Defense Technical Information Center Compilation Part Notice

ADP014189

TITLE: Collaborative Design Environment for Space Launch Vehicle Design and Optimization

DISTRIBUTION: Approved for public release, distribution unlimited
Availability: Hard copy only.

This paper is part of the following report:

TITLE: Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation [La reduction des couts et des delais d'acquisition des vehicules militaires par la modelisation avancee et la simulation de produit virtuel]

To order the complete compilation report, use: ADA415759

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP014142 thru ADP014198

Collaborative Design Environment for Space Launch Vehicle Design and Optimization

Mark D. Stevenson
2856 G Street, B79
Wright Patterson AFB, OH 45433, USA

Alicia R. Hartong
1970 Monahan Way, B11A R021
Wright Patterson AFB, OH 45433, USA

Jeffrey V. Zweber, Amarshi A. Bhungalia
2210 8th St., B146 R301
Wright Patterson AFB, OH 45433, USA

Ramana V. Grandhi
College of Engineering and Computer Science
Dept. of Mechanical and Materials Engineering
3640 Colonel Glenn Hwy.
Dayton, OH 45435, USA

The design of a hypersonic cruise or space launch vehicle is a large undertaking requiring the team effort of many engineers having expertise in the areas of aerodynamics, propulsion, structures, flight control, performance and mass properties. As the design takes shape, specialists are requested to design such things as the crew station, landing gear, interior layout, weapons location, and equipment installation. The completed vehicle design is a compromise of the best effort of many talented engineers. It should be clear that the design process is a complex integration effort requiring the pulling together and blending of many engineering disciplines.

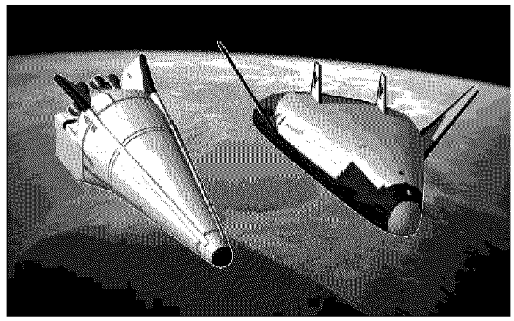

Figure 1: Trans Atmospheric Vehicles

Like all organizations, the Air Force is interested in conducting its vehicle studies as quickly as possible with as high fidelity an analysis as is feasible and with a proven, repeatable design and analysis process. This research is in support of an approach formulated by engineers at Wright Patterson Air Force Base who seek to integrate design and analysis tools into a collaborative, network-distributed design environment. The benefits of using an integrated design environment to reduce the time and potential errors associated with the transfer of data between design and analysis codes are well documented.[1,2] This research presents the integration of an initial set of space access and future strike vehicle analysis codes designed to improve the entire conceptual-level design process and documents the advantages of using the tools in a collaborative, network-distributed environment. This paper focuses on the design environment including geometry modeling, object design, discipline interactions, and design tools built for this effort including weight, propulsion, and trajectory analysis.

REUSABLE LAUNCH SYSTEMS

Both the US Air Force and NASA have indicated that next-generation reusable launch systems are needed within the next few years. Indications of the area's high importance can be seen through funding of projects like the X-33 and Hyper-X experimental launch concepts. At this stage of the study program, similar technologies and vehicle concepts are being examined to meet both the space access and future strike requirements. Consequently, rapid assessment of a Reusable Military Launch Systems is becoming increasingly important. There is a large array of RMLS options and promising configurations must be selected quickly for higher fidelity analysis. Furthermore all proposals must be analyzed uniformly using the same base-lined analysis tools and objective constraints.

Paper presented at the RTO AVT Symposium on "Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation", held in Paris, France, 22-25 April 2002, and published in RTO-MP-089.

The initial user of the web-based, collaborative application for launch vehicle design is the Air Force's Reusable Military Launch System (RMLS) analysis team. The core of this team has members from five different organizations that are located in four different buildings at two different bases. The team focuses on capability assessment for both future strike and space access vehicles. The goal is to impartially judge RMLS designs without restrictions on mode of operations. These modes include Horizontal Takeoff-Horizontal Landing, Vertical Takeoff-Horizontal Landing, and Vertical Takeoff-Vertical Landing. The team will also judge vehicle configuration options such as air breathing vs. rocket based propulsion and Two Stage to Orbit vs. Single Stage to Orbit.[3,4] A better understanding of the RMLS design space will dictate future areas of research and development needed to increase the viability of promising configurations. Because of the distributed nature of the team, the initial method used to conduct analyses was to pass files manually via email and a web site bulletin board. This system is sufficient for the relatively small team. However it has obvious areas of inefficiency in communication. Moreover there exists the possibility of errors being introduced due to data translation and loss of configuration control. An improved design and analysis process was needed to prevent these potential errors and to allow the RMLS team to efficiently interact with technology experts from other government agencies, industry and academia.

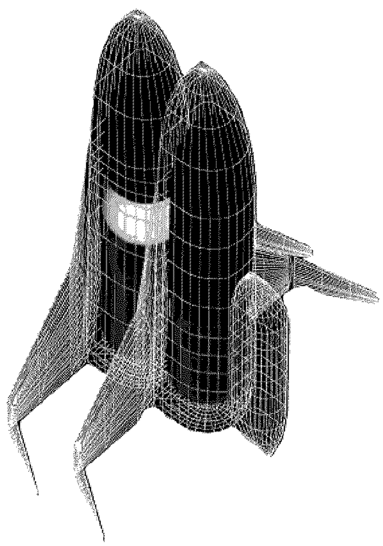

Figure 2: Reusable Military Launch System

The current vehicle under study is an in-house design of a fully reusable TSTO. The design (Figure 2) is a departure from the Bimese concept of identical booster and orbiter stages arranged "piggy-back" with an external payload mounted on the orbiter. The in-house concept consists of a booster and orbiter with a similar aeroshape but internal differences. Future vehicles under consideration include a stacked (serial burn) version of the Bimese concept and an air-breathing design.

LAUNCH VEHICLE DESIGN ENVIRONMENT

The conceptual-level design process for hypersonic and space access vehicles is dominated by geometric modeling, aerodynamics, aerothermodynamics, engine performance (air-breathing or rocket) analysis, trajectory simulation, mass properties analysis and cost modeling. This process is shown in Figure 3 as a design structure matrix. A design structure matrix is used to graphically display the interactions between the various disciplines in a design process. [5] Each block in Figure 3 represents a different analysis code. These codes could be further associated with different engineers, different computers or even computer platforms.

The process starts with a designer formulating a possible outer moldline of the vehicle. This can be done anywhere from a "back of the envelope" sketch to lofted model in a CAD package. From the geometry, the aerodynamic, propulsion and mass properties analysts generate their models. Using the results of these analyses, a set of trajectories or missions is simulated to determine if the concept vehicle will meet its requirements. Then, from the results of the trajectory simulation, an aerothermoelastic analysis can be performed to determine the heating loads on the vehicle and subsequently

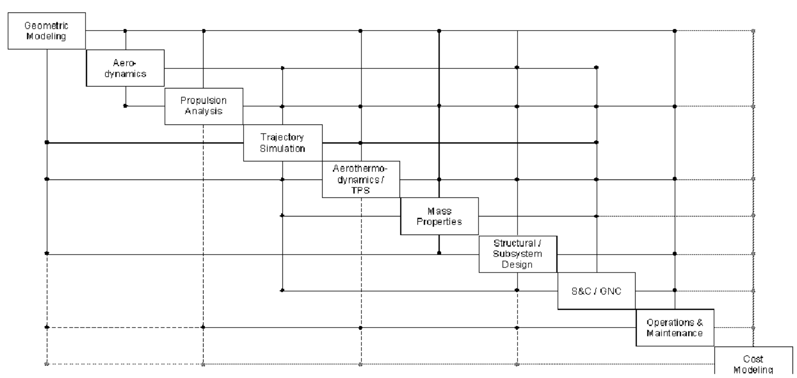

Figure 3: Design Structure Matrix

size the thermal protection system (TPS) and internal structure. The TPS size affects the geometric model by reducing the available internal volume for fuel and payload. Conventionally, this design cycle is repeated, varying geometric parameters, until the size and shape of the vehicle converges to the smallest vehicle that will perform a given set of missions.

One of the well-known shortcomings of this process is that it takes far too long for the design to progress to a point where operations, logistics and life cycle cost analyses are performed.[6] The long-term goal of this research is to demonstrate that, by integrating all the launch vehicle design disciplines into a collaborative design environment, the design data can be fed to the cost and operations disciplines sooner. In addition, by capturing the design process, the results of these analyses can be fed back to the conventional, conceptual design disciplines. By removing the manual data transfer steps, more design iterations can be accomplished in the same amount of engineering time.

The current status of the project is that some tools for the geometric modeling, aerodynamic analysis, propulsion analysis, trajectory simulation and mass properties disciplines have been integrated. Structural weight and aerodynamic results are calculated directly from an initial geometry specification, with the total weight being determined by adding the thermal protection system (TPS), propulsion system, payload and propellant weights. These three disciplines (mass properties, aerodynamics, and propulsion) provide the data that is needed by the trajectory simulation code to determine if the vehicle meets mission requirements (altitude and inclination angle). Finally, an iterative process is employed to vary the vehicle's fuel fraction ratio, and consequently the overall size of the vehicle, to correctly size the vehicle and propulsion system for a specified mission, or to determine that a specific vehicle class will not work for the required payload and orbit.

The Adaptive Modeling Language

For this effort, the Adaptive Modeling Language (AML) developed by Technosoft Inc., was selected as the design-modeling environment. AML is a framework for Knowledge Based Engineering that provides the ability to capture the launch vehicle design and analysis process and manage the data transfer between codes. It is by using the logical functions and calculations in AML, to capture process knowledge and design intent, that the significant timesavings in performing repeated analyses on a family of designs can be achieved. Previous research has demonstrated this knowledge capture in AML models for structural analysis and cost modeling. [7] The current version of AML has a wide variety of features that make it well suited for developing applications to capture a complex, multidisciplinary design process.[8] Perhaps the most important and least unique feature of AML is that it is an object-oriented language. A consensus has been reached in the software industry that object-oriented programming is vital for ease of software development and reuse. By applying the object-oriented paradigm to engineering models, AML allows the reuse of these models (objects). A well-formulated model will represent the component in general, parametric terms. For instance, the 747 and F-16 have very different wing shapes and sizes, but both wings can be represented by the same set of parameters (i.e., aspect ratio, root chord, taper ratio, airfoil section, twist distribution, dihedral and sweep angle). By developing a wing model this way, the same object can be used to model both aircraft.

A second important feature of AML (inherited from its Allegro Common LISP infrastructure) is its hierarchical, dynamic part-model. This feature is what makes AML "adaptive"; that is, models do not need to be recompiled to change the object hierarchy. The subobjects can be added interactively or specified in the definition of the class that was chosen for the top-level model (or in the definition of classes that were added as subobjects). This capability also allows objects and their properties to be added, edited or deleted independent of the order of instantiation. Included in the hierarchical structure is the Unified Part Model paradigm. This paradigm allows the model of a given component, the wing for example, to contain all the data about the wing that will be required by the various analyses. For instance, the wing model could contain a panel aerodynamic model, which would be used for low-speed calculations; a finite-element model of the wing box, which would be used for structural analysis; a second aerodynamic model that includes control surfaces, which would be used for stability analysis; and a thermal model, which would be used to size the wing's thermal protection system.

This modeling paradigm allows the model to grow as the design matures and new parts are created or new analyses are required. By keeping all the design information in a unified model, the "bookkeeping" of the data can be simplified. AML has built-in dependency tracking and demand-driven calculation capabilities to assist in this data management. *Dependency tracking* is important for ensuring that each discipline of the model is working with the current set of design parameters. With a manual design or configuration management system, it is easy for the various discipline specific models to get out of sync. AML automatically builds and maintains a list of dependencies. This list is updated as the objects are instantiated or deleted; or as the formulas associated with a property are changed. AML's dependency tracking also works in the other direction. That is, AML maintains a list of the properties that are affected by each property. The *demand-driven calculation* feature is complimentary to the dependency tracking capability. While the dependency tracking capability notifies all the parts of the model that have been affected by a change in a design parameter, the demand-driven calculation feature ensures that the only calculations to be performed are those needed for the current item of interest.

The last important feature of AML that will be covered here is the Graphical User Interface (GUI) included in AML. AML provides the powerful ability to automatically generate GUI's from an objects coding, eliminating the need for a designer to specifically develop a GUI structure. When writing an object, a developer specifies which parameters should be included in the user interface with only minor modifications in the parameter classes used. AML builds the GUI's during runtime. This eliminates a substantial volume of required coding from an object and reduces object development time. Additionally, when a design is being run over a network, form information does not need to be transmitted because the forms are part of an objects code, and generated on each individual client machine.

Collaborative design requires a distributed set of users running various analyses, possibly hundreds of miles apart. Bringing together a set of analysis tools under a unified environment is only a first step in achieving a fully integrated collaborative environment. Because of the large number of disciplines, an application would be extremely inefficient if limited to a single computer. A new feature being added to AML, under an Air Force Dual Use Science and Technology program termed Web-Based Design Environment (WDE) allows users to be distributed over a wide area network.[9] Users log into a server that contains the vehicle model via a standard WDE browser. Vehicle geometry modification and analysis can then be performed real-time over the network. The browser is platform independent and can access analysis codes on any computer across the entire network. By allowing pieces of the model to reside on different machines, each computer can specialize in a single discipline. This reduces the number of analysis codes needed and can save money by reducing the required software licenses and simplifying the system administration. The tool only passes parameter values of the model, which means that a high-fidelity graphical model requires a very small bandwidth.[10] Security and design configuration control issues are addressed within the modeling environment.

DESIGN DISCIPLINES

Design begins with geometry or an array of geometric considerations. Preferably the geometry object should be fully parametric, allowing the user to change shape into any other shape under consideration. However, the author has found that a single geometric object capable of all design configurations is not desired. The large number of parameters (e.g. number of fuselage cross sections, cross section geometries, cross section positions, wing type, and wing location) for a design forces a user interface to be complicated and unwieldy. There are a number of design possibilities, creating a huge array of very different vehicle designs. A series of parametric models tailored for each vehicle class (e.g. 2-D air-breathing and rocket based lifting body) is being created as part of the ongoing RMLS research. Using only a few parameters these models can be rapidly changed to any vehicle design within a given class. When a desired vehicle falls outside a class, other classes may have to be used or built to accommodate the new vehicle. A new parametric model takes about two weeks to create. The Bimese parametric vehicle class developed in conjunction the RMLS team at WPAFB for the current research with the help of TSI is shown in Figures 2, 8, 10, and 12. The model is able to be non-photographically stretched for vehicle sizing and includes links to previously mentioned analysis tools. The geometry objects developed for this class will also be used for future horizontally stacked configurations.

Rocket Engine Design Code

A focus for any launch vehicle design is centered on the propulsion system. Engine selection impacts several crucial design decisions including fuel type and associated fuel tank selection. Fuel fractions for SA/FS vehicles can be as high as 90% so fuel selection becomes a very important issue. Hydrogen fuels have a higher ISP (a measure of the overall energy contained in a rocket) but are less dense and require cryogenic tanks. Hydrocarbon fuels require smaller fuel pumps that reduce the size and weight of the rocket engine. Trade-offs for both fuel types require detailed analysis to determine the best fuel type for a specific rocket configuration. The importance of the propulsion system requires a rapid rocket design and performance analysis tool for vehicle modeling. The Parametric Rocket Model [11], developed at Wright Patterson AFB, uses a historical data trend approach primarily taken from “Design of Liquid Propellant Rocket Engines”.[12]

The author chose to incorporate the simple Parametric Rocket Model into the AML environment because of its simplicity and fast run times. Additionally it provides information required for other analysis codes with a minimal input. The basic procedure for designing the propulsion system using the Parametric Rocket Model is as follows:

1. Select a specific rocket type and fuel, the characteristic velocity and combustor pressure, ratio of specific heat, propellant flow per unit throat area and characteristic combustor length based on previous engine designs are set. This represents the performance level of the engine class.
2. Given the specified nozzle expansion ratio(s) and nozzle type (1 position, 2 position, or dual bell) a nozzle thrust coefficient is calculated as a function of altitude.
3. Thrust at a reference throat area is then calculated as a function of altitude.
4. Given the specified thrust at a specified altitude, a scale factor is calculated that is applied to the reference thrust function to obtain the specified thrust.
5. The scale factor is also applied to the reference throat area to properly scale the geometry.

An example of how engine performance parameters are calculated are the equations used for exit nozzle pressure. The theoretical nozzle expansion ratio is calculation using Equation 1, where γ is the specific heat for a given fuel type, p_e is an assumed exit pressure and p_{cns} is the chamber (nozzle stagnation) pressure for a given fuel. This doesn’t include boundary layer displacement correction, heat transfer or shifting γ effects, but it is close to actual values. The exit pressure is then calculated using Equation 2, where ε is the desired expansion ratio. Equations 1 and 2 are related to each other so a Newton-Rhapson iteration method is used for convergence. The iteration is performed on $1/\varepsilon$ because it is more linear than ε.

$$\varepsilon_{th}(p_e, p_{cns}, \gamma) := \frac{\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}}\left(\frac{p_{cns}}{p_e}\right)^{\frac{1}{\gamma}}}{\sqrt{\frac{\gamma+1}{\gamma-1}\left[1-\left(\frac{p_e}{p_{cns}}\right)^{\frac{\gamma-1}{\gamma}}\right]}}$$

Equation 1: Theoretical Expansion Ratio

$$p_e(\varepsilon, p_{cns}, \gamma) := \mathrm{root}\left(\frac{1}{\varepsilon} - \frac{1}{\varepsilon_{th}(p_e, p_{cns}, \gamma)}, p_e\right)$$

Equation 2: Exit Nozzle Pressure

A plot of engine performance (given by thrust coefficient) for several nozzle types vs. altitude is plotted in Figure 4. The plots are characteristic of typical engine performance curves. The discontinuity in the graph for the Space Shuttle Main Engine (SSME) 150 2p (two position) nozzle is a result of moving a secondary nozzle into position at a specific altitude. The method has been correlated with advanced LH-LOX and RP-1-LOX engines. This simple model calculates thrust and Isp as a function of altitude, weight and geometry of the engine based on thrust at a specified altitude, rocket type, nozzle type (1-position, 2-position, or dual bell nozzle), and expansion ratio.

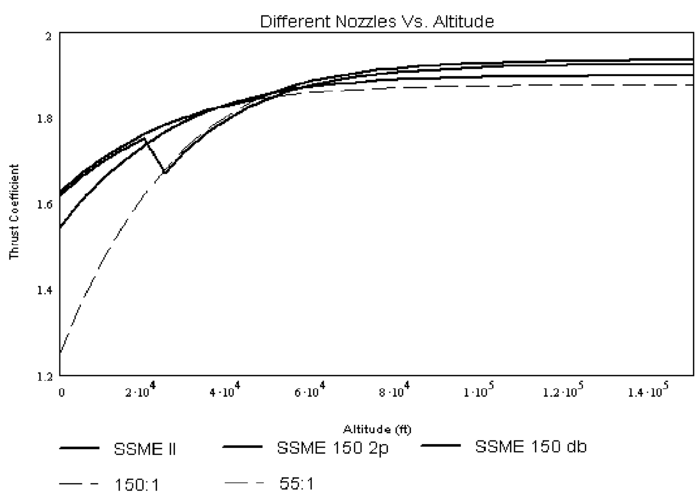

Figure 4: Engine Performance

Simply changing one parameter such as fuel type can radically change the engine geometry; Isp, thrust, and weight are also affected.

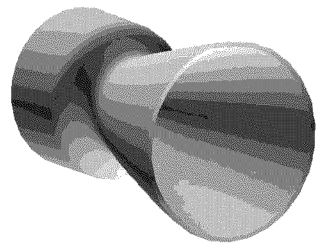

Figure 5: Engine Geometry

Weight Analysis

Weight analysis is a crucial aspect of RMLS design. Too much vehicle dry mass and fuel fractions will never be high enough to get a payload to orbit. Additionally, weight and aerodynamic parameters such as G-loading, calculated from trajectory and aerodynamic analysis, drive structural sizing.

Weight analysis equations tend to be strictly proprietary information tightly held by their parent organizations. Consequently no commercial off the shelf weight estimation software was found that suited the RMLS design group. Weight estimation software should be simple, use available information associated with the model and track the physics well. To build weight estimation software, engineers at WPAFB compiled historical trends in launch vehicle design as a way to predict future vehicle designs. Data was compiled from Air Force Flight Dynamics Lab reports produced in the 1970's and 1980's including the Space Shuttle, NASP and BETA vehicle.[13,14,15,16,17,18]

Weight Estimation

The weight analysis software was written directly into the AML environment, and highly coupled with the geometry. Component weights are generally calculated from a vehicle's gross weight, empty weight or geometry (also a function of gross weight). For example, Figure 6 plots the relationship of tail area with tail weight. The relationship is almost linear for a variety of vehicles. The vehicles used for this comparison are the XB-70 Valkyrie (Mach 3 USAF experimental bomber 1964-1969), STS (Space Shuttle), F-106A Delta Dart (supersonic USAF operational interceptor 1956-1960), B-58A Hustler (Supersonic USAF Operational Bomber 1960-1970), F-4 MK-2 Phantom (Supersonic USAF Operational Fighter 1965-1992). The actual relationship used for the weights equation (Equation 3) was chosen to match the Space Shuttle data. Because this weights equation is based on geometry, which is based on gross vehicle weight, iteration of the overall vehicle is required to close the vehicle size and weight calculations. Component weights can be known values, such as an electrical system power supply that has been set at 770 lbs based on Space Shuttle requirements. Setting a weight to a deterministic value is equivalent to pulling a known power supply off the shelf and adding it to the model. Component weights can also be a simple equation or expanded into geometrical objects depicting sub-system placement. Components can be further broken down into constituent parts for increased model fidelity. The basic procedure for calculating an overall vehicle weight using the system is as follows:

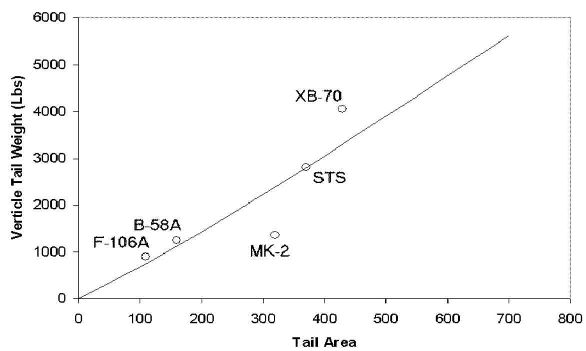

Figure 6: Historical Weight Trends

$$Weight = 5 * Svt^{1.09} * 0.89$$

Equation 3: Tail Weight Estimation

1. Guess the empty weight fraction
2. Calculate component weights based on initial guess
3. Sum the weights and determine difference in empty weight calculations
4. Size the vehicle and adjust the empty weight guess
5. Iterate until vehicle closure

Once the weight estimation and sizing procedure are complete, the model is run through trajectory analysis that is used to update the propellant fraction. The weight estimation procedure is then rerun iteratively with trajectory analysis until overall vehicle closure. This research has discovered that only two to three iterations are required to close the vehicle.

Thermal Protection System Weight Estimation

As part of the weight estimation process, Thermal Protection System (TPS) weight must be addressed. The model uses a simple water-line scheme to estimate what TPS types are needed in what areas. With the knowledge that the vehicle will re-enter the atmosphere at a specified angle (i.e., 30°) surfaces that are in direct line with or at specified increments from the stagnation points are calculated. With the knowledge that surfaces nearest the stagnation points will require the highest temperature TPS, a lookup table of TPS materials (based on Shuttle tiling) is used to place tiles in specific regions. The density and thickness of the tiles is then used to calculate the entire weight of the TPS.

A more physics based approach for predicting TPS design is currently under development through an Air Force Small Business through Innovative Research (SBIR) program. Using high fidelity aerodynamics and heating analysis calculated directly from the geometry of the rocket design and its trajectory profile, the transient heating profile will be coupled with a TPS optimization routine. The thickness of the TPS is varied so that a maximum temperature on the inner rocket structure is not exceeded throughout the trajectory. The heating loads are then applied to the Finite Element model of the inner vehicle structure for sizing. The updated vehicle weight can then be sent back to trajectory analysis in an iterative cycle until vehicle closure. This will not only yield higher fidelity TPS design, but will also include the transient effects in the heating profile. Currently most TPS designs are sized to the point in the trajectory that yields the highest temperature; this overestimates the required TPS and consequently increases the weight of the vehicle.

Weight Estimation Error

There are errors in the weight estimation routines. The vehicles currently being analyzed are roughly five percent under weight, based on historical vehicle designs. The additional weight is accounted for using a weight correction factor, but additional work needs to be done to model vehicle weight better. Five percent under estimation is a considerable factor considering that the RMLS type vehicles may have growth factors of 30 or more. The higher fidelity methods previously discussed could be used to refine the weight synthesis equations for future increased model accuracy. Additionally, members of the RMLS team have modeled aerospace partners design to compare and verify the analysis. Results have shown a good comparison between the reports. Additionally, the comparison found that a few parameters in the model weight were not feeding back into the weight estimation scheme. Future studies will allow higher confidence in the model. Despite these errors, the current weight estimation routines are a good start to capturing vehicle weight, and accurate enough for the level of fidelity desired.

The weight estimation software developed is only for preliminary design. The author knows there are dangers to base weight estimation on historical data trends. This is especially true when the only data point that has been built and flown is the Space Shuttle that was designed for an immense 80,000-pound payload and is an operational nightmare. The Space Shuttle is not a good data point, but it is widely used because it is the only point available. Future work may incorporate higher fidelity tools, which will benefit from the vehicle-sizing starting point this tool gives. Additionally, the physics in the higher fidelity tools could then be captured to increase the accuracy of the preliminary weights equations developed.

Trajectory Analysis

As previously discussed, trajectory analysis is an integral part of RMLS design. The two main trajectory analysis codes used within the industry are OTIS (Optimal Trajectories by Implicit Simulation) and POST (Program to Optimize Simulated Trajectories). Within the aerospace industry, the author has found that new codes are not easily accepted, and various organizations (even within the RMLS team) live and die by their selected code with no thought of change. Consequently both codes have been integrated into the environment using program wrappers. However, the author favors OTIS because its solutions have yielded better results, coupled with the ability to use more parameters and constraints.

OTIS 3.0 is a FORTRAN77 program for simulating and optimizing the point mass trajectories of a wide variety of aerospace vehicles. The version used at Wright Patterson AFB was recently compiled for

use on NT-based windows machines. The most advanced simulation uses implicit integration to generate an open-loop optimal control of a prescribed vehicle.[9] OTIS was designed more like a math program; give it a series of parameters (possibly hundreds), constraints and objectives, and it will solve for the optimal mathematical solution. POST is also a FORTRAN 77 program for a generalized point mass with discrete parameter targeting and optimization.[19] POST behaves more like a traditional trajectory program; give it a series of parameters (under 100), constraints, objectives and a trajectory that the user thinks is good, and it will yield a slightly better trajectory. POST has the benefit of being fast but is hampered by only running in DOS mode on PC-based machines. The POST integration uses a LISP function to traverse the tree to collect data, reformatting it into an input text file required by POST. The text file must then be sent to the trajectory analyst to run POST and send back the updated fuel fractions.

The OTIS 3.0 integration currently only contains the specific information relevant to a particular RMLS class of vehicles. The properties allowed in a specific model are tailored such that a limited set of trajectories can be performed, reducing the incredibly large array of options OTIS 3.0 allows. This reduces the strain on a user of the tool by reducing the number of properties understood and checked during program execution. The few properties relevant to a given design are easily accessible within the design environment. However, the initial trajectory file relevant to a particular vehicle class is required to be generated by an expert user of OTIS 3.0. Trajectory analysis is extremely complicated, and eliminating the expert entirely from the design process would be impossible. Vehicle configuration properties such and aerodynamics, weight, engine propulsion are automatically formatted into the OTIS format, and the updated fuel fractions are automatically read back into the collaborative environment for automated iterative design.

Aerodynamic Analysis

The aerodynamic analysis application used for the Bimese trade studies was Missile DATCOM. DATCOM requires geometry to be broken down into simple known components and then uses empirical equations of the known shapes to calculate the desired aerodynamic coefficients for the overall vehicle. Consequently only simple geometry can be modeled using DATCOM. Multiple bodies also pose a problem becuase they are not handled in DATCOM. The author chose to model the orbiter and the booster separately, with the payload treated as a protuberance on the orbiter. The drag of the orbiter and booster is then summed. The calculated drag using this method ignores whatever interference exists between the wings, which adds to the drag calculation. But this decreases at higher Mach numbers and is not unreasonable to ignore. To check this assumption, a CFD model is being run for the concept. However, the results are not expected soon because of the huge computational expense of CFD analysis.

The analysis shown in Figure 7 demonstrates the expected drag rise going through Mach 1.0; the large increase is a result of the NACA 0012 airfoil chosen for the Bimese concept. The analysis is consistent with predictions on how the model should behave, allowing confidence in the aerodynamic analysis.

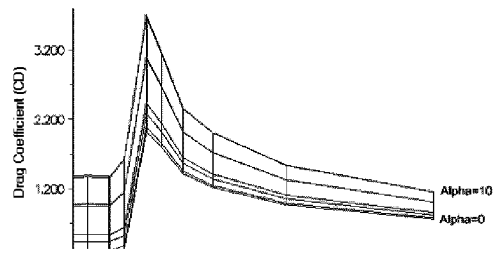

Figure 7: Coefficient of Drag Calculation Using DATCOM

For a sanity check a more detailed analysis could be performed using PANAIR, an example of an analysis of the Bimese concept is shown in Figure 8. PANAIR is a linear aerodynamic solver using the technique of boundary elements (commonly referred to as aerodynamic paneling). Surface geometry is "body-fitted" with an array of quadrilateral panels. Continuous surface singularities (both sources and doublets) are distributed using a number of schemes to meet a number of needs.[20] The program is accurate but requires longer run times, and is not applicable to the quick trade studies desired for the RMLS team. Additionally, PANAIR requires a continuous structured body grid that is difficult to model around protuberances such as wings in an automated fashion. The RMLS Bimese model was not constructed with PANAIR in mind so the wings could not be included in the PANAIR analysis. Consequently, only the body is analyzed in Figure 8 and the analysis cannot be compared with DATCOM. Both aerodynamic analysis objects contain information on how to break the smooth geometry of the model into their respective application inputs. No additional user work is required to run the analysis within the limits of the Bimese concept.

Future work will include adding ZONAIR to the list of aerodynamic analysis tools included in the environment. ZONAIR is a panel method aerodynamic solver based on ASTROS for very accurate results with limited computational time. A benefit of ZONAIR is that meshing can be unstructured, allowing input grids to be automatically generated. Additionally, multiple wingsets will be able to be modeled.

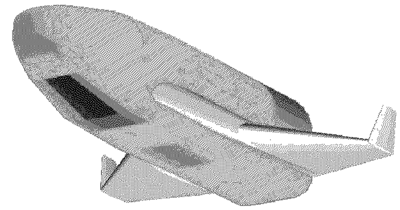

Figure 8: PANAIR Pressure Distribution on RMLS Bimese Vehicle

EXCERSIZING THE MODEL

The majority of the research has focused on the design environment development, and, as a result, the majority of this research is concerned with the environment and associated analysis modules. However the environment is only a foundation for rapid trade studies. Using this tool, the author performed various trade sweeps of the Bimese concept. The vehicle sizing routine incorporating weights and propulsion takes 60-180 seconds (sizing both the orbiter and booster) running on a Pentium III processor with 500 Mbytes RAM. The time difference depends on how many sizing iterations are required (which depends on how close original model sizing guess is to final design). Initial trajectory analysis using POST must be run offline because of the limitations of POST (which must be run in DOS mode), so trajectory and its required aerodynamics analysis are not run in an automated fashion. The input file required for the automated OTIS 3.0 analysis has recently been built and will be used to run through the series of designs the RMLS team wants to look at. With the limited number of analysis tools incorporated (weights, aerodynamics, propulsion, and trajectory) only a few trade study parameters can be considered. But the parameters considered are critical to design formulation. Trade study parameters able to be handled by the model include payload sizing, thrust to weight ratio, fuel selection for both booster and orbiter, wing thickness, rocket nozzle type, and staging velocity. The author will limit discussion to the first three trade studies mentioned.

Payload Sizing

Payload size comes from mission requirements. The payload size trade study performed shows what a top-level mission change will cost in terms of vehicle weight for a given design. In this study, the author changed the payload weight from 4k to 64k pounds, sized both the orbiter and booster vehicles and plotted the resulting overall vehicle fractions in Figure 9. In this plot, the propellant fraction (Pf) plotted is 1-Pf (i.e., about 76% of the vehicle weight). If the ordinate was scaled to one, the entire area between the Pf curve and one would represent the propellant fraction. Payload fraction is the difference between propellant fraction and empty weight. The empty weight plotted is the true empty weight of the vehicle. The increase in the empty weight fraction at the lower vehicle gross weight is largely a result of nearly constant TPS weight, resulting in greater weight fractions. The propellant fraction was held constant at 76% for the payload sweep; the slight decrease seen is a result of

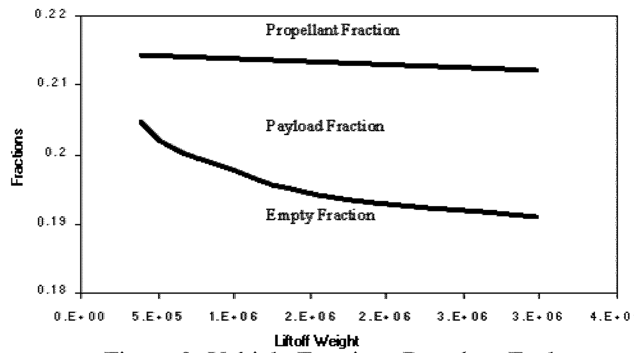

Figure 9: Vehicle Fractions Based on Fuel

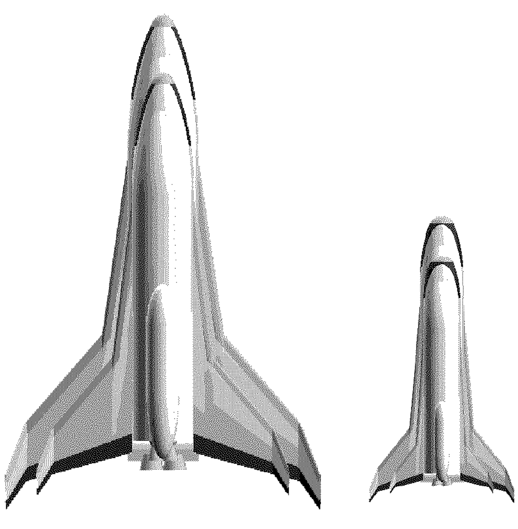

Figure 10: Payload Sizing Comparison

the weight equations not summing the weights properly. Within the range of payloads that were analyzed, greater vehicle efficiency is realized with larger payloads. At a lower liftoff weight, no payload is able to fit within the vehicle.

The vehicles at the extremes of the analysis (4k and 64k pound payloads) are shown in Figure 10. Notice that the sizing is not photographic. The wings grow at a faster rate in comparison with the fuselage. This is a result of the wings depending on the empty weight of the vehicle to maintain an acceptable rate of sink, span loading, and wing loading during landing conditions. The weight mainly depends on the size of the fuel tanks, the engine, and thrust structure, which depend on the fuel volume. Volume is a cubic function, so a small change in the fuselage will lead to a large increase in weight. The planform area only grows by the square of the increase in fuselage size, so if the fuselage grows by a factor of two, the weight increases by a factor of eight, and the wings increase a factor of four.

Thrust-to-Weight Optimization

In the second analysis sweep the thrust-to-weight ratio of the orbiter was varied and plotted as a function of vehicle dry weight (Figure 11). Thrust-to-weight and propellant fractions are closely related; higher thrust to weight ratios require less vehicle fuel fractions. Iteration was required with the trajectory analysis to solve for the fuel fraction. The results ranged from 77.5% at a thrust to weight ratio of 1.0 to 74.5% at a thrust to weight ratio of 1.8. Because the orbiter operates at high altitudes, the thrust to weight effect on vehicle weight is mostly a result of gravity losses (a factor of ΔV). Consequently there is only a small shift in dry weight and slight differences in vehicle design. The increase in dry weight at higher thrust to weight ratios results from limiting the G-loading on the vehicle. Additional increases in thrust only add additional engine weight to the vehicle. An optimum thrust-to-weight ratio is found to be between 1.3 and 1.7.

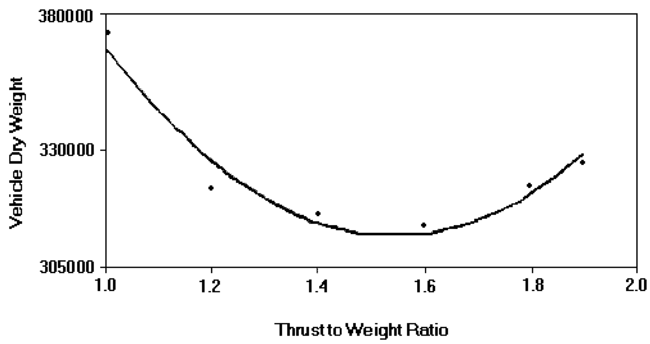

Figure 11: Thrust to Weight Optimization

Fuel Selection

In this study, the fuel of both the orbiter and booster where set to either a hydrocarbon (kerosene) or hydrogen based fuel with a LOX (liquid oxygen) for the oxidizer. The vehicles in Figure 12 show that the hydrogen-fueled concept is much larger than the hydrocarbon design. This is a result of the very low density of hydrogen, which requires a larger volume for the same propellant mass, increasing the volume required to store it. However, the vehicle dry weight is still roughly the

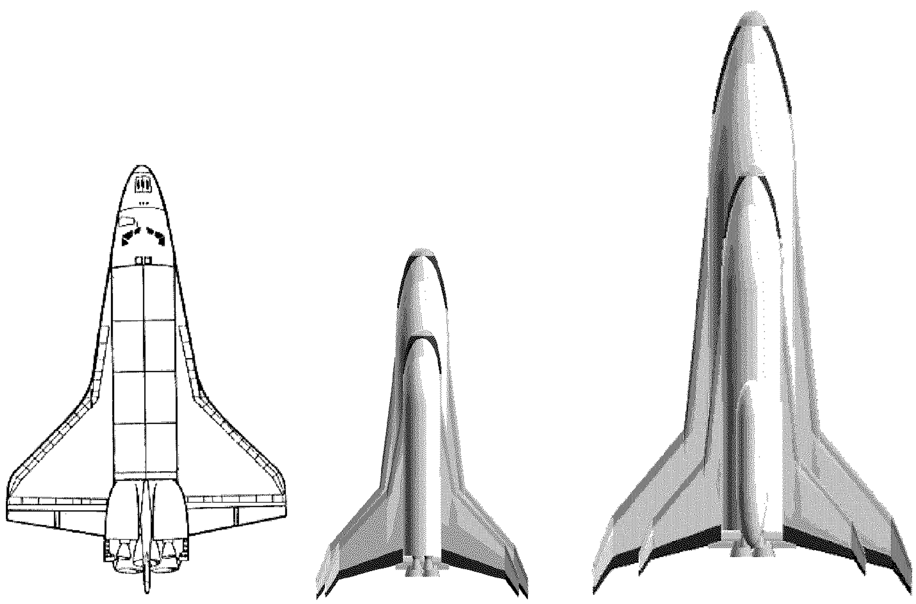

Figure 12: Fuel Selection Vehicle Comparison

same for both concepts. In fact, further studies have shown that the hydrocarbon design could be made lighter than the hydrogen concept by increasing the staging Mach number. The non-photographic scaling in the vehicle concepts results from the same sizing constraints of the payload trade study. Further analysis is required, particularly in the operations area, to determine the optimal staging mach number. The scaled picture of the Space Shuttle is included only as a yardstick as to the size of these concepts.

SUMMARY

The Reusable Military Launch System design environment under development at WPAFB has demonstrated dramatic design and analysis timesavings. The collaborative design environment currently incorporates parametric geometry, aerodynamics, mass properties, aeroheating, rocket propulsion, and trajectory analysis disciplines for the Bimese rocket configuration currently under study by the RMLS team. Continuing Research will incorporate additional analysis tools and optimization techniques for complete vehicle formulation. As the number of analysis objects grows, the usefulness and efficiency of the tool will increase. Further trade study analysis will define optimal vehicle design. These trade studies include:

- Load-factor
- Allowable wing loading
- Number of engines (engine out capability)
- Engine type
- Fuel selection
- Parallel vs. serial burn
- Staging Mach number

However, performing various single degree of freedom trade studies does not necessarily produce optimal results. The interconnectivity of the various disciplines found in the design structure matrix almost guarantees non-linear results that must be analyzed as a whole. Future work will include optimization across the disciplines to produce optimal vehicles for particular mission categories. The research reported here has created a design environment for rapid design analysis at a conceptual level. This work will be useful for assessing optimal design solutions and will dictate future air force requirements and direction for building RMLS vehicles. This is only the beginning of a much larger process. With additional object creation, higher fidelity analysis will be achieved.

BIBLIOGRAPHY

1. Bhungalia, A. A., Zweber, J. V. and Stevenson, M. D., “Integrated Aerodynamic and Geometric Modeling for Hypersonic Vehicle Design”, ASME paper DETC/DAC-14267, presented at the 26th ASME Design Automation Conference, Baltimore, MD, September 10-13, 2000.

2. Stevenson, M. D., Bhungalia, A. A. and Zweber, J. V., “Integrated Trajectory Analysis for Transatmospheric Vehicle Design”, AIAA paper 2000-4817, presented at the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, September 6-8, 2000.

3. Stanley, D., *et al.,* “Rocket Powered Single Stage Vehicle Configuration Selection and Design”, AIAA 93-1053, Feb. 1993.

4. Freeman, D., Stanley, D., Camarda, C., Lepsch, R., Cook, S., “Single Stage to Orbit: A Step Closer”, AIF 94-V3.534, Oct 1994.

5. Olds, J., Bradford, J., Charania, A., Ledsinger, L., McCormick, D., and Sorensen, K., "*Hyperion*: An SSTO Vision Vehicle Concept Utilizing Rocket-Based Combined Cycle Propulsion", AIAA paper 99-4944, presented at the 9th International Space Planes and Hypersonic Systems and Technology Conference, Norfolk, VA, November 1-5, 1999.

6. Prasad, B., *Concurrent Engineering Fundamentals: Integrated Product and Process Organization,* Volume I, Prentice-Hall, Upper Saddle River, NJ, 1996.

7. Zweber, J. V., Blair, M., Kamhawi, H., Bharatram, G. and Hartong, A., "Structural and Manufacturing Analysis of a Wing Using the Adaptive Modeling Language", AIAA paper 98-1758, presented at the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Long Beach, CA, April 20-23, 1998.

8. *Adaptive Modeling Language Reference Manual: Version 3.0*, TechnoSoft Inc., Cincinnati, OH, 1999.

9. "Web-based Design Environment", DUS&T Program Report to Congress, Air Force Projects for Fiscal Year 1999, http://www.dtic.mil/dust/cgr/af00cgr.htm#airforce19

10. Meltzer, Pete, "Web-based tool speeds up weapon system design", Skywrighter, WPAFB, OH, 26 May, 2000.

11. Livingston, John, "A Parametric Rocket Model", USAF, WPAFB, OH, 2000.

12. Huzel D., Huang D., Rocketdyne Div, N. America Aviation Inc., "Design of Liquid Propellant Rocket Engines", NASA SP-125, 1967.

13. Klich, P., MacConochie, I., NASA Langley, "Mass Estimating Techniques for Earth-To-Orbit Transports with Various Configuration Factors and Technologies Applied", SAWE Paper No. 1315, 38th Annual Conf. Of SAWE, NY, NY 7-9 May 1979.

14. MacConochie I., Klich P., NASA Langley, "Techniques for the Determination of Mass Properties of Earth to Orbit Transportation Systems", NASA TM-78661, June 1978.

15. NASA Johnson Space Center, System Definition Branch: Technology and Project Implementation Office, "Design Mass Properties II: Mass Estimating and Forecasting for Aerospace Vehicles Based on Historical Data", JSC-26098, November 1994.

16. HYCAD – Equations developed by ASC/XR 1985 and 1988.

17. Olds, John, Dissertation NC State 1993, Appendices A and B.

18. Forbis, J., Woodhead, G., Boeing Military Airplanes, "Conceptual Design and Analysis of Hypervelocity Aerospace Vehicles: Vol 1. Mass Properties, Part 1 Transatmospheric Vehicle Weights (TAVWTS)", WL-TR-91-6003, July 1991.

19. Powell, R. and Braun, R., "Program to Optimize Simulated Trajectories (POST)", NASA Langley Research Center, Hampton VA, Oct. 1997.

20. Blair, M., Moorhouse, D., Weisshaar, T., "System Design Innovation Using Multidisciplinary Optimization and Simulation", AIAA 2000-4705