**AIAA-2000-4885**

# Comparison of Collaborative Optimization to Conventional Design Techniques for a Conceptual RLV

T. Cormier
A. Scott
L. Ledsinger
D. McCormick
D. Way
J. Olds

Georgia Institute of Technology
Atlanta, GA

8th AIAA/USAF/NASA/ISSMO Symposium on
Multidisciplinary Analysis and Optimization
6-8 September 2000
Long Beach, California

# Comparison of Collaborative Optimization to Conventional Design Techniques for a Conceptual RLV

Timothy A. Cormier† , Andrew Scott†, Laura A. Ledsinger†, David J. McCormick† , David W. Way†, John R. Olds*
Space Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0150

## ABSTRACT

Initial results are reported from an ongoing investigation into optimization techniques applicable to multidisciplinary reusable launch vehicle (RLV) design. The test problem chosen for investigation is neither particularly large in scale nor complex in implementation. However, it does have a number of characteristics relevant to more general problems from this class including 1) the use of legacy analysis codes as contributing analyses and 2) non-hierarchical variable coupling between disciplines. Propulsion, trajectory optimization, and mass properties analyses are included in the RLV problem formulation. A commercial design framework is used to assist data exchange and legacy code integration.

The need for a formal multidisciplinary design optimization approach is introduced by first investigating two more conventional approaches to solving the sample problem. A rather naive approach using iterative sublevel optimizations is clearly shown to produce non-optimal results for the overall RLV. The second approach using a system-level response surface equation (RSE) constructed from a small number of RLV point designs is shown to produce better results when the independent variables are judiciously chosen. However, the response surface method (RSM) approach cannot produce a truly optimum solution due to the presence of uncoordinated sublevel optimizers in the three contributing analyses.

Collaborative optimization (CO) appears to be an attractive multidisciplinary design optimization approach to solving this problem. Initial implementation attempts using CO have exhibited noisy gradients and other numerical problems. Work to overcome these issues is currently in progress.

## NOMENCLATURE

| | |
|---|---|
| $A_e$ | Engine Exit Area ($in^2$) |
| ACC | Advanced Carbon-Carbon |
| $C_f$ | Thrust coefficient ($T/P_cA_t$) |
| CCD | Central Composite Design |
| CO | Collaborative Optimization |
| DSM | Design Structure Matrix |
| $I_{sp,vac}$ | Engine Vacuum Specific Impulse (sec) |
| MER | Mass Estimating Relationship |
| MR | Mass ratio (gross weight/burnout weight) |
| $P_c$ | Engine Chamber Pressure (psia) |
| r | Engine mixture ratio |
| RLV | Reusable Launch Vehicle |
| RSE | Response Surface Equation |
| RSM | Response Surface Method |
| $S_{ref}$ | Wing Planform Area ($ft^2$) |
| SQP | Sequential Quadratic Programming |
| SSTO | Single-Stage-to-Orbit |
| $T_{vac}$ | Engine Vacuum Thrust (lbf) |
| $T_{sl}$ | Engine sea-level Thrust (lbf) |
| $T_{sl}/W_e$ | Engine sea-level thrust-to-weight ratio |
| $T_{sl}/W_g$ | Vehicle sea-level thrust-to-weight ratio |
| TABI | Tailorable Advanced Blanket Insulation |
| TUFI | Toughened Uniform Fibrous Insulation |
| $W_{dry}$ | Vehicle dry weight (lb) |
| Wg | Vehicle gross weight (lb) |
| ΔV | Velocity increment (ft/sec) |
| ε | Nozzle expansion ratio |

## DESIGN PROBLEM STATEMENT

The RLV to be designed is a second-generation SSTO vehicle that is launched from the Kennedy Space Center to the International Space Station (target orbit is 220 nmi x 220 nmi x 51.6º). The propulsion system is to be comprised of five high thrust-to-weight engines. Primary orbit insertion occurs at a transfer orbit of 50 nmi x 100

---

† - *Graduate Research Assistant, School of Aerospace Engineering, Student member AIAA.*

* - *Assistant Professor, School of Aerospace Engineering, Senior member AIAA.*

nmi x 51.6º. An on-orbit ΔV of 1100 fps is included to transfer the vehicle to the final orbit, rendezvous with the space station, and deorbit. The unpiloted vehicle is required to carry a 25,000 lb payload to orbit. The vehicle is also classified as a Generation 2 reusable launch vehicle, therefore all technologies are commensurate with 2005 technology freeze date for the first flight in 2010. The propellant tanks are to be made of an aluminum-lithium alloy. Graphite-epoxy is used in the exposed wing and carry through structure, as well as the primary, secondary, and payload structures. The thermal protection system uses ACC, TUFI tiles and TABI blankets. A schematic of the reference RLV is shown in Figure 1.

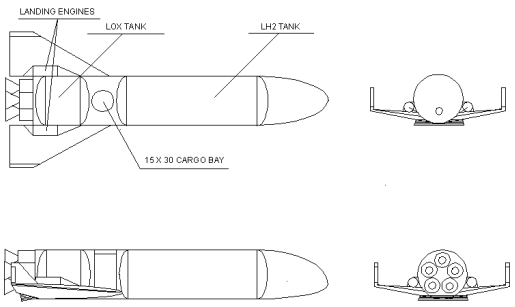


**Figure 1. Schematic of reference RLV**

## DESIGN TOOLS

### PROPULSION

SCORES (SpaceCraft Object-oriented Rocket Engine Simulation) is a web-based rocket engine analysis tool developed at Georgia Tech [1]. This tool suitable for use in conceptual design, provides propulsion metrics such as thrust and specific impulse. Only top-level propulsion parameters are required for input. These parameters include mixture ratio, chamber pressure, throat area, and expansion ratio. The SCORES web-based tool is public and can be accessed at the Uniform Resource Locator (URL) address listed below:

http://titan.cad.gatech.edu/~dwway/SCORES

SCORES may be run from the web or interactively from the UNIX operating system For the purposes of this exercise, the UNIX version of SCORES was coupled with Phoenix Integration's Model Center computational framework [2].

SCORES models a rocket engine in two parts. First, the chemical processes occurring in the combustion chamber are analyzed. Second, the expansion of hot gases in the convergent-divergent nozzle is analyzed. The combustion process is assumed to occur adiabatically and at constant pressure. Additionally, all of the molecular species involved in the combustion are assumed to be thermally perfect gasses. Finally, the initial velocity of the reactants is taken to be zero, thus assuming an infinite-area combustor. Therefore, the temperature and pressure in the combustion chamber are taken to be total values. The initial temperature of all the reactants is assumed to be 500K. The composition of the product gasses is then determined through chemical equilibrium calculations.

For the convergent-divergent nozzle, the flow is assumed frozen at the equilibrium conditions calculated for the combustion chamber. The expansion process is then modeled as a steady, inviscid, quasi-1D, isentropic flow. Because of the quasi-1D assumption, cross-sectional area and expansion ratio are the only geometry variables. A detailed description of the nozzle contour is not necessary. The combustion products are assumed to be a mixture of calorically perfect gasses.

Thrust and $I_{sp}$ are calculated from the determined nozzle exit conditions. These estimates typically overpredict the thrust and $I_{sp}$. This over-prediction is due to the ideal nature of the assumptions. Statistical performance efficiencies derived from existing flight hardware are then used to simulate losses by correcting downwardly adjusting the ideal thrust and $I_{sp}$ values.

SCORES provides an option to sizing the nozzle throat area to match a required thrust. Because the thrust is linear with throat area, the required throat area is simply the guessed value, 1 sq.in. by default, multiplied by the ratio of required thrust to calculated thrust. Therefore, no iteration is required, making the sizing option just as rapid as the analysis option. A low-fidelity estimation of thrust-to-weight (T/W) is also provided. This estimation is based on the premise that the engine will develop a constant power-to-weight (P/W), where power, defined in Equation 1, is based on the chamber and exit enthalpies.

$$P = \dot{m}(h_c - h_e) \quad (1)$$

Power is easily calculated within the same routines that predict thrust and Isp. If the P/W is known, then the T/W is found easily from the thrust and power by Equation 2.

$$\frac{T}{W} = \left(\frac{P}{W}\right)\frac{T}{P} \quad (2)$$

To allow for differences in technology levels, SCORES provides a user input to select the T/W relative to other engines. The user may select "high", "average", or "low" from a pull-down menu. A selection of "average" uses a P/W of 0.017 MW/lb, while a selection of "high" or "low" uses a P/W of 0.023 or 0.015 MW/lb respectively.

## PERFORMANCE

The tool used to simulate the trajectories of the RLV was the Program to Optimize Simulated Trajectories, POST [3]. POST is a Lockheed Martin and NASA code that is widely used for trajectory optimization problems in vehicle design. POST is a generalized event-oriented code that numerically integrates the equations of motion of a flight vehicle given definitions of aerodynamic coefficients, propulsion system characteristics, atmospheric tables, and gravitational models. Guidance algorithms used in each phase are user-defined. Numerical optimization is used to satisfy trajectory constraints and minimize a user-defined objective function by changing independent steering and propulsive variables along the flight path. POST runs in a batch execution mode and depends on an input file (or input deck) to define the initial trajectory, event structure, vehicle parameters, independent variables, constraints, and objective function.

## MASS PROPERTIES

The weights and sizing analysis uses a photographic scaling on a set of parametric mass estimating relationships (MER's) that have a NASA Langley heritage. This analysis is performed on a Microsoft Excel spreadsheet. Using the results of the trajectory analysis, the booster is photographically scaled up or down until the available mass ratio on-board the currently sized vehicle (MR_avail) and the required mass ratio from POST (MR_req). Since changing the vehicle scale changes the gross weight, sea-level thrust requirements, etc., the disciplines in the main iteration loop must be iterated until the vehicle size converges. This typically takes 4 to 5 iterations.

Primary booster structural materials include aluminum lithium alloy for the propellant tanks and graphite-epoxy composite for other structure such as exposed wings, the wing carry through, and verticals. Other subsystem highlights include an autonomous flight control system, electrohydraulic actuators, high power density fuel cells, lightweight avionics, and environmentally safe LOX-ethanol orbital maneuvering system (OMS) propellants.

The weights and sizing analysis provides a great deal information to the other analyses. Gross weight and wing reference area are given to the trajectory analysis and required sea-level static thrust is used by the propulsion analysis.

## COMPUTATIONAL FRAMEWORK

In addition to the tools used for each discipline, a fourth tool, Phoenix Integration's Model Center software package, was used to coordinate the system level analysis. Model Center a program that facilitates cross-platform analysis integration. For each of the disciplines, a wrapping script was written to respectively send and collect the inputs and outputs of each tool. POST and SCORES were set up to run on UNIX machines while the weights spreadsheet was run in Microsoft Excel on a Windows NT machine. Once each of the tools was properly wrapped and set up, one could easily link the inputs and outputs of the three disciplines to one another from within Model Center. These links were setup as appropriate to the design problem and the optimization method used to solve it. With the setup complete, Model Center is capable of transferring the appropriate information between the tools as well as coordinating their execution.

Model Center also provides an optimization package, which was used in the methods that required system level optimization. This package is based on the popular optimization code, DOT [4]. For the collaborative optimization, sequential quadratic programming was used as the optimization algorithm. As system level optimization progresses, Model Center neatly records and organizes all the desired variables and constraints at each system level iteration. Model Center was also very helpful in the data collection for the RSM method by evaluating and collecting the results of multiple runs as setup by the user. All data collected by Model Center is easily exportable to Excel, making it easy to analyze the final results.

Without the use of a program like Model Center a great deal of user interaction is required to manually run each analysis. Manually running each analysis requires the user to change input variables, run the tool and appropriate collect the results. This process can be very time consuming and tedious. With so much interaction being required by the user to manually run each tool, there is also an increased likelihood of making mistakes. Though using Model Center certainly takes more time to set up, the cost of doing so is negligible when compared to the timesavings and error reduction gained when doing the actual analysis. However, it

should be mentioned that in using Model Center it was imperative that the disciplinary analyses be very robust. For disciplinary tools that traditionally require a bit a tweaking and interaction (such as POST), a little extra effort is required during setup to ensure that they perform consistently and accurately.

Overall, the use of Model Center greatly increased the speed and efficiency with which the analyses were performed. In fact, with methods such as collaborative optimization that require many iterations and a great deal of computational time, it is difficult to imagine if manual implementation of such methods is even realistically feasible or desirable.

## DESIGN METHODOLOGY

The design structure matrix (DSM) is dependent upon the design method used. In conventional methods, variables are passed from one discipline to the others. A DSM for conventional methods is shown in Figure 2.

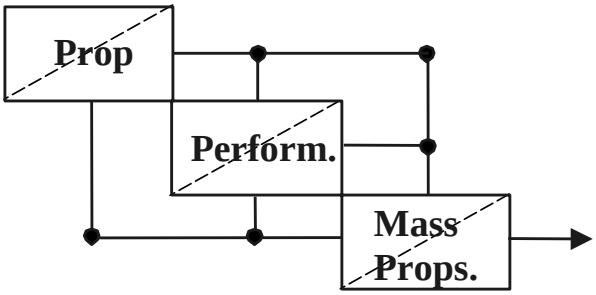


**Figure 2. Conventional Method Design Structure Matrix**

The diagonal dotted lines indicate that local optimization occurs within a given CA. A table depicting the flow of variables through each of the CA's is shown in Table 1.

ITERATIVE OPTIMIZERS METHOD

Simple iteration of the disciplinary optimizers follows the DSM depicted in Figure 2. Each disciplinary tool modified its own local variables to obtain local optimization based on the inputs received from the other disciplines. The influence of local optimization on the system level is easily seen as the overall design converged at a suboptimal configuration.

The first step in this process is to let the first CA make design decisions. The propulsion expert will typically select a design that will maximize his objective function, $I_{sp}$. This being the case, the propulsion expert will chose the lowest mixture ratio possible with the highest possible expansion ratio. Once the propulsion analysis has been completed, the performance expert will perform an analysis to minimize the fuel consumed, thus minimizing the mass ratio. Finally, the mass properties analyst will use the weights and sizing sheet to minimize vehicle dry weight.

**Table 1. Coupling variables in the conventional DSM**

| Variable | Propulsion | Performance | Mass Properties |
|---|---|---|---|
| Tsl | input | | |
| Tvac | output | input | input |
| r | output | | input |
| Ae | output | input | input |
| Isp,vac | output | input | |
| Tsl/Weng | output | | input |
| Sref | | input | output |
| Wg | | input | output |
| MR | | output | input |
| Wdry | | | output |

RESPONSE SURFACE METHOD

A central composite design (CCD) matrix was used to set values for the global variables for the response surface method. Table 2 illustrates a generic CCD matrix.

**Table 2. Generic Central Composite Design Matrix**

| $x_1$ | $x_2$ | $x_3$ |
|---|---|---|
| -1 | -1 | -1 |
| -1 | -1 | 1 |
| -1 | 1 | -1 |
| -1 | 1 | 1 |
| 1 | -1 | -1 |
| 1 | -1 | 1 |
| 1 | 1 | -1 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| $-\alpha$ | 0 | 0 |
| $\alpha$ | 0 | 0 |
| 0 | $-\alpha$ | 0 |
| 0 | $\alpha$ | 0 |
| 0 | 0 | $-\alpha$ |
| 0 | 0 | $\alpha$ |

The results of the 15 runs of the array were used to fit a response surface to the design space. The constraints placed on the global variables were taken into account in the DOE by limiting the range of testing. The alpha values were set as the maximum range for the three design variables instead of the typical –1 and 1. The coded variables are shown in Table 3.

**Table 3. Response Surface Equation Coded Variables**

| | ε ($x_1$) | r ($x_2$) | $T/W_g$ ($x_3$) |
|---|---|---|---|
| $-\alpha$ | 30.00 | 4.500 | 1.20000 |
| -1 | 38.11 | 5.615 | 1.24054 |
| 0 | 50.00 | 7.250 | 1.30000 |
| 1 | 61.89 | 8.885 | 1.35946 |
| $\alpha$ | 70.00 | 10.000 | 1.40000 |

Once the response surface was determined, Matlab's constrained function optimizer (a sequential quadratic programming based algorithm) was used to find the best design.

## COLLABORATIVE OPTIMIZATION

In collaborative optimization the general strategy is to obtain the optimal system configuration for a given objective function while allowing each CA to remain as independent and focused as possible while still maintaining consistency amongst disciplines. The primary characteristic of collaborative optimization is that it allows the CA's to maintain their discipline level optimization capabilities. Normally, this would present a problem as is likely that the disciplinary objective functions are not consistent with the system level object function. For example, SCORES normally tries to maximize $I_{sp}$, but this however drives the engine weight up and may result in an engine configuration that does not lend itself to a system level configuration that minimizes dry weight.

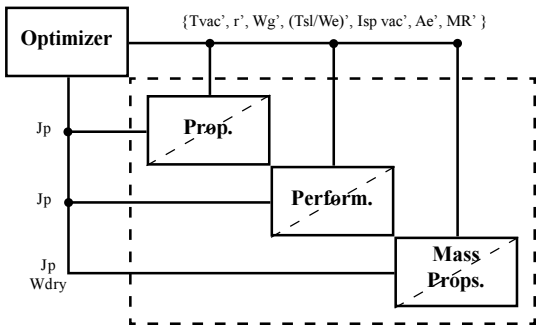


**Figure 3. Collaborative Optimization Design Structure Matrix**

To avoid this conflict, collaborative optimization replaces the objective functions of each disciplinary optimization. The new objective function attempts to minimize a newly defined error function, known a J term. These J terms measure the relative error between the output variables of the disciplinary tool and corresponding target values. These target values are set by the system level optimizer, which is configured to minimize a system level objective function under the constraint that the J terms of each discipline are kept below a certain tolerance. Each disciplinary tool is allowed to vary all of its usual inputs and local variables to minimize its own objective function. This allows for disciplinary experts to focus on their domain-specific issues while maintaining interdisciplinary compatibility. Table 4 summarizes the changes made for collaborative optimization while Figure 3 show the modified DSM.

**Table 4. Variable breakdown for Collaborative Optimization**

| **System Level** | |
|---|---|
| Objective Function | Minimize $W_{dry}$ |
| Variables | $T_{vac}$', r', ε', $T_{sl}/W_e$', $I_{sp}$', $A_e$', $W_g$', MR' |
| Constraints | $J_p$ <= tolerance<br>$J_t$ <= tolerance<br>$J_w$ <= tolerance |

| **Propulsion** | |
|---|---|
| Objective Function | Minimize $J_p$:<br>$J_p = [1-(T_{vac}/T_{vac}')]^2 + [1-(r/r')]^2$<br>$+ [1-(A_e/A_e')]^2 + [1-((T_{sl}/W_e)/(T_{sl}/W_e)')]^2$<br>$+ [1-(I_{sp}/I_{sp}')]^2$ |
| Input Variables | $T_{vac}$', r', ε', $T_{sl}/W_e$', $I_{sp}$', $A_e$' |
| Local Variables | $T_{vac}$, r, ε, $P_c$ |
| Calculated Variables | $T_{sl}/W_e$, $I_{sp}$, $A_e$ |

| **Trajectory Performance** | |
|---|---|
| Objective Function | Minimize $J_t$:<br>$J_t = [1-(T_{vac}/T_{vac}')]^2 + [1-(W_g/W_g')]^2$<br>$+ [1-(A_e/A_e')]^2 + [1-(S_{ref}/S_{ref}')]^2$<br>$+ [1-(I_{sp}/I_{sp}')]^2 + [1-(MR/MR')]^2$ |
| Input Variables | $T_{vac}$', $W_g$', $A_e$', $S_{ref}$', $I_{sp}$', MR' |
| Local Variables | $T_{vac}$, $W_g$, $A_e$, $S_{ref}$, $I_{sp}$, Azimuth, pitch angles |
| Calculated Variables | MR |

| **Mass Properties** | |
|---|---|
| Objective Function | Minimize $J_w$:<br>$J_w = [1-(T_{vac}/T_{vac}')]^2 + [1-(W_g/W_g')]^2$<br>$+ [1-(A_e/A_e')]^2 + [1-(S_{ref}/S_{ref}')]^2$<br>$+ [1-(MR/MR')]^2 + [1-(r/r')]^2$<br>$+[1-((T_{sl}/W_e)/(T_{sl}/W_e)')]^2$ |
| Input Variables | Tvac', Wg', Ae', Sref', MR', r',Tsl/We' |
| Local Variables | $A_e$, r, $T_{sl}/W_e$, scale factor, $T_{sl}/W_g$ |
| Calculated Variables | $T_{vac}$, $W_g$, $S_{ref}$, MR |

In the implementation of this method, it is crucial to normalize all the target variables and the objective function. After various trials, it was found that the convergence speed and accuracy of the method was particularly sensitive to the tolerance allowed for the J terms. If the tolerance was too loose, the system level optimizer would converge on a design that was not realistically feasible. If the tolerance is too tight, the optimizer may converge at a suboptimal level or merely take an excessively long time find the optimal answer. If the initial guess is particularly bad, the optimizer may, again, have difficulty finding a solution.

ADAPTING SCORES FOR CO

To implement collaborative optimization, each contributing analysis must have its own optimizer. DOT was chosen as the optimizer to be integrated with SCORES. Since SCORES is normally run through an input file using the web interface, the code had to be modified somewhat to allow the program to be called in a subroutine fashion. Several input options for this problem were fixed: $H_2/LO_2$ combustion, staged-combustion engine cycle, bell-shaped convergent-divergent nozzle, sea-level (1 atm) ambient conditions, high T/W, and English units. In addition, the initial guess for throat size was set at 1 $in^2$. Three of the remaining inputs supplied by the system level optimizer. These were required sea-level thrust, nozzle area ratio, and propellant mixture ratio. The remaining input, combustion chamber pressure, was then the only design variable in the local optimizer's scope. The SCORES subroutine then provided specific impulse given the chamber pressure.

The optimization problem was then to maximize specific impulse subject to the constraint that there be no flow separation in the nozzle. This constraint was assessed by comparing the sea-level thrust coefficient ($C_f$) to the thrust coefficient at separation conditions, ($C_{f,sep}$). Separation was deemed to occur if $C_f$ exceeded $C_{f,sep}$. The separation thrust coefficient was found from the maximum thrust coefficient (the thrust coefficient at perfect nozzle expansion to atmospheric pressure) by a curve fit which is a function of the nozzle area ratio (ε). The equation used was:

$$CF_{sep} = .85\ CF_{\max}\left(1 - e^{-0.3\varepsilon}\right) \qquad (3)$$

DOT, a FORTRAN subroutine, was compiled separately and then linked with the SCORES $C^{++}$ code. The options used for the optimization were Sequential Quadratic Programming (SQP), automatic scaling, and central difference evaluation of gradients. The optimization of Isp within SCORES is also limited by the constraint that the chamber pressure does not exceed 3100 psia.

ADAPTING POST FOR CO

For both the iterative optimizers method and the response surface method, the trajectory simulation is identical. Beginning with launch from Kennedy Space Center, the RLV is controlled by its initial heading and then with several pitch angles, a total of five controls. At the end of the simulation, the vehicle is constrained to obtain an orbit of 50 nmi x 100 nmi x 51.6° at a flight path angle of zero degrees. (Through use of the OMS engines, the ISS orbit will be achieved.)

For these methods, the trajectory was optimized every iteration of each vehicle design. The trajectory optimization method used for the RLV simulation was the accelerated projected gradient algorithm. The objective of the optimization was to maximize the final weight (in effect, minimize fuel consumed).

The trajectory simulation requires several inputs: vacuum specific impulse, gross lift-off weight, wing planform area, total vacuum thrust, and total exit area of the engines. The output used was the required mass ratio, which is burnout weight divided by gross lift-off weight.

The trajectory simulation for the CO method had a few similarities to that of the iterative method and RSM. POST still optimized using the accelerated projected gradient algorithm. In addition, all of the local constraints (the orbital termination criteria) exist and had to be met.

There are many differences however. As previously stated, the objective for the trajectory simulation in the CO method was one of the constraints at the system-level, the error, $J_p$, as listed in equation 1. The optimizer in POST tries to match the targets variables, those which are 'primed,' to the local versions of those variables.

In order to set this up correctly in POST, the code's special calculations subroutine had to be employed. The local versions of the target variables were recorded/analyzed at the appropriate moments and used in the calculation of $J_p$. Additionally, the local versions of the target variables became added control variables in the POST deck. Only five more controls were added because the required mass ratio, as an output, was calculated internally.

ADAPTING THE WEIGHTS SHEET FOR CO

For collaborative design, the typical practice of scaling the vehicle length until the mass ratio from the trajectory analysis is abandoned in favor of a constraint matching approach. In this case, the local optimizer, instead of minimizing just the error in mass ratio, minimizes the error for the target variables from the top-level optimizer. At the local level, these target variables can be either local input variables (length, engine exit area, lift-off T/W, etc.) or local output variables (mass ratio, gross weight, etc.) What is important is that the analysis finds a valid (within weights and sizing) design that minimizes the error of the target vector.

The system-level optimizer requires the value of the J error for the analysis, the value of the objective function dry weight and the gradients of both these variables with respect to the targets in the J vector. To find the gradient of the J error with respect to the targets, the target values are varied within mass properties, then the local optimization is repeated. Because the objective function for the entire analysis is calculated by mass properties, the gradient of this value is calculated simultaneously.

## RESULTS

Of primary concern when comparing the various methods were, first, the quality of the answer obtained, second, the speed with which it was obtained, and finally, the overall difficulty in implementing the method. To ensure that the results of each method were fairly comparable to one another, a fixed set of initial guesses was used for each method. These guesses are summarized in Table 5.

ITERATIVE OPTIMIZERS

As expected, the results from the iterative optimizers method were quick and easy to obtain. To ensure convergence, six loops were made through the iterative optimizers DSM with the results of the previous iteration being the inputs for the next. The results obtained, however, were very poor. In its effort to maximize $I_{sp}$, SCORES drives the mixture ratio to its lower limit of 4.5 and the expansion ratio to its upper limit of 70. Though a high Isp is generally good for driving dry weight down, a high expansion ratio results in a larger, heaver nozzle that drives dry weight up. Similarly, it is not necessarily the case that POST's efforts to minimize the consumed fuel and drive the MR down are in conjunction with the system level objective function of minimum dry weight. The convergence history of the iterative optimizers method is shown in figure 4.

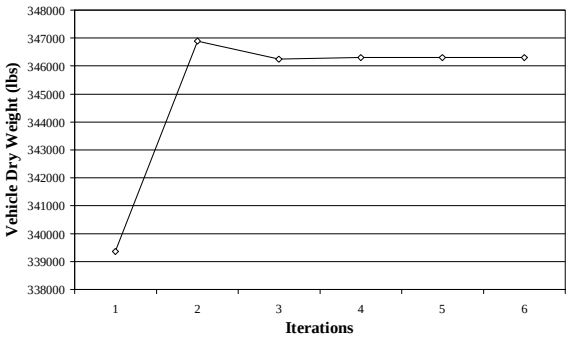


**Figure 4. Dry weight vs. iteration history for the iterative optimizers method**

**Table 5. Initial Guesses for Optimization**

| Initial guesses by: | | SI | RSM | CO |
|---|---|---|---|---|
| $T_{sl}$ | 3,000,000 lbs | x | x | x |
| r | 6.500 | | | x |
| $I_{sp}$ | 440.00 s | | | x |
| $A_e$ | 250.00 sq. ft. | | | x |
| $Tsl/W_e$ | 50.000 | | | x |
| $W_g$ | 2,500,000 lbs | x | x | x |
| $S_{ref}$ | 4000 sq. ft. | x | x | x |
| $Tsl/W_g$ | 1.20 | | | x |

RESPONSE SURFACE METHOD

The response surface method provided a near-optimal solution. By picking out select points in the design space a representative response surface equation was generated. A list of the runs performed to complete the CCD matrix and their solutions is shown in Table 6. The general form of the equation and a table of its coefficients are shown below.

$$W_{dry} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1^2 + \beta_5 x_2^2 + \beta_6 x_3^2 + \beta_7 x_1 x_2 + \beta_8 x_1 x_3 + \beta_9 x_2 x_3 \quad (4)$$

The RSE was determined using a least squares regression program written in Matlab. A comparison of the dry weight determined using the RSE and the optimal dry weight found by using the design tools is shown in Table 8.

**Table 6. Coded variables and the corresponding results for the RSM**

| $x_1$ | $x_2$ | $x_3$ | $W_{DRY}$ |
|---|---|---|---|
| 38.11 | 5.615 | 1.24054 | 330,315 lbs |
| 38.11 | 5.615 | 1.35946 | 348,862 lbs |
| 38.11 | 8.885 | 1.24054 | 552,152 lbs |
| 38.11 | 8.885 | 1.35946 | 608,054 lbs |
| 61.89 | 5.615 | 1.24054 | 318,200 lbs |
| 61.89 | 5.615 | 1.35946 | 340,845 lbs |
| 61.89 | 8.885 | 1.24054 | 519,361 lbs |
| 61.89 | 8.885 | 1.35946 | 581,221 lbs |
| 50 | 7.25 | 1.3 | 378,252 lbs |
| 30 | 7.25 | 1.3 | 420,351 lbs |
| 70 | 7.25 | 1.3 | 384,450 lbs |
| 50 | 4.5 | 1.3 | 357,490 lbs |
| 50 | 10 | 1.3 | 782,711 lbs |
| 50 | 7.25 | 1.2 | 359,818 lbs |
| 50 | 7.25 | 1.4 | 405,678 lbs |

**Table 7. Response surface equation coefficient values**

| Coefficient | x term | Value |
|---|---|---|
| $\beta_0$ | --- | 375370 |
| $\beta_1$ | $x_1$ | -13008 |
| $\beta_2$ | $x_2$ | 118436 |
| $\beta_3$ | $x_3$ | 12964 |
| $\beta_4$ | $x_1^2$ | -6047 |
| $\beta_5$ | $x_2^2$ | -217 |
| $\beta_6$ | $x_3^2$ | 7777 |
| $\beta_7$ | $x_1x_2$ | 6086 |
| $\beta_8$ | $x_1x_3$ | 65626 |
| $\beta_9$ | $x_2x_3$ | -763 |

**Table 8. Comparison of the RSE results to a verification run using the design tools**

| | INPUTS | | | OUTPUT | |
|---|---|---|---|---|---|
| | e | r | Tsl/Wg | Wdry | % Difference |
| RSE | 57.4717 | 6.0051 | 1.2 | 306725 | 2.65 |
| DSM | | | | 315065 | |

Because no system level optimizer is used in the RSM, the difficulty involved in setting up and implementing this method is relatively small. Because each run is completely independent from the others, problems that may occur in the individual CA's are not nearly as detrimental as it is fairly easy to make adjustments and reevaluate a specific run.

It should be noted, however, that this method requires a good amount of previous knowledge about the problem. The number of runs required to parameterize the design space rises exponentially with the number of variables used. Because of previous experience, it was known which variables the local optimizers would drive in conflicting directions with the overall objective of minimum dry weight. These variables were the ones chosen for the parameterization variables. Without such knowledge, one would be required to use more variables to explore the design space. This of courses would require more runs making the RSM method less feasible for problems in which a good deal of knowledge of the variable effects is not already known.

## COLLABORATIVE OPTIMIZATION

Unfortunately the latest attempts to obtain a solution for this problem using collaborative optimization have been met with limited success. Several problems and difficulties have been noticed in the general implementation of this method. Perhaps the greatest limitation of collaborative optimization is that it does not appear to be a quickly converging method, requiring several iterations to minimize the objective and meet constraints. Additionally, POST calculations take a great deal more time as the number of independent variables is increased from 5 to 10. Furthermore, as the target values set by the system level optimizer can fluctuate quite a bit, several repetitions of POST are often required to obtain optimality with POST. These limitations in speed were surpassingly detrimental to time required for debugging and tweaking the process in general.

Limited success has been achieved by using a "hot" starting point that is known to be already converged. Using the general initial guesses specified in table, a refined starting point was obtained by iterating through the regular CA's until convergence was obtained. Figure 5 shows this convergence history. CO's convergence rate seems to be fairly sensitive to the tolerances allowed on the error constraints. These results took approximately 14 hours and were obtained using a tolerance of .0001 corresponding to a total of 1% error between the targets and outputs of each CA. This tolerance was initially thought to be acceptable, but what was found was that this error would tend to gather in a few variables.

To further compound the problem, these errors were usually in the most sensitive of the design variables, such as $I_{sp}$. Consequentially, these results are were unacceptable as the final design variables would vary significantly between CA's. This effect is well illustrated in Figures 6 & 7 which show the convergence history of $I_{sp}$ and MR respectively. Though the errors in these variables are technically satisfactory according to the tolerance allowed on the J terms, these errors are in reality quite significant. Small differences in the values of Isp and MR are known to greatly influence dry weight.

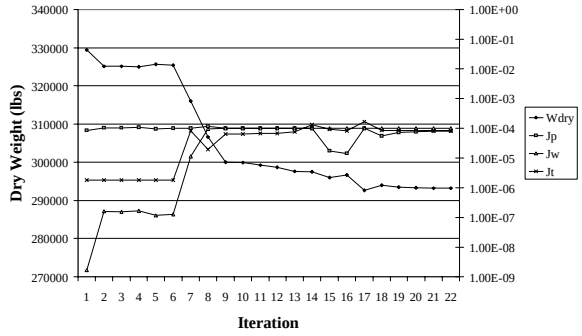


**Figure 5. Dry weight and J-term convergence for collaborative optimization**

Attempts were made to use tighter tolerances, but this resulted in either excessively long calculation times or premature convergence at a suboptimal solution. Unfortunately, there has been no success to date obtaining any kind of results using a "cold" starting point that is not necessarily an already converged design configuration.

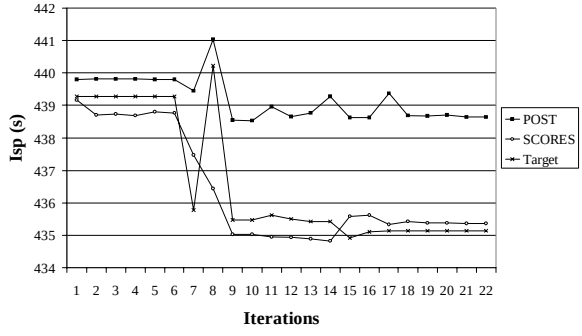


**Figure 6. Isp convergence for collaborative optimization**

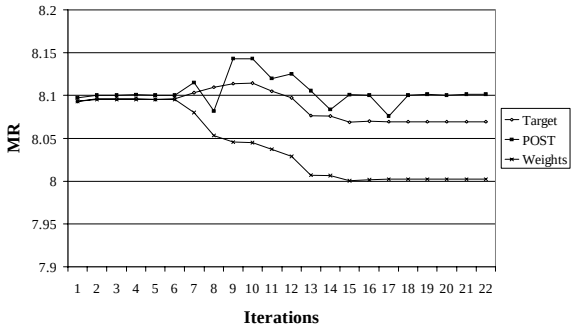


**Figure 7. MR convergence for collaborative optimization**

**Table 9. Final configuration comparison**

| | Iterative Optimizers | RSM |
|---|---|---|
| $T_{SL}$ | 3,937,930 lbs | 3,966,746 lbs |
| e | 70.00 | 57.47 |
| r | 4.500 | 6.005 |
| $P_C$ | 210.88 atm | 210.88 atm |
| $I_{SP}$ | 472.10 s | 449.47 s |
| $A_E$ | 400.81 sq. ft. | 318.93 sq. ft. |
| $T/W_E$ | 67.53 | 73.66 |
| $T_{VAC}$ | 4,786,110 lbs | 4,641,680 lbs |
| MR | 7.11 | 7.77 |
| $W_G$ | 3,281,608 lbs | 3,305,621 lbs |
| $S_{REF}$ | 5725 sq. ft. | 5179 sq. ft. |
| $T_{SL}/W_G$ | 1.20 | 1.20 |
| $W_{DRY}$ | 346,297 lbs | 315,695 lbs |

## CONCLUSIONS

Final configuration results and computations times are summarized in tables 9 & 10 respectively. Additionally, table 11 summarizes the constraint and variable breakdown for each of the methods.

The answer obtained by the RSM is better than that of the iterativeoptimizers method. By allowing the indiviual CA's to freely optimize their local objective functions without any regard for the system level objective, the answers obtained from the iterative optimizers method and RSM are inherently suboptimal. However, the RSM results are far better due to the fact that the ability to vary the most sensitive variables is removed

**Table 10. Computation comparison**

| | Iterative optimizers | RSM |
|---|---|---|
| Calls to Scores | 6 | 90 |
| Calls to Post | 6 | 90 |
| Calls to Weights | 6 | 90 |
| System Iterations | 6 | 6/per run |
| Approx Time | 9 min | 1.5 hours |

**Table 11. Variable & constraint summary**

| **Propulsion** | | | |
|---|---|---|---|
| | SI | RSM | CO |
| Inputs | 1 | 3 | 4 |
| Local | 3 | 1 | 4 |
| Constraints | 1 | 1 | 1 |

| **Performance** | | | |
|---|---|---|---|
| | SI | RSM | CO |
| Inputs | 5 | 5 | 6 |
| Local | 5 | 5 | 10 |
| Constraints | 4 | 4 | 4 |

| **Weights & Sizing** | | | |
|---|---|---|---|
| | SI | RSM | CO |
| Inputs | 4 | 4 | 7 |
| Local | 2 | 1 | 5 |
| Constraints | 2 | 2 | 1 |

from the local CA's. The detriment of providing the CA's free control over all their variables is easily seen by looking at how the iterative optimizers method drove the expansion ratio to its maximum limit and the mixture ratio to its lower limit. Although this configuration results in a very efficent engine, it is obviously in conflict with the objective of minimum dry weight.

As the collaborative approach uses a system level optimizer and directs the optimization of the CA's, it should not suffer from the same problems as the RSM and iterative optimizers. Unfortunately, satisfactory results using collaborative optimization have not as yet been obtained due to various numerical problems. Work to correct these problems is still in progress.

## FUTURE WORK

The limited results obtained from the Collaborative Optimization study provide the opportunity to explore several alternative methods. It has been hypothesized that the use of Sequential Quadratic Programming as a system-level optimizer my present problems as the J terms become significantly small. Being a gradient-based optimizer, SQP encounters difficulty once the derivates approach zero.

As previously mentioned, certain variables in the J terms are extremely sensitive and hold the majority of the error. To alleviate this, it is proposed that the errors of these sensitive variables be weighted to offset their effect on the J term. Other alternatives include choosing an optimizer that doesn't use constraints in conjunction with penalty functions and using the advanced capabilities of POST to determine gradients for performance.

## REFERENCES

1. Way, D. W., Olds, J. R., "SCORES: Web-Based Rocket Propulsion Analysis Tool for Space Transportation System Design," AIAA 99-2353, 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Los Angeles, CA, June 20-24, 1999.

2. Phoenix Integration, URL: "http://www.phoenix-int.com"

3. Brauer, G. L., Cornick, D. E., and Stevenson, R., *Program to Optimize Simulated Trajectories (POST)*, Final Report for NASA contract NAS1-18147, Martin-Marietta Corp., September 1990.

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

5. R. D. Braun, *Collaborative Architecture for Large-Scale Distributed Design*, PhD Thesis, Stanford University, June 1996