AIAA 2002-5587
A Distributed Framework for Probabilistic Analysis

D. J. McCormick
Boeing Phantom Works
Huntington Beach, CA

J. R. Olds
Space Systems Design Laboratory
Georgia Institute of Technology
Atlanta, GA

2002 AIAA/ISSMO SYMPOSIUM ON MULTIDISCIPLINARY ANALYSIS AND DESIGN OPTIMIZATION

Sept. 4-6, 2002
Atlanta, GA

A Distributed Framework for Probabilistic Analysis

Dr. David J. McCormick[†]
Boeing Phantom Works
Huntington Beach, CA

Dr. John R. Olds[††]
Space Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0150

ABSTRACT

Probabilistic Multidisciplinary Design Optimization promises to incorporate critical design uncertainty in order to create optimal products with a high probability of meeting design constraints under a wide variety of circumstances. Several methods of accelerated probability analysis are available to designers. What is not available is a formal method for tying contributing analysis-level probability analysis into an integrated design framework capable of optimization. This would allow probability methods to be tailored to the characteristics of a particular contributing analysis as well as potentially reduce the dimensionality of the problems considered. This research presents such a method, then tests it on a conceptual launch vehicle design problem.

This probabilistic optimization problem consisted of 84 noise variables and 4 design variables. This problem setup consistently found system optimums in 6-8 hours. It utilized several probability approximation methods run in an iterative manner to generate probabilistic vehicle sizing information. Once the probabilistic optimum was identified and confirmed using this process, a system-level Monte Carlo random simulation of the vehicle design was conducted around the optimum point to confirm the accuracy of the distributed approximation method. Because this simulation was prohibitively expensive, it was only conducted at the single optimum point. Following this accuracy confirmation, a comparison to a deterministic optimization of the same problem illustrated the difference between the probabilistic and deterministic optimums.

NOMENCLATURE

A_E	exit area
AR	area ratio
DPOMD	Discrete Probability Optimal Matching Distribution
DSM	Design Structure Matrix
GLOW	gross liftoff weight
I_{sp}	specific impulse
MER	mass estimating relationship
MR_{avail}	mass ratio available from vehicle
MR_{req}	mass ratio required by mission
OML	outer mold line
P/W_e	power-to-weight ratio of engine
P_{ch}	chamber pressure
r	oxidizer to fuel ratio
RLV	reusable launch vehicle
RSE	response surface equation
SSME	Space Shuttle Main Engine
SSTO	single stage to orbit
STS	Shuttle Transportation System
T/W_e	thrust-to-weight ratio of engine
T_{SL}	thrust at sea level condition
T_{vac}	thrust at vacuum condition
VB	Visual Basic
WBS	weight breakdown structure

[†] - *Senior Engineer/Scientist, Member AIAA.*
[††] - *Associate Professor, School of Aerospace Engineering, Senior member AIAA.*

INTRODUCTION

Recent trends in aerospace conceptual design have lead to the use of distributed computational models covering multiple disciplines. Unfortunately at the level of conceptual design, many simplifying assumptions must be made because of the lack of design maturity. The problem with this lies in that current computational models are for the most part deterministic. They do not allow a designer to express uncertainties from such sources as user assumptions, computational model error and physical unknowns. Methods are therefore sought to use these distributed multidisciplinary deterministic models to generate probabilistic designs that can more accurately express the future performance of a concept.

At the same time, these methods should be compatible with optimization. To this end, this paper presents a <u>formal method for optimization that ensures a true optimum in the face of uncertainty while at the same time retaining traditionally distributed analyses</u>. This will allow disciplinary experts to retain an integral role in all aspects of model creation. Therefore, the experience of the expert in probabilistic simulation of a particular discipline can be fully utilized to generate accurate and efficient estimates.

The distinguishing feature of this new method is that it performs uncertainty calculations at the contributing analysis level. By reducing the communications requirements and lowering the dimensionality of the local uncertainty analyses, the distributed method has advantages for many classes of multidisciplinary design problems.

<u>Research Goals and Objectives</u>

The overriding goal for this activity is to introduce a new formal method for distributed probabilistic conceptual launch vehicle design. An important aspect of this is to demonstrate probabilistic analysis at the contributing analysis level, therefore reducing the dimensionality of each of the analyses. This demonstration will also reveal how the methods can be tailored to the particular problems, according to the experience and knowledge of disciplinary experts. The following list of goals and corresponding objectives was the guide for this research:

- Demonstrate a distributed probabilistic multidisciplinary framework.

This distributed philosophy of multidisciplinary design optimization has many advantages (Refs. 1-4). Among these are local ownership of analyses, compatibility with existing analysis infrastructure and distributed computational effort. If the goal of demonstration of a distributed probabilistic framework for conceptual launch vehicle design optimization is met, then all of these advantages will be enjoyed. At the same time, probabilistic information, which is crucial at the conceptual design phase will be brought into conceptual launch vehicle optimization.

The demonstration of the conceptual launch vehicle design framework will include a detailed account of the procedure to construct it. It will include any problems or hazards encountered and assist those who wish to create similar problem setups for other conceptual design problems. In this light, ease of setup is a high priority. Problem setup should be on the order of or less than two man-weeks. Anything longer than this amount of time represents a significant investment and is unlikely to be implemented by industry.

Another objective used to measure the success of this demonstration will be the amount of probabilistic analysis able to be undertaken at the contributing analysis level. The degree to which probabilistic analysis can be done at this lower level will indicate how distributed the computational effort and responsibility has become. The objective for success was the creation of design framework that performed all significant calculations at the contributing analysis level.

- In this framework, multiple heterogeneous computer platforms will be utilized on a conceptual launch vehicle design problem.

By utilizing analysis integration software packages, a repeatable framework for analysis across heterogeneous platforms will show that application of this method is possible in established engineering environments with several, discrete analyses running on separate platforms.

This demonstration will show that a heterogeneous platform approach is feasible in combination with this probabilistic framework. Success will be measured here by utilizing more than one computing platform in the conceptual launch vehicle example problem.

- This distributed probabilistic framework should have a significant computational expense savings when compared to Monte Carlo simulation.

To make this method competitive with other options for system-level probabilistic analysis, a two or greater order of magnitude improvement in speed when compared to a Monte Carlo simulation is required. This will ensure that the analysis method is competitive in terms of other options.

- Optimization should be able to be completed in a reasonable amount of time.

To ensure that this method is applicable in real engineering situations, an entire optimization should be able to be completed overnight. This should simplify manpower tasking while waiting for results, as another task does not need to be found for the engineer if the optimization can be run during off hours.

- Optimization using this method should be repeatable.

To measure the success of this goal, several confirmation optimizations of the primary optimization beginning with different initial guesses must find the same optimum point. Success here is all of the confirmation optimizations finding the same answer.

- The distributed probability approximations should arrive at accurate values.

This objective will be measured against a confirmation Monte Carlo simulation. The approximate framework should be no more than 5% off on the important problem constraints and objective function evaluation. The other secondary output parameters should also have errors within the calculated error bound for the Monte Carlo simulation.

- Uncertainty sources for conceptual launch vehicle design will be identified and reasonable distribution assumptions will be made.

The goal of uncertainty identification is crucial to an accurate representation of the conceptual launch vehicle design problem. This will involve identification of the sources of uncertainty, both environment and human based. Therefore, this research should identify and quantify as many open-source launch vehicle uncertainty parameters as is possible. These sources can then be represented by appropriate input distributions and included in the conceptual launch vehicle example problem. This work should be helpful to future launch vehicle designers who wish to include a formal mechanism to account for modeling uncertainties. However, most practicing organizations should be able to better quantify these uncertainties than this research given the competitive nature of many of the quantities of interest and the experience of most manufacturers.

Techniques

This method of multidisciplinary design optimization is best described as a set of requirements on the variable communication between disciplines as well as the accuracy of the results from each of the analyses. To be a valid multidisciplinary analysis technique, the proposed method must have certain characteristics related to passing accurate information between the disciplines. In addition to accurate data passing, it must also ensure that all the disciplines have all the required inputs for analysis. For deterministic analysis, this is a relatively simple task. When translating this to probabilistic analysis there are some important considerations. Finally, there should exist some capability for handling objective functions from several disciplines. This capability should also exist in deterministic design frameworks, but probabilistic optimization introduces new possibilities, as described in the background chapter, that require more computational effort than most deterministic multi-objective formulations.

The requirement of accurate variable communication between the contributing analyses meant the transfer of data should include enough distribution information in the coupling variables to give each contributing analysis a good idea of the probability map exiting the other disciplines as they occurred together. This means that at a minimum, second order moments should be transferred. To do this, a set of standard deviations for each variable along with a correlation coefficient for each variable combination that is an input to any contributing analysis. The correlation is required because these variables are not assumptions, they are fits to the multivariate probabilistic solution of another analysis. In many cases, these outputs are highly correlated and ignoring this information would lead to faulty results for subsequent contributing analyses.

This correlation information requirement means that the inputs and outputs of a probabilistic multidisciplinary analysis problem will be slightly different from that of a deterministic one. In order to generate correlation coefficients required by other analyses, extra variables need to be added to certain contributing analyses that do not require them for deterministic optimization. The Design Structure Matrix (DSM) in Fig. 1 illustrates this point. Contributing analyses for propulsion and trajectory both feed multivariate normal distributions into the mass properties analysis, but a correlation between propulsion and trajectory cannot be generated in this situation, yet it exists and is an important input to the mass properties analysis. Therefore, an extra link is created between propulsion and trajectory so that the propulsion analysis can calculate the required correlation between the output of the trajectory contributing analysis and its own outputs.

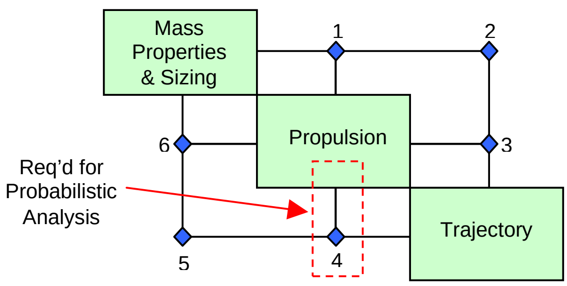

Figure 1 – Launch Vehicle Design Structure Matrix

This technique of adding links between disciplines should yield accurate simulation results, assuming that the results of the contributing analyses are in fact normal. This assumption should therefore be tested using a Monte Carlo analysis on each contributing analysis at a typical design point, before optimization begins. The type of output observed by each should then indicate what type of distribution is most appropriate to fit. These variables should still be fit in a multivariate manner, with correlation coefficients recorded and transferred to the other disciplines so that they can be simulated accurately as inputs.

One of the advantages of this type of loosely coupled formulation is that there are minimal requirements placed on the subsystem probabilistic analysis other than accuracy and the ability to handle its own probabilistic constraints. Besides making sure that confidence levels for constraint satisfaction and the coupling variables are consistent within the system, there is very little else the system level implementation must handle. The consistency between coupling variables is ensured by requiring the contributing analysis to provide information for its outputs based on updated information about its inputs. This delegation of authority is one of the primary advantages of the loosely coupled approach.

Example Problem

An example problem using these techniques has been completed. It was a single stage to orbit (SSTO) reusable launch vehicle (Fig. 2) conceptual design optimization problem with a large number of uncertainty variables (83).

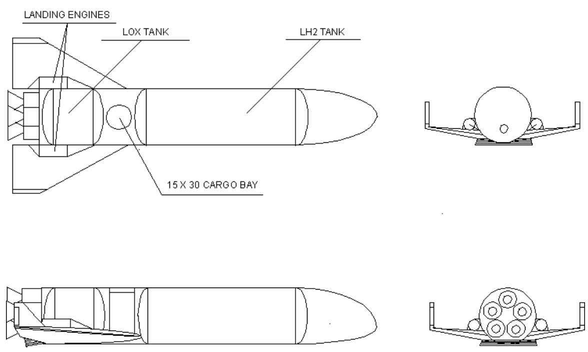

Figure 2 – Single Stage to Orbit Reusable Launch Vehicle

This problem was analyzed using three distinct contributing analyses. These can be seen in design structure matrix in Fig. 1. In a design structure matrix, each link represents an information flow. The links above the diagonal represent feedforward flows, while the links below represent feedback flows.

Table 1 - Coupling Variables for Distributed Probabilistic Launch Vehicle Design

Coupling Location	Parameters Passed
Mass Properties and Sizing – Propulsion (1)	GLOW mean, GLOW std. dev.
Mass Properties and Sizing – Trajectory (2)	GLOW mean, GLOW std. dev., Sref
Propulsion – Trajectory (3)	Tvac mean, Tvac std. dev., Ispvac mean, Ispvac std. dev., Ae mean, Ae std. dev., GLOW-Tvac corr., GLOW-Ispvac corr., GLOW-Ae corr., Tvac – Ispvac corr., Ae – Tvac corr., Ae – Ispvac corr.
Trajectory – Propulsion (4)	MR_{req} mean, MR_{req} std. dev., MR_{req} - GLOW corr.
Trajectory – Mass Properties and Sizing (5)	MR_{req} mean, MR_{req} std. dev.
Propulsion – Mass Properties and Sizing (6)	Tvac mean, Tvac std. dev., Ae mean, Ae std. dev., T/W_e mean, T/W_e std. dev., MR_{req} – Ae corr., MR_{req} – Tvac corr., MR_{req} – T/W_e corr., Ae – Tvac corr., Ae – T/W_e corr., Tvac – T/W_e corr.

The first contributing analysis, mass properties and sizing, determined the weight components and then photographically scaled the vehicle to meet sizing requirements to a desired confidence level. Propulsion similarly analyzed and sized the main propulsion system to match constraints to a given confidence. Trajectory simulated the ascent of the vehicle in order to propellant and loading requirements.

The distribution communication between these disciplines can be found in Table 1.

Mass Properties and Sizing

The mass properties and sizing algorithm was originally constructed using an Excel© spreadsheet. This particular spreadsheet calculated the vehicle weight parameters for a given set of mass estimating relationship (MER) assumptions and a vehicle outer mold line (OML) length. Because these MER's were highly interrelated, fixed-point iteration with no relaxation was used to solve the system. This has proven in this past to be a simple and reliable method for solving these sets of equations.

For this sizing analysis, the vehicle in question had a 20 klb. payload, a 350 fps. orbital maneuvering capability and a five minute powered landing capability provided by a pair of hydrogen turbofans. Also, zero weight growth margin was assumed. This is the traditional safety factor method for accounting for uncertainty in the MER's and performance estimates. Because this job is now done using more advanced probabilistic methods, this margin was no longer necessary. These assumptions were constant and have a significant impact on the size and weight distribution of the vehicle.

A compiled Matlab© function generated the Discrete Probability Optimal Matching Distribution (DPOMD) (Ref. 5) tables in a format that could be read by the previously described Visual Basic (VB) script. This function took text file inputs for the mean, standard deviations and correlation coefficients for the input coupling variables and a separate file for triangular distribution information. This information, along with a reduction factor for the fractional factorial DPOMD method comprised the inputs used to generate the run table.

Embedded in the spreadsheet was a VB macro that read in a run table for the assumptions and input variables, executed the spreadsheet for all the cases, then calculated output parameters such as means, standard deviations and confidence levels. These responses included all the items in the generated weight breakdown structure (WBS) as well as overall variables such as gross liftoff weight (GLOW), vehicle dry weight and mass ratio available (MR_{avail}). These parameters were then sent to cells in the spreadsheet where they would be available to the Analysis Server© wrapper utility.

To bring the contributing analysis into the ModelCenter© environment, two different wrappers were created. The first step was to create a wrapper around the Excel©-based mass properties analysis. This wrapper provided inputs for deterministic variables such as vehicle length, vehicle thrust-to-weight ratio, etc. The output parameters for the probabilistic process were also wrapped during this step, after being calculated by the VB script in the spreadsheet described in the previous paragraph.

The second wrapper provided inputs to the DPOMD analysis in a compiled Matlab© function. This wrapper provided data to the text files for input distribution information. The output from the DPOMD program, however, was not handled by ModelCenter©. DPOMD executable and input files were placed in the same directory as the mass properties spreadsheet so that when the analyses were executed in order, the data was available to the mass properties analysis. This avoided having to send the rather large amounts of data through across the network to the ModelCenter© control panel.

Once both parts of the analyses were wrapped, they were connected together in ModelCenter©. This meant that input distributions could be provided inside ModelCenter, along with a vehicle size and corresponding output parameters could be generated. The next step was to construct the sizing process. This was done using the ModelCenter-integrated version of DOT (Ref. 6). Using the goal seek method, the 80% confidence level on the difference between the MR_{req} and MR_{avail} by changing the vehicle length. This goal seek was only necessary on the mass properties spreadsheet, as none of the input distributions changed with the vehicle length directly. This added the sizing element to the process and completed the mass properties and sizing contributing analysis as it was incorporated here. This analysis was then inserted into the framework described earlier. A screen shot of the completed set of components in ModelCenter© can be seen in Fig. 3.

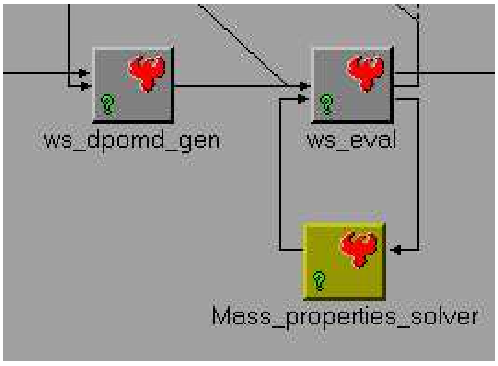

Figure 3 – Probabilistic Weights and Sizing in ModelCenter©

Propulsion

The propulsion analysis consisted of three layers. The innermost layer was the SCORES (Ref. 7) analysis, which sized each engine scenario. Wrapped around this analysis was a Perl script that took inputs from a text table, then executed the table for each scenario listed. The outer layer was a Matlab function responsible for generating the DPOMD runs for the propulsion analysis to execute. This is shown in Fig. 4.

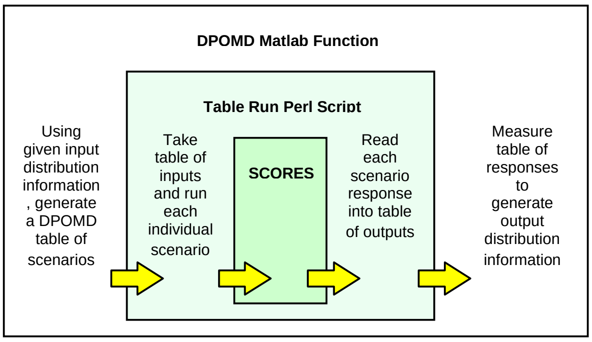

Figure 4 – Overview of Probabilistic Propulsion Analysis

The input table consisted of the inputs for the propulsion analysis. This meant that variables added for the purpose of generating correlations such as the MR_{req} parameters did not need to be included in the list. The variables that were included in the list were Tsl, Pch, P/W_e, AR and r. The last two were deterministic variables, so these did not vary with the runs in the table. For each of the scenarios listed in the table, SCORES constructed an engine with matching sea level thrust and reported the vacuum engine performance and thrust, exit area and whether or not there is a shock in the nozzle.

To generate the table, a Matlab© function producing a full factorial version of the DPOMD method was constructed. This function took distribution information from input files and then generated a DPOMD table corresponding to those inputs, minus those random variables that were added for the purpose of generating correlation coefficients. Once the script was done generating the table, it executed the Perl script described above. Once the Perl script had run SCORES for the required table of scenarios, the output responses were read from a text file of responses generated by the Perl script. Using these responses, distribution information was calculated and placed in another text file. This final text file would eventually be read by ModelCenter© and used as outputs to other contributing analyses.

The final step of including this analysis in the overall system consisted of first wrapping the correlated normal coupling variable distribution parameters for GLOW and MR_{req}, the uncorrelated normal input P_{ch} and finally the deterministic inputs T/W_v, AR and r. The outputs wrapped were the correlated normal coupling variables parameters for I_{spvac}, T_{vac}, A_e and T/W_e plus correlations to selected input variables as listed in Table 1.

Trajectory

The trajectory contributing analysis consisted of a Monte Carlo simulation performed on a response surface of propellant-optimized trajectories. A ten variable on-face central composite experiment with full factorial box points was used. This design had 1,045 runs that took approximately two days evaluate.

$$f(x) = \sum_{alli} \sum_{allj} a_{ij} x_i x_j + \sum_{alli} b_i x_i + c \qquad (1)$$

Once the responses to the experiment were generated, a quadratic function of the form in Eqn. 1. was fit in a least squares sense using the software package JMP© (Ref. 8). For a slightly improved fit, a stepwise regression was done for each response. This technique performs an F significance test (Ref. 8) on the terms of the polynomial to determine which ones are essential to the fit and which ones create numerical error. The essential terms of the polynomial are retained and the others are discarded, ignoring the overall fit. These tests determined which terms of the equation would be used. This process improves the fit of the equation by eliminating terms that only slightly contributed to the response. The fit parameters resulting from this process are given in Table 2.

Table 2 – Results of Stepwise Regression for Trajectory RSE

Fit Parameter	Value
R-Square Value	0.9995
Adjusted R-Square Value	0.9995
Number of Terms Selected	33
Number of Terms Eliminated	33

Once the coefficients of the equation were determined, they were put into a text file that was read by a C++ program designed to evaluate a response surface for a text file list of inputs. Now the probabilistic analysis was ready to be run using Monte Carlo simulation. Because of the inherent batch nature of the RSE evaluation program, this process did not require a Perl script step like the propulsion test. Instead, a Matlab© function that generated a list of inputs from a previously created list of pseudo-random numbers was wrapped directly around the RSE executable.

The random number generation process used a table of random numbers generated offline, then relied on a transform to give the trials the proper distribution characteristics. First, for the normal distributions, samples from independent standard normal distributions were generated offline. This saved the expense of having to perform a costly inverse cumulative probability function call for each variable in each trial. For each subsequent Monte Carlo simulation, this list of standard normal samples was transformed into the required multivariate normal distribution by means of an inverse Hasofer-Lind (Ref. 9) transform.

For the triangular distributions, a more standard approach was taken. Here, a list of uniform [0,1) samples was generated offline. Then, for each different Monte Carlo simulation, an inverse triangular cumulative probability function was used to determine the samples from each of the triangular random variables. This was more time consuming than the normal distribution generation, but there were only two triangular random variables, so this extra expense was not noticeable. In addition, the distribution parameters for triangular distributions did not lend themselves to a simple transform like the normals.

Once the table of random inputs for the RSE program was ready, the program was run and the responses recorded to a separate file. These responses were then read in by the outer Matlab function and the parameters for the output distributions calculated. In this case, the sole output distribution was the MR_{req}, but the correlation of this variable with GLOW was also calculated. These parameters were then written to a text file where they would be easily accessible to ModelCenter©. This process is illustrated in Fig. 5.

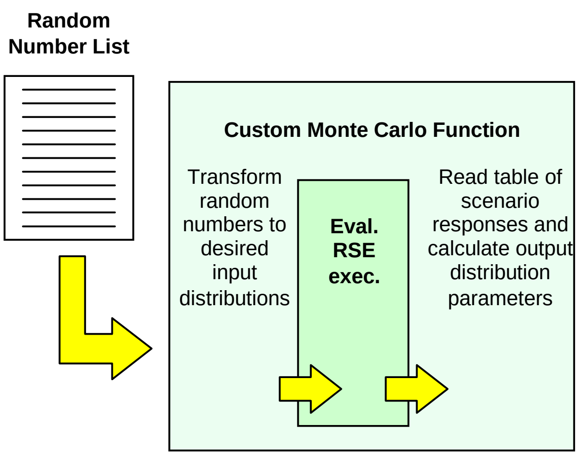

Figure 5 – Overview of Probabilistic Trajectory Analysis

This high dimensionality in the random variables space, along with limited communication requirements between the contributing analyses made this problem an ideal candidate for distributed probability analysis.

This problem was then optimized using the DOT© (Ref. 6) gradient-based optimization tool.

RESULTS

Optimization

To test the feasibility of these distributed probability methods to generate a probabilistically optimum launch vehicle, an optimization of the launch vehicle design system described in the previous section was undertaken. This optimization altered four design variables traditionally known to have a large effect on all-rocket SSTO reusable launch vehicle (RLV's). These were:

- Engine Mixture Ratio (r) – This variable is the ratio of oxidizer mass to fuel mass burned in the engine. This variable primarily affects engine efficiency and propellant bulk density. For optimization, the value of this variable was limited to values from 5 to 7.
- Engine Area Ratio (AR) – This is the ratio of the engine exit area to the engine throat area. It primarily affects how the engine performs with changing altitude. For the system optimization, the value of this variable was constrained to be from 40 to 85.
- Vehicle Liftoff Thrust to Weight (T/W_v) – This ratio determines how much vehicle thrust is present at liftoff. It primarily affects the relative weight of the engines on the vehicle, the trajectory gravity losses and required throttle level on ascent. The limits on this variable were set to be from 1.2 to 1.6. 1.2 was the minimum to ensure safety while clearing the launch tower, while 1.6 was chosen because of throttling concerns near burnout.
- Mean Engine Combustion Chamber Pressure (Pch) – This variable was assumed to be a somewhat controllable noise. The mean value was varied while the standard deviation around that mean was assumed to be a constant 4 atm. This parameter primarily affects engine weight

and performance. It was limited to values from 150 atm. to 210 atm.

Each time these values were altered, the resulting vehicle was sized according to the rules and techniques laid out in the previous section. When this sizing was completed, the objective function, the 95th percentile dry weight, was returned to the optimizer. The particular optimizer used for this process was the DOT package using conjugate gradient unconstrained optimization method.

Because numerical noise from the sizing process was a concern, a finite difference derivative test was undertaken at a particularly difficult point. This point was identified by launching the optimizer using the default settings for finite difference gradients and observing the area where the optimizer could not generate a viable search direction. The defaults for this preliminary search were forward difference gradients with a relative step size of 0.001. This point around which the derivative test was conducted can be seen in Table 3.

Table 3 – Testing Point for Finite Difference Derivatives

Design Variable	Value
Mixture Ratio (r)	5
Area Ratio (AR)	40
Vehicle Thrust to Weight (T/W_v)	1.2
Engine Chamber Pressure (Pch)	210 atm.

Once a trouble point was located, several step sizes using both forward and central differences were taken. These were used to calculate both forward and central difference partial derivative estimates. A sweep of step size can be seen in Fig. 6.

The sweep in Fig. 6 illustrates how the decision on step size was made. The general direction of the large step sizes was considered to be correct. This was assumed because the error here is in the finite difference approximation, which is unlikely to reverse the direction of the estimate for reasonably large derivative values. However, the magnitude of these large step sizes was assumed to be faulty. At the other extreme, the numerical noise in the sizing process can create huge errors in the derivative estimate. So to minimize the error due to both the finite difference approximation and numerical noise, the smallest step size that did not have obvious numerical noise was taken. Because the other three derivative sweeps exhibited similar behavior to the one shown in Fig. 6, a relative step size of 0.005 was chosen. Because this was a rather large step size, central difference gradient estimates were used in place of forward difference. This change made sure that the gradients supplied to the optimizer were as accurate as was possible. It will be shown later that this step size enabled the system optimizer to consistently find the optimum design.

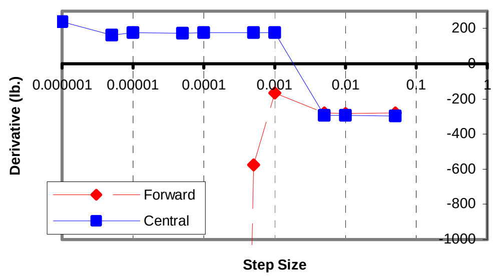

Figure 6 – Step Size Sweep for Derivatives with Respect to Area Ratio

Once the derivative test was complete, the optimization process was started at several points, each one with its own rationale. The baseline point was an SSTO starting out with an engine design similar to the Space Shuttle Main Engine (SSME). This baseline optimization data can be found in Table 4. This baseline point optimized quickly, taking about six hours to find the best solution. An iteration history of the objective function, the 95% confidence level of dry weight, can be seen in Fig. 7.

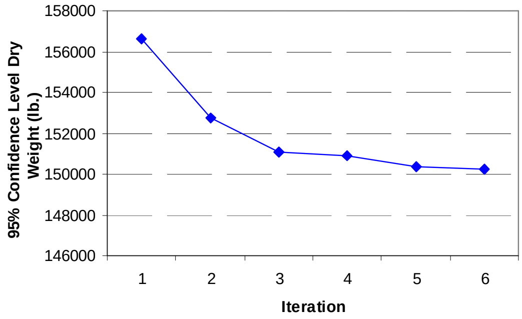

Figure 7 – Objective Function History for Initial Probabilistic Optimization

The design variables in this case went to optimum settings common for this type of problem. The area ratio went to a lower value, from 77.5 to 53. This is to be expected, since the engines must provide the entire vehicle thrust from the pad to orbit. This differs from the Shuttle Transportation System (STS) in that the STS delivers the majority of its main engine impulse at higher altitudes. The lower mixture ratio also shows indicated the high demand for engine performance provided by this variable. The other variables did not change very much, instead going to their constraints for this problem.

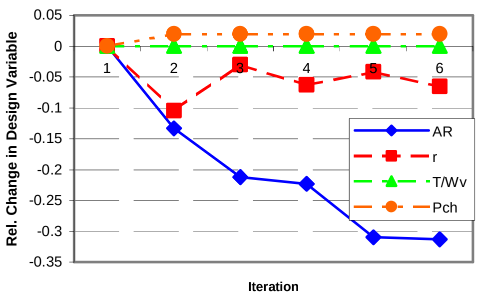

Figure 8 – Design Variable History for Initial Optimization

To confirm that in fact this was the true optimum, two other starting points were run. The first re-optimization began from the engine manufacturer's dream design. This point maximized area ratio for the best vacuum performance, minimized mixture ratio for the highest propellant efficiency, maximized chamber pressure in order to push the limits of power head technology and maximized vehicle thrust to weight in order to sell more engines.

This set of design variables then proceeded to the system optimum found in the initial optimization in about the same amount of time, even though it started further away. The overall comparison of start and end points can be seen in Table 4.

The next point to be optimized came from the standpoint of using a slightly lower propellant performance, but also try to take advantage of a higher bulk density and higher thrust, lower gravity loss trajectory. This corresponded to a low engine chamber pressure, a high mixture ratio, a high area ratio and mid-level vehicle thrust to weight. This optimization also traveled to the same optimum as the others. This fact can be seen in Table 4. Each of these system-level optimizations took six hours to complete on a combination of Windows NT© and SGI Octane© workstations.

Table 4 – Optimum Confirmation Run Results

Design Variable	Start Pt. 1	Opt. Pt. 1	Start Pt. 2	Opt Pt. 2	Start Pt. 3	Opt. Pt. 3
Mixture Ratio (r)	6	5.61	5	5.60	7	5.62
Area Ratio (AR)	77.5	53.2	85	52.1	85	51.3
Vehicle Thrust to Weight (T/W_v)	1.2	1.2	1.6	1.2	1.4	1.2
Engine Chamber Pressure (Pch)	206 atm.	210 atm.	210 atm.	210 atm.	150 atm.	210 atm.
Objective: 95% C.L. Dry Weight	156.7 klb.	150.2 klb.	237.2 klb.	150.2 klb.	286.1 klb.	150.1 klb.

The two design variables that were not limited by constraints generally fell to the same value. While there are slight differences in the optimum results, these all resulted in negligible differences in the 95% confidence level value for dry weight, so they can be considered the same result. To further illustrate the convergence of these two variables, Fig. 9 shows a path iteration history of mixture ratio versus area ratio.

The results in Fig. 9 show that consistent probabilistic optimums can be generated using this distributed probabilistic technique in a reasonable amount of time. This was one of the key goals of the research and the objectives relating to this goal have been met.

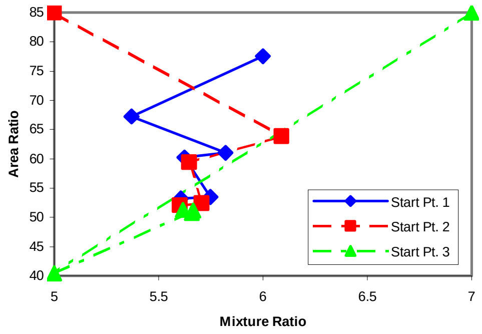

Figure 9 – Design Variable Paths for Area Ratio and Mixture Ratio

Monte Carlo Confirmation

To confirm that the generated probabilistic optimum has been accurately approximated, a Monte Carlo simulation of the optimum point was executed. This involved removing authority over the local noise variables and bringing them into ModelCenter© and applying a central Monte Carlo simulation. This tested the accuracy of the overall process, including all of the approximations made. This simulation consisted of 1,000 trials, all of which had to be run on the heterogeneous framework. It is important to not that this was not an optimization process.

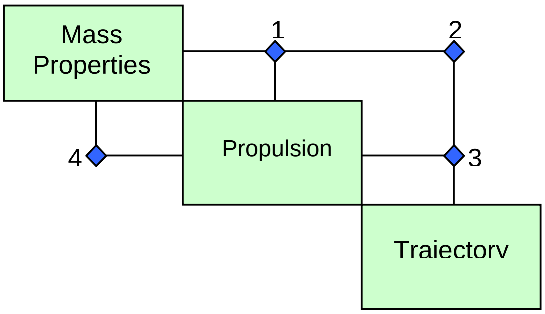

Figure 10 – Design Structure Matrix for Monte Carlo Confirmation

The process for each trial consisted of iteratively sizing the propulsion system with the mass properties analysis, then feeding the resultant scenario to the combination of POST (Ref. 10)/GRAM99 (Ref. 11). This system therefore tested the atmospheric approximation GRAM99 as well as the probabilistic methods. A design structure matrix and coupling table illustrating the confirmation process data flow are shown in Fig. 10 and Table 5.

The coupling variables in Table 5 were all deterministic, since the probabilistic analysis would be handled at the system level. The DSM above was only responsible for evaluating the scenarios given to it by the top-level Monte Carlo simulation.

Table 5 – Coupling Variables for Monte Carlo Confirmation

Coupling Location	Parameters Passed
Mass Properties – Propulsion (1)	GLOW
Mass Properties – Trajectory (2)	GLOW, Sref
Propulsion – Trajectory (3)	Tvac, Ae, Ispvac
Propulsion – Mass Properties	T/W_e

The mass properties algorithm was different from the previous probabilistic optimization in two ways. First, there was no sizing step. This meant that the analysis took place for a single length vehicle, the same length that was the solution to the probabilistic optimization. Second, the probabilistic analysis does not take place at the contributing analysis level, so the functions to generate runs and pass them to the mass properties spreadsheets were eliminated. The steps taken to implement this were to eliminate the VB script to run several scenarios for mass properties and then wrap all of the assumptions corresponding to random variables into ModelCenter© as inputs, and all of the responses corresponding to output variables required by other analyses and output distributions.

The propulsion contributing analysis in this case was a simple wrapping of the SCORES analysis, with no run generating function or Perl wrapper. Inputs were taken for Tsl (generated by T/W_v and GLOW) and the settings of the design variables at the optimum point, and a sized engine was sent back to the mass properties. These analyses were iterated until the engine size was consistent, then the propulsion data for the chosen scenario was sent to the trajectory analysis.

The Monte Carlo trial ended with a call to the trajectory contributing analysis. Because there was no sizing involved, a feedback to mass properties was not necessary. The trajectory contributing analysis here consisted of the original call to POST using a random atmosphere generated by the GRAM99 range model. This was not the parameterized atmosphere used by the approximation methods.

This system of contributing analyses was run 1,000 times for random scenarios picked from the POST / GRAM99, SCORES and mass properties random inputs. For the POST analysis, the random variables that were required for input were the aerodynamic multipliers and the random atmosphere. The atmosphere generation required a different random seed each time, along with a random date and time of launch. SCORES required random values for the engine chamber pressure and power-to-weight ratio while the mass properties took random parameters corresponding the assumptions in the MER's. The settings for these random variables were identical to those in the respective analysis test baseline Monte Carlo simulations. The simulation took approximately 24 hours running on a combination of Windows NT and SGI Octane workstations. This makes it far too slow to even use in a sizing process, much less an optimization.

Table 6 reveals some important facts about the approximations used for optimization in this research. Because there was no feedback from the trajectory optimization, inaccuracies in the results of this analysis must either come from an inaccurate input or a bias in the analysis approximation itself. In converse, inaccuracies in the trajectory outputs had no effect on the results from the other contributing analyses. Keeping this in mind, Table 6 shows that the propulsion and mass properties analyses matched well with the Monte Carlo simulation, while the trajectory analysis output parameters had values outside the error bands for the Monte Carlo. This was due to a problem modeling the trajectory during optimization, this fortunately only lead to a slight underestimate of the probability of success with regards to meeting propellant requirements in the sizing process. This meant that the size of the vehicle was slightly overestimated when compared to the confidence level indicated by the result of the Monte Carlo simulation. However, this error was small enough that it was still well within the 95% confidence level error bands calculated for the Monte Carlo simulation.

Table 6 – Results of Full Monte Carlo Confirmation of Optimum

System Parameter	Approx. Estimate	Monte Carlo Estimate	95% C.I.	Rel. Error
Dry Weight μ	142,190 lb.	142,190 lb.	± 300 lb.	0 %
Dry Weight σ	4,880 lb.	4,840 lb.	± 213 lb.	0.826 %
P(Dry > 150,200 lb.)	95 %	94.7 %	± 11 %	0.317 %
MRReq μ	8.0496	8.0400	± 0.0016	0.119 %
MRReq σ	0.0309	0.0261	± 0.0011	18.4 %
MRavail μ	8.2331	8.2446	± 0.013	0.139 %
MRavail σ	0.217	0.215	± 0.010	0.930 %
P(MRReq - MRavail > 0)	80 %	82.4 %	± 5 %	2.91 %

To show the accuracy of the propulsion and mass properties approximations, Figure Fig. 11 shows a plot of the approximate dry weight for the fixed OML optimum vehicle generated using the approximation compared to the Monte Carlo result for the same random variable. As is shown in Table 6, the result here has negligible error. The probability density values for the Gaussian plot were scaled to match the frequency plot for the Monte Carlo simulation.

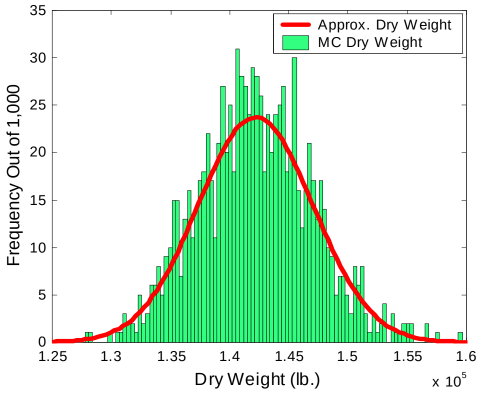

Figure 11 – Dry Weight Approximation and Monte Carlo Confirmation

Fig. 12 shows the error in mass ratio required and the impact it had on the overall system. While the errors are significant to the estimate for required mass ratio, the large spread of the mass ratio available probability minimizes the impact of the error and puts the mass ratio difference error back in the confidence level for the Monte Carlo simulation. For the Gaussian line plots, the probability density was scaled to match the frequency axis of the Monte Carlo simulation results.

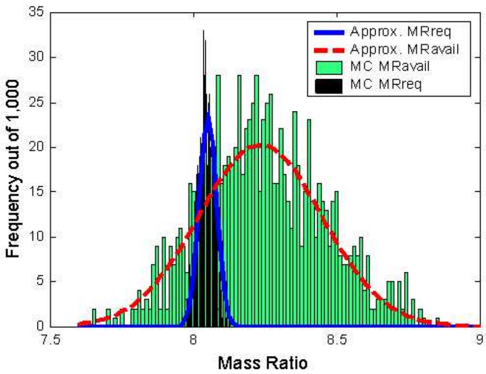

Figure 12 – Sizing Histogram for Full Monte Carlo Confirmation

While the cause of this small error does lie in the trajectory analysis, the cause could be any number of approximations made by this algorithm. Because the error was not a problem to the overall synthesis estimates, this was still a good method for probabilistic optimization. However, it is clear where room for improvement in accuracy in terms of contributing analysis probability estimation lies.

Deterministic Optimization Comparison

To see if there was any difference between the deterministic optimum and the probabilistic optimum solution, a deterministic optimization using the same models and design variables was conducted. The models used were the direct versions of the propulsion and mass properties/sizing algorithms and the response surface version of the trajectory analysis. The results of this were compared to the probabilistic optimum for 95% confidence level dry weight to determine their differences.

These results were compared using two criteria. First, it was determined if the settings of the design variables were any different from the probabilistic optimization. This should give relative information about which design variables are related to robustness in weight growth. Second, the difference will be shown in the results of the two optimizations, such as the reported dry weight value and OML size. This will show how using probabilistic information can tailor the reported results to the desired risk level of the program. If a high risk is tolerable, the confidence levels can be reduced and more optimistic results can be reported. If it is not tolerable, the confidence levels can be raised and correspondingly less optimistic results reported. The key is that the risk is expressed in easy to translate terms, such as the probability of not having enough propellant for the mission, or the probability of a dry weight value larger than 150,000 lb. It is easier to quantify this risk than the risk associated with changing dry weight margins.

Table 7 – Results of Deterministic Optimization Comparison

Design Var.	Start Pt. 1	Opt. Pt. 1 (Prob./Det)	Start Pt. 2	Opt Pt. 2 (Prob./Det)	Start Pt. 3	Opt. Pt. 3 (Prob./Det.)
Mixture Ratio (r)	6	5.61 / 5.65	5	5.60 / 5.65	7	5.62 / 5.66
Area Ratio (AR)	77.5	53.2 / 52.3	85	52.1 / 52.7	85	51.3 / 54.0
Vehicle Thrust to Weight (T/W_v)	1.2	1.2 / 1.2	1.6	1.2 / 1.2	1.4	1.2 / 1.2
Engine Chamber Pressure (Pch)	206 atm.	210 / 210 atm.	210 atm.	210 / 210 atm.	150 atm.	210 / 210 atm.
95% C.L. Dry Weight / Dry Weight	158 / 139 klb.	151 / 134 klb.	186 / 164 klb.	151.0 / 133.8 klb.	250.4 / 213.7 klb.	151.1 / 133.9 klb.
Prob. Length / Det. Length	147 / 142 ft.	147 / 142 ft.	152 / 147 ft.	147 / 142 ft.	168 / 161 ft.	147 / 142 ft.

The results in Table 7 show a decided similarity between the design variable settings for the probabilistic and deterministic optimizations. The one small difference that is consistent is the difference in mixture ratio. The optimizations found that a slightly lower mixture ratio was favorable for the probabilistic optimum. This is surprising because most of the uncertainty in sizing is due to the mass properties algorithm, an analysis that usually favors higher bulk density for system robustness (Ref. 12). The shift in mass ratio required due to the slightly higher engine specific impulse seems to have offset this effect.

These optimizations took around 10 minutes each to complete. While this was a very short amount of time, it took only 40 times longer (~6 hrs.) to optimize the probabilistic system. Considering the number of noise variables (84), this is a below linear real-time cost scale-up to use the distributed probabilistic system. This was a major cost savings over existing methods.

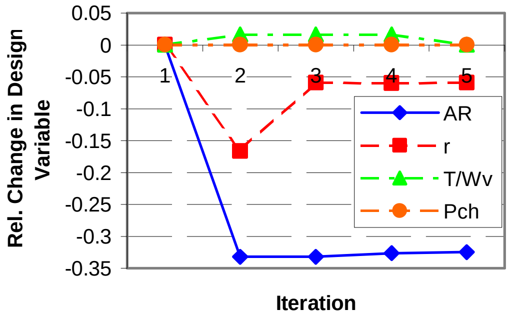

Figure 13– Design Variable History for Initial Deterministic Optimization

The convergence of the deterministic optimization was slightly better than that of the probabilistic optimization. This was most likely due to the more accurate gradients, since a smaller step size was possible for finite difference estimates. It is clear from a comparison of Fig. 8 and Fig. 13 that the deterministic optimization process was close to the optimum much sooner than the probabilistic process. The results here show that there is a small difference in the optimum variable settings for the two optimizations, but a large difference in the conservatism of the final answer.

To determine to which confidence levels the deterministic optimum corresponded, a Monte Carlo simulation of the final deterministic optimum was conducted. This simulation was similar to the confirmation of the probabilistic optimum except that it was conducted at a smaller length for slightly different settings of the design variables. The results of this simulation show that the probability of the reported dry weight being equal to or lower than 133.8 klbs. was only 53%. In addition, the confidence level associated with having enough propellant to perform the mission was only 54%. When compared to the error associated with a 1,000 trial Monte Carlo simulation, these can be both considered around 50%. These values are much lower than the 95% and 80% confidence levels reported for the probabilistic optimum. Histograms of mass ratio required and mass ratio available for the deterministic optimum can be seen in Fig. 14. The curves represent the predictions for the probabilistic optimum. As can be seen in Fig. 14, the probabilistically sized vehicle has a much higher probability of the mass ratio available exceeding the mass ratio required.

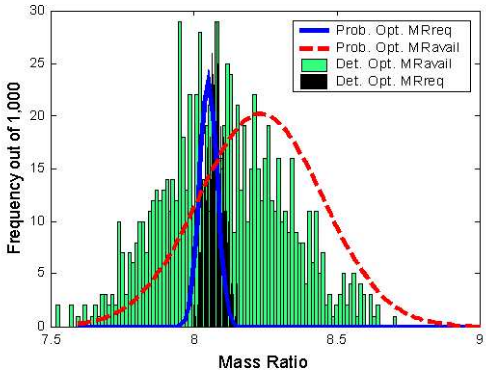

Figure 14 – Sizing Histogram of Deterministic Optimum

Apparently, the deterministic assumption of 15% dry weight margin corresponds to a fairly high risk level when compared to what was considered acceptable for the probabilistic phases of this study. The acceptable probability of success with regards to meeting propellant requirements for orbit was only 80%, and the probability of coming in under dry mass budget was 95%. These probabilities were not outrageously high, and in the case of the propellant requirement constraint, actually quite low. These results combined with the results of the Monte Carlo simulation of the deterministic optimum suggested that a 15% margin was very risky with respect to the assumptions made here about noise variables.

CONCLUSIONS

The specific goals attained were as follows:

- A new distributed probabilistic framework for launch vehicle conceptual design was demonstrated.

This was evidenced by the detailed process account given in this research. Also, now that the technique has been demonstrated, it should allow for this process to be re-implemented in less than two man-weeks. This implementation can also be done in parallel by individual disciplinary experts and as a result improve the overall process setup time. Finally, all of the uncertainty analysis was conducted at the contributing analysis level. This was important to retaining the advantages of distributed analysis.

- The utilization of heterogeneous computing platforms in a distributed probabilistic framework was demonstrated by the inclusion of executable programs, Perl scripts, Matlab scripts and Excel worksheets in a single automated framework.

These codes were based on both SGI workstations and Windows NT PC's. While this was done using the ModelCenter© commercial analysis integration package, the compatibility of this package with the distributed probabilistic launch vehicle design problem was a key to the utility of the technique.

- The goal of a multiple order of magnitude improvement in speed over a Monte Carlo simulation method was demonstrated.

This was shown in the fact that an entire probabilistic vehicle sizing process could be accomplished in about thirty minutes using the distributed approximations, while just a single length evaluation process took around 16 hours using a non-distributed Monte Carlo simulation. Accounting for the repeated simulations that would be required for a sizing process, this is a three order of magnitude improvement over a direct, system-level Monte Carlo process.

- Optimization utilizing the distributed probability analysis method was shown to be fast.

The goal of overnight optimization was met by the fact that the demonstration optimizations took between 6 and 8 hours each. Considering the amount of time that the non-distributed Monte Carlo simulation took for just a single length analysis, this is a huge savings.

- The test optimization was confirmed, showing that the problem formulation was sound and that the noise in the sizing process was not so great as to interfere with accurate gradient generation. This was shown by the optimization finding the same point from three separate starting locations.

- The accuracy of the distributed approximation was also found to be quite good.

It exceeded the objective of 5% accuracy set for constraint satisfaction by a comfortable margin for both the propellant required and dry weight confidence level calculations. This was determined by comparison of the found optimum to a single length Monte Carlo simulation. In addition to this, all of the critical output parameters, along with the majority of all parameters were well within the error bounds calculated for the Monte Carlo simulation.

- While the deterministic and probabilistic optimum design variable settings were not very different, the reported vehicle size by the two processes was vastly different. This means that the reported size of the analysis corresponds to a specific user confidence level, not just an arbitrary growth safety factor.

All in all, a new fast and efficient method for probabilistic optimization of conceptual launch vehicle designs was presented, along with test results verifying its speed and accuracy. This new architecture has the potential to allow for the practical probabilistic optimization of reusable launch vehicles in inherently distributed environments where it was impractical before.

REFERENCES

1. Cormier, T., Scott, A., Ledsinger, L., McCormick, D., Way, D., and Olds, J., "Comparison of Collaborative Optimization to Conventional Design Techniques for a Conceptual RLV," AIAA paper 2000-4885, Sept. 2000.

2. Ledsinger, L., "Solutions to Decomposed Branching Trajectories with Powered Flyback Using Multidisciplinary Design Optimization," Ph.D. Thesis, Georgia Institute of Technology, July 2000.

3. Braun, R. D., Kroo, I. M., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," International Congress on Industrial and Applied Mathematics, August 1995.

4. Budianto, I., "A Collaborative Optimization Approach to Improve the Design and Deployment of Satellite Constellations," Ph.D. Thesis, Georgia Institute of Technology, 2000.

5. McCormick, D., Olds, J. R., "A Design of Experiments- Based Method for Point Selection in Approximating Output Distributions," AIAA Paper 2002-5535, Sept. 2002.

6. *DOT™ User's Manual, Version 4.20.*, Vanderplaats Research & Development, Inc., Colorado Springs, CO, 1995.

7. Way, D. W., Olds, J. R., "SCORES: Web-Based Rocket Propulsion Analysis Tool for Space Transportation System Design," AIAA paper 99-2353, June 1999.

8. *JMP Statistics and Graphics Guide*, Cary, NC: SAS Institute, Inc., 1995.

9. Hasofer, A. M. and Lind, N. C., "Exact and Invariant Second Moment Code Format," *Journal of Engineering Mechanics*, ASCE, Vol. 100, No. EM1, February, 1974.

10. Brauer, G. L., Cornick D. E., Olson, D., Peterson, F., and Stevenson, R., "Program to Optimize Simulated Trajectories (POST) Formulation Manual," NAS1-18147, September 1990.

11. Justus, C. G., Johnson, D. L., "The NASA/MSFC Global Reference Atmospheric Model - 1999 Version (GRAM-99)," NASA Technical Memorandum TM-1999-209630, May 1999.

12. McCormick, D. and Olds, J., "System Robustness Comparison of Advanced Space Launch Concepts," AIAA paper 98-5209, Oct. 1998.