

# Application of Design of Experiments and Surrogate Modeling within the NASA Advanced Concepts Office, Earth-to-Orbit Design Process

Mathew R. Zwack\* and Patrick D. Dees†

*Jacobs ESSSA Group, Huntsville, AL, 35806, United States*

James B. Holt‡

*NASA, George C. Marshall Space Flight Center, AL, 35812, United States*

Decisions made during early conceptual design have a large impact upon the expected life-cycle cost (LCC) of a new program. It is widely accepted that up to 80% of such cost is committed during these early design phases.<sup>1</sup> Therefore, to help minimize LCC, decisions made during conceptual design must be based upon as much information as possible.

To aid in the decision making for new launch vehicle programs, the Advanced Concepts Office (ACO) at NASA Marshall Space Flight Center (MSFC) provides rapid turnaround pre-phase A and phase A concept definition studies. The ACO team utilizes a proven set of tools to provide customers with a full vehicle mass breakdown to tertiary subsystems, preliminary structural sizing based upon worst-case flight loads, and trajectory optimization to quantify integrated vehicle performance for a given mission.<sup>2</sup> Although the team provides rapid turnaround for single vehicle concepts, the scope of the trade space can be limited due to analyst availability and the manpower requirements for manual execution of the analysis tools. In order to enable exploration of a broader design space, the ACO team has implemented an Advanced Design Methods (ADM) based approach. This approach applies the concepts of Design of Experiments (DOE) and surrogate modeling to more exhaustively explore the trade space and provide the customer with additional design information to inform decision making.

This paper will first discuss the automation of the ACO tool set, which represents a majority of the development effort. In order to fit a surrogate model within tolerable error bounds a number of DOE cases are needed. This number will scale with the number of variable parameters desired and the complexity of the system's response to those variables. For all but the smallest design spaces, the number of cases required cannot be produced within an acceptable timeframe using a manual process. Therefore, automation of the tools was a key enabler for the successful application of an ADM approach to an ACO design study.

Following the overview of the tool set automation, an example problem will be given to illustrate the implementation of the ADM approach. The example problem will first cover the inclusion of Ground Rules and Assumptions (GR&A) for a study. The GR&A are very important to the study as they determine the constraints within which a trade study can be conducted. These trades must ultimately reconcile with the customer's desired output and any anticipated "what if" questions. The example problem will then illustrate the setup and execution of a DOE through the automated ACO tools. This process is accomplished more efficiently in this work by splitting the tools into two separate environments. The first environment encompasses the structural optimization and mass estimation tools, while the second is focused on trajectory optimization. Surrogate models are fit to the outputs of each

---

\*Systems Engineer, Advanced Concepts Office, George C. Marshall Space Flight Center/ED04, Huntsville, AL, AIAA Member.

†Trajectory Engineer, Advanced Concepts Office, George C. Marshall Space Flight Center/ED04, Huntsville, AL, AIAA Member.

‡Lead Aerospace Engineer, Advanced Concepts Office, George C. Marshall Space Flight Center/ED04, Huntsville, AL

environment and are integrated via connection of the surrogate equations. Throughout this process, checks are implemented to compare the output of the surrogates to the output of manually run cases to ensure that the error of the final surrogates is at an acceptable level.

The conclusion of the example problem demonstrates the utility of the ADM based approach. Using surrogate models gives the ACO team the ability to visualize vehicle sensitivities to various design parameters and identify regions of interest within the design space. The ADM approach can thus be used to inform concept down selection and isolate promising vehicle configurations to be explored in more detail through the manual design process. In addition it provides the customer with an almost instantaneous turnaround on any “what if” questions that may arise within the bounds of the surrogate model. This approach ultimately expands the ability of the ACO team to provide its customer with broad and rapid turnaround trade studies for launch vehicle conceptual design. The ability to identify a selection of designs which can meet the customer requirements will have the potential to lower LCC of launch vehicle designs originating from ACO.

## Nomenclature

|               |   |
|---------------|---|
| <i>ACO</i>    | Advanced Concepts Office                      |
| <i>BBC</i>    | Baseball Card                                 |
| <i>ETO</i>    | Earth-to-Orbit                                |
| <i>GLOW</i>   | Gross Liftoff Weight                          |
| <i>INTROS</i> | Integrated Rocket Sizing                      |
| <i>LCC</i>    | Life Cycle Cost                               |
| <i>LVA</i>    | Launch Vehicle Analysis                       |
| <i>MBO</i>    | Burnout Mass                                  |
| <i>MER</i>    | Mass Estimating Relationship                  |
| <i>MSFC</i>   | George C. Marshall Space Flight Center        |
| <i>NASA</i>   | National Aeronautics and Space Administration |
| <i>OFR</i>    | Oxidizer to Fuel Ratio                        |
| <i>POST</i>   | Program to Optimize Simulated Trajectories    |
| <i>SLS</i>    | Space Launch System                           |
| <i>STS</i>    | Space Transportation System                   |
| <i>TLI</i>    | Trans-Lunar Injection                         |
| <i>VBA</i>    | Visual Basic for Applications                 |

## I. Introduction

Decisions made during early conceptual design have a large impact upon the expected life-cycle cost (LCC) of a new program. It is widely accepted that up to 80% of such cost is committed during these early design phases.<sup>1</sup> Therefore, to help minimize LCC, decisions made during conceptual design must be based upon as much information as possible.

To aid in the decision making for new launch vehicle programs, the Advanced Concepts Office (ACO) at NASA Marshall Space Flight Center (MSFC) provides rapid turnaround pre-phase A and phase A concept definition studies. The ACO team utilizes a proven set of tools to provide customers with a full vehicle mass breakdown to tertiary subsystems, preliminary structural sizing based upon worst-case flight loads, and trajectory optimization to quantify integrated vehicle performance for a given mission.<sup>2</sup> Although the team provides rapid turnaround for single vehicle concepts, the scope of the trade space can be limited due to analyst availability and the manpower requirements for manual execution of the analysis tools. In order to enable exploration of a broader design space, the ACO team has implemented an Advanced Design Methods (ADM) based approach. This approach applies the concepts of Design of Experiments (DOE) and surrogate modeling to more exhaustively explore the trade space and provide the customer with additional design information to inform decision making.

The following section will outline the current design process and tools utilized by the ACO team. Section III will discuss the approach for implementation of design of experiments and surrogate modeling. Sections III.A and III.B will detail the work done to automate and integrate the toolset. An example problem is given in Section IV to demonstrate the use of DOE and surrogate modeling during a design study.

## II. Current Process and Tools

The current Earth-to-Orbit (ETO) vehicle design process consists of three primary tools; INTROS, LVA, and POST. The first tool, INTEgrated ROcket Sizing (INTROS), is based in Excel and implements subroutines and functions written in Visual Basic for Applications to aid the user in vehicle definition and initial sizing. The INTROS tool first provides the user with options for vehicle setup including architecture definition, propulsion system parameters, and vehicle dimensions. Based upon the vehicle setup, mass properties are established using an existing list of typical launch vehicle systems, subsystems, and propellants. The mass calculations are completed using a set of mass estimating relationships (MERs) derived from historical launch vehicle and missile data. The mass calculation accuracy for existing and historical vehicles has been verified to be within 5%.<sup>3</sup> At the conclusion of a study, functions within INTROS are used to generate visualizations for inclusion on a “baseball card” (BBC) summary of pertinent information for each vehicle.

The Launch Vehicle Analysis (LVA) tool provides quick turnaround preliminary structural design and analysis. It is written in Visual Basic and utilizes direct solution methods based on material properties, load factors, aerodynamic loading, stress, stability, etc. Single combined worst-case loads are used to determine the primary structural mass of a launch vehicle concept, which is passed back into INTROS. The LVA tool has been in use at NASA MSFC for over 25 years.<sup>3</sup>

The final tool in the ACO process is the Program to Optimize Simulated Trajectories (POST), which is a FORTRAN 77 based legacy code originally developed at NASA Langley.<sup>3</sup> It uses a direct solution method to approximate the control function for an optimal trajectory.<sup>4</sup> POST provides the capability to optimize point mass trajectories for powered or unpowered vehicles near an arbitrary, rotating, oblate planet, and has been used successfully to solve a wide variety of atmospheric ascent and entry problems.<sup>5</sup>

The tools described above are implemented within the standard ACO ETO design process, which can be seen in Figure 1. The process typically begins by iterating with the customer to develop a set of Ground Rules and Assumptions (GR&A) for the study. This set of GR&A helps pare down the design space for the team and specifies any key requirements for both the vehicle being designed and the final deliverables of the study.

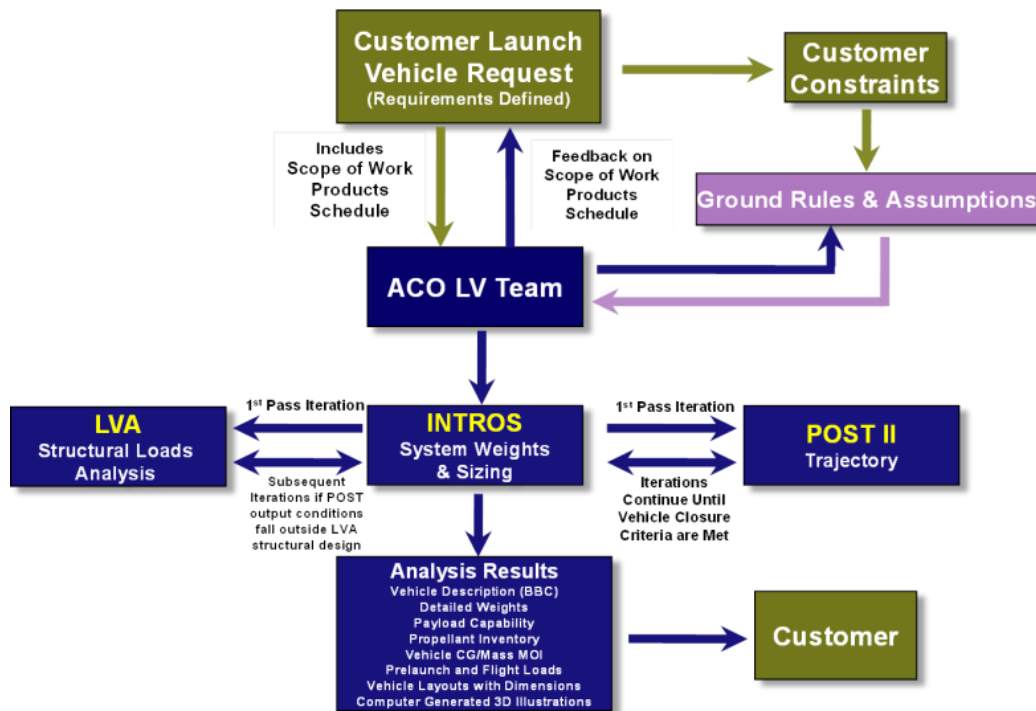


Figure 1: ACO Earth-to-Orbit vehicle design process

The design work begins with the definition of the vehicle architecture within INTROS, which provides an initial estimate for the vehicle masses. Using the vehicle burnout mass (mbo), propellant masses, and propulsion system information, an initial trajectory is flown in POST to determine the ascent environment. The vehicle dimensions and ascent loads from POST are then used in LVA to develop the primary structural masses. The updated structure masses are then fed back to INTROS, which then updates the overall vehicle mass properties. The vehicle is then re-flown in POST to determine its performance. Iteration is typically required to ensure that the loading used to design the structure in LVA is not violated during the ascent trajectory in POST. The iterative process between INTROS, LVA, and POST is performed for each vehicle concept within a study. After closure of a vehicle, the mass properties within INTROS are used to generate a BBC for the concept, which includes weights, payload capability, pertinent trajectory info, and vehicle dimensions. The analysis results are then provided to the customer at the study conclusion.

### III. Design Process Augmentation

The ACO ETO design process has been successfully applied on recent Heavy Lift Launch Vehicle (HLLV) and Small Launch Vehicle (SLV) studies.<sup>3,6,7</sup> It is particularly well suited for studies that look at a handful of discrete point designs. The manual nature of the process, however, tends to limit the scope of new studies based on analyst availability. For this reason, improvements were desired to augment the current process and provide a means for broader trade space exploration. In addition, these improvements looked to expand the ability of the team to make parametric vehicle trades and to visualize performance sensitivities to the metrics of interest.

In order to carry out such parametric trades and sensitivity analyses, an improvement in case throughput was required. The first step in improving throughput was to automate the design process, which is discussed in Section III.A. As discussed in Section III.B, the automated tools were then integrated to further improve throughput. Although automation and integration improve case throughput drastically, they do not necessarily enable full parametric trades. Therefore, the concept of surrogate modeling was applied in the augmented process to allow for nearly instantaneous analysis of vehicle concepts. Surrogate models also allow for visualization of vehicle sensitivities and enable probabilistic design. The implementation of surrogate modeling by way of DOE will be illustrated in the example problem in Section IV.

#### III.A. Automation

The first step in improving case throughput of the ETO tools was to automate their execution. This allowed for the completion of large sets of data without the supervision of an analyst. In addition, the automation allowed for the execution of the tools during periods of downtime for the analyst (i.e. nights and weekends).

Each of the tools were first automated separately to ensure they were being executed properly by the wrappers. Throughout the process, manual cases were compared to automated cases from each tool to verify correct operation. This exercise also helped refine the tool wrappers and optimize their runtime to improve case throughput.

##### *III.A.1. INTROS Wrapper*

The primary challenge of automating INTROS was the vast amount of options available to the user, which can be used to represent many unique vehicle configurations. During typical setup, the INTROS tool implements various pop-up dialogs to aid the user in identifying the appropriate settings to use for the new vehicle. In addition to pop-ups, various macros are executed along the way to help calculate dimensions, propellant loadings, etc. At the end of the tool setup, the appropriate MERs are identified by the user and are then assembled into a mass properties summary sheet. This sheet provides the vehicle mass breakdown to tertiary subsystems, if included. An additional complication associated with tool setup is that multi-stage vehicles are represented in separate Excel workbooks. Thus, a two-stage vehicle would have a single workbook for the core stage and another for the upper stage.

Due to the sheer number of options available to the user during INTROS setup, it was determined that the tool automation could not be practically accomplished for all unique vehicle configurations. Instead, it was decided that an INTROS expert would still need to be involved in workbook setup. The user defines the discrete architecture options for the vehicle and the automation executes the analysis and performs trades

on the continuous parameters associated with the concept. This allowed the team to keep the tool wrappers relatively simple, while still ensuring flexibility in the vehicles that can be represented.

The INTROS wrappers were written in Python due to its flexibility and relative ease of use. Multiple packages exist that provide functions for interacting with Excel documents and even executing VBA macros from Python. In addition, Python packages were used for multi-processing and easy integration of the wrapped tools.

The INTROS wrappers consist of a single Python file for each individual INTROS file; each Python file contains a function for execution of the specific INTROS file. This function first takes in a dictionary of vehicle input variables and places them in the appropriate cell in the Excel file. This list needs to be identified prior to execution of a study and is based upon the study's GR&A. After placing the tool inputs, the Python wrapper executes macros built into the Excel document to complete the analysis. A list of desired outputs is used to gather data from the Excel workbook and enter it into an output dictionary within Python. Input and output dictionaries were implemented in Python to ensure easy integration of the automated tools. As long as a standard naming scheme was adhered to in each of the wrappers, integrating the tools simply consisted of passing the dictionaries between wrapper files.

### *III.A.2. LVA Wrapper*

The LVA tool is written in Visual Basic and operates using simple text input/output files. It can also be executed via command line, which made it significantly easier to work with than INTROS. The LVA wrapper was split into three segments: one to deal with input file creation, one for tool execution, and one for output file parsing.

As with INTROS, LVA is capable of representing many unique launch vehicle architectures and likewise, it utilizes unique input files for separate stages (i.e. core stage, upper stage, liquid booster). For this reason, an input template approach was used for the LVA input wrapper. This approach requires the user to define the discrete vehicle architecture but allows the wrapper to change the continuous variables. Within the template text file, the user includes placeholders for DOE variables. The variable names in the template file must match the names found in the Python input dictionary. If matched, Python uses a *join* operation to replace the placeholders with their value from the input dictionary.

After placement of the input values into the template files, LVA is executed via the second wrapper segment. Within this segment the input files are executed through LVA in a specific order that is dependent upon the vehicle architecture. If the vehicle contains any suspended components (typically seen on upper stages) "UpStart" analysis is required to size the components. If the vehicle utilizes liquid boosters, an iteration is necessary to perform structural analysis on the boosters and the core stage file. For example, a two-stage vehicle with liquids boosters would consist of three files. The core stage and liquid boosters are contained within their own LVA file, while the suspended components of the upper stage are contained in an UpStart file. To execute this case, the UpStart tool is executed first, which then passes upper stage masses to the core file. After passage of the updated masses, an iteration is necessary to produce the liquid booster and core component masses. This iteration consists of running the core stage file, the liquid booster file, and finally the core stage file.

Following execution of LVA, the final wrapper segment parses the output files. An output file exists for each unique input file and contains the component masses for the defined stage. The component masses are listed in descending order from top to bottom of the stage. Using this standard setup the output file wrapper identifies the component masses and assigns them to the appropriate vehicle components based upon the architecture in use. For example, the top-most component of the UpStart file for SLS is the upper stage intertank. The component masses are assigned in this manner by the output wrapper, which ultimately outputs another Python dictionary.

### *III.A.3. POST Wrapper*

A great deal of work has been done to automate the execution of POST for ACO launch vehicle studies.<sup>4, 8-11</sup> The challenges associated with automating POST can be broken into two primary categories: successfully executing POST without an expert in the loop and ensuring a global optima has been found for a given vehicle. The initial version of the POST wrapper addressed the first challenge by using expert-based heuristics to help close cases in an automated fashion.<sup>8</sup> To execute POST manually, an analyst needs to develop an initial control vector (u-vector) guess and then, based upon how the trajectory fared, adjust the guess

accordingly to close a failed case or improve an already converged one. The heuristics implemented in the original POST wrapper were developed based upon the operations typically performed by a POST analyst. Using the built-in heuristics, the original wrapper was able to successfully execute POST for the purpose of surrogate modeling.<sup>8</sup>

The second challenge associated with automating POST was addressed via repetitions in the original POST wrapper. For each vehicle case, a Monte Carlo was run on the u-vector parameters to generate the initial guesses. These initial guesses were all run automatically through POST, resulting in a set of converged trajectories for each vehicle case. The trajectory that produced the most optimal result, maximum payload in this case, was identified as the optimum for the given vehicle. This approach is highly dependent upon the number of successful repetitions for each vehicle. As the number of converged repetitions increases, the confidence in the optimum increases. However, the required run-time also increases drastically. Research is currently being conducted in an attempt to identify the appropriate number of repetitions needed in order to reach a desired confidence in the optimal result for a given vehicle.<sup>11</sup>

From the standpoint of case success rate, the repetitions-based approach is very inefficient. Depending upon the ranges for the initial u-vector guess, less than 10% of the total submitted repetitions may successfully close, meaning a great amount of time is spent executing failed cases.<sup>9</sup> For this reason, additional work was done to try and improve the efficiency of the POST wrapper by implementing a graph-based approach for generating initial u-vector guesses.<sup>9</sup>

The graph-based approach emulates the manner in which a POST analyst would develop the initial u-vector guess for a new vehicle. If possible, the analyst will try to utilize a previously converged vehicle's u-vector as a starting point for the new vehicle. If these vehicles are similar enough, an initial closed trajectory can be found very quickly for the new case. The analyst can then fine tune the u-vector to improve the vehicle's performance.

The graph-based POST wrapper develops a graph using the euclidean distances of the DOE vehicle cases in the input space. The closer the vehicles are in the input space, the more likely it will be that the closed u-vector of one case will be an acceptable starting point for the other. Therefore, the graph-based POST wrapper can use seed points to string together vehicle cases via their converged u-vectors. This can greatly minimize the effort required to get an initial closed trajectory for each DOE case. It has also been shown to increase the goodness-of-fit of POST surrogate models.<sup>9</sup>

### III.B. Integration

Following successful automation of the tools, integration was desired in order to complete the entire ETO design process in an automated fashion. In order to successfully integrate the toolset, a strategy was developed to identify the order of operations and key interfaces between each tool. To develop the strategy, typical sets of GR&A as well as the standard design process were considered.

Typical GR&A for ETO studies include vehicle dimensions, propulsion parameters, target orbits, payload fairing configurations, acceleration limits, and any other customer constraints. From these inputs the team delivers vehicle masses and performance in terms of a maximized payload delivered to the target orbit. Considering the standard process in Figure 1, all three tools are iterated to convergence. This iteration had two major implications when considering tool integration for the purpose of applying DOE and surrogate modeling.

First, the iteration between the tools could cause DOE input variable values to change during runtime. For example, if POST is unable to fly a vehicle to orbit then an increase in propellant loading, and thus a change in vehicle tank dimensions, may be required. In this case, if a DOE were run that included tank dimensions as an input, some of the original DOE cases would be changed during tool iteration. This is undesirable for fitting surrogate models as the user loses some control over the placement of DOE cases within the trade space. Depending upon the iterations, the vehicle cases may end up clumping in certain regions of the design space, which will adversely affect the goodness-of-fit of the model.

The second implication of iteration is that it implies the need for multiple executions of each tool for a given vehicle case. This means that any wasted iterations due to failed cases will result in an increase in the time required to close each vehicle. This is especially important when considering the difficulties associated with closing cases in POST. A vehicle case may be successfully run through INTROS and LVA only to find that POST is unable to close the trajectory. In this case, the time required for INTROS and LVA to run as well as the time required to determine that POST is unable to close, has been wasted. This also means that a case from the original DOE is essentially dropped from the set. If many unsuccessful DOE cases are

encountered, large un-sampled regions may exist within the design space, further degrading surrogate model goodness-of-fit.

At this point in the integration planning it was decided that INTROS and LVA would be integrated together and POST would be executed separately. Two benefits resulted from this approach. First, the required run time to close a single vehicle case was reduced due to minimization of failed cases. It also allowed for the simultaneous execution of INTROS/LVA and POST. Second, the execution of a DOE in this manner avoided difficulties in fitting surrogate models due to iteration between the tools.

With INTROS/LVA and POST set up to run separately, the “integration” of the tools was achieved by linking surrogate models together. This approach required that surrogates for INTROS/LVA be fit such that their outputs were inputs to the POST surrogate model. Thus, the INTROS/LVA surrogates would represent the masses of a given vehicle that are required as inputs by POST. The POST surrogate would then take these masses and produce a predicted value for payload delivered to the desired orbit. It was considered undesirable to require any iteration between the surrogates as this could cause issues concerning propagation of error. Thus, care was taken to ensure that no iteration was required between the integrated surrogates.

To achieve successful surrogate integration, some of the typical study constraints needed to be adjusted. For example, the standard process shows an iteration between INTROS/LVA and POST to ensure that the structure is sized to the actual loads seen throughout the trajectory, such as maximum dynamic pressure (Q) and maximum acceleration (G). In order to implement the separate approach, these loads were used as inputs to LVA and as constraints in POST. This ensured that the trajectory was constrained to not violate the loads used to size the vehicle structures, which means no iteration between INTROS/LVA and POST was required.

As discussed previously, another part of the iteration between INTROS/LVA and POST included updates to propellant loading in the case where the vehicle could not deliver the desired payload to the desired orbit. Removal of direct iteration between INTROS/LVA and POST implied that there would no longer be such feedback. It was accepted that INTROS/LVA, running separately, would more than likely size vehicles that could not fly or were sub-optimal. However, the goal of the surrogate model was to represent the response of the tools within the given design space. Therefore, sub-optimal cases from INTROS/LVA still represented useful vehicles for ensuring an accurate surrogate model fits for the predicted masses. From the POST perspective, a sub-optimal vehicle’s masses from INTROS/LVA would simply produce a trajectory that does not provide sufficient payload capability. Thus, such cases could be efficiently eliminated while executing the integrated surrogates by filtering out cases with undesirable payloads.

For example, consider a vehicle run through INTROS/LVA using very high values for the trajectory loads inputs, max Q and max G. Now assume that this vehicle, when actually flown in POST, does not reach those constraints. This means that the structure of the vehicle has been sized to more constraining loads than actually seen during ascent. The resulting vehicle has a higher burnout mass and thus a lower payload delivered to orbit. When linking the surrogate models, adjusting the max Q and max G inputs to lower values would approach the case where the structure was sized to the actual loads from POST. This adjustment would show an improvement in payload delivered and thus would be of more interest to the user.

The integration of INTROS and LVA required an internal loop between the tools to ensure closure. This loop was related to the liftoff thrust-to-weight value that is used in LVA to help size the structures. After inputting the initial vehicle parameters into INTROS, an estimate of the vehicle Gross Liftoff Weight (GLOW) is generated. This GLOW determines the initial liftoff thrust-to-weight used in LVA; however, after LVA re-sizes the structures, the GLOW and liftoff thrust-to-weight change. In order to get an updated GLOW using LVA structures masses, INTROS is updated with the LVA output. The iterative process between INTROS and LVA can be seen in Figure 2. Note that for a typical case only one iteration is necessary to close the vehicle. The iteration is considered converged when the INTROS GLOW is within 100 pounds of the LVA updated GLOW.

In order to improve the case throughput of INTROS/LVA, parallel computing techniques have been implemented. The current setup allows for parallel execution of individual DOE cases, with a pool of worker processes each executing INTROS/LVA as depicted in Figure 2 for a given vehicle case. Due to the Excel-based nature of INTROS, the number of worker processes is limited to be no more than half the number of processor cores available on the machine executing the DOE. Through testing of the integrated INTROS/LVA environment it was found that utilizing more cores results in Excel instabilities, which can cause complete failure of the DOE execution.

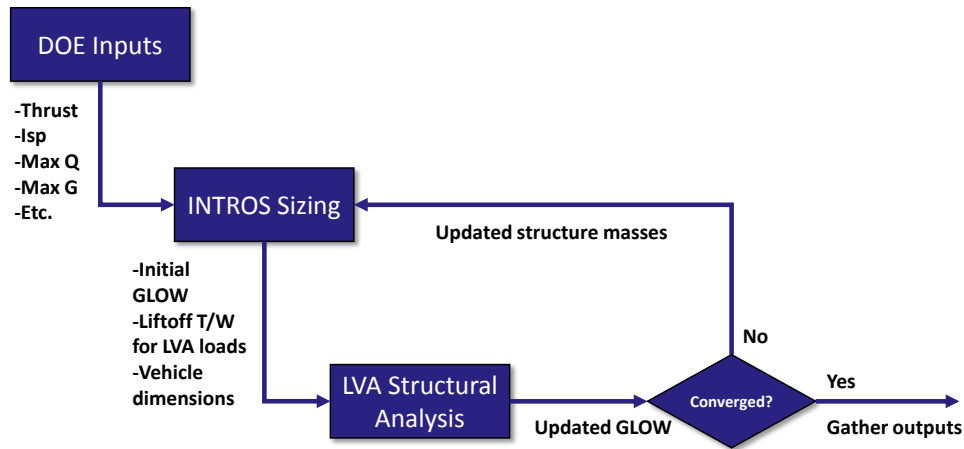


Figure 2: INTROS and LVA integrated process

Using the parallelized version of the INTROS/LVA environment, a test set of 7,200 unique vehicle cases was run in order to compare the run time. This set of cases was run using a total of 8 cores on a standard desktop workstation computer within ACO. The whole set completed in approximately 16.5 hours, which averages to about 1 case closed every 8 seconds. For comparison, a single string run of the same set would require approximately 60 hours to complete as a typical INTROS/LVA case is completed in around 20 to 30 seconds. Attempting to run a similar set manually would be impractical as a typical case can be closed in approximately 20 to 30 minutes.

### III.C. Automated Design Process

The automation and integration work described in the previous sections resulted in a new automated design process that is utilized by ACO. A simplified depiction of the automated process is given in Figure 3. When implemented in tandem with the standard process in Figure 1, the automated process fits into the boxes currently occupied by the manual LVA, INTROS, and POST executions.

As noted in Figure 3, the automated process begins with definition of the GR&A for the study. This GR&A is nearly identical to the standard process with the addition of DOE and surrogate modeling considerations. These considerations are needed to identify the variables that will be used as DOE inputs as well as the outputs that will require surrogate model fits. The GR&A primarily drives the setup of the INTROS/LVA DOE inputs and ranges, but also provides trajectory constraints for the POST DOE.

Following identification of the desired DOE inputs and ranges for INTROS/LVA, a screening test is run. This screening test executes the corner points of the design space in order to determine minimum and maximum values for the vehicle masses. The ranges for vehicle mass are required for any input variable in the POST DOE. For example, vehicle burnout mass is typically used as an input in POST. Thus the range of possible vehicle burnout masses from INTROS/LVA are needed to ensure that the INTROS/LVA surrogate output can match up with the POST surrogate input. If the INTROS/LVA surrogate outputs a mass outside of the original DOE ranges of the POST surrogate, a valid response cannot be obtained from the POST surrogate for that case.

After the DOE inputs and ranges for INTROS/LVA and POST have been determined, the DOE execution step is entered. This step typically consists of two components: manual verification cases and automated DOE execution. For new vehicles or non-standard GR&A, the manual verification component is carried out prior to execution of the DOE. This step ensures that the INTROS/LVA and POST wrappers are successfully running the tools in the same manner as the analyst within the manual process.

Manual verification of INTROS/LVA typically consists of selecting a handful of DOE cases, running them by hand, and then comparing the mass outputs between the manual and automated runs. For POST, the verification step first involves check-out of the input deck to ensure it runs properly. After verifying the POST input deck, a handful of cases are run and compared between the manual and automated processes. If the verification tests are completed successfully, the DOEs are executed through the automated tools.



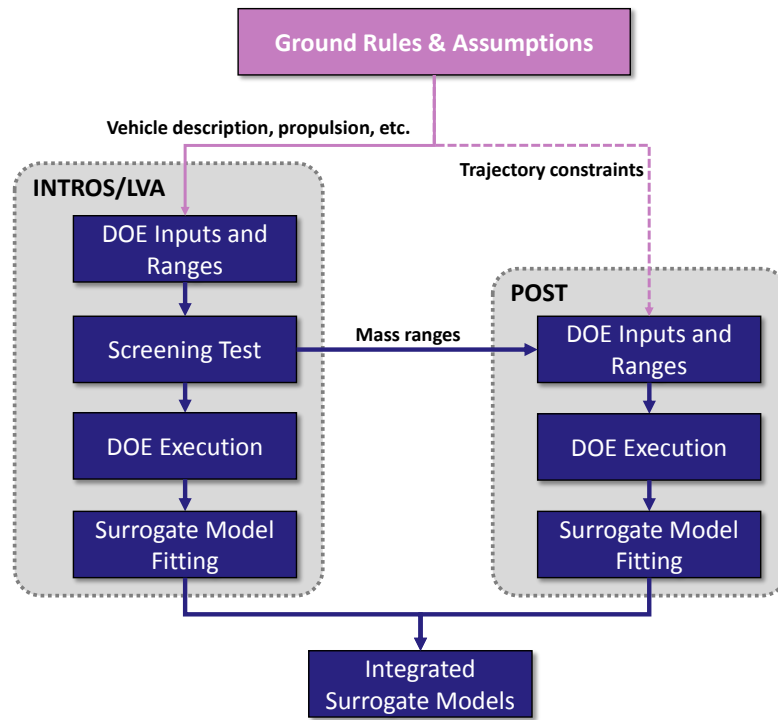


Figure 3: ACO Earth-to-Orbit automated vehicle design process

The final step of the automated process is to generate the individual INTROS/LVA and POST surrogates and integrate them together. As stated earlier, the INTROS/LVA inputs consist of vehicle dimensions, propulsion parameters, boil-off rates, materials, etc. The outputs of interest from INTROS/LVA are the mass inputs required by POST. As will be shown in the example problem, these masses are typically the burnout mass and propellant mass of each stage as well as payload fairing mass if the vehicle is in cargo configuration. Thus the INTROS/LVA surrogates are vehicle masses as a function of vehicle parameters, for example:

$$\text{Stage 1 mbo} = f(\text{tank dimensions, number of engines, thrust, } I_{sp}, \text{ etc.})$$

Using the POST data a single surrogate model is fit, representing payload delivered as a function of vehicle masses and trajectory constraints. Since payload is optimized by POST, the surrogate model is fit to the maximum payload delivered for the specific vehicle case. For this reason it is necessary to complete repetitions for each unique DOE case in order to ensure that a global maximum payload is found for the vehicle. Efficiently finding such a global maximum for each vehicle has been the topic of other work and will not be discussed in detail herein. A more in-depth discussion can be found in the references.<sup>4,8-11</sup> The resulting POST surrogate model outputs payload delivered as a function of masses and constraints:

$$\text{Payload delivered} = f(\text{Stage 1 mbo, Stage 1 prop., Max } G, \text{ etc.})$$

Integration of the surrogate models is achieved in the statistical software JMP. The models can be simply linked by matching up the POST surrogate input variable names with the output names of the INTROS/LVA surrogates. The JMP software allows the user to expand the full equation, which then gives the POST payload delivered as a function of the INTROS/LVA inputs:

$$\text{Payload delivered} = f(\text{tank dimensions, number of engines, thrust, } I_{sp}, \text{ etc.})$$

After obtaining the integrated surrogate model, JMP provides many graphical tools for exploring the design space. The most commonly used is the prediction profiler, which provides a visualization of the sensitivity of the payload delivered with respect to the various inputs. The profiler also provides capabilities for executing large Monte Carlo runs for the purpose of inverse design, probabilistic design, performance

risk assessment, or technology assessment. Some of the profiler’s capability will be demonstrated in Section IV.C.

## IV. Example Problem

An example problem will be given to illustrate the execution of the ACO’s automated process for ETO vehicle design. The example problem will be based on the Space Launch System (SLS), which is currently being developed by NASA as the next generation heavy lift vehicle that will enable manned exploration missions to the moon and beyond.<sup>12</sup> The SLS was chosen as a relevant example because design trades are currently being performed, looking at future block upgrades of the vehicle upper stage and boosters.

The SLS is a 2.5 stage vehicle employing liquid hydrogen and liquid oxygen propellant for its core and upper stages, and two STS derived five-segment solid rocket boosters.<sup>12</sup> In order to narrow the scope of the example problem, no modifications will be made to the solid rocket boosters. In addition, the core stage of the vehicle will be fixed to the baseline configuration with four RS-25 engines. Therefore, the example problem will focus on capturing design trades for the upper stage of an SLS-like vehicle.

The following section will discuss the development of the DOE inputs and ranges for INTROS/LVA and POST. Section IV.B will discuss the surrogate modeling techniques that were utilized for the example problem. This section will also illustrate the goodness-of-fit tests that were completed for the surrogate model fits of INTROS/LVA and POST. Section IV.C will give an example of the uses of the integrated surrogate model within the JMP software.

### IV.A. Design of Experiments

The DOE inputs and ranges for INTROS/LVA can be developed based upon trades of interest for an upper stage of an SLS-like vehicle. For the upper stage, trades of interest may include number of engines, thrust per engine, specific impulse, engine mass, engine dimensions, and component materials. For the purpose of the example problem, only the engine parameters and necessary stage dimensions will be changed. The engine parameters include number of engines, thrust per engine, specific impulse ( $I_{sp}$ ), and oxidizer-to-fuel ratio (OFR). Note that the OFR and  $I_{sp}$  will have an effect on the propellant loading of the stage. Thus, the tank dimensions must be changed within the DOE as well.

To help simplify the list of tank dimension parameters within the INTROS/LVA DOE, the SLS baseline diameters were assumed. The liquid hydrogen tank diameter is the same as the SLS core stage, while the liquid oxygen tank is nearly 3 meters smaller in diameter. With these parameters fixed, only the two tank cylinder lengths can be adjusted to change the stage’s propellant loading. However, with OFR as an input, the mass ratio between the propellants is known and therefore the volume ratio between the tanks is also known. For this reason, only one of the tank lengths needs to be input and the remaining length can be calculated using the tank diameters and OFR. The oxygen tank cylinder length was chosen as the desired DOE input as it is the smaller of the two tanks in terms of diameter.

The final parameters for the INTROS/LVA DOE relate to the trajectory constraints that will be used in POST. As discussed earlier, these parameters are used as inputs to LVA in order to size the vehicle structures. The same parameters are then implemented as constraints in POST to avoid overloading the vehicle on ascent. Two parameters were used to represent the trajectory loads, maximum dynamic pressure and maximum acceleration. The complete list of INTROS/LVA DOE inputs and ranges can be seen in Table 1.

Since the core stage of the vehicle was assumed to be fixed for the example problem, multiple parameters related to the core stage propulsion and solid boosters had to be defaulted. In addition, the vehicle payload configuration and intermediate orbit were fixed. A list of fixed parameters for the INTROS/LVA and POST DOEs can be seen in Table 2. Note that the core engine parameters were derived from the RS-25 engine specifications published by Aerojet Rocketdyne.<sup>13</sup> The solid booster parameters were taken from the five-segment reusable solid rocket motor shown in the ATK space propulsion products catalog.<sup>14</sup>

Using the inputs shown in Table 1, a screening test was run through INTROS/LVA to derive the ranges for the POST DOE. As discussed earlier, this step is necessary in order to match up the output range from the INTROS/LVA surrogates with the input range of the POST surrogate. In addition to the mass inputs, the trajectory constraints are added to the DOE. The vehicle destination in terms of characteristic energy, or  $C_3$ , was also added as a POST DOE parameter. The POST DOE inputs and ranges can be seen in Table

3.

Table 1: INTROS/LVA Upper Stage DOE Inputs

| Variable                | Range               |
|-------------------------|---------------------|
| Number of engines       | 2, 3, 4, 5          |
| Thrust per engine       | 20,000 - 65,000 lbf |
| Engine $I_{sp}$         | 440 - 470 sec       |
| OFR                     | 5.4 - 6.0           |
| Ox tank cylinder length | 3 - 12 ft           |
| Max G                   | 3 - 4               |
| Max Q                   | 540 - 700 psf       |

Table 2: INTROS/LVA and POST Fixed Parameters

| Parameter                        | Value             |
|----------------------------------|-------------------|
| Core number of engines           | 4                 |
| Core engine sea level $I_{sp}$   | SLS nominal       |
| Core sea level thrust per engine | SLS nominal       |
| Core propellant mass             | SLS Baseline      |
| Booster burnout mass             | SLS Baseline      |
| Booster propellant mass          | SLS Baseline      |
| Payload configuration            | Crewed            |
| Intermediate orbit               | 100 nmi x 100 nmi |

Table 3: POST DOE Inputs

| Variable                    | Range                 |
|-----------------------------|-----------------------|
| Upper stage burnout mass    | 40,000 - 70,000 lbm   |
| Upper stage propellant mass | 205,000 - 375,000 lbm |
| Upper stage T/W             | 0.2 - 0.75            |
| Upper stage $I_{sp}$        | 440 - 470 sec         |
| Upper stage total thrust    | 60,000 - 325,000 lbf  |
| Core burnout mass           | 200,000 - 255,000 lbm |
| Max G                       | 3 - 4                 |
| Max Q                       | 540 - 700 psf         |
| Destination ( $C_3$ )       | -15 - 3 $km^2/s^2$    |

#### IV.B. Surrogate Modeling

Following successful DOE execution through the automated tools, surrogate models were fit for each of the data sets. The software JMP was used to complete the model fitting. This software provides tools for fitting many types of models including Kriging, Gaussian process, neural networks, and response surface equations.<sup>15,16</sup> Response Surface Equations (RSE) are the desired model type for launch vehicle studies due to their relative simplicity as compared to Gaussian process models or neural networks. Data from INTROS and LVA have been found to be very easy to fit using RSEs. Although the trajectory response space is typically multi-modal and non-linear, higher-order RSEs have also been used to successfully fit results from POST.<sup>8</sup>

The RSEs are typically fit using the method of least squares to estimate the regression coefficients.<sup>17</sup> Equation 1 gives the generic form of a second order RSE, where R is the response,  $x$  is an input parameters,

and  $\beta$  is a regression coefficient. Note that the model is linear in the  $\beta$  parameters and is therefore a linear regression model, regardless of the order of the input parameters.<sup>17</sup>

$$R = \beta_0 + \sum_{j=1}^k \beta_j x_j + \sum_{i=1}^{k-1} \sum_{j=i+1}^k \beta_{ij} x_i x_j + \sum_{j=1}^k \beta_{jj} x_j^2 \quad (1)$$

As noted above, higher-order RSE models have been successfully used to fit INTROS/LVA and POST output data. However, as the order of the model is increased, the total number of terms in the equation also increases. Since  $k + 1$  cases are required to regress the  $\beta$  coefficients for a model of  $k$  terms, this also increases the total number of DOE cases that must be completed through the automated tools. In order to keep the number of required cases as well as the overall length of the equation to a reasonable level, stepwise regression was utilized in JMP.

Stepwise regression uses a statistical significance level to determine which terms in the regression model are most beneficial to the model's ability to predict the data. During model fitting, terms are added or removed from the model based upon their calculated significance levels. The result is a model containing only terms that are statistically significant for predicting the response, which tends to be much more compact than the full model.

In addition to stepwise regression, k-folds cross validation was implemented when fitting the models in JMP. K-folds cross validation splits the full set of cases into  $k$  different sets or "folds". The model is fit to  $k - 1$  sets of data, leaving the remaining set for model validation. After fitting a total of  $k$  models, the one with the best validation statistics is returned as the final model.<sup>15</sup>

When a surrogate model is returned, multiple goodness-of-fit tests are applied to accept or reject the fit. First, the coefficient of multiple determination, or  $R^2$ , is used as an initial indicator of fit acceptability. The  $R^2$  value is a measure of the reduction in variability of the response obtained from the model.<sup>17</sup> Based upon experience with fitting INTROS/LVA and POST data, an  $R^2$  value greater than 0.99 warrants further goodness-of-fit testing. Any value below 0.99 will require re-fitting after investigation of outliers or execution of additional DOE runs.

Additional goodness-of-fit testing first involves viewing the actual-by-predicted and the residual-by-predicted plots for the surrogate. The actual-by-predicted plot shows the actual output value of the tool plotted against the predicted output of the surrogate for each DOE point. This figure should show a near straight line for an acceptable fit. The residual-by-predicted plot shows the predicted response value for each point plotted against its residual as compared to the actual tool's output value. The desired shape of this plot is a "shotgun spread" of points with no clear trends, an example of which is shown in Figure 4. Clumping or non-linear trends within either of the plots are a good indication that the surrogate fit is unacceptable.

The final goodness-of-fit test is used to calculate the percent error of the surrogate model across the entire data set. This percentage encompasses both the points used to fit the model itself and the points held back for validation purposes. After calculating the error, the characteristics of the percent error distribution are used as a final acceptance check of the model. Typically, the mean is desired to be as close to zero as possible and the standard deviation must be below one.

The goodness-of-fit tests described above were implemented for the example problem surrogates. Figure 5 gives an error distribution of the upper stage burnout mass surrogate from INTROS/LVA. This distribution is representative of all of the INTROS/LVA surrogates. Figure 6 shows the error distribution for the POST surrogate. As illustrated by the figures, the POST surrogate has a much larger error than the INTROS/LVA surrogates. This is due to the inherent difficulties associated with closing cases in an automated fashion in POST. Although the error distribution in Figure 6 does meet the requirements for acceptance, upgrades of the POST wrappers are being implemented to help reduce the noise in the POST output, thus improving the goodness-of-fit.<sup>9</sup>

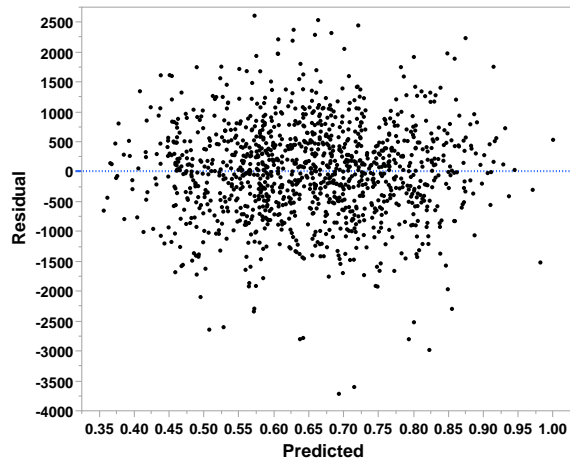


Figure 4: Sample residual-by-predicted plot

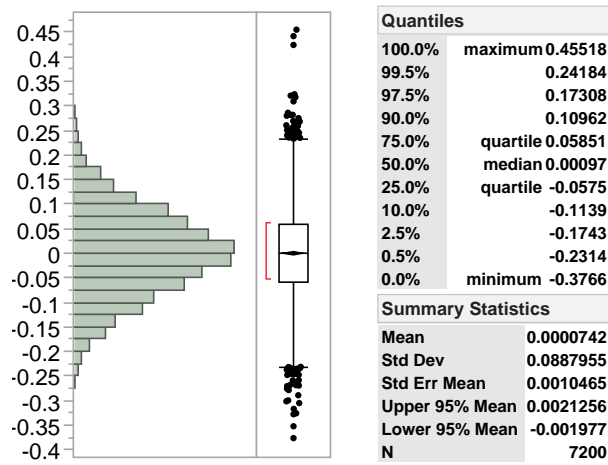


Figure 5: Percent error distribution for the INTROS/LVA upper stage burnout mass surrogate

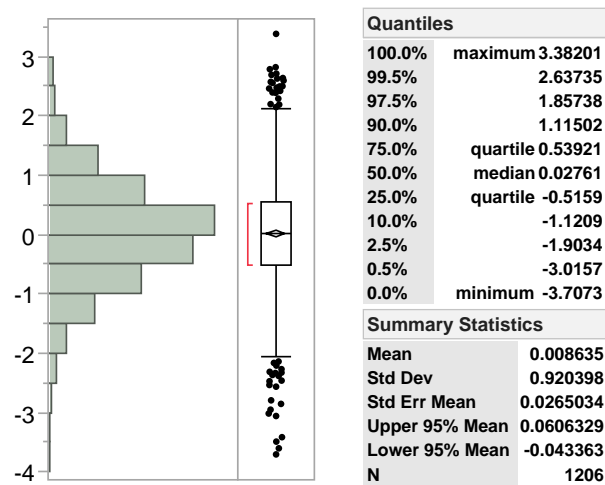


Figure 6: Percent error distribution for the POST payload delivered surrogate

## IV.C. Results

After successfully fitting the INTROS/LVA and POST surrogates, the integrated surrogate can be used for design space exploration. The typical first step for an ETO study is to look at the surrogate model profiler in JMP. The profiler for the example problem integrated surrogate model can be seen in Figure 7. As can be seen in the figure, the profiler gives the user an initial look at the sensitivity of the output to changes in the input parameters. For example,  $C_3$  shows an intuitive result where payload delivered decreases with increasing  $C_3$ . The trends seen in the profiler can also be used as a final check of surrogate accuracy. For example, an expected result is shown where an increase in  $I_{sp}$  produces an increase in payload delivered. If such a trend were reversed, the user would immediately know that an issue exists with the INTROS/LVA surrogate, POST surrogate, or the integrated surrogate itself. Typically if an unintuitive trend is encountered, some manual investigation is implemented to confirm or deny the accuracy of the result.

The primary power of the profiler is the ability to evaluate a given vehicle case nearly instantaneously. The surrogate model behind the profiler predicts the response of the INTROS, LVA, and POST tools for the given vehicle. Therefore, any inputs plugged into the profiler represent the result one would expect from the process in Figure 1 within the error bounds described by Figures 5 and 6. This is especially useful during a design review or out-brief to the customer. If the customer has questions regarding the vehicle configuration, an estimate for vehicle performance can be provided real-time as opposed to taking an action to be completed for a follow-up meeting. For example, the profiler in Figure 7 shows an upper stage with four 35,000 lbf engines, which appears to be the optimal in terms of payload for the given destination ( $C_3$ ) and propellant loading (ox cyl length). However, if the customer has a different engine in mind, say of 22,000 lbf, the thrust per engine variable can be changed real-time to show the delta. Figure 8 reflects this change in thrust, which shows that either the number of engines would need to increase or the propellant load would need to decrease to improve the payload delivered to the given  $C_3$ .

Rapid execution of vehicle cases using the JMP profiler can also be used to carry out a Monte Carlo simulation for the purpose of probabilistic design, inverse design, technology evaluation, or risk assessment. A Monte Carlo can be executed in the JMP profiler using the simulator, which can be seen in Figure 9. The simulator allows the user to select the desired distribution type and identify the distribution parameters for each input variable. The example in Figure 9 shows a distribution applied to every input, however, variables can also be defaulted to a single value. The additional output box at the right of the profiler gives the user a quick visual of the resulting distribution of payload delivered given all of the input distributions. The results of the simulation can also be exported to an additional table for more analysis.

Probabilistic design, technology evaluation, and risk assessment are carried out in a similar manner using the simulator. For example, consider assessing the performance risk associated with the engine parameters. If the desired engine is a new build versus already operational, there will be some uncertainty associated with the final thrust and  $I_{sp}$  the engine will produce when it reaches operations. This uncertainty can be captured via distributions in the simulator, which could then produce a distribution for payload delivered. From this distribution, the probability of successfully delivering a specific payload mass can be estimated. A similar approach is used for technology evaluation, where the impact of the technology on given inputs is captured via simulator distributions.

The process of inverse design involves execution of large Monte Carlo runs, which are exported to an additional JMP table. Using the newly generated table and filters based upon constraints or vehicle requirements, the user can find a vehicle that best fits the needs of the customer. For example, the customer may have a specific payload delivery requirement in mind. Instead of manually adjusting the profiler input values the analyst can set up uniform distributions across all of the input variables. A large Monte Carlo will then capture variations of the vehicle within the entire trade space. The analyst then filters the cases using the desired payload value from the customer and can identify a subset of vehicles of interest.

Following the conclusion of a study, such as demonstrated by the example problem, the study GR&A and goodness-of-fit results are compiled into a deliverable for the customer. Depending upon the customer's requirements, various visualizations are created in JMP for the study out-brief. The profiler is usually the go-to visualization for out-briefs as it provides the customer with instant feedback on any "what-if" questions that may arise. It also gives the customer a very clear picture of the effects of the inputs on the performance metric of interest and is a powerful addition to the standard vehicle BBCs provided by the ETO team.

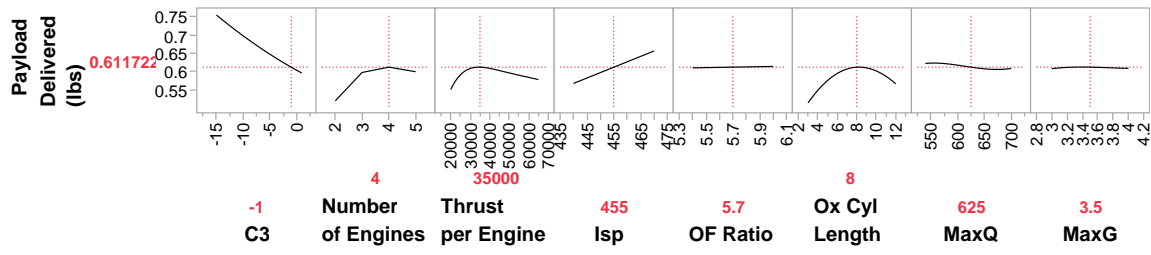


Figure 7: Surrogate model profiler in JMP

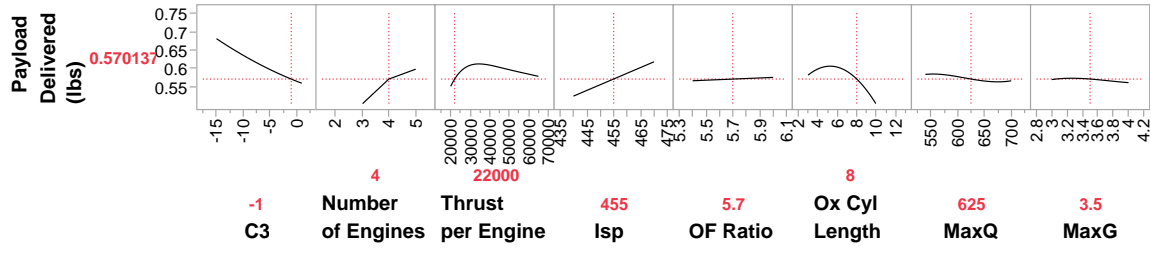


Figure 8: Surrogate model profiler in JMP

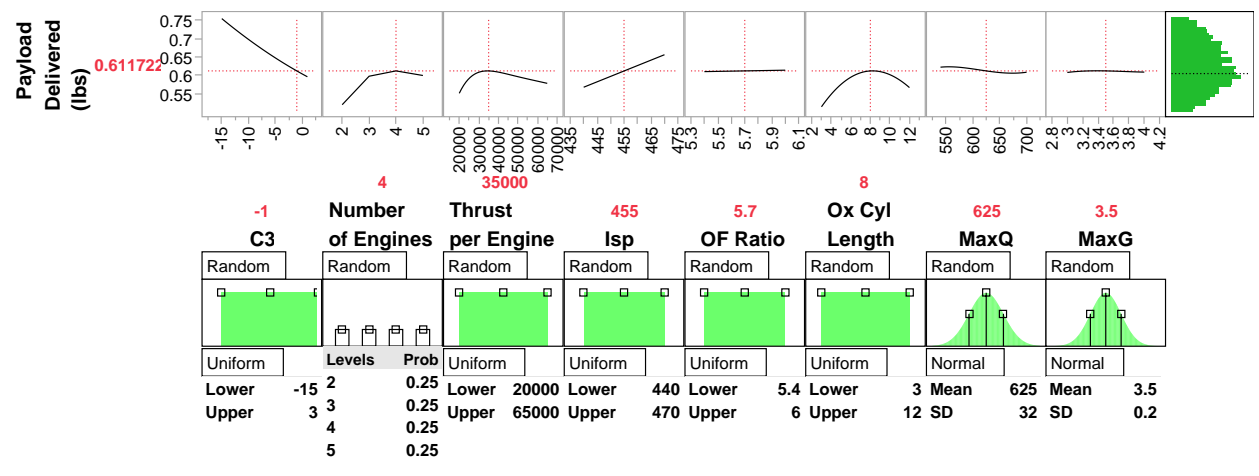


Figure 9: Surrogate model profiler with simulator

## V. Conclusion

The primary goal of this work was to extend the capabilities of the ACO ETO vehicle design process. Although this process has been successfully implemented for many years, an expansion was desired in order to more fully explore the design space for a given study. To achieve this goal, the standard process was first augmented by automating and integrating the design tools. The automation of the tools ultimately allowed for the implementation of DOE and surrogate modeling techniques.

The example problem demonstrated the utility of the DOE and surrogate modeling approach. The use of surrogate models gives the ACO team the ability to visualize vehicle sensitivities to various design parameters and identify regions of interest within the design space. In addition, it provides the customer with an almost instantaneous turnaround on any what-if questions that may arise within the bounds of the surrogate model. When coupled with the standard process, the automated process provides ACO's customers with a powerful tool for decision making during early conceptual design.

## References

- <sup>1</sup>Blair, J., Ryan, R., Schutzenhofer, L., and Humphries, W., "Launch Vehicle Design Process: Characterization, Technical Integration, and Lessons Learned," Tech. rep., National Aeronautics and Space Administration, May 2001.
- <sup>2</sup>Waters, E., Garcia, J., Beers, B., Philips, A., Holt, J., and Threet, G., "NASA Advanced Concepts Office, Earth-To-Orbit Team Design Process and Tools," *AIAA SPACE 2013 Conference & Exposition*, 2013.
- <sup>3</sup>Creech, D., Threet, G., Philips, A., Waters, E., and Baysinger, M., "Modular Approach to Launch Vehicle Design Based on a Common Core Element," AIAA SPACE 2010 Conference & Exposition, Anaheim, California, September 2010.
- <sup>4</sup>Steffens, M. J., Edwards, S. J., and Mavris, D. N., "Capturing the Global Feasible Design Space for Launch Vehicle Ascent Trajectories," *AIAA SciTech*, Kissimmee, FL, January 2015.
- <sup>5</sup>NASA Langley Research Center, "POSTII Overview," <http://post2.larc.nasa.gov/overview.htm>.
- <sup>6</sup>Beers, B. R., Waters, E. D., Philips, A. D., and Threet Jr, G. E., "Small Launch Vehicle Concept Development for Affordable Multi-Stage Inline Configurations," *AIAA SPACE 2013 Conference and Exposition*, 2013, p. 5529.
- <sup>7</sup>Creech, D., Threet, G., Philips, A., and Waters, E., "Launch Vehicle Demonstrator Using Shuttle Assets," AIAA, Long Beach, California, September 2011.
- <sup>8</sup>Dees, P. D., Zwack, M. R., Steffens, M., Edwards, S., Diaz, M. J., and Holt, J. B., "An Expert System-Driven Method for Parametric Trajectory Optimization During Conceptual Design," AIAA SPACE 2015 Conference & Exposition, Pasadena, CA, 2015.
- <sup>9</sup>Dees, P. D., Zwack, M. R., Steffens, M., and Edwards, S., "Augmenting Conceptual Design Trajectory Tradespace Exploration with Graph Theory," AIAA SPACE 2016 Conference & Exposition, Long Beach, CA, September 2016.
- <sup>10</sup>Steffens, M., *A Combined Global and Local Methodology for Launch Vehicle Trajectory Design-space Exploration and Optimization*, Master's thesis, Georgia Institute of Technology, 2014.
- <sup>11</sup>Steffens, M., *Trajectory-Based Launch Vehicle Performance Analysis for Design-Space Exploration in Conceptual Design*, Ph.D. thesis, Georgia Institute of Technology, August 2016.
- <sup>12</sup>National Aeronautics and Space Administration, "NASA facts: Space Launch System," June 2012. [http://www.nasa.gov/pdf/664158main\\_sls\\_fs\\_master.pdf](http://www.nasa.gov/pdf/664158main_sls_fs_master.pdf).
- <sup>13</sup>Aerojet Rocketdyne, "RS-25 Engine," Online, <http://www.rocket.com/rs-25-engine>.
- <sup>14</sup>Alliant Techsystems Inc., "ATK Space Propulsion Products Catalog," September 2012. [http://www.orbitalatk.com/flight-systems/propulsion-systems/launch-abort-motor/docs/orbital\\_atk\\_motor\\_catalog\\_\(2012\).pdf](http://www.orbitalatk.com/flight-systems/propulsion-systems/launch-abort-motor/docs/orbital_atk_motor_catalog_(2012).pdf).
- <sup>15</sup>*JMP 11 Fitting Linear Models*, SAS Institute Inc., Cary, NC, 2013.
- <sup>16</sup>*JMP 11 Specialized Models*, SAS Institute Inc., Cary, NC, 2013.
- <sup>17</sup>Myers, R., Montgomery, D., and Anderson-Cook, C., *Response Surface Methodology*, Wiley, 3rd ed., 2009.