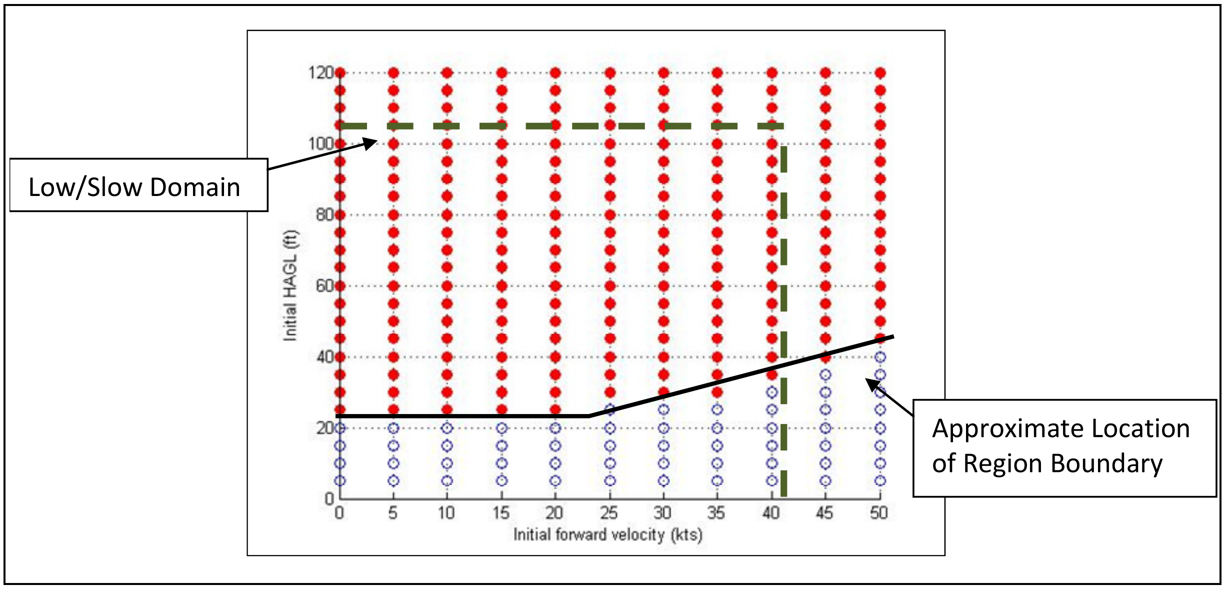

Figure 5. $P_{k/d}$-related scatterplot output.
Note: Each data point is assigned a new value of 1 (red) or 0 (white) based on whether the original value was greater or less than the velocity criteria for the relevant kill category. In this case, "Attrition" kills (red) were assigned to cases where the vertical impact velocity exceeded 24 ft/s. Criteria are platform- and analysis-dependent. Annotations to the figure show the extent of the flight envelope over which $P_{k/d}$ is calculated, in this case the "low/slow" domain, and the approximate region of Attrition kills; the Attrition $P_{k/d}$ is then simply the percentage of the low/slow domain comprised by the Attrition region. Note that the H/V domain considered in a DESCENT analysis is not identical to a $P_{k/d}$ domain. Frequently, the former will be larger so that trends extending beyond the $P_{k/d}$ domain are discernible to the analyst.

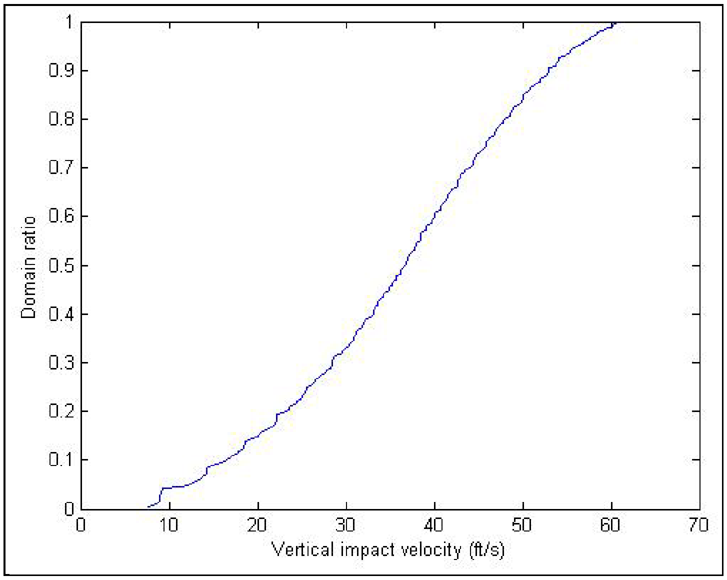

Figure 6. Domain ratio plot.
Note: The "domain ratio" plot is the final piece of standard DESCENT output. It shows the portion of the H/V domain wherein a vertical impact velocity of a given value or less is possible. The domain ratio, then, is the probability of *survival* for a given threshold velocity (or in other words, the opposite of the kill probability). The data is created by pairing each impact velocity listed in the output file (from lowest to highest) with a cumulative count of the number of H/V cases paired, then dividing by the total number of cases. This output is useful both as a measure of sensitivity to velocity threshold and as a troubleshooting tool. Large breaks or discontinuities in the plot are often signs of a malfunctioning optimization algorithm.

5. Simulator Data Extension

One challenge to comprehensive V&V of the DESCENT model is the lack of a large, diverse set of autorotation test data from which to compare modeling results. Ideally, a systematic method for creating data on demand would exist, as new vehicle configurations and mission conditions are unpredictable. Real world testing is prohibitively costly (or dangerous, depending on the flight conditions under consideration), so a pilot-in-the-loop simulator model has been identified as an acceptable substitute.

High-fidelity simulators of most, if not all, Army inventory rotorcraft are already in operation for purposes including pilot training and hardware evaluation. The DESCENT extension will be able to harness these existing uses of simulators for its own data collection purposes. During an autorotation, it will simply monitor and record vehicle state and control level data as it is created. The simulator model's equations of motion, assumptions, and other intellectual property will be undisturbed, and in fact, will be mostly transparent to the data collection process.

Upon completion of the training exercise, a record of the autorotation and pertinent helicopter properties will be collated into a test case data file.

Once the test case data is available, the software will save it to a local database for export to ARL/SLAD. This might be done automatically over a network connection or periodically through e-mail or other means. ARL/SLAD will then be able to organize the test case by vehicle characteristics and flight conditions; over time, statistical analysis of a "median" maneuver will become possible for most common conditions. Crucially, if an unusual aspect of the model requires V&V, a set of "customized" test case data could be procured for a relatively low cost from the organization that hosts the simulator. Thus, the extension provides not only a continuous stream of new V&V data, but the flexibility to create unique data matched to the needs of the analysis.

5.1 Data Collection Requirements

The simulator data extension will require both "cause" and "effect" data, i.e., a record of both the control settings and the resulting helicopter state through time.

On the control side, since DESCENT simplifies pilot controls as the magnitude of the lift coefficient and the lift vector's pitch-wise orientation, this process must be able to similarly produce two-degree-of-freedom inplane control variables, analogous to collective and longitudinal pitch, respectively. Options for achieving this include recording blade angle-of-attack as a function of azimuth, swashplate position and orientation, or the settings of the pilot control inputs. The exact implementation strategy will depend on which group of control variables lend themselves most consistently to a transfer function for translating to DESCENT control variables.

On the state side, the data collection process should record all of the data that DESCENT produces. These quantities include HAGL, fuselage pitch orientation, velocity, rotor speed, and engine power supplied to the rotor. Each of these will necessarily be calculated in the context of the normal application of the simulator, ensuring no extra capabilities or processing burdens will be required of the simulator model in order to fulfill the needs of the data collection process.

Finally, the process must record pertinent information about the specifics of the autorotation event. Initial flight conditions, environmental variables, severity of power loss, vehicle size and drag characteristics, and other similar quantities should be recorded to ensure that the DESCENT results are comparable to the simulator output.

5.2 Application Examples

Potential applications of this extension fall into two broad categories: improved V&V of the DESCENT model, or assistance in transitioning it from purely a "best-case" optimization model to a code that can also predict likely real world maneuvers with reasonable fidelity.

Model V&V primarily refers to validation of what could be referred to as the qualitative "overall" solution maneuver produced by DESCENT. The need for a large data set to supplement evaluation of the model becomes apparent when the potential diversity of successful maneuvers is considered. At the H/V boundary, where safe autorotation becomes impossible, only one maneuver—the optimal—results in a successful landing. Near this boundary, a small family of similar maneuvers is successful. Far from the boundary, at high initial speeds and/or at large HAGL, many diverse strategies can result in a successful autorotation, and indeed one or more outright mistakes from the pilot can be compensated for. (It is not coincidental that most flight testing of expensive rotorcraft platforms occurs in the latter region.) However, this flexibility makes validation difficult with limited data. If DESCENT produces a maneuver that is much more aggressive, much longer, or differently ordered than the flight test data, is the model prediction wrong (per se, as in nonphysical, or otherwise impossible), or just differently right (figure 7)? It is extremely advantageous to have a large set of flight test data to work from so that a modeling result that does not resemble *any* flight test can be more confidently characterized as unlikely, if not erroneous.

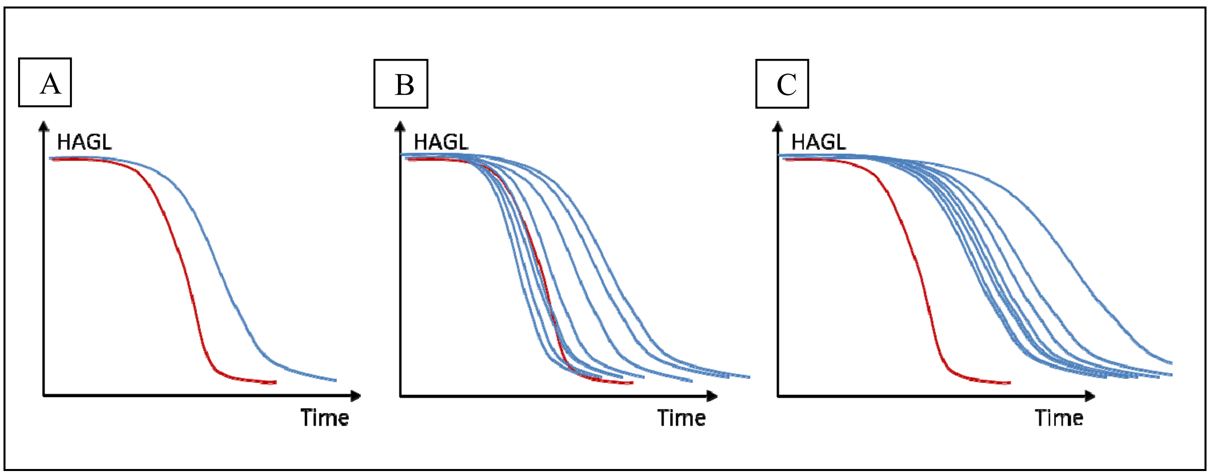

Figure 7. Hypothetical plots of HAGL vs. time for DESCENT output (red) and validation data (blue).
Note: With only limited flight test data (plot A, left), it is difficult to tell whether somewhat differing results are actually divergent. Using simulator data, it becomes apparent whether the DESCENT solution is within the range of plausible solutions (plot B, center), or outside of it (plot C, right).

Another facet of model V&V is the valuation of parameters that are set, if not arbitrarily, then with a great deal of analyst discretion. These are usually vehicle characteristics that require surrogation of another vehicle, or an informed assumption by a user. For example, the maximum rate of change in the rotor disk's lift coefficient (dC_l/dt_{max}) is a quantity that is difficult to calculate from vehicle control stick rates, is often unavailable from the manufacturer, and in practice is often defaulted to a standard value. Using a typical flight test instance, the analyst can record the greatest dC_l/dt observed *in that flight*. This one-time maximum is useful for verifying that the model parameter is not lower than the observed value, but is unlikely to reflect the true performance boundary of the vehicle. The aggregation of many flights in a simulator, however, is much more effective at arriving at a limiting approximation of the true maximum. This is especially true because the pilot-in-the-loop can be instructed to use one or more runs to push the limits of the parameter of interest; i.e., perform an extremely aggressive flare.

Beyond V&V, the simulator data extension has extensive application to finding a real world outcome via what is otherwise inherently a “best-case” model. Since the flight path is simply the helicopter’s equations of motions applied to the control history, nonoptimal autorotation is the result of nonoptimal pilot input. Given enough pilot input histories, emergent trends can be characterized (figure 8). These nuances can be encouraged in DESCENT’s objective function, and the model will then begin to replicate actual autorotations more closely. For instance, if the real world steady descent velocity during autorotation is typically somewhat slower than what the optimization algorithm predicts, this fact will become apparent in the aggregated data. Then DESCENT can be modified so that it evaluates solutions based on matching the empirical result as well as prior criteria. This will ultimately lead to $P_{k|d}$ output that reflects the realistic tendencies of the entire pilot-driven system and not just the theoretical capabilities of the hardware.

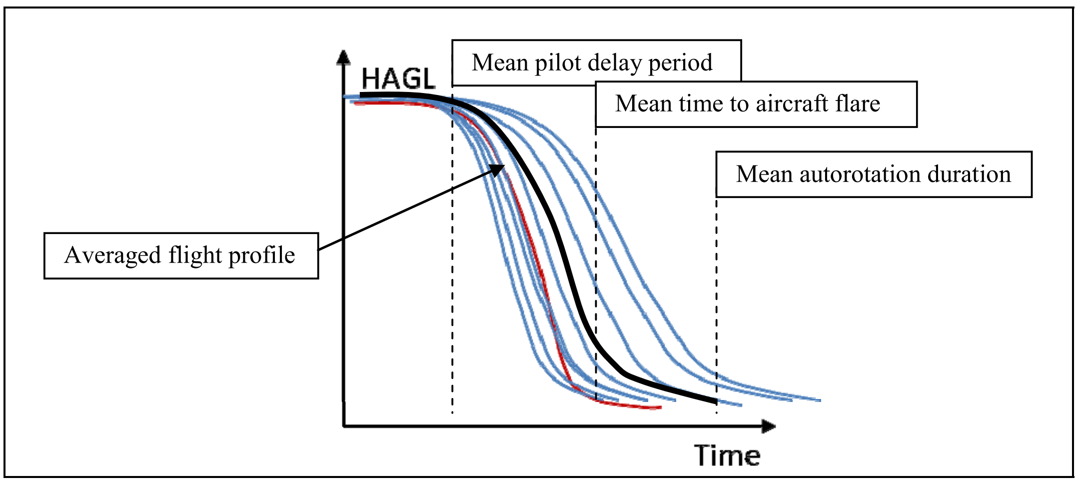

Figure 8. Averaged flight profile from hypothetical data.
Note: Trends in simulator data can be used to characterize the “average” maneuver for the purposes of emulating it in the objective function. The DESCENT-calculated optimal solution (red) will usually differ from the typical real world maneuver (black), so editing the objective function will be necessary if replicating a “real” outcome is required for the analysis.

A related “real world” application of simulator data is assistance with the description of randomness and probability distributions in the maneuver (figure 9). This is related to quantifying pilot tendencies, but expresses the timing or magnitude of an action in terms of a probability distribution, rather than a mean value. DESCENT is otherwise a deterministic model. By allowing it to randomly pick values for parameters, such as pilot reaction time delay, a range of output is produced over a number of executions. This delivers a stochastic analysis similar to that available from other ARL S/V codes. Additionally, it provides an understanding of the sensitivity of the results to changes in values of variable quantities.

Parameters that have little effect on the eventual range of outcomes can be either neglected or de-emphasized in future model enhancements, increasing the efficiency of the code. Parameters shown to be very sensitive to the solution end-state will be prioritized for increased scrutiny.

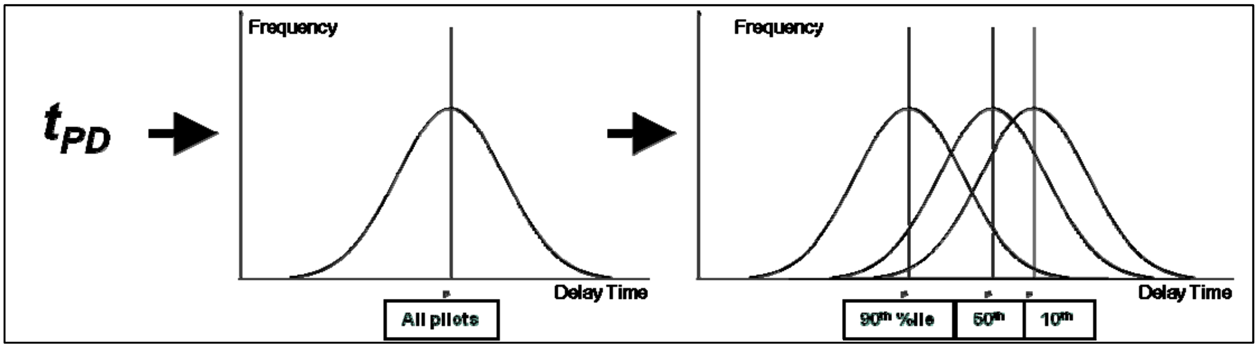

Figure 9. Probability distribution generation.
Note: A constant value for pilot delay (t_{PD}, left) can be replaced with a probability distribution that lets DESCENT play the behavior with a certain degree of randomness (center). Given enough information about the pilots creating the data, a further refinement is possible, so that an analyst would be able to choose the ability level of the pilot (right). The latter step enables analyses to be biased towards more conservative, or more optimal outcomes, as required.

Finally, simulator data will assist in the extension of the code to considerations beyond its explicit capabilities. For example, as a two dimensional aerodynamic/dynamic model, DESCENT does not consider out-of-plane motions, such as roll and yaw of the fuselage. Clearly, however, the rolled state of the fuselage at impact will affect occupant loading and survivability; this in turn affects the kill probability of the overall system. The current solution is to assume an unrolled impact. A better solution is to attempt to define a set of loading "penalties" through trends observed in structural dynamic modeling and simulator data. The penalties would likely take the form,

$$C_i = L_i P_i, \tag{1}$$

where C_i is the penalty constant applied to a loading quantity; L_i is the multiplying factor for a particular state (such as fuselage roll), and might not be a constant; and P_i is either the probability of a state occurring, or a random draw on that probability to achieve a binary value. These penalties could be computed stochastically if the analyst desires to account for them.

Other possible penalties could be applied to account for unmodeled aspects of the problem, such as wind speed and direction, terrain type, pilot encumbrance (due to smoke, darkness, etc.), or landing gear damage.

5.3 Progress Status

Agreement has been reached with the U.S. Army Research Development and Engineering Command Communications-Electronics Research, Development, and Engineering Center

(CERDEC) on the feasibility of implementing a data-gathering patch in their Aircraft Survivability Equipment Integration & Digital Simulation Laboratory (ASEIL) simulator. The ASEIL is not platform-specific and is designed to be able to represent major helicopter systems in the Army inventory. A proposal has been submitted for completing this work in FY14 (see section 2); limited data-gathering is also possible with qualified CERDEC personnel.

In future years, the intention is to expand this project to platform-specific high-fidelity flight simulators with greater exposure to different pilots. Program offices for the T-BOS (Black Hawk) and LCT (Apache Longbow) training simulators have been contacted regarding a similar software patch implementation and cost estimates have been delivered. Once this implementation is complete, adoption by training locations will be the final step before the simulator data extension is fully functional. This is not seen as a difficult hurdle, and the effort will likely be assisted by presentation of the benefits of the existing CERDEC partnership.

6. Occupant Outcome Prediction Extension

The nature of the DESCENT model is that it relies on external sources of information for mapping impact conditions to damage levels, i.e., setting the threshold criteria for assigning kill categories. This is necessary because "damage" must be defined flexibly based on the needs of the analysis. Unfortunately, comprehensive information on how impact loading is transmitted through the vehicle to different onboard systems is often difficult to come by. As a result, a simplifying assumption is necessary.

This assumption is that landing gear failure is a proxy for sufficient fuselage damage to ensure that at least one critical system, somewhere onboard, is irreparably damaged. Therefore, instead of trying to model loading paths through the structural members of the vehicle, the analyst can simply look for failure in a single component in direct contact with the ground. Instead of calculating stress or strain levels in the landing gear, explicitly, a failure quantity—such as kinetic energy absorption—is used to find an associated critical impact velocity. In this way impact conditions are connected to vehicle damage state.

One problem created by this approach is that onboard components, which are sensitive to different forms, or lesser magnitudes of loading conditions than the landing gear, are effectively excluded from the analysis. In particular, human occupants (pilot[s], other crew members, and passengers) are vulnerable to a wide spectrum of injuries due to a "rough" landing that does not cause a structural attrition of the helicopter; these outcomes would usually not be captured by the core DESCENT process. This blind spot in DESCENT's analysis capability is viewed to be increasingly unacceptable as occupant outcome becomes a central concern of Army S/V analysis. But to explicitly consider occupant outcomes, an entirely new extension of the code is necessary, including additional data sources and analysis methods.

6.1 Data Flow Overview

Driving this extension is the observation that occupants and structural components are vulnerable to nonidentical—although considerably overlapping—sets of impact conditions. This means that, (1) the portion of the H/V domain where a survivable landing is possible differs for occupants versus structure, and (2) the objective function that optimizes impact conditions will differ from occupants to structure. Identifying those differences requires that:

1. Before the analysis, a structural dynamics model of the vehicle (figure 10) is created, such as with a finite-element analysis (FEA) tool; MADYMO* has been used in previous work. This model must represent load transmission and structural-deformation through the landing gear and fuselage to occupant locations with reasonable fidelity. The FEA model is populated with manikin entities to record loading at the occupant locations, and a contact function is created to represent the ground properties of the impact site.

2. The model is "crashed" repeatedly in a parametric study of different impact characteristics. Fuselage pitch and roll orientations, the two components of the impact velocity vector, and other parameters of interest are varied during the FEA runs. Occupant loads (lumbar compression, neck torque, tibia stress, etc.) are recorded so that expressions for each loading type as a function of the impact conditions (figure 11) can be created. These expressions have minima at the optimized impact conditions for occupant outcome.

* MAthematical DYnamic MOdels, http://www.tassinternational.com/madymo.

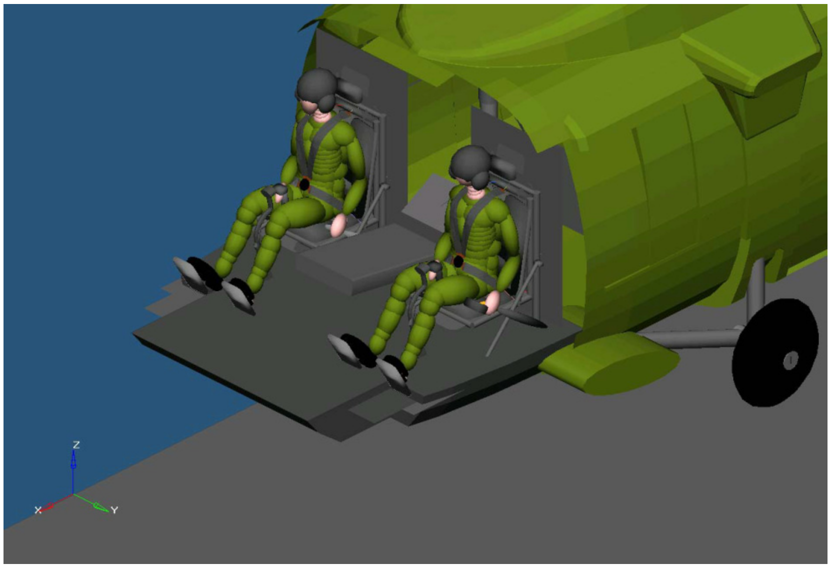

Figure 10. H-60 MADYMO model.
Note: View of the H-60 MADYMO model used in the 2009 Naval Air Systems Command (NAVAIR) analysis (see section 4.3). Occupants are representative models of the Hybrid-3 manikin (sensor-enhanced dummy) and automatically record all loading measurements that the actual manikin would take.

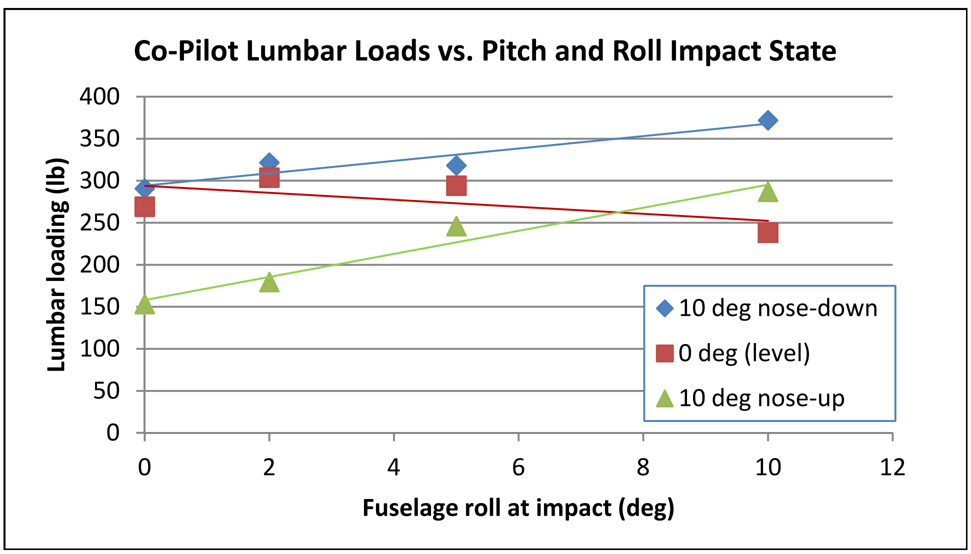

Figure 11. Lumbar loading due to impact.
Note: This analysis used a refined model of Bell 206 helicopter with a 25 ft/s horizontal, 25 ft/s vertical impact velocity vector. Linear trends in lumbar loading for impacts at different pitch and roll orientations. This plot shows that, for this impact vector, optimal occupant outcome is likely found at a flat roll orientation and slightly nose up pitch orientation. In all cases, loading was well below the 1800 lb threshold for injury.

3. DESCENT is executed for the scenario under analysis. The DESCENT output is optimized outcome for the vehicle structure, at least according to the assumptions laid out previously. It will be referred to here as a "$P_{k|d}$-vehicle" quantity and an "H/V-vehicle" diagram.

4. The relevant impact conditions at each H/V location are collected and input into the loading expressions, creating an "H/V-occupant" diagram. It is important to remember that this is *occupant* outcome when the autorotation is done with the default objective function, i.e., with the *structural* outcome primarily in mind.

5. Injury lookup tables (maintained by ARL/SLAD) are consulted to determine where on the H/V-occupant diagram unacceptable injuries occur. Mapping loading to injury, then combining the regions related to various unacceptable injuries and dividing by the size of the entire analysis domain, creates a "$P_{k|d}$-occupant" quantity.

6. The H/V regions that correspond to an occupant injury and a vehicle kill are compared (figure 12). The underlying motivation of explicit occupant outcome prediction is that the $P_{k|d}$-occupant is just as important a quantity as the $P_{k|d}$-vehicle, even if the former is unrelated to formal S/V metrics. If (as is likely) the regions differ, one of several actions may be undertaken in order to reconcile the measures into a single $P_{k|d}$ output.

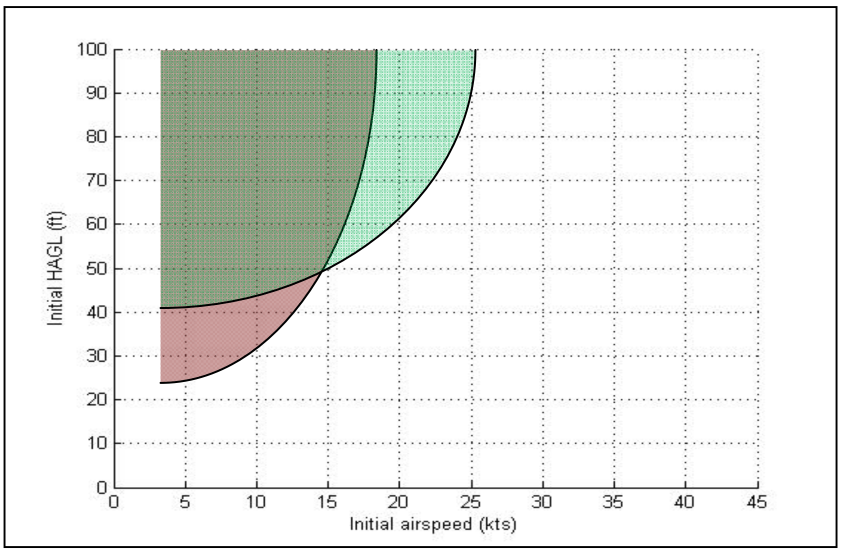

Figure 12. Kill region comparison.
Note: Kill regions for a typical vehicle (red) and hypothetical occupants (green) must be compared to determine the next step in the analysis. Where the regions overlap, both a vehicle-kill and occupant injury are predicted to occur. In the green-only region, the occupant-focused analysis process can determine if a differently prioritized autorotation maneuver might produce a better outcome.

- 6a. For analyses where one $P_{k|d}$ is specifically required, simply use that one.
- 6b. If the occupant injury region is a subset of the vehicle kill region within the overall H/V analysis domain, it is suggested that the more conservative (higher) $P_{k|d}$-vehicle quantity be used.
- 6c. If the regions overlap imperfectly, or the occupant injury region is more extensive, software script will attempt to resolve the discrepancy automatically. First, the script will list the H/V locations where further analysis is desired. Then, *one location at a time*, the script will compare the impact conditions output by DESCENT with the parametric study expressions discussed in step 2. If there is a significant difference, the script will change the operative objective function in DESCENT to encourage the more beneficial impact conditions. For example, if a greater fuselage pitch appears advantageous, the objective function will be edited to increase the target impact pitch and increase the weighting of the relevant term.

  DESCENT is re-executed. Occupant and vehicle outcomes are both recorded and compared to threshold criteria. It is expected that the tradeoff for advantaging occupant outcome will be an impact of greater magnitude; the script will have to determine whether this tradeoff is worthwhile. In some analyses, a purely occupant-centric evaluation might be desired, whereas in others, altering the objective function to the point that a structural kill is incurred will be a signal to backtrack.

  The objective function will continue to be perturbed along the guidelines of the parametric study expressions and the results of previous iterations. Eventually, a stopping point will be reached, and the next H/V location is considered.

  Once the entire region of initial discrepancy has been re-analyzed (and, hopefully, reduced considerably) the analyst must decide how to resolve outstanding discrepancies between the occupant injury and vehicle kill regions.

7. DESCENT output is reproduced, with a separate output file listing the objective function values operative at each H/V location where a re-analysis took place. This will assist in reproducing the results at a later time and (if an application presents itself) discerning trends in how optimal objective function values change with H/V location.

The data flow of the overall process described above is diagrammed below in figure 13. Its place within the context of the integrated DESCENT product is shown in section 8 (figure 21).

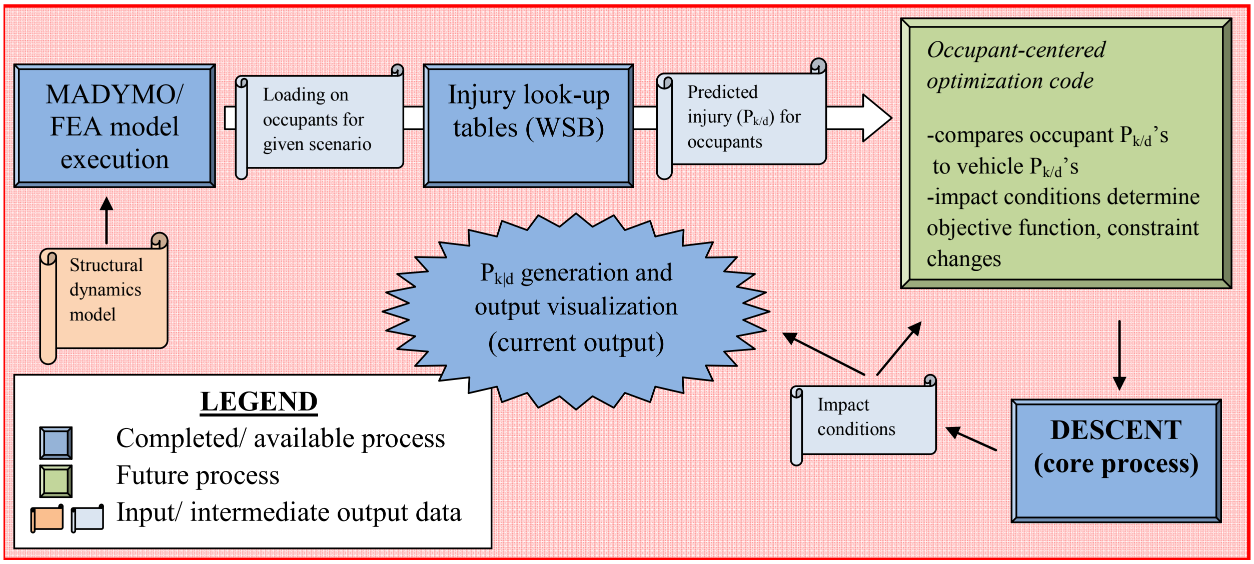

Figure 13. Occupant outcome data flow diagram.
Note: Occupant $P_{k|d}$ prediction (top middle), is already possible as per NAVAIR analyses (section 6.3); DESCENT-produced vehicle $P_{k|d}$ generation is discussed in section 4. Linking the two capabilities is the focus of work (green square) proposed for FY15–16 and discussed above as steps 6–7, and especially, 6c.

6.2 Modeling and Validation Issues

Difficulty in the creation and maintenance of a sufficiently high-fidelity FEA model is the chief obstacle to success in this project. To date, a basic model of an OH-58 variant (updated somewhat by NAVAIR) has been the workhorse for demonstrating the sequence of data processing in the proposed analysis process. However, this is the current extent of the FEA model inventory.

It appears clear that providing information in a timely manner for regular S/V analyses will require that base models already exist for the relevant vehicles and that they be quickly editable to reflect recent changes. The initial investment in model generation might be cost-prohibitive if unsupported by other stakeholders, and it is unclear how much cooperation from vendors and/or other parties is available for analysis-specific updates at the time of testing.

Furthermore, it is not readily apparent exactly how much investment in model creation is required. Validation of the models is problematic since real world crash scenarios are difficult to reproduce. One strategy might be to continue refining, or adding detail, to the FEA model until loading output converges—but there is no guarantee of that convergence leading to a broadly correct solution. Even the definition of what constitutes a “correct” solution will change based on the differing needs of each new analysis. Along these lines, priorities for further development of this extension will have to be the definition of an FEA model’s accuracy from an occupant-loading standpoint, and agreement on a suitable test strategy for the created models; this should be done prior to any new systematic model creation.

6.3 Progress: 2009–2012

Beginning in 2009, ARL/SLAD and NAVAIR cooperated in demonstrating the practicality of using DESCENT output to define a set of impact conditions that would ultimately lead to injury predictions.[3,4] A mostly rigid-body-model of an H-60 variant (figure 10) was "crashed" in the MADYMO structural dynamics code at several horizontal and vertical impact velocities. Multiple forms of loading were measured on the manikin entities—including at the neck, head, chest, spine, pelvis, and legs (figure 14). The data was compared to ARL injury tables to create injury, or incapacitation, predictions based on the nominal loading data. Although the H-60 FEA model was not sophisticated enough to be validated, the project successfully demonstrated the sensitivity of injury predictions to variations in impact conditions within the range normally produced by DESCENT optimizations.

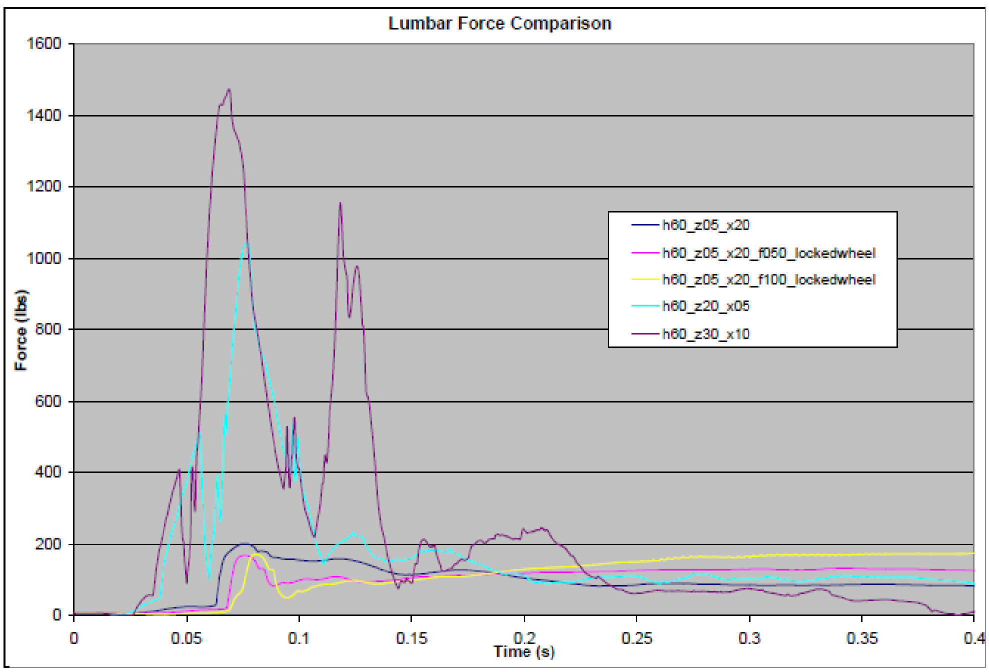

Figure 14. Lumbar force resultant experienced by the manikin under different impact conditions.

Note: The test cases are: 5 ft/s vertical and 20 ft/s horizontal impact speed (no ground friction, dark blue line; some friction, magenta line; high-friction, yellow line); 20 ft/s vertical and 5 ft/s horizontal impact speed (no ground friction, cyan line); and lastly, 30 ft/s vertical and 10 ft/s horizontal impact speed (no ground friction, maroon line). The influence of ground friction was negligible at low-impact speeds. Changing the direction of the impact velocity vector from mostly horizontal (dark blue) to mostly vertical (cyan), without increasing its magnitude, increased the peak lumbar loading upwards of 500%. Peak loading continues to increase proportionally to vertical impact speed as speed increases to 30 ft/s (maroon).

---

[3] Kitis, L., Sieveka E., Paskoff, G. *Injury Analysis of MADYMO Simulations of 50th percentile Hybrid III Manikin Under Controlled Sink Rate Conditions*, 7 August 2009.

[4] Shukla, N. *Crew Casualty Assessment for Descent of H60 Helicopter*, 1 July 2009.

Further work in 2010 and 2012 was also based on the goal of demonstrating the sensitivity of output to different input variations without focusing on the accuracy of the numbers, per se. Again constrained by a lack of pre-existing developed FEA models, a more advanced Bell 206 (basis for the OH-58) was chosen in 2010 and refined heavily in 2012, for demonstration projects. As in 2009, a focus was placed on demonstrating both the practical throughput of data through the process and the plausible sensitivity of output-to-input perturbations at the relevant margins. Producing "usable" results—in the sense of applying the product itself to an S/V analysis—will have to wait until progress discussed in the previous section is made towards generating more detailed FEA models of the vehicles.

In 2010, the Bell 206 was subjected to a number of different impact vectors at level, nose up, and nose down fuselage orientations. Seats were modeled as either locked or stroking to give a look at how load-mitigating safety equipment might change the modeled outcome. Some clear trends in the data, shown in figure 15, were visible despite the limitations of the 206. This work formed the basis of the 2012 study, which used a better-developed version of the same 206 model and added roll orientation to the parameterization. The goal was to demonstrate that creating parametric study expressions for loading as a function of a state variable was feasible—even when the state was not one accounted for in the DESCENT model (figures 11 and 16). That work is ongoing, but preliminary results are promising.

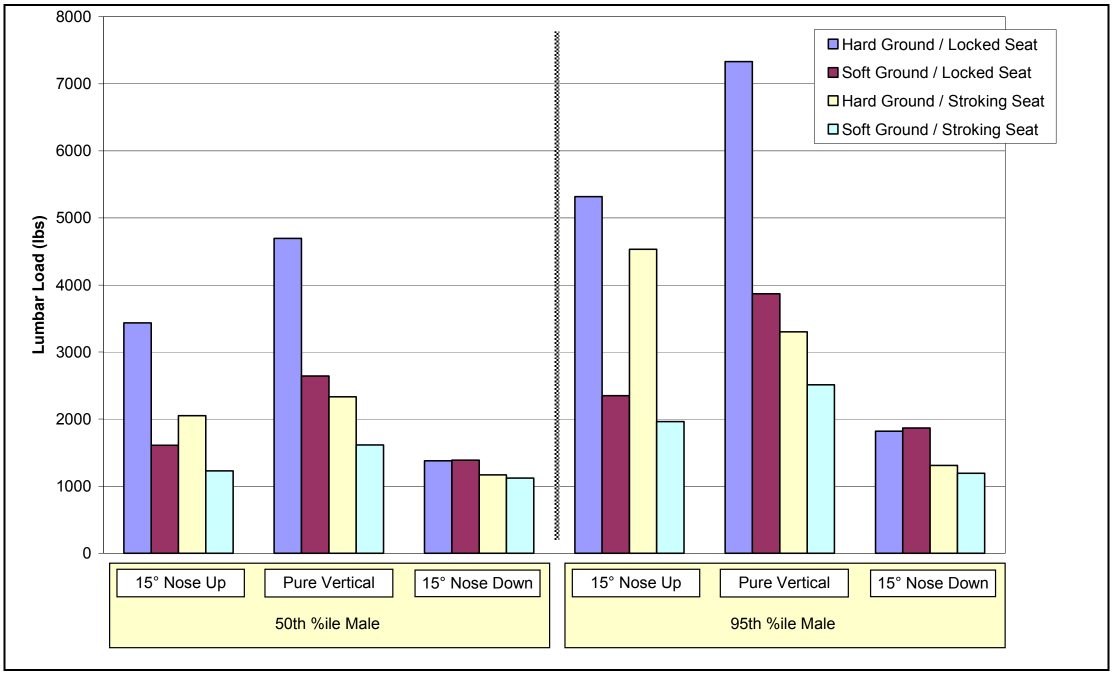

Figure 15. Lumbar force resultant experienced by medium and large manikins under different impact conditions.
Note: Impact velocity was held constant, while fuselage pitch orientation was varied. Note the discernible effect of a stroking seat, most dramatic vs. a hard ground surface. Also, it is clear that a nose down impact orientation that minimizes the pitching-forward motion after impact is most advantageous for the occupants in this case. The result appears to vary based on the impact velocity vector chosen. Investigating these trends with a higher quality FEA model will be a priority in the future.

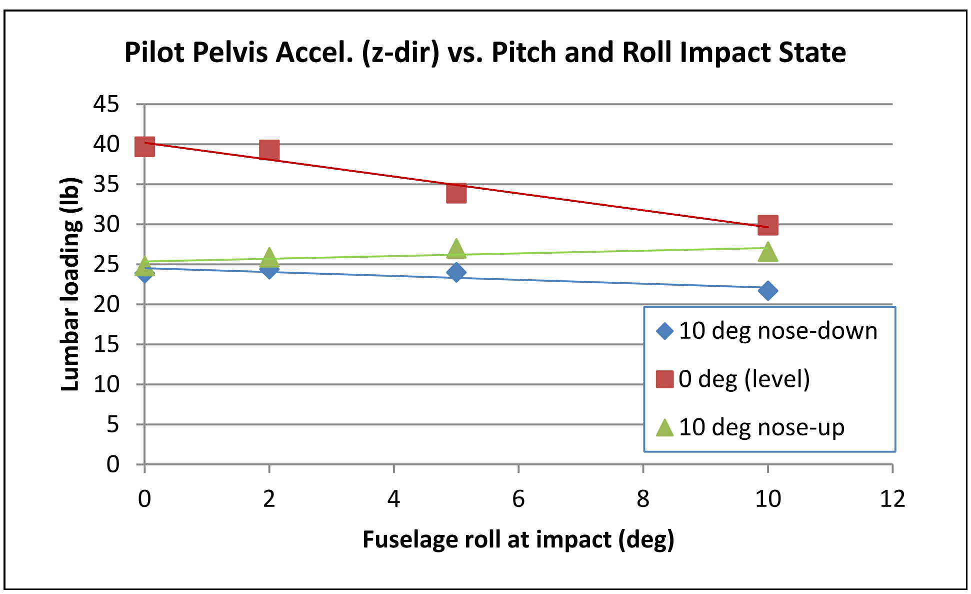

Figure 16. Refined Bell 206 model with a 25 ft/s pure vertical impact velocity vector.
Note: Linear trends in pelvic acceleration for impacts at different pitch and roll orientations. This plot suggests that some forms of occupant loading become consistently less severe with increasing fuselage roll. Successfully avoiding occupant injury will require finding “sweet spots” that minimize the entirety of experienced loading.

6.4 Future Development and Application

The first priority for future development of this extension is resolution of the FEA model creation problem. The lack of sufficiently detailed vehicle models will hinder future progress on this extension indefinitely. Once this problem is addressed, applications of the work can be explored.

One such application comes from parametric studies of occupant safety equipment, similar to those routinely conducted for civil automobile design. For a range of impact conditions predicted by DESCENT, the change in occupant loading due to force-mitigating seats, harnesses—or even occupant positioning or posture—can be analyzed. These results could be combined with the design space analysis extension (discussed in the following section) to determine whether the weight penalty of additional equipment overwhelms its load-mitigation benefit from an S/V perspective.

Another potential application is the study of pilot behavior and how traditional autorotation control technique affects occupant survivability differently than vehicle survivability. With simulator data available to characterize "typical" maneuvers, it may be shown that landing conditions pursued by the pilot with the goal of minimizing damage to the airframe actually place additional risk on the occupants. For example, in the 2010 NAVAIR study, one finding was that the nose up landing commonly thought to minimize structural loading was not optimal for some forms of occupant loading at certain impact vectors (figure 15). In fact, a slightly nose down landing minimized the whiplash effect of the vehicle pitching forward upon impact. If this result were to hold up with more reliable modeling inputs, it may have implications for piloting, strategy, and training. The ability to explore our assumptions about "textbook" autorotation techniques is valuable for both occupant and vehicle survivability maximization.

7. Design Space Analysis Extension

DESCENT's contribution to S/V analysis has typically been near the end of the testing process, when the vehicle configuration has been largely finalized and a verification of its survivability-related capabilities is required. The design space analysis extension is based on the belief that survivability is a vehicle characteristic that can (and should) be traded against others during the earlier design phases of the vehicle's lifecycle. This extension uses design of experiments (DOE) and full-factorial parametric analysis to map out the relationship between DESCENT-produced $P_{k|d}$'s and one or more analysis parameters, by executing the code in advance, using sets of plausible values for each parameter. The result is an instantaneous evaluation of how changing vehicle characteristics like weight or rotor solidity will impact survivability capability; available so that survivability can be scored against considerations like range, maximum takeoff weight, or armament payload, when evaluating vehicle designs.

7.1 Creation of a Design Space

Although, in theory, any quantifiable characteristic of the helicopter is variable, several key characteristics stand out as both frequently affected by common design decisions and influential to the vehicle's survivability capability. These include, primarily, *gross weight*, *engine power*, and *rotor inertia*.

The design space is defined as the set of plausible values that these variables might take, assuming mutual independence (figure 17). The characteristics of the realized rotorcraft design will be one of the possible combinations of values within this space. For an *n*-dimensional space, the $P_{k|d}$ of the hypothetical vehicle can be expressed as a function of *n*+1 variables:

$$P_{k|d} = f(v_{crit}, x_1, x_2, \dots, x_n). \quad (2)$$

This expression is found by populating a sizable portion of the design space with $P_{k|d}$ values (by executing DESCENT iteratively), then data-fitting to create an empirically-based survivability function.

Quantity	Nominal Value	Minimum	Maximum
Gross aircraft weight (lb)	18000	14000	20000
Remaining engine power (hp)	1250	1000	1600
Rotor inertia (lb-ft^2)	4200	4000	4500
Pilot reaction time delay (s)	2	0	4
Velocity weighting	1.0	0.1	5.0
Critical velocity (ft/s)	10	4	25

Figure 17. Typical design space for a medium-large helicopter.
Note: Nominal value refers to the "actual" value in the present design, or most likely value for a future design. Minimum and maximum values can be determined by surrogation from similar vehicles, alternative design scenarios, or plausible limits. The core DESCENT analysis is currently run only at the nominal value, as requested by the customer; this extension will enable analyses nearly simultaneously throughout the design space.

The basis of DESCENT's prototype five-dimensional design space was chosen to consist of the vehicle characteristics mentioned previously, a quantity that reflects pilot skill (*reaction time delay*), and the semi-arbitrary analysis parameter *velocity weighting*, which is the relative optimization priority of the vertical and horizontal descent velocity components. Velocity weighting was chosen because it is has poorly understood effects on the final output, it likely has different "ideal" values at different regions of the H/V domain, and there is no clear rationale for choosing a given value. Adding velocity weighting will allow an assessment of what value produces a minimum $P_{k|d}$ for rotorcraft of various characteristics, and hopefully lead to a more systematic guidance for choosing weighting values in the future. It is possible and expected that the number of dimensions in the design space will increase beyond five as resources allow and requirements dictate.

Analysis of this type quickly suffers from "the curse of dimensionality," or as *n* increases, modeling all possible combinations of an *n*-dimensional problem becomes exponentially more resource-prohibitive. As a first step, the five design space dimensions are being modeled in a full-factorial (all possible combinations) scheme. A very coarse grid is used to contain the number of DESCENT executions required. Ultimately, a finer grid is desired to help capture nonlinearities in the relationships between variables, and a transition is envisioned to a DOE-based method of populating the design space. A variety of techniques, such as kriging, are available for intelligently choosing domain points to test. Choosing a mathematical model and implementing the automation script is planned for FY15. When this is completed the design space can be expanded dimensionally, or populated more finely, without placing substantial burdens on computing resources.

7.2 Data Reduction and Visualization

The design space is populated by an automating script that re-executes DESCENT as needed and stores the output systematically. Thus, in total there are several hundred or more points in the design space that each contain several hundred autorotations at unique initial H/V values. When the output visualization is created, a significant amount of data reduction is required for the information to be usable.

The first step is reducing each execution instance (an execution instance is a single call to DESCENT, i.e., autorotation is modeled for a single combination of the design space variable values over the entire H/V domain) to a relationship between the $P_{k|d}$ output and its criteria. Initially, valid $P_{k|d}$ criteria will be limited to the vertical impact velocity, called critical velocity,

$$P_{k|d} = g(v_{crit}). \quad (3)$$

This function will be derived from data similar to figure 5. In order to do this automatically, nonconvergent values would have to be eliminated or intelligently modified; otherwise, the data-reduction algorithm will overestimate the prevalence of more severe kill categories. The coefficients of function *g* are then stored in the cell $[x_1, x_2, \ldots x_n]$ for its design space variable values in an *n*-dimensional matrix.

For full-factorial runs, each combination of design space variables is executed from the beginning and the $P_{k|d}$ matrix is fully populated. A single instance of data-fitting is done with all the values of *g* known throughout the design space. For a DOE-based run, only a small subset of the possible combinations is executed. Data-fitting is done and tested for variance from the data and excessive nonlinearity. If warranted, more data points are judiciously selected for additional measurements and the process is repeated. For both full-factorial and DOE-based strategies, the end product is the data-fit function mentioned in section 5.1,

$$P_{k|d} = f(v_{crit}, x_1, x_2, \ldots, x_n). \quad (4)$$

It is expected that traditional contour plots will best visualize the $P_{k|d}$ trends. Thus, *n*-1 variable values (including the desired critical velocity) must be specified by the analyst at the time of plotting. The remaining two variable ranges are the x and y coordinates of the plot.

With *f* calculated in advance, the analyst can change specified values interactively to see how the $P_{k|d}$ trend changes at different locations in the design space.

Ideally, this will be presented in the context of a graphical user interface (GUI), or similar application that is sufficiently quick-running, to provide a responsive, interactive experience (figure 18). While using the GUI, the analyst will have access to assumed constant parameter values—such as nominal rotor speed—for reference. One additional option is to save a representative subset of H/V diagrams from the iterative DESCENT executions. This would provide insight into where in the analysis region the kills that constitute the $P_{k|d}$ are coming from, but possibly at the cost of excessive data storage requirements. If the H/V diagrams are made available, it is likely that they would be called on by selecting the contour plot. The locations on the contour plot where detailed H/V data are available will appear when the contour plot is selected. The H/V diagram (such as figure 1b) would be shown elsewhere in the GUI, or in a separate window.

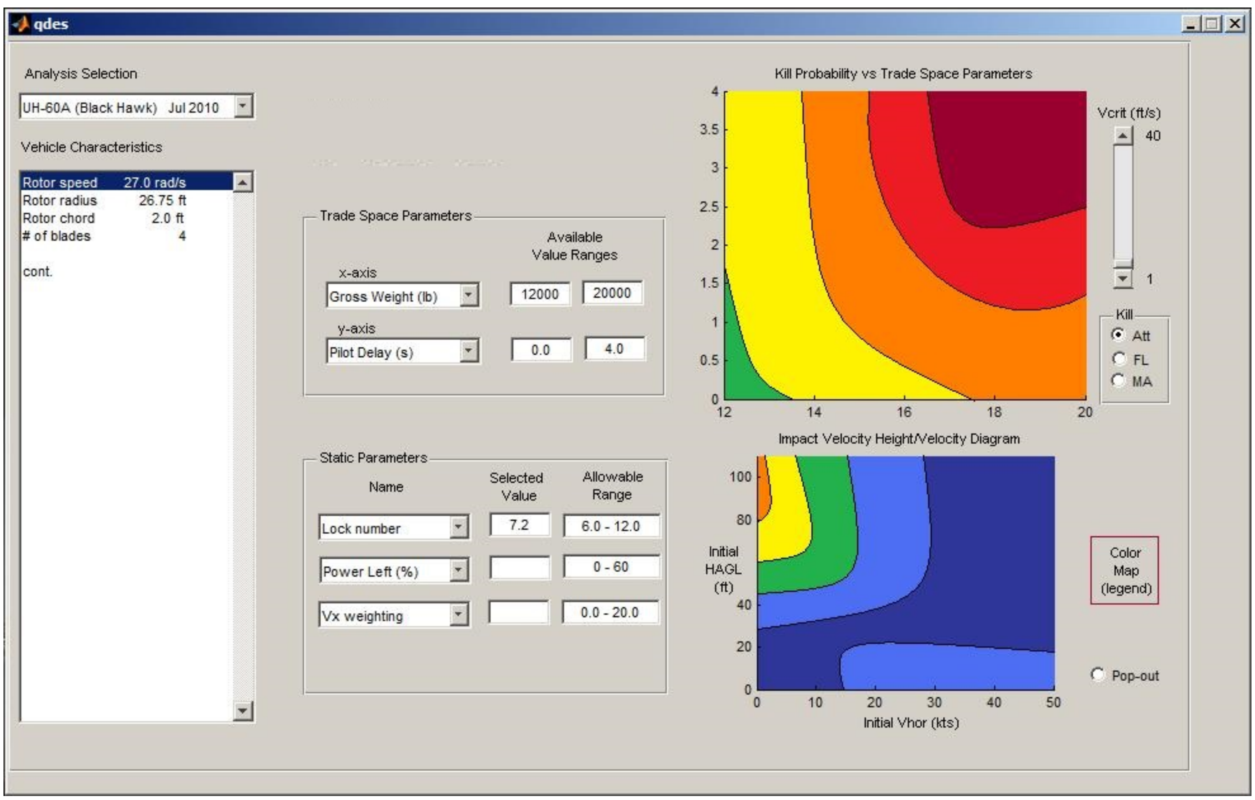

Figure 18. Notional GUI for querying the design space, also referred to as a “trade space.”
Note: The analyst inputs a single value for the static parameters (bottom, middle column) and a max/min pair for the variable parameters (top, middle column), that will serve as the axes limits of the surface plot. That plot appears on the top right. Bottom right is the H/V diagram at a selected point. Parameters outside of the design space (and thus not adjustable by the analyst) are listed on the left side menu.

7.3 Application Examples

In order to demonstrate the capabilities of design space modeling, consider the hypothetical example of a platform with three competing engine choices. Option A, the incumbent engine; option B, increases available power by 50 hp, but weighs 200 lbs more; and option C, decreases weight by 200 lbs, but requires additional manual control actions that effectively add one second to pilot reaction time. From a survivability perspective, it is not immediately obvious which engine is the most advantageous choice.

Currently, DESCENT would be executed for each set of conditions—depending on the density of the H/V domain grid, a run might take several hours or more—and the $P_{k|d}$ output used for survivability comparisons. However, since all of these scenarios are reasonably close to the incumbent design, a design space might have been created ahead of time that encompasses all three options. In that case only the platform's $P_{k|d}$ data-fit function would need to be loaded in the analysis GUI. With available power and gross weight selected as the contour plot axes, the output might look like figure 19.

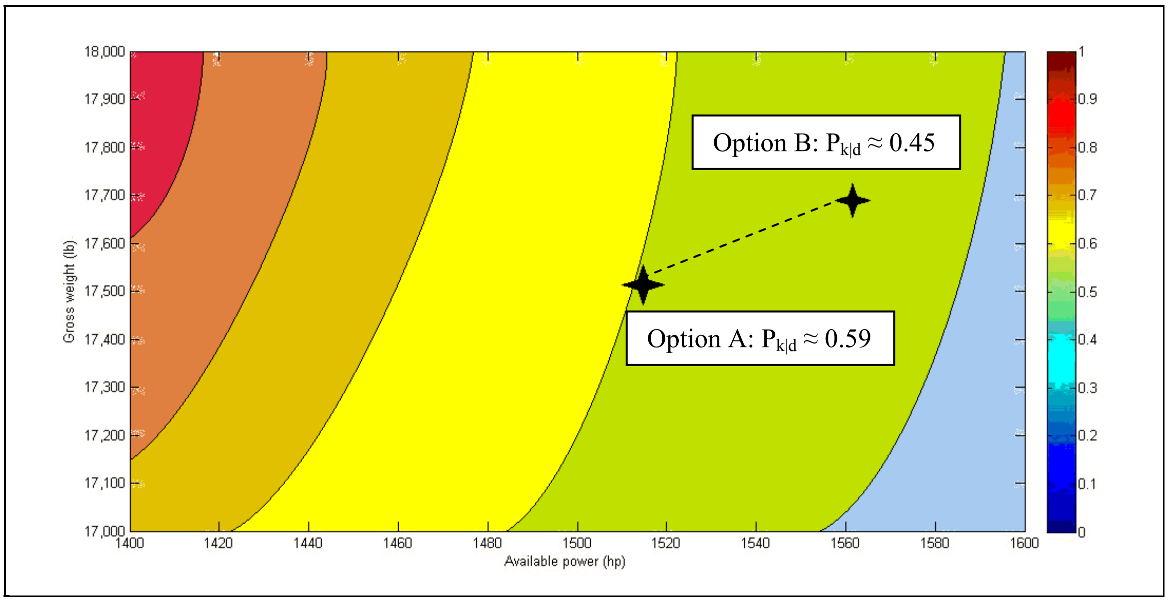

Figure 19. Engine options A and B on the $P_{k|d}$ vs. power and weight surface plot.
Note: The color bar on the right refers to the local $P_{k|d}$ value. Static parameter (e.g., rotor inertia) values are not shown.

It is evident from the plot that the more powerful, but heavier, engine is slightly more survivable than the incumbent engine. Now, to compare options 2 and 3, the axes are switched to pilot delay time and gross weight, and the specified parameters are changed to match the third scenario. The output plot might look like figure 20.

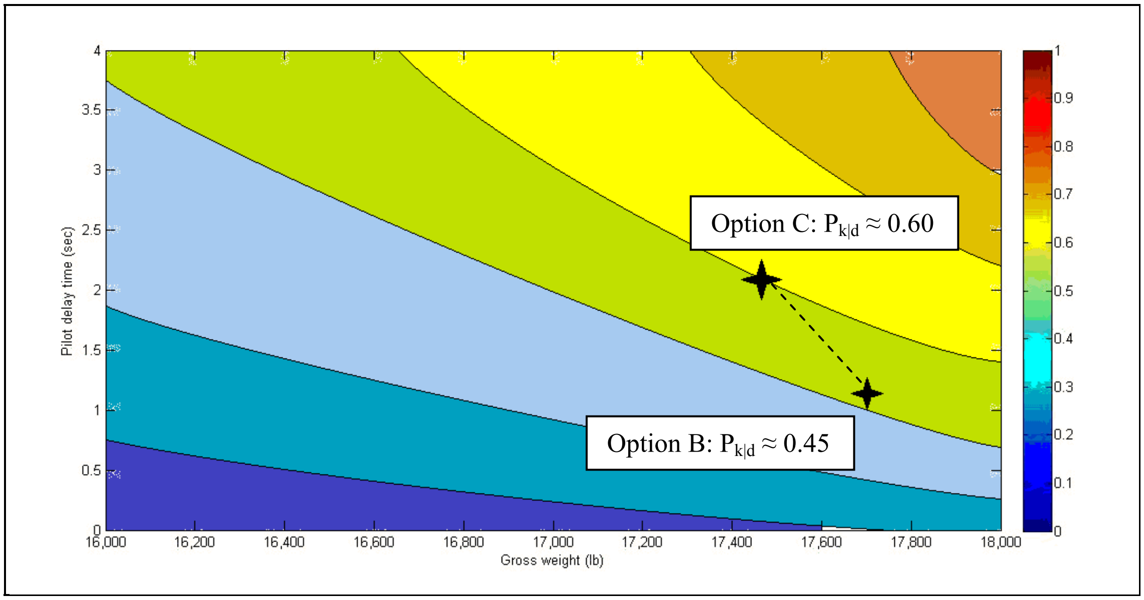

Figure 20. Engine options B and C on the $P_{k|d}$ vs. delay and weight surface plot.
Note: Option A no longer appears because horsepower becomes a static parameter (set by the user in the background) when pilot delay becomes a variable parameter, and the value chosen for comparing B and C is not applicable to A.

Therefore, option C is less advantageous than option B, having a $P_{k|d}$ closer to 0.6 as opposed to the latter's 0.45. The solution is apparent almost immediately. This, by itself, constitutes a marked analysis efficiency improvement, but larger gains accrue when more questions are asked. As examples:

- How much would the landing gear need to be improved to bring option A into survivability parity with option B? To answer, start with figure 19. Slowly change the value of v_{crit} (perhaps by manipulating a slider on the GUI) until option A's new $P_{k|d}$ matches the original value for option B—the impact velocity the better gear needs to withstand.
- All else equal, what is the effect of a high-energy rotor system design? To answer, use *rotor inertia* and *gross weight* as the design space axes. Weight will climb slightly, but inertia should improve dramatically as the mass of the rotor blades moves outward. This tool should be able to show if (and where) it is a positive tradeoff for survivability.
- Does switching to fly-by-wire make sense? Automating the initial aspects of the autorotation via feedback loops from the rotor to the engine will reduce pilot delay time, but add weight and complexity to the power system. The delay penalty is especially important near the ground or near the autorotation boundary, so more detailed output is needed. In this case, call the H/V diagram (figure 18, bottom right) at design space points corresponding to standard and reduced pilot delay values. If the vehicle is unlikely (because of mission requirements) to be in a region where the predicted outcome is affected by a lower pilot delay value, the improvement may not be advantageous.

Since many characteristics of the vehicle are "on the table" at one point or another of the design process, it is desirable to make the design space analysis tool as flexible as possible. The limit on how many dimensions can comprise the design space is the feasibility of processing an exponentially increasing number of DESCENT cases. However, with some lead time, an extra variable could be added to an existing space, especially if one or more standard variables could be converted to assumed constants. This sort of *ad hoc* space creation provides a balance between comprehensive flexibility and resource limitations.

8. Conclusions: The Integrated DESCENT Application

DESCENT is planned to be configured as a software suite for releases beginning in about FY16. The package should include:

- A "wrapper" code. This will be the executable for the overarching package. It launches a GUI that receives analyst input and calls one or more of the other scripts.
- Design space visualization tool. This is a graphics window for observing output from the design space extension. It will display previously created design spaces without calling DESCENT, or if a new space is required, it will interact with the core code.
- Occupant outcome tool. This is a GUI that allows the user to choose parametric study expressions (section 6; input obtained separately) for objective function refining. It will display unmodified DESCENT output and occupant-optimized output for comparison purposes. It will interact with the core code.
- DESCENT core code. This is the optimization code currently in use. It will be callable from the wrapper code or executable independently. It outputs the graphs discussed in section 4 for S/V analyses.
- Input/output visualization tool (possible). This is simply a window for reading text-based input and output files in a user-friendly way. Features would include explanations of the terms and plotting shortcuts.

The typical user will execute the wrapper code and choose either a single-execution analysis (the current DESCENT capability), review of an existing design space, creation of a new design space, or a single-execution analysis with occupant outcome optimization. The GUI will direct the user to enter appropriate input data (file names) and the code will output data as appropriate. Because some processes take significant time to complete, an effective status meter will be more important in this integrated code than it is currently.

Note that both the design space and occupant outcome extensions are intrinsically linked to the core code. This creates the data-flow dependence shown in figure 21, where both application extensions rely on DESCENT output for their functionality, and in turn provide feedback to the core code as necessary. The simulator data extension, as a validation and "tuning" tool, operates independently of DESCENT, but informs its parameter values.

When fully extended, DESCENT is planned to have significantly greater capabilities and applications than it does today. It will be relevant for S/V assessments throughout the design process and will be able to compare its predictions to the tendencies borne out in simulators with more sophisticated aerodynamic models.

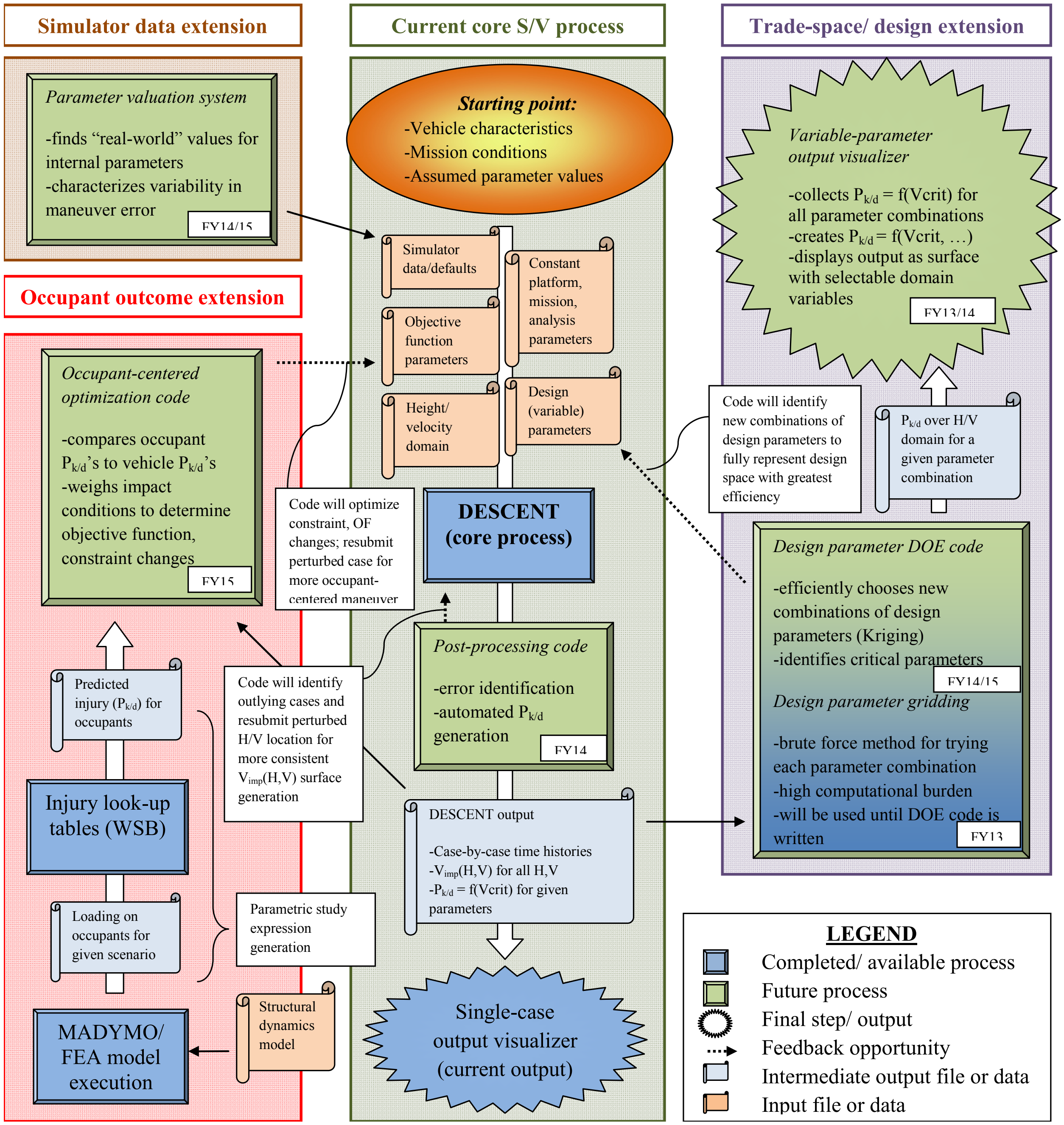

Figure 21. DESCENT data flow diagram including proposed extensions.

INTENTIONALLY LEFT BLANK.

Appendix A. DESCENT Input Template

This appendix appears in its original form, without editorial change.

```
!* * * * * * * * * * *
!DESCENT Helicopter Autorotation Model
!Input File
!* * * * * * * * * * *

!* * * * * * * * * * *
!Variable listing & value assignment section

!Execution case labeling (*note 1)
&descent
CaseName = 'testpkd',             !Case name
CaseLabl = 'testpkd',

!Definition of analysis domain and critical velocity (*note 2)
AMin = 140.,               !Altitude min/max (ft)
AMax = 160.,
Ainc = 10,                 !Altitude increment (ft)
VMin = 0.,                 !Velocity min/max (kts)
VMax = 20.,
Vinc = 20,                 !Velocity increment (kts)
Vcritical = 10.,                  !Critical velocity (ft/s)
stepsize = 0.1,            !Initial time-wise stepsize (s)
showgui = 1,               !Show progress GUI? (0/1 = no/yes)
showplots = 1,                    !Show output plots?

!Definition of special regions (boxes) within domain
VLSmin = 0,                !Low/Slow box (kts, feet)
VLSmax = 40,
ALSmin = 0,
ALSmax = 100,
VHFmin = 80,               !High/Fast box
VHFmax = 160,
AHFmin = 100,
AHFmax = 600,

!Helicopter blade and rotor characterization (*note 3)
helo%c = 2.,                   !Main rotor blade chord (ft)
helo%Nb = 4,                   !Number of MR blades
helo%omega = 27.004,           !Nominal (goal) rotor speed (1/s)
helo%R = 26.75,                !MR radius (ft)
helo%irotor = 5700.,           !MR inertia (slug-ft^2)
helo%a = 6.38,                 !MR blade lift-curve slope (Cl/deg)
helo%cd0 = 0.008,              !MR blade min Cd
helo%ctsigmas = 0.16,          !MR blade Ct/sigma at stall
helo%z0 = 16.67,               !MR hub height above landing gear (ft)
helo%kappa = 1.15,             !MR inflow coefficient (est.)
helo%ns = 30,                  !Rotor stall sharpness (arb.)
helo%ss = 10,                  !Rotor stall scale (arb.)
helo%ctsigdotmax = 0.16,       !Max d/dt of Ct/sigma (1/s)
helo%alphadotmax = 0.5,        !Max d/dt of alpha (rad/s)
helo%alphamax = 0.5,           !Max MR TPP pitch displacement (rad)
helo%ctsigmax = 0.18,          !Max Ct/sigma (hard constraint)
helo%ctsigmin = 0.01,          !Min Ct/sigma (hard constraint)
helo%omegamin = 0.7, 0.5,  !Min MR speed (norm.) at beginning, end
helo%omegamax = 1.1, 1.05, !Max MR speed (norm.) at beginning, end

!Helicopter drag and weight balance characterization (*note 3)
helo%Ax = 32.7,                !Horizontal drag area (ft^2)
helo%Az = 272.,                !Vertical drag area (ft^2)
helo%W = 18000.,               !Gross weight (lb)
```

```
!Mission and analysis conditions (*note 4)
opcond%rho = 0.00193,         !Air density (*note 5)
opcond%g = 32.174,            !Gravity (slug-ft/s^2, constant)
opcond%vc = 0,                !Initial vertical velocity (ft/s, + down)
opcond%Wx = 0.1,          !Weighting factor (*note 6)
opcond%hp0 = 0,           !Total power available after loss (hp)
opcond%nt = 200,          !Number of timesteps
opcond%delay = 0.0,       !Pilot reaction-time delay (s)
opcond%taueng = 1.15,            !Torque decay constant (higher = faster)
opcond%const_list = 0,           !Constraint set choice (*note 7)
opcond%engratio = 0.5,           !Amount of power remaining (*note 8)

!Definition of initial conditions other than nominal (*note 9)
opcond%setalpha0 = 0,            !"set" vars: 0 = no, 1 = yes
opcond%setct0 = 0,
opcond%setomega0 = 1,
opcond%alpha0 = 0.0,             !Same units as alpha, ct, omega
opcond%ct0 = 0.0,
opcond%omega0 = 21.2,

!Objective function weighting factors (*note 10)
weighting%avf = 0.0,             !Vertical velocity (%avg=1 as const.)
weighting%auf = 0.0,             !Horizontal velocity
weighting%aug = 1.0,
weighting%atf = 1.0,             !Fuselage pitch
weighting%atg = 0.0,
weighting%awf = 1.0,             !MR speed
weighting%awg = 1.0,
weighting%adf = 1.0,             !Rate smoothing
weighting%adg = 1.0,
weighting%asf = 0.0,             !Miscellaneous (likely should be zero)
weighting%asg = 0.0,

!Autorotation maneuver targets (*note 11)
udesc = 0.0,              !Steady-state descent velocity (horiz.)
vdesc = 0.0,              !(vert.)
fus_pitch = 0.0,          !Desired fuselage pitch at impact
!* * * * * * * * * * *
```

Documentation section (notes 1-11) redacted for brevity but is available upon request. Objective function weighting values (variables weighting%...) refer to α rows of the chart in appendix B.

INTENTIONALLY LEFT BLANK.

Appendix B. Objective Function Terms and Values

Objective function terms are grouped by three parameters. Designated in the first column, *f* terms are evaluated throughout the flight path and integrated for a total value; *g* terms are evaluated only at the impact condition (final time step). In the second column, α terms refer to the global relative weighting (priority) of the term compared to others; γ terms modify the weighting of *f* terms on a time-wise basis throughout the flight path; β terms describe the underlying quantity being evaluated. Designated in the second row, each subscript (*u*, *v*, *t*, *w*, *d*, and *) refers to the vehicle states described above them.

For example, the input variable *weighting%atf* refers to the objective function term with parameters *a* (α), *t* (θ), and *f*. This is the global weighting of fuselage pitch during the maneuver.

Table B-1. Suggested weighting table for DESCENT objective function.

Quantity		**Vertical Velocity**	**Horizontal Velocity**	**Fuselage Pitch**	**Main Rotor Speed**	**Smoothing**	**Misc.**
Label		J_v	J_u	J_θ	J_ω	J_Δ	J_*
f	α	0.01			0.2	0.1	—
	γ	$\sqrt[3]{cos(\pi(\zeta - 0.5))}$		$20\,\zeta$	$(1-\zeta)\sqrt[4]{\mu} + \zeta\,\mu^2$	1	
	β	$\lvert v - v_{desc}\rvert$	$\lvert u - u_{desc}\rvert$	$\lvert\theta\rvert$	$^{\omega}/_{\omega_{nom}}$	$\left(\frac{\varphi_{i+1}+\varphi_{i-1}-2\varphi_i}{2\cdot\Delta_{\varphi max}}\right)^4$	
g	α	1	W_x	0.2	—	—	—
	β	v	u	$\lvert\theta\rvert$			

Notes: ζ is the nondimensionalized time variable, i.e., the percentage of the total maneuver time elapsed, as elsewhere. μ, which appears in the definition of $\gamma_{\omega f}$, is defined as $\mu = \left(\frac{cos(\pi\zeta\ )}{2} + 0.5\right)$. φ, which appears in the definition of $\beta_{\Delta f}$, is the pitch setting denoted by θ in “DESCENT Smoothing Function” (19 July).
The symbol was changed to avoid confusion with the fuselage pitch variable. Dashed entries denote quantities that do not appear in the objective function at this time. They can be coded as α=0. J_* is a placeholder for terms added in the future.

Appendix C. Sample DESCENT Output

DESCENT's text-based output is comprised of the following sections:

Header, including input data

```
'Namelist after reading:'
&DESCENT
 HELO%AX=  79.10000    ,
…
```

Individual case histories

```
**********************************
 Begin case    1 of  299
 Altitude:  10.0, Airspeed:   0.0
 **********************************

 'Insufficient power for flyaway, attempting forced landing.'


************************
Output of State Variables
************************

   Time      Vz        Vx     Omega      Z     Ct/sigma  Alpha    Ct-dot  Alphadot
Pengine
     0.00     0.00     0.00   99.931     10.0  0.10936     0.00  0.00000     0.00
28552.
     0.01     0.00     0.00   99.755     10.0  0.10936     0.00  0.00000     0.00
28614.
     0.02     0.00     0.00   99.579     10.0  0.10936     0.00  0.00000     0.00
28676.
…
```

Summary table

```
***************************
Summary of Impact Velocities
***************************

 Case   Initial Altitude   Initial Velocity   Impact Velocity
    1           10.00000            0.00000          14.29615
    2           10.00000            5.00000          14.23636
    3           10.00000           10.00000          14.09701
    4           10.00000           15.00000          13.86358
    5           10.00000           20.00000          13.48060
…
```

Appendix D. Post-Processing Script

This appendix appears in its original form, without editorial change.

```
%* * * * *
%Script to convert new DESCENT output to Pkd and Charts
%* * * * *
%enter the file name of the output file (must be in the same directory) and
%find summary of impact velocities at the end of the file
fid=fopen('mo3.out');
line=fgetl(fid);
k=strfind(line,'Summary');
while (isempty(k)) && (~feof(fid))
    line=fgetl(fid);
    k=strfind(line,'Summary');
end;
waste1=fgetl(fid);
waste2=fgetl(fid);
waste3=fgetl(fid);
%read in and compute values
%  A is matrix of values read directly from file (#, h0, v0, v_imp)
%  B is A with case numbers stripped out (A(2:4))
%  Bpk is B with v_imp replaced by Pk (boolean of v_imp > criterion)
i=1;
while ~feof(fid)
    A(:,i)=str2num(fgetl(fid));
    i=i+1;
end;
B=A(2:4,:)';
Bpk=A(2:3,:)';
%criterion=0 for Bpk -> FL vs MA, criterion=vc for Bpk -> Att vs FL
criterion=0;
for i=1:length(Bpk)
    if B(i,3)>criterion
        Bpk(i,3)=1;
    else
        Bpk(i,3)=0;
    end;
end;
%plotting section
%  xvals, yvals are unique values of h0, v0 for creating plot axes
xvals=unique(B(:,1));
yvals=unique(B(:,2));
%  zvals is impact values reshaped to m x n format for surface plot
v_imp_surf=reshape(B(:,3),length(yvals),length(xvals));
%  Figure 1 is a surface plot of impact velocities
figure(1)
surf(xvals,yvals,v_imp_surf)
%  Figure 2 is a scatter (point) plot of impact velocities
figure(2)
scatter3(A(2,:),A(3,:),A(4,:),40,A(4,:),'filled')
%  Figure 3 is a scatter plot of Pk contributions (1 or 0)
figure(3)
scatter3(Bpk(:,1),Bpk(:,2),Bpk(:,3))
%  Figure 4 plots Pk vs vc for sensitivity analysis
impact_list=sort(A(4,:)');
num_cases=length(impact_list);
impacts=[impact_list zeros(num_cases,1)];
for i=1:num_cases
    impacts(i,2)=i/num_cases;
end;
figure(4)
plot(impacts(:,1),impacts(:,2))
```

List of Symbols, Abbreviations, and Acronyms

ARL	U.S. Army Research Laboratory
ASEIL	Aircraft Survivability Equipment Integration & Digital Simulation Laboratory
CERDEC	Communications-Electronics Research, Development, and Engineering Center
DOE	design of experiments
FEA	finite-element analysis
GUI	graphical user interface
HAGL	height above ground level
H/V	height/velocity
NAVAIR	Naval Air Systems Command
$P_{k/d}$	kill probability given damage
SLAD	Survivability/Lethality Analysis Directorate
S/V	survivability/vulnerability
V&V	verification and validation

NO. OF COPIES	ORGANIZATION
1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDFS)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL RDRL IMAL HRA RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	USARL RDRL SLE R FLORES

ABERDEEN PROVING GROUND

1 (HC)	DIR US ARMY EVALUATION CTR HQ TEAE SV P A THOMPSON 2202 ABERDEEN BLVD 2ND FL APG MD 21005-5001
4 (2 HC 2 PDFS)	DIR USARL RDRL SL J BEILFUSS P TANENBAUM RDRL SLB A M PERRY (PDF only) RDRL SLB D A DRYSDALE