AIAA SPACE 2009 Conference & Exposition
14 - 17 September 2009, Pasadena, California

AIAA 2009-6563

GT-FAST: A Point Design Tool for Rapid Fractionated Spacecraft Sizing and Synthesis

Jarret M. Lafleur* and Joseph H. Saleh†
Georgia Institute of Technology, Atlanta, Georgia 30332

In July 2007, DARPA issued a Broad Agency Announcement for the development of System F6, a flight demonstration of an architecture in which the functionality of a traditional monolithic satellite is fulfilled with a fractionated cluster of free-flying, wirelessly interconnected modules. Given the large number of possible architectural options, two challenges facing systems analysis of F6 are (1) the ability to enumerate the many potential candidate fractionated architectures and (2) the ability to analyze and quantify the cost and benefits of each architecture. One element necessary in enabling a probabilistic, value-centric analysis of such fractionated architectures is a systematic method for sizing and costing the many candidate architectures that arise. The Georgia Tech F6 Architecture Synthesis Tool (GT-FAST) is a point design tool designed to fulfill this need by allowing rapid, automated sizing and synthesis of candidate F6 architectures. This paper presents the internal mechanics and some illustrative applications of GT-FAST. Discussed are the manner in which GT-FAST fractionated designs are specified, including discrete and continuous-variable inputs, as well as the methods, models, and assumptions used in estimating elements of mass, power, and cost. Finally, the paper concludes with sample outputs from GT-FAST for a notional fractionated architecture, an example of GT-FAST's trade study capability, and a partial validation of GT-FAST against the Jason-2 and TIMED satellites. The ease with which GT-FAST can be adapted to new fractionated spacecraft applications is highlighted, and avenues for potential future expansion of GT-FAST are discussed.

Nomenclature

$C_{add/replace}$	=	average cost of adding or replacing component	P	=	total power requirement
$C_{i,existing}$	=	cost of adding component via an existing module	t	=	time on-orbit
$C_{i,separate}$	=	cost of adding component via a dedicated module	V	=	average orbital velocity
f_{100}	=	smoothing function near 100 W power boundary	β	=	average ballistic coefficient
f_{500}	=	smoothing function near 500 W power boundary	ΔV	=	velocity change requirement
n	=	number of fractionatable components in architecture	ρ	=	average atmospheric density

I. Introduction

IN July 2007, the U.S. Defense Advanced Research Projects Agency (DARPA) released a Broad Agency Announcement soliciting proposals for development of System F6 (Future Fast, Flexible, Fractionated, Free-Flying Spacecraft united by Information eXchange).[1] DARPA's goal for F6 is ultimately a flight demonstration of an architecture in which the functionality of a traditional "monolithic" satellite is fulfilled with a "fractionated" cluster of free-flying, wirelessly interconnected modules. The potential benefits of the F6 approach include enhanced responsiveness in delivering initial capabilities to commercial or government (especially defense) customers, greater flexibility in responding to mid-life changes in requirements, and superior robustness against internal failure and external attack (i.e., enhanced survivability).

Two systems analysis challenges that are especially critical for the flexible and architecturally complex F6 concept are (1) the ability to thoroughly and systematically generate candidate fractionated architectures and, more

*Ph.D. Candidate, Daniel Guggenheim School of Aerospace Engineering, Student Member AIAA.
†Assistant Professor, Daniel Guggenheim School of Aerospace Engineering, Associate Fellow AIAA.

importantly, (2) the ability to assess and quantify the cost and benefits of each architecture, and in so doing to order-rank the different proposed architectures according to the right metrics. System attributes such as flexibility and survivability, which are essential for systems operating in distinctly uncertain and rapidly changing environments, are not properly captured and valued in the traditional cost- or performance-centric mindsets of system design and acquisition (e.g., achievement of a given level of performance for the least cost, the preferred policy of former Defense Secretary Robert McNamara[2,3]). As a result, a value-centric approach is required to properly assess and benchmark the benefits of fractionation compared with those of the traditional monolith spacecraft. Value-informed decisions regarding F6 architectures hinge upon analysis of uncertainties and value generation throughout the life of the system.

One element necessary in enabling such a probabilistic, value-centric analysis of F6 architectures is a systematic method for enumerating, sizing, and costing the many candidate architectures that are introduced by fractionating subsystems or resources. For example, in one previously published design for F6,[4] twelve instances of six distinct types of fractionatable components are distributed among seven free-flying modules. However, this distribution of components is just one of many possibilities. As shown in Ref. 5, if only six components exist in the system and each can be independently placed in any of up to six modules, 203 distinct cluster configurations exist. If the number of components increases to twelve (akin to the design in Ref. 4), the number of possible configurations explodes to over 4.2 million. Furthermore, these numbers do not include the multitude of launch manifesting options.[‡] Clearly there is a need to be able to evaluate more than a handful of these alternative configurations in order to make an informed decision on the design of an F6 architecture. The Georgia Tech F6 Architecture Synthesis Tool (GT-FAST) is a point design computer tool designed to help solve this problem by allowing rapid, automated sizing and synthesis of candidate F6 architectures.

The primary function of GT-FAST is to convert a user-defined configuration of fractionated components (i.e., a specification of which fractionatable components are assigned to which modules) and launch manifest (i.e., which modules are carried on which launch vehicles) into a point design.[§] The information output by GT-FAST for each point design is a mass, power, and cost budget for the cluster and for each module in the cluster. Also integral to GT-FAST's sizing procedures are user inputs for continuous variables such as orbit altitude, inclination, module design lifetime, and assumptions such as engine specific impulse (I_{sp}), payload mass, and payload power. Because GT-FAST automatically (and relatively quickly) sizes an F6 design, the tool is well-suited for trade studies and has a built-in capability to run a series of input sets and track any number of user-defined output metrics.[**]

Due to the amount of material to be covered, the trade study process and sample results are the subject of a separate companion paper.[5] The details of the sizing procedures and assumptions are the focus of this paper, which is organized as follows: First, the manner in which a GT-FAST point design is specified is described, covering all major inputs but especially focusing on the manner in which architecture configurations are specified. This is illustrated with an example point design that is used throughout the entire paper. Second, the current models for mass, power, and cost are discussed, and sample outputs are then provided. These outputs include a program cost budget plus a mass and power budget for each of four modules in the example point design. Additionally, a two-objective Pareto front is shown to partially illustrate the trade study capability that is expanded upon in Ref. 5.

II. Defining a Design in GT-FAST

GT-FAST is currently implemented in Microsoft Excel with approximately 3,200 lines of supporting Visual Basic code. The selection of Excel/Visual Basic as a programming language is due largely to the ability of Excel to automatically iterate among circular references that may exist, a common occurrence in sizing programs. Additionally, this choice allows a great deal of portability in allowing the code to be distributed and used by a large number of engineers in various organizations, if necessary. Computing time depends on the complexity of the

‡ The nomenclature distinguishing components from modules, clusters, and designs is presented in Section II.

§ As a rapid sizing and synthesis point design tool, GT-FAST is similar in concept to numerous others in academia and industry, such as FLOPS[6], ATLAS[7,8], PESST[9,10], EXAMINE[11], and ROSETTA models[12]. GT-FAST is unique in that it is specifically designed for fractionated satellite architectures.

** These input sets are analogous to experiments that the designer might like to run to characterize his design space and determine an optimum design, if one such design exists. If all inputs into GT-FAST were continuous variables, this process would be well-suited to a classical design-of-experiments approach[13].

design in question and on processor speeds, but in the trade study covered by Ref. 5, computational time was demonstrated at an average of about 20 seconds per point design.

The first step in any execution of GT-FAST in its point-design mode is the definition of the point design itself. This is accomplished through specification of both discrete and continuous inputs. Because of the size of the combinatoric design space, the discrete inputs have been the focus of GT-FAST F6 analyses and will be covered in the most detail in this paper.

A. Discrete (Fractionation Scheme) Inputs

The principal discrete inputs into GT-FAST deal with specification of which fractionatable components are present in which modules and which modules are carried on which launch vehicles. On this point, it is important to clarify issues of nomenclature. In this paper, the basic unit of fractionation is called a fractionatable component, or a component for short. Depending on the resolution one desires in examining fractionated designs, these components can be subsystems (as in Ref. 14) or resources/payloads (as in Ref. 4). As will shortly be described, the current version of GT-FAST uses the latter as definitions of components.

Next, a compilation of components (and any required essential support subsystems, such as structure, thermal, and others) into a single free-flying vehicle is called a module. A compilation of modules into an independent on-orbit F6 system is called a cluster or architecture. Finally, a cluster with the specification of their launch manifest (e.g., on what vehicle each module is launched, acknowledging that multiple modules may launch on the same launch vehicle) is called a design. This nomenclature is illustrated graphically in Fig. 1.

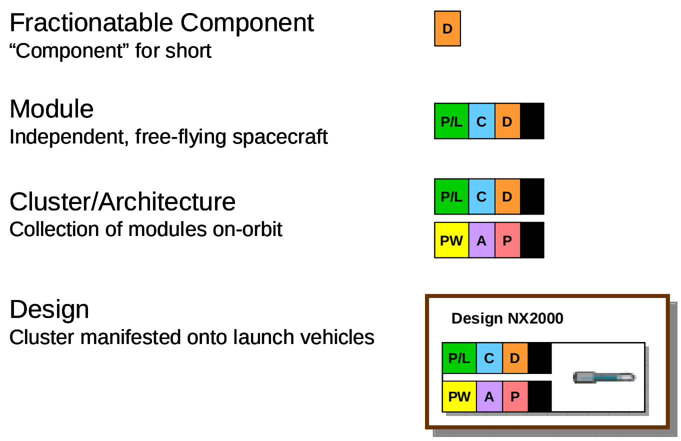

Figure 1. Nomenclature for F6 designs used in this paper.

1. *Fractionatable Components currently modeled in GT-FAST*

The current implementation of GT-FAST uses five different classes of fractionatable components, consistent with those of Ref. 4. An architecture can contain up to three payloads, up to two 24/7 communication units, up to two high-bandwidth downlinks, a solid-state recorder, and a mission data processor. Icons used in this paper to represent these nine individual fractionatable components are shown in Fig. 2. Payloads are specified by their mass, sunlight and eclipse power requirements, and pointing requirement. Unlike the Air Force Satellite Control Network (AFSCN) communications unit which every module is sized to include, a 24/7 communication unit provides near-continuous communications capability through a relay satellite such as one of the NASA Tracking and Data Relay Satellites (TDRSs). High-bandwidth downlink units allow for high-volume downlinks that could not otherwise be provided with AFSCN or 24/7 links. A solid state recorder allows high-volume data storage, and a mission data processor is a resource allowing for onboard high-speed computing.

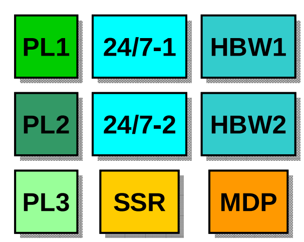

Figure 2. Icons for fractionated components currently implemented in GT-FAST.

2. *Example Specification of Fractionation Scheme in GT-FAST*

To illustrate the way in which an arbitrary architecture can be input into GT-FAST, we use the example design shown in Fig. 3. In this design, there are four modules. The first holds Payload #1, the primary solid state recorder, and the primary mission data processor. The second module holds one of two high bandwidth downlink units within the architecture. The third module holds Payload #2 and the second high bandwidth downlink unit, and the fourth module holds Payload #3 and a 24/7 communication unit. Note that there is only one 24/7 communication unit within this architecture even though the current version of GT-FAST can support up to two 24/7 communication units (i.e., in general, the fact that a component is available does not mean that it must be used in a module or a cluster). The black block on each module signifies that all modules also include all essential support subsystems, such as structure, thermal, power, and others.

Figure 3 also represents that Modules #1 and #2 are manifested to be flown on the same launch vehicle. Modules #3 and #4 each launch separately. Note that launch order is not represented (or needed) by GT-FAST; that is, the representation in Fig. 3 does not preclude Module #4 from launching first or second. Also, as will be discussed, the actual launch vehicle is selected by GT-FAST based on required launch mass, launch vehicle payload capabilities, and launch costs.

The example design shown in Fig. 3 is specified within GT-FAST through two matrices.[††] The first, shown in Fig. 4, maps the fractionatable components (columns) to the modules that carry them (rows). Thus, each row represents the configuration of a single module and is color coded to appear similar to the representation in Fig. 3. Each element of the matrix is allowed to take one of three character values: P, F, or N. The letter "P" indicates that the particular component exists in the design and is present on the corresponding module. The letter "F" indicates that the component exists in the design but is not present on the module. The letter "N" indicates that the component in question does not exist in the design. Thus, any column which is not filled entirely by the letter "N" is allowed to have only one "P" (and all other elements of the column must have the letter "F").[‡‡] Thus, the first row of the matrix in Fig. 4 shows that Module #1 carries Payload #1, the solid state recorder (SSR), and the mission data processor (MDP), just as indicated by Fig. 3. Note that the column for the second 24/7 communication unit is filled with the letter "N" since the second 24/7 communication unit does not exist in this example design.

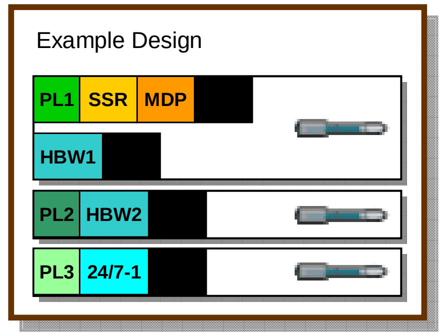

Figure 3. Architectural depiction of example design used in this paper.

The second matrix, shown in Fig. 5, maps the modules (rows) to the launch vehicles that carry them (columns). Thus, each column shows the modules that launch on a given launch vehicle. Each element of the matrix is allowed to take one of two character values: O or N. The letter "O" indicates that a particular module is carried onboard a particular launch vehicle. The letter "N" indicates that a particular module is not carried aboard a particular launch vehicle. Thus, the element in the first row and first column of the matrix in Fig. 5 is marked "O", indicating that Module #1 is carried by Launch Vehicle #1. By necessity, all other elements in the first row are marked "N", since Module #1 can only be launched on one vehicle.

ARCHITECTURAL

Number of Modules 4 Consistency Check

	Payload 1	Payload 2	Payload 3	24/7 Comm 1	24/7 Comm 2	High BW 1	High BW 2	SSR	MDP
Module 1	P	F	F	F	N	F	F	P	P
Module 2	F	F	F	F	N	P	F	F	F
Module 3	F	P	F	F	N	F	P	F	F
Module 4	F	F	P	P	N	F	F	F	F
Module 5	F	F	F	F	N	F	F	F	F
Module 6	F	F	F	F	N	F	F	F	F
Module 7	F	F	F	F	N	F	F	F	F
Module 8	F	F	F	F	N	F	F	F	F
Module 9	F	F	F	F	N	F	F	F	F

Create this Architecture!

Figure 4. Input Matrix Mapping of Components to Modules for the Example Design.

[††] Although the matrices in the current implementation of GT-FAST are limited in dimension to 9 × 9, this can be easily modified for future implementations involving more fractionatable components.

[‡‡] It is reasonable to ask why there is a need to distinguish between the "F" and "N" designations since this implementation of GT-FAST focuses on the distribution of payloads and resources (i.e., to size a module, all that is necessary to know is whether a particular component is onboard, regardless of whether it is present on another module. The distinction between "F" and "N" does, however, become useful if the components are subsystems, as in Ref. 14. If we take the case of a fractionated power subsystem through power beaming, for example, we see that an "F" indicates that power is produced in another module and beamed to the module in question, so this module must carry power receiving hardware. An "N", however, would indicate that no power beaming occurs in the design at all, so the power subsystem could be sized in a more traditional manner. Thus, although the "F" vs. "N" distinction is unimportant in this implementation of GT-FAST using payloads/resources, the nomenclature is retained for future flexibility of the tool.

LAUNCH									
Number of Launches	3	Launch Consistency Check							
	LV 1	LV 2	LV 3	LV 4	LV 5	LV 6	LV 7	LV 8	LV 9
Module 1	O	N	N	N	N	N	N	N	N
Module 2	O	N	N	N	N	N	N	N	N
Module 3	N	O	N	N	N	N	N	N	N
Module 4	N	N	O	N	N	N	N	N	N
Module 5	N	N	N	N	N	N	N	N	N
Module 6	N	N	N	N	N	N	N	N	N
Module 7	N	N	N	N	N	N	N	N	N
Module 8	N	N	N	N	N	N	N	N	N
Module 9	N	N	N	N	N	N	N	N	N
Tot. Launch Mass (kg)	340.61	148.06	134.49	0.00	0.00	0.00	0.00	0.00	0.00
Launch Cost ($FY08M)	22.00	22.00	22.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 5. Input Launch Manifest Matrix for the Example Design.

An additional note to make about Fig. 4 and Fig. 5 is that, prior to any execution of the GT-FAST sizing program, a series of consistency checks are performed on both of the two matrices to ensure that no implicit constraints are violated. For the matrix in Fig. 4, this involves verifying that the following conditions hold:

- The number of modules input above the matrix (four in Fig. 4) agrees with the number of non-blank rows (modules) within the matrix.
- Components are assigned to modules sequentially starting with Module #1 (i.e., if any rows are left blank, they are at the bottom of the matrix).
- If any column is not filled by N's, then there must be exactly one element in that column marked with the letter "P" (and all other elements in the column must contain the letter "F").
- An SSR and MDP must be present in any design to provide data storage and processing capability; thus, the last two columns in the Fig. 4 matrix cannot contain any N's.
- At least one high bandwidth downlink unit must be present in any design; thus, at least one of the two high bandwidth downlink columns in Fig. 4 must not contain N's.
- At least one 24/7 communication unit must be present in any design; thus, at least one of the two 24/7 communication columns in Fig. 4 must not contain N's.
- At least one payload must be present in any design; thus, at least one of the three payload columns in Fig. 4 must not contain N's.

For the launch manifest matrix in Fig. 5, the consistency check is somewhat simpler. This check involves verifying that the following conditions hold:

- The number of launches input above the matrix (three in Fig. 5) agrees with the number of non-blank rows (modules) within the matrix.
- Modules are assigned to launch vehicles sequentially starting with Module #1 (i.e., if any rows are left blank, they are at the bottom of the matrix).
- No modules may be assigned to multiple launch vehicles; thus, a maximum of one letter "O" may exist per row in Fig. 5.
- All existing modules defined in Fig. 4 are assigned to a launch vehicle in Fig. 5. If four modules are described in Fig. 4, then all four must be manifested in Fig. 5.

In concluding this discussion of the discrete fractionation scheme input into GT-FAST, it is important to note that the example used in Fig. 3 is just one of many possible fractionation schemes that an F6 design could take. The combinatorics involved in placing components into modules and modules into launch vehicles results in the fact that the possible designs for this problem actually number in the millions.[5] A clear advantage of a tool like GT-FAST is that, when automated, it can allow for a rapid sizing, synthesis, and trade-space evaluation even for large numbers of possible designs. Topics related to such a trade-space evaluation are covered in Ref. 5.

B. Continuous Inputs

In addition to the discrete inputs involving fractionation scheme, several inputs to GT-FAST are directly controllable from the main input sheet (additional continuous-variable parameters are documented as assumptions in Section III and can also be changed if necessary by modifying the models used). These inputs can be grouped into the three broad categories of orbit, payload, and margin.

In terms of orbit-related inputs, the GT-FAST user must specify the altitude and inclination of the desired orbit for the F6 cluster. The baseline implementation of GT-FAST assumes that the orbit is a circular low-Earth orbit (LEO), although the program has been demonstrated to be adaptable to non-LEO orbits. These altitude and inclination inputs allow GT-FAST to select launch vehicles and to budget propellant for orbit maintenance against atmospheric drag. If a higher-fidelity power subsystem model is used by GT-FAST in the future, this information can also be used to estimate the percentage of an orbit in eclipse (i.e., for battery charging and discharging). The example design used throughout this paper assumes a 370 km altitude and 28.5° inclination.

In addition to orbit altitude and inclination, the estimation of orbit maintenance ΔV requires inputs for mission duration and vehicle ballistic coefficient. As will be documented in Section III, the propellant estimation model for GT-FAST also includes attitude control propellant and residuals; any propellant that does not fit one of these three categories can be input by the user as a ΔV. Engine I_{sp} is required to convert all ΔV values to propellant masses. Currently all of these inputs are assumed to be the same for each module, although future versions of GT-FAST may allow for non-homogeneous mission durations, ballistic coefficients, orbital elements, etc. The example design used throughout this paper assumes a 2-year mission duration, 110 kg/m² ballistic coefficient (based on average values from Ref. 15), 300 s specific impulse (representative of a bipropellant hypergolic thruster), and no additional user-defined ΔV requirements.

Payload inputs include the mass, power, and pointing requirement for each of the up to three payloads allowed in the current GT-FAST implementation. Power inputs are divided into both sunlit and eclipse requirements, allowing a user to input a low or zero eclipse power requirement, for example, if a payload is a visual imager. Mass and power inputs directly feed into the mass and power budgets for the modules carrying the corresponding payloads. Pointing requirements (coupled with a fourth non-payload pointing requirement which could be used to account for communications antenna pointing, for example) affect attitude determination and control system (ADCS) cost estimates from the Small Satellite Cost Model 2007 (SSCM07).[16] It deserves note that the GT-FAST requirement of only four inputs per payload allows portability in that only minimal information need be passed between payload designers and GT-FAST users. In the example design for this paper, Payload #1 is modeled after the NOAA-N Search and Rescue Repeater (SARR) instrument,[17] Payload #2 is modeled after the transponder payload of the Orbcomm LEO communications satellite,[15,18] and Payload #3 is modeled after the science sensor payloads of the recent Interstellar Boundary Explorer (IBEX) spacecraft.[17] Although this payload set is notional, it highlights the potential for F6 to accommodate a variety of diverse payloads within a single fractionated infrastructure.

Finally, the user may specify four independent margin percentages to be used for dry mass, propellant mass, power, and cost. These margins are added to each of the respective budgets for each module to account for possible growth during the development, production, and operations of the program. Special notes to make are that the cost margin is not applied to the launch vehicle, and the mass margin is not applied to the launch adapter mass. In the example design for this paper, 25% margin is used for dry mass, propellant mass, power, and cost.

Table 1. Assumed Payload Characteristics for Example Design.[15,17,18]

Payload No.	Payload Description	Flight Heritage	Mass (kg)	Sunlit & Eclipse Power Requirement (W)	Pointing Requirement (deg.)
1	Search & Rescue Repeater	NOAA-N	24.0	53	1.0
2	LEO Transponders	Orbcomm	8.4	10	5.0
3	Sensors and Electronics Unit	IBEX	26.0	16	0.5

III. Sizing and Costing Models

At the core of GT-FAST is a set of mass, power, and cost estimating relationships constructed primarily from Refs. 15 and 16 and complemented by estimates from one satellite manufacturer. In this section, we survey the sizing and costing models used by GT-FAST. First, we survey the power and mass models by subsystem. Second, we survey the cost models by line item, including a discussion of launch vehicle selection. Although this section describes the GT-FAST models as currently implemented, it should be kept in mind that these models are modular and can be (and have been) adapted if a user prefers to use a model better suited for a particular application.

A. Mass and Power Modeling

Individual modules are sized to be independent, free-flying spacecraft, allowing for the application of mass and power estimating relationships from sources such as Ref. 15. The mass and power models for the majority of subsystems (propulsion, attitude determination and control, thermal, power, and structures) aboard each module are no different from typical models for conceptual design which will be described next. Depending on the components present on a given module, the communications subsystem and command and data handling (C&DH) subsystem may use modified mass and power models, which will also be described next.

1. Models for Typical Subsystems

Since the only fractionatable components in this implementation of GT-FAST involve communications, data storage, and data processing, the subsystems of propulsion, attitude determination and control, thermal, power, and structures are sized as usual for a first-order conceptual design. In terms of mass, this means that historical percentages[15] are used which relate a subsystem mass to the total dry mass of the module. For example, using historical data for LightSats,[15] the structural mass of a module is expected to be 22.7% of the total dry mass. In the example design of Fig. 3, the resulting dry mass of Module #3 is 88.3 kg (before margin is applied); correspondingly, the structures subsystem mass estimate is 20.0 kg. This method of modeling based on historical percentages also applies to the communications and C&DH subsystems when no high bandwidth downlink or 24/7 communication units are included on a given module.

In terms of power, typical subsystems use a set of power estimation relationships from Ref. 15. These relationships are more complex than the mass percentages and use different models depending on the total power requirement of the module. If the total module power requirement is below 100 W, Ref. 15 recommends a particular fixed power level for each subsystem. If the total power is between 100 W and 500 W, a percentage of the total power is recommended, and if the total power is above 500 W, a different percentage is recommended.

To avoid convergence issues near the 100 W and 500 W boundaries and to provide continuity in the power estimate, a smoothing function is applied to the power model in the vicinity of the boundaries. The smoothing function *f* below is a third-order polynomial which describes the relative weighting between the two power estimates of the Ref. 15 model in the vicinity of a boundary. At the boundary itself (i.e., 100 W or 500 W), the two estimates are equally weighted and $f = 0.50$. At 20% above the boundary (i.e., 120 W or 600 W), $f = 1$, and at 20% below the boundary (i.e., 80 W or 400 W), $f = 0$. Thus, *f* describes the weighting on the power estimate above the boundary; as a result, the weighting on the power estimate below the boundary is $1-f$. The polynomials that describe *f* as a function of total power *P* are shown in Eq. (1) and plotted in Fig. 6. As an example, Fig. 7 shows the result of smoothing on one representative subsystem (ADCS) power requirement. Note the C^1 (and C^0) continuity[§§] provided by the smoothing function as opposed to the original discontinuous model from Ref. 15.

Note that it is assumed that these power relationships apply both to sunlit and eclipse periods; thus if the power requirement of the payload for a given module is also constant between sunlight and eclipse, the sunlight and eclipse power requirements are identical. Additionally, it should be noted that this model has no coupling between power and mass estimates (although higher-fidelity, coupled models could easily be implemented in the future).

$$\begin{aligned} f_{100} &= -0.00003125P^3 + 0.009375P^2 - 0.90P + 28 \\ f_{500} &= -0.00000025P^3 + 0.000375P^2 - 0.18P + 28 \end{aligned} \tag{1}$$

[§§] For a brief discussion of simple C^0 continuity and C^1 first-derivative continuity, the reader is referred to Ref. 19.

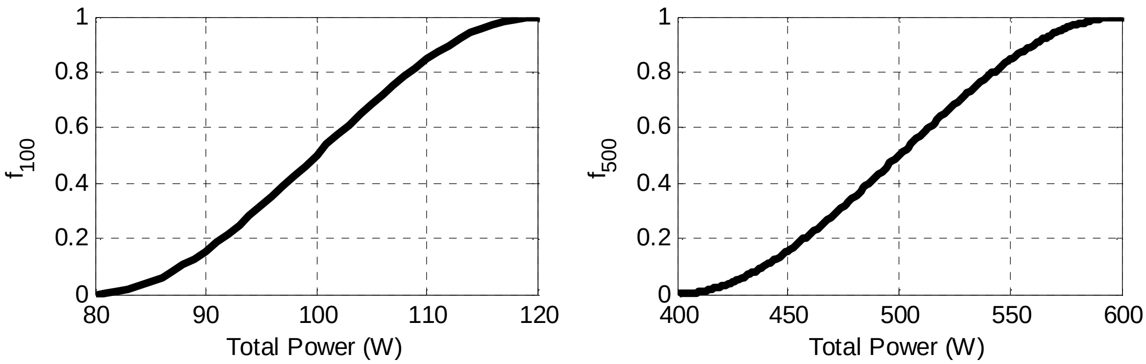

Figure 6. Smoothing functions for the 100 W boundary (left) and 500 W boundary (right).

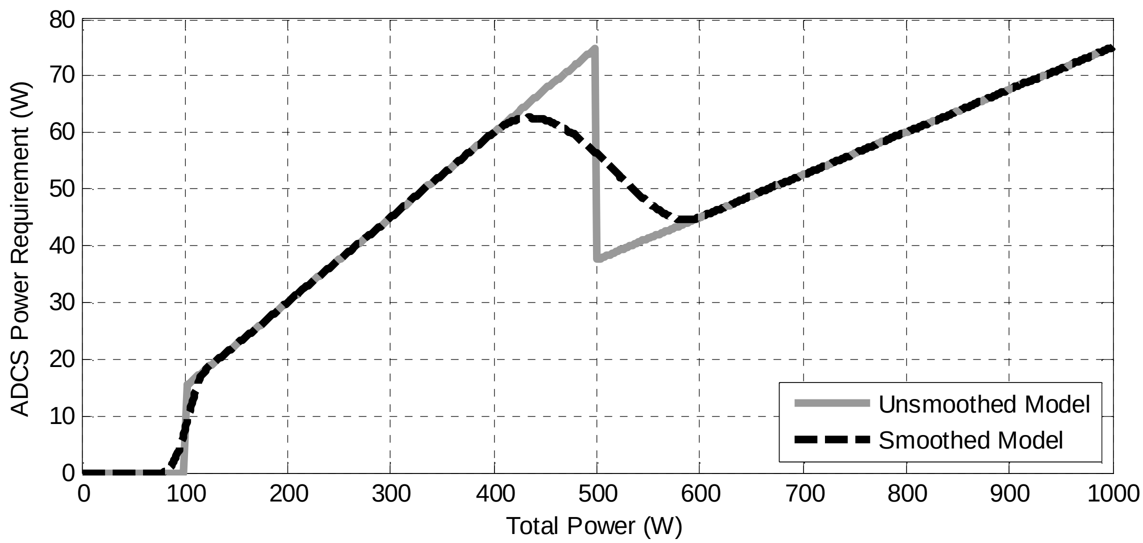

Figure 7. Example (ADCS) variation in power requirement with total spacecraft power.

2. Models for Fractionation-Affected Subsystems

In the current implementation of GT-FAST, the sizing of the communications and C&DH subsystems is directly affected by fractionation, and these cannot be properly sized based on historical data (since no such data exist for fractionated systems). Instead, these subsystems are sized using a set of rules which define the subsystem components that are present on a module given the fractionation scheme.

The components of the communications subsystem for any given module include the high bandwidth downlink and 24/7 communication units allocated to that module as well as an intra-cluster wireless unit and AFSCN link equipment. The intra-cluster wireless unit and AFSCN link equipment are included by default for all modules; the former allows for wireless communications between modules, and the latter allows low-bandwidth communication with an AFSCN-equivalent ground station network. The only exception to the inclusion of the intra-cluster wireless unit on all modules is that the unit is excluded in instances where only one spacecraft exists in the architecture (in which case there is presumably no need for wireless communications between modules). The mass of the communications subsystem is the sum of the masses of each component present. The power requirement of the communications subsystem is based on the assumption that the module always uses the intra-cluster wireless unit and only one external link at a time. Thus, if a module carries a 24/7 communication unit, high bandwidth downlink unit, and AFSCN link equipment, only the largest of these three power requirements is added to the power required by the intra-cluster wireless unit. No distinctions are made between sunlit and eclipse periods, so the power requirements during sunlight and eclipse are equal.

The command and data handling subsystem for any given module consists of the solid state recorders (SSRs) and mission data processors (MDPs) allocated to that module as well as a minimum C&DH unit providing basic processing and storage capabilities. The minimum C&DH unit, which has a mass of 5.5 kg and power requirement of 15.5 W based on Ref. 15, is present on all modules to provide basic functionality even if an SSR or MDP is not present on that module. The mass of the C&DH subsystem is the sum of the masses of each component present, and the power requirement is the sum of the power requirements for each component present. As with the communications subsystem, no distinctions are made between sunlit and eclipse periods.

3. Propellant Mass Estimation

Propellant mass for each module created in GT-FAST is budgeted among the five areas of orbit maintenance, additional maneuvers, attitude control, residuals, and margin. Orbit maintenance propellant is estimated through the expression for ΔV in Eq. (2), which uses a module's mission duration (*t*), altitude, and estimated ballistic coefficient (β). Altitude is used to estimate velocity (*V*) assuming a circular orbit, and altitude is also used to estimate atmospheric density (ρ) under the conservative assumption of a solar maximum period using data from Ref. 15. The ΔV for additional maneuvers not associated with any of the other four categories is left as a user input. This ΔV is converted to a propellant mass through the ideal rocket equation. In the example design shown throughout this paper, the ΔV for orbit maintenance is 110.0 m/s per year, totaling to 220.0 m/s overall. The ΔV associated with additional maneuvers is assumed to be zero for the example application.

The remaining three areas of propellant budget are estimated as percentages of propellant mass rather than ΔV values. Attitude control propellant is estimated at 6.5% of the total propellant budget, and residuals are estimated at 1.5% of the total budget according to Ref. 15. Propellant margin is user-defined, and in the example used in this paper, propellant margin is set at 25%. As a result of these three additional propellant requirements, the total propellant budget for each module in the example design is 322.6 m/s.

The propellant mass estimation model described here is applied to all modules in a GT-FAST architecture, although specialized assumptions (for example, different mission durations for different modules) can be applied in future implementations of the tool. In one example of a modified GT-FAST tool, a geosynchronous communications satellite was modeled; modifications to the propellant estimation models involved the addition of geosynchronous-orbit-specific stationkeeping requirements and disposal orbit requirements.

$$\Delta V = \frac{\rho V^2 t}{2\beta} \tag{2}$$

B. Cost Modeling

In terms of cost modeling, GT-FAST in its current implementation for Earth-orbiting F6 designs primarily draws upon the Small Satellite Cost Model 2007 (SSCM07)[16], although other models are used for estimates that SSCM07 does not provide (for example, software, ground segment development, and launch costs). One challenge in using traditional satellite cost models for fractionated architectures is that these models are regressions from previous programs and are inherently biased toward architectures consisting of a single spacecraft. As a result, the regressor variables in the traditional cost estimating relationships (CERs), which are often subsystem masses, refer to properties of a single monolithic spacecraft and not to a spacecraft cluster. Thus, some of these CERs are reasonably applied to properties of individual modules in the cluster, while others are more reasonably applied to properties of the cluster as a whole. For example, it would make little physical sense to apply the propulsion subsystem CER, which uses propulsion subsystem dry mass as the regressor, at the cluster level (which would imply that several small propulsion subsystems on independent, free-flying modules should have the same cost as one large subsystem on a monolithic spacecraft). On the other hand, the program management/systems engineering CER would be more appropriately applied to overall metrics of the entire cluster (i.e., applying this CER on a per-module basis would imply that program management on each vehicle is independent, which would likely be an overestimation). Thus, costs are divided into costs estimated at the module level and costs estimated at the cluster level.[***] In the present implementation, all costs are reported in fiscal year 2008 dollars ($FY08).

1. Module-Level Cost Estimation

GT-FAST accounts for subsystem, payload, and assembly, test, and launch operations (ATLO) costs on a per-module basis. In terms of subsystem costs, SSCM07 is used almost exclusively. For module wet masses below 125 kg, the SSCM07 Micro Satellite CERs are used, and for all other wet masses the Small Satellite CERs are used. Typical inputs into an SSCM07 subsystem cost model include the dry mass of the subsystem and subsystem-specific

[***] This modeling strategy can be rigorously refined when actual fractionated spacecraft are flown and cost data becomes available.

parameters. For example, the propulsion subsystem cost model requires as inputs the propulsion subsystem dry mass as well as type of propellant (cold gas, hydrazine monopropellant, or hydrazine bipropellant). Within GT-FAST, propulsion subsystem dry mass is known from mass sizing described in Section III.A of this paper, and propellant type is automatically inferred from the continuous input of I_{sp} also described in Section II. The SSCM07 equation relating these inputs to an estimated cost (which can be broken into recurring and nonrecurring parts) is typically nonlinear. The only deviation from SSCM07 in terms of subsystem CERs is due to the fact that SSCM07 costs the C&DH and communications subsystems as a single unit based on the total mass of the two subsystems; for accounting purposes, this total cost is split such that 58% is assigned to the C&DH subsystem and 42% is assigned to the communications subsystem based on dry mass split data from Ref. 15.

Payload cost is estimated as 40% of the module bus cost based on a CER from Ref. 15. Assembly, test, and launch operations (ATLO) cost is estimated for each module based on SSCM07. By the SSCM07 definition, ATLO consists of the combination of the categories of integration, assembly, and test (IA&T) and launch and orbital operations support (LOOS). ATLO cost estimation inputs include module wet mass (calculated from the mass sizing of Section III.A) to determine whether to use the Micro Satellite or Small Satellite CERs, design lifetime (from user inputs), and module power (from the larger of either sunlit or eclipse operating power calculated as in Section III.A).

2. Cluster-Level Cost Estimation

GT-FAST accounts for program management/systems engineering (PMSE), flight software development, ground segment development, operations, and launch costs at the cluster level. PMSE cost is estimated using SSCM07 relationships. Inputs to the PMSE cost model are the total cost of all module buses (each calculated on a per-module basis as described in Section III.B.1), the total ATLO cost for all modules (also each calculated on a per-module basis), and the total wet and dry masses of all modules (calculated as described in Section III.A).

Flight software cost is estimated based on relationships available in Ref. 15 for cost per thousand lines of code. Nominally, it is estimated based on Ref. 15 that each module requires 26,000 lines of flight software code, and GT-FAST scales lines of code directly with the number of modules in the cluster. This is likely to produce a conservative estimate of software cost since, although each module would have unique components aboard, there may be cost savings due to elements of commonality.

Ground segment development cost is also estimated based on Ref. 15. The ground segment development cost includes ground station facilities, equipment, software, logistics, and system-level costs. The breakdown between each of these various components of the cost is given by a set of typical percentages from Ref. 15, and the absolute magnitude of the ground segment development cost is anchored upon a ground software cost estimate under the assumption of 100,000 lines of code from Ref. 15. GT-FAST can also allow the user to override the Ref. 15 ground segment development cost model with a custom estimate (for example, if new ground stations do not need to be developed).

Operations cost is estimated based on a publicly-available mission operations cost model from NASA.[20] Inputs into this first-order model include the investment cost of the system (total development and production cost, excluding launch costs) and duration of the mission (a user input described in Section II.B). Normally, use of the NASA model requires specification of mission type; in this implementation of GT-FAST, estimates are produced for both Earth observation and communication mission types and averaged since either (or both) of these missions may be executed by a fractionated design, depending on the payloads carried.

Launch cost estimation is accomplished through a three-step process for each launch in the prescribed manifest (e.g., see Fig. 5). First, the total mass capability required by the launch vehicle is calculated by adding the individual wet masses of each module aboard. In order to account for structural mating of each module to the launch vehicle, an adapter mass of 18.8 kg is added to each individual module mass based on example vehicles in Ref. 15. Second, a database of launch vehicle capability relationships[†††] is used to compute the maximum payload deliverable to the user-specified orbit for each of 20 expendable launch vehicles[‡‡‡]. Third, GT-FAST identifies the launch vehicles with deliverable payload capabilities greater than or equal to the mass to be launched and selects the lowest-cost option based on a launch vehicle cost database compiled from Refs. 22 and 23. This three-step procedure is repeated for each scheduled launch in the manifest. Note that this assumes launch vehicles are purchased independently for each of the launches at the prescribed price (no discounts are assumed, for example, if

[†††] For an example of such relationships using response surface equations (RSEs), the reader is referred to Ref. 21.

[‡‡‡] Currently, GT-FAST's launch vehicle database is limited to American launch vehicles, but this database will be expanded in the future to include foreign options.

all launches use the same vehicle). Although launch vehicle reliability does not factor into the computer's automated selection of launch vehicles, GT-FAST does allow the user to exclude launch vehicles from consideration (for example, if reliability is too low to merit consideration).

C. Model Integration

As mentioned in Section II, one reason for selecting Microsoft Excel as a platform for GT-FAST was its automatic iteration capability. As a result, circular references among cells can be made and automatically evaluated without explicit programming of iteration procedures. This capability is utilized extensively by GT-FAST. Each module in a given architecture is represented by a power, mass, and cost breakdown in a dedicated worksheet within the tool (see Fig. 8), and formulae within the cells of each worksheet are allowed to reference other values within the worksheet or within other worksheets. Use of Visual Basic code within Microsoft Excel allows worksheets to be automatically created and configured according to the number of modules, components contained within modules, and other inputs specified by the user.

As a result, once a user inputs a fractionation scheme and series of continuous inputs as described in Section II, GT-FAST creates a sizing worksheet for each module and automatically iterates both within worksheets and among worksheets in order to determine the mass, power, and cost breakdown for each module and for the entire architecture. It should be noted that most sizing and costing relationships are based on parametric scaling relationships and not discrete unit masses, power requirements, or costs.

Figure 8. Worksheets from GT-FAST for the example design in this paper.

IV. Example Outputs

In this section, examples of GT-FAST outputs are provided. Shown first are the mass, power, and cost budgets for a fully-sized point design (the example design used throughout this paper). Next, it is demonstrated how GT-FAST can be used to conduct trade studies among program cost and user-defined metrics. Finally, results of a partial validation of GT-FAST against two monolithic satellites are presented.

A. Example Point Design

Tables 2-5 show the mass, power, and module-level cost budgets for each of the four modules for the example design used throughout this paper. Recall that the configuration of this design is shown by Fig. 3, the payloads it carries are defined in Table 1, and it is assumed to be in a 370 km circular orbit at 28.5° inclination for a two-year mission. Table 6 documents the estimated cost budget for the entire system, which includes costs estimated at the module level and cluster level. These mass, power, and cost budgets represent the typical core outputs of GT-FAST.

Figure 9 graphically shows the cost breakdown of Table 6. Note that the GT-FAST cost models here assume a ground segment development cost and margin, which together comprise 33% of this budget; these items are particularly easy to adjust if the user wishes to use custom estimates. As shown in Fig. 10, the launch vehicle selected for all three launches is the Pegasus XL at a cost of $22 million (FY08).[§§§] The Pegasus XL's 450 kg payload capacity to the desired orbit was sufficient for all launches, and $22 million was the lowest launch cost in the database used for this launch vehicle selection (foreign and under-development vehicles were excluded).

[§§§] Note that GT-FAST does not require all launches to use the same launch vehicle; this coincidence is due to the particular payload requirements for this set of launches.

Table 2. Mass, Power, and Cost Budgets for Module #1

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	24.0	53.0	53.0	16.0
2.0	Bus Subsystems				
2.1.	Propulsion	4.4	0.1	0.1	6.4
2.2.	Attitude Control	18.4	61.1	61.1	2.9
2.3.	Communications	10.1	95.0	95.0	7.8
2.4.	Command & Data Handling	26.0	70.4	70.4	10.8
2.5.	Thermal	2.8	20.4	20.4	0.5
2.6.	Power	40.0	108.7	108.7	9.2
2.7.	Structures & Mechanisms	36.9	0.0	0.0	2.6
	Pre-Margin Subtotal	*162.4*	*408.7*	*408.7*	*56.2*
	Margin	40.6	102.2	102.2	[See Table 6]
	Post-Margin Subtotal	*203.0*	*510.9*	*510.9*	
3.0	Propellant	23.5			
	Loaded Mass	*226.5*			
4.0	Adapter	18.8			
	Boosted Mass	*245.3*			
	ATLO Cost				5.5

Table 3. Mass, Power, and Cost Budgets for Module #2

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	0.0	0.0	0.0	0.0
2.0	Bus Subsystems				
2.1.	Propulsion	1.8	0.0	0.0	5.4
2.2.	Attitude Control	7.4	58.2	58.2	1.0
2.3.	Communications	18.8	177.6	177.6	0.3
2.4.	Command & Data Handling	5.5	15.5	15.5	0.4
2.5.	Thermal	1.1	19.4	19.4	0.5
2.6.	Power	16.1	117.3	117.3	0.8
2.7.	Structures & Mechanisms	14.9	0.0	0.0	2.8
	Pre-Margin Subtotal	*65.6*	*388.0*	*388.0*	*11.1*
	Margin	16.4	97.0	97.0	[See Table 6]
	Post-Margin Subtotal	*82.1*	*484.9*	*484.9*	
3.0	Propellant	9.5			
	Loaded Mass	*91.6*			
4.0	Adapter	18.8			
	Boosted Mass	*110.4*			
	ATLO Cost				2.0

Table 4. Mass, Power, and Cost Budgets for Module #3

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	8.4	10.0	10.0	4.7
2.0	Bus Subsystems				
2.1.	Propulsion	2.4	0.0	0.0	5.7
2.2.	Attitude Control	10.0	59.5	59.5	1.0
2.3.	Communications	18.8	177.6	177.6	0.3
2.4.	Command & Data Handling	5.5	15.5	15.5	0.4
2.5.	Thermal	1.5	19.8	19.8	0.6
2.6.	Power	21.7	114.0	114.0	0.9
2.7.	Structures & Mechanisms	20.0	0.0	0.0	2.8
	Pre-Margin Subtotal	*88.3*	*396.4*	*396.4*	*16.4*
	Margin	22.1	99.1	99.1	[See Table 6]
	Post-Margin Subtotal	*110.3*	*495.5*	*495.5*	
3.0	Propellant	12.8			
	Loaded Mass	*123.1*			
4.0	Adapter	18.8			
	Boosted Mass	*141.9*			
	ATLO Cost				2.0

Table 5. Mass, Power, and Cost Budgets for Module #4

	Subsystem	Mass (kg)	Sunlit Power (W)	Eclipse Power (W)	Cost (FY08$M)
1.0	Payload	26.0	16.0	16.0	14.8
2.0	Bus Subsystems				
2.1.	Propulsion	3.4	0.0	0.0	6.1
2.2.	Attitude Control	14.3	44.6	44.6	3.1
2.3.	Communications	15.2	95.0	95.0	6.9
2.4.	Command & Data Handling	5.5	15.5	15.5	9.6
2.5.	Thermal	2.1	14.9	14.9	0.4
2.6.	Power	31.0	111.6	111.6	8.6
2.7.	Structures & Mechanisms	28.6	0.0	0.0	2.3
	Pre-Margin Subtotal	*126.2*	*297.6*	*297.6*	*51.8*
	Margin	31.5	74.4	74.4	[See Table 6]
	Post-Margin Subtotal	*157.7*	*372.1*	*372.1*	
3.0	Propellant	18.3			
	Loaded Mass	*176.0*			
4.0	Adapter	18.8			
	Boosted Mass	*194.8*			
	ATLO Cost				4.5

Cost Element	Cost (FY08$M)
Module-Level Costs	
Module #1	61.6
Module #2	13.1
Module #3	18.4
Module #4	56.3
Program Management	23.4
Software	51.7
Ground Segment Development	75.6
Operations	33.1
Pre-Margin Subtotal	*333.2*
Margin (25%)	83.2
Post-Margin Subtotal	*416.4*
Launch	66.0
Total	**482.4**

Table 6. Overall Example Design Cost Budget

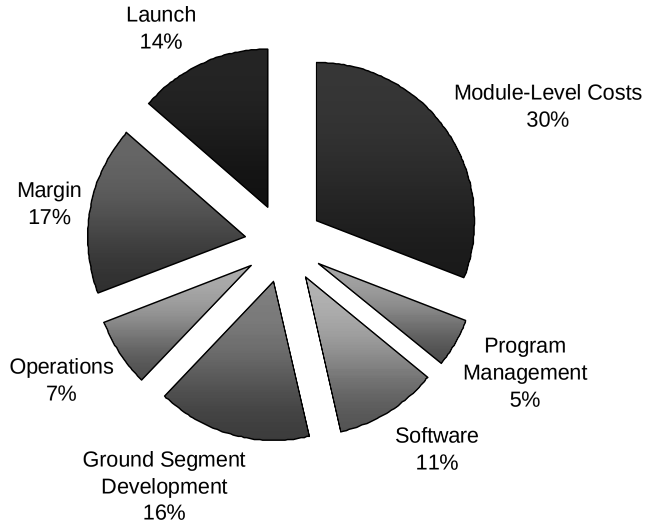

Figure 9. Breakdown of Costs from Table 6.

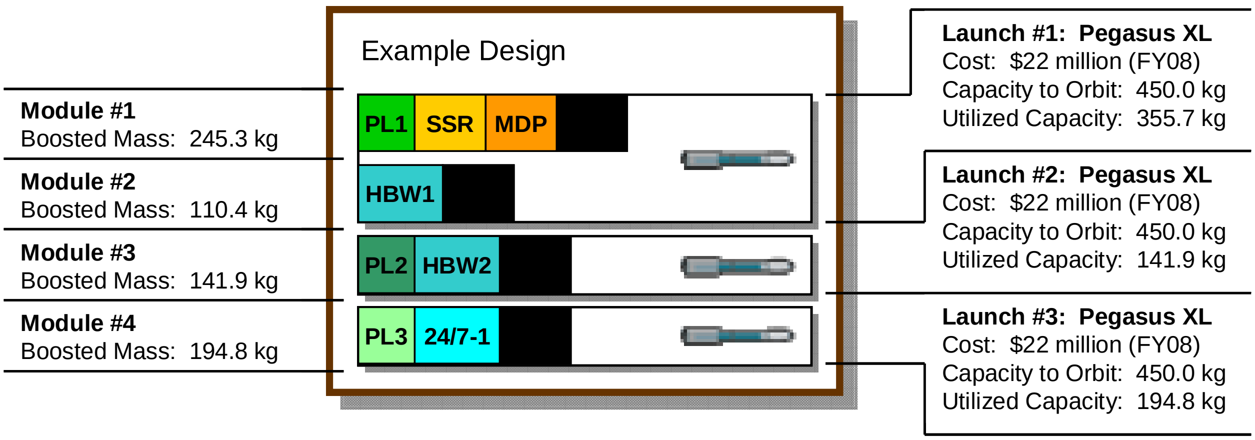

Figure 10. Launch Summary for Example Design.

B. Example Trade Study Capability

In addition to mass, power, and cost budget outputs for a single point design, GT-FAST can be used to conduct rapid trade studies among these budget outputs and user-defined output metrics. Shown here is an example of trade study between the parameters of program cost and the average cost to add or replace a fractionatable component. In this trade study, only 6 components were considered (see Fig. 11; compared to Fig. 2, the third payload was omitted, as were the redundant 24/7 communication and high-bandwidth downlink units). The payloads were a 20-kg, 50-W class payload (PL1) and a 10-kg, 70-W class payload (PL2), and the mission was at a 370 km altitude circular, 28.5° inclination orbit with a duration of two years.

Figure 11. Icons for fractionated components in example trade study.

1. Program Cost Distribution

For the purposes of the trade study example shown here, the variation considered among designs is the configuration itself (i.e., the many different combinatorial ways of filling out the matrices such as in Figs. 4 and 5). Details on the enumeration of these various designs are available in Ref. 5; in this trade study, 3190 designs are enumerated and considered. Thus, 3190 different total costs (i.e., the bottom line of budgets as in Table 6) are output by GT-FAST. These can be formed into a histogram, shown on the top of Fig. 12. In this particular problem, the minimum program cost is $249 million and the maximum is $548 million, with a median program cost of $419 million. Recall again that this includes ground segment development costs and a 25% margin.

2. User-Defined Output Metric Distribution

The second component required of this example trade study is a measure of the benefit or relative value of each of the 3190 designs considered. A representative metric chosen here is the average cost of adding or replacing a component of the cluster. This metric, $C_{add/replace}$ as defined in Eq. (3), considers the fact that a given single component i can be added to the cluster in one of two practical ways. First, the user could choose to launch the needed component as part of a module that is a duplicate of one that is already on-orbit. This strategy takes advantage of the fact that no research, development, test, and evaluation (RDT&E) costs are incurred since the module has already been manufactured before. The cost to implement this option is reflected as $C_{i,existing}$ in Eq. (3). The second option for the user is to simply launch a module with the single component i that is needed (for an example of a single-component module, see Module #2 in Fig. 10). This strategy takes advantage of the low cost associated with a small, single-component module but has the disadvantage that, unless this module had been developed for the original cluster, RDT&E costs are incurred. The cost of this option is $C_{i,separate}$ in Eq. (3).

The $C_{add/replace}$ metric is based on the idea that a user would prefer the lowest-cost option when it comes to adding or replacing a single component. However, since it is not obvious which components will require addition or replacement, the average is taken over all the n possible components of the lowest-cost addition/replacement options. This is reflected in Eq. (3), and this metric is evaluated in GT-FAST for each of the 3190 designs considered.[****] These can be formed into a histogram, shown on the right of Fig. 12. In this particular problem, the minimum $C_{add/replace}$ is $42.5 million and the maximum is $83.5 million, with a median of $52.2 million. If only this objective were considered, a fully-fractionated design (consisting only of single-component modules) would be optimal since each single-component module is pre-developed.

$$\overline{C}_{add\,/\,replace} = \frac{1}{n}\sum_{i=1}^{n}\min\left(C_{i,separate}, C_{i,existing}\right) \tag{3}$$

3. Cost and User-Defined Output Metric Trades

The central region of Fig. 12 shows how the two parameters of program cost and $C_{add/replace}$ compare to each other for each of the designs evaluated for this trade study. Each blue “x” in Fig. 12 represents one of the 3190 designs evaluated. The six red circles in the figure indicate the non-dominated, or Pareto-optimal, designs in the trade space, and the red line indicates the interpolated Pareto frontier. It is first quite notable that this simple analysis has narrowed the trade space to just 6 designs (a 530-fold reduction in the number of designs to consider).[††††]

Also shown in Fig. 12 are the configurations of three designs on the Pareto frontier. The first design, Design A, is a monolithic spacecraft with only PL2. This design has the lowest total program cost but has one of the highest average component replacement costs. The third design, Design C, consists of a fully fractionated cluster in which every component has a dedicated module. As a result, it has a high program cost but a very low average component replacement cost since each single-component module is pre-developed. The most interesting design, Design B, lies at a very sharp corner (or knee) of the Pareto frontier and has a low program cost (though slightly higher than Design A) and a low average component replacement cost (though slightly higher than Design C). This design fractionates the payload and solid state recorder each into single-component modules but permits the 24/7 communication unit, high bandwidth downlink unit, and mission data processor to remain in the same module. This particular compromise between the economies of scale of the traditional monolith and flexibility of the fully fractionated spacecraft presents a very appealing design. Of course, it should be emphasized that Design B cannot be called “optimal” since, strictly speaking, the notion of optimality does not exist in a multi-objective problem.

As indicated earlier, this simple two-objective demonstration is meant only as an example of the type of multi-attribute trade space analysis that can be conducted for fractionated spacecraft with the capabilities of GT-FAST. Clearly additional objectives exist, and these considerations and others (including details on how designs are combinatorially enumerated) are addressed in Ref. 5.

[****] It deserves emphasis that this metric is just one of many that a user may define and track through GT-FAST.

[††††] Although this assumes the user is only interested in the two metrics shown, this same procedure and concept can be applied to additional objectives and still result in vast reductions in the trade space.

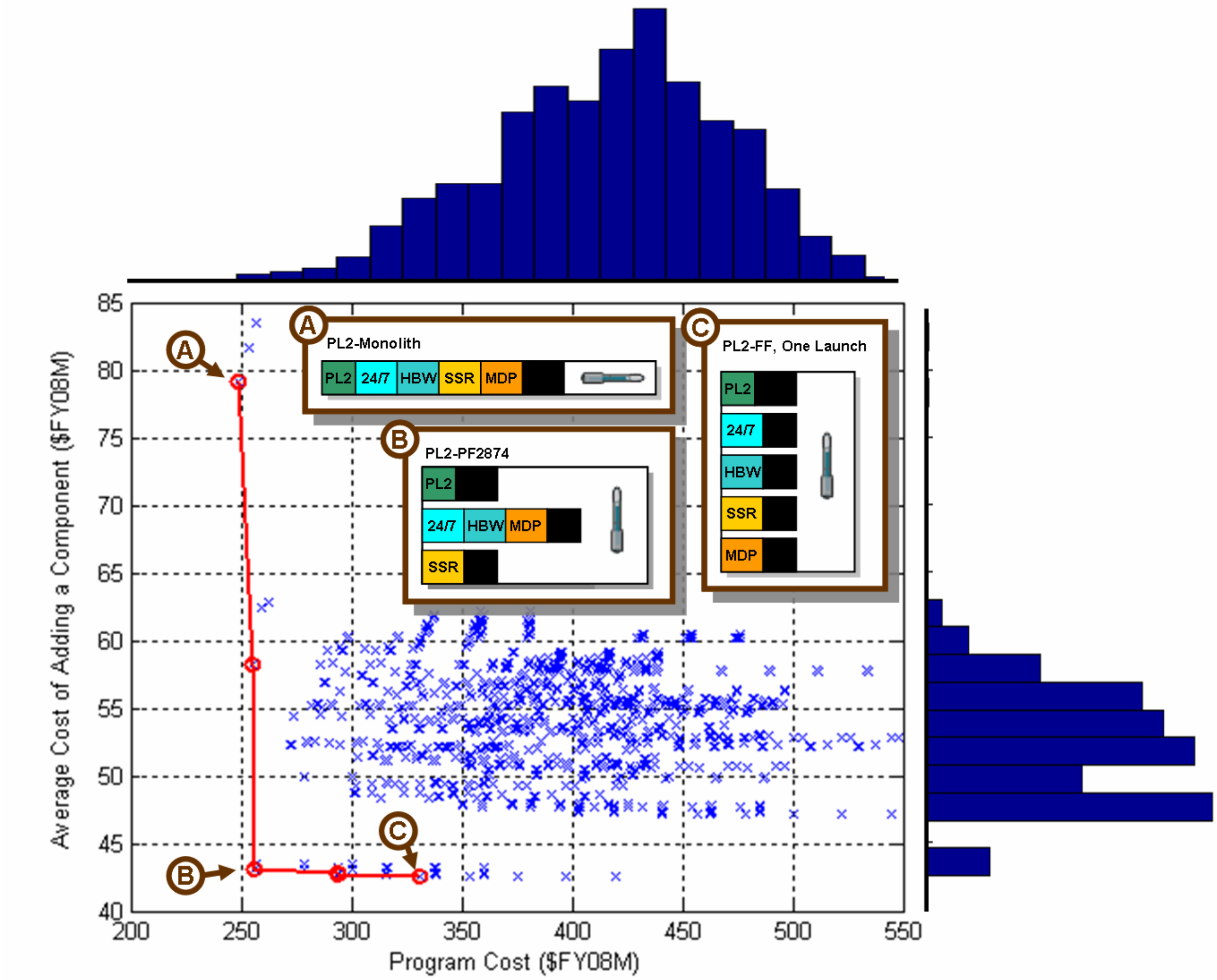

Figure 12. Pareto Frontier and Pareto-Optimal Designs for Example Trade Study.
Here, each blue "x" represents one of 3190 possible designs, the red circles indicate the non-dominated designs, and the red line indicates the interpolated Pareto frontier. Three interesting designs are noted by the letters A, B, and C.

C. Comparison against Operational Monolithic Spacecraft

While a fractionated spacecraft has yet to launch, it is possible to partially demonstrate the accuracy of GT-FAST in comparison with existing monolithic spacecraft. Used in this comparison are the Jason-2 and TIMED spacecraft, both of which are approximately of the small-satellite class and are currently operational in orbit. Neither spacecraft was used in the generation of the models in Ref. 15 that GT-FAST draws upon for several mass and power estimates. Jason-2 (see Fig. 13) is a follow-on mission to Jason-1, aiming to continue the data record of Jason-1 and measure sea surface levels to a 2.5 cm accuracy[17]. Jason-2 is a cooperative undertaking between NASA, NOAA, CNES, and EUMETSAT, and it has orbited at a circular orbit altitude of 1336 km and inclination of 66° since its launch in June 2008.[17,25] TIMED (see Fig. 14), launched on the same Delta II rocket as Jason-1 in December 2001, operates in a circular 625 km altitude, 74.1° inclination orbit.[17,27] TIMED is sponsored by NASA and was designed, built, and is operated by the Johns Hopkins University

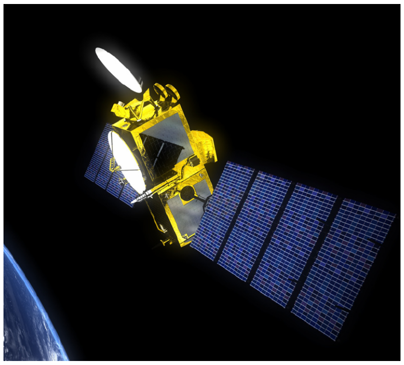

Figure 13. Artist's Concept of Jason-2 Satellite.[24]

Applied Physics Laboratory. Its mission is the global study of the physical and chemical processes acting within Earth's upper atmosphere.[17]

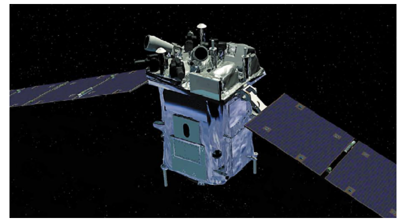

Figure 14. Artist's Concept of TIMED Satellite.[26]

In order to complete this comparison, GT-FAST is used to size a single-module cluster (i.e., a traditional monolithic spacecraft). As a result, the intra-cluster wireless unit (which has no use in a monolithic spacecraft) is automatically excluded. Additionally, the 24/7 communication units and high-bandwidth downlink units are excluded since these do not represent the actual components flown on TIMED or Jason-2.[‡‡‡‡] Thus, the remaining communication subsystem in GT-FAST consists of an AFSCN-equivalent link. In terms of command and data handling, the SSR and MDP are both included in the GT-FAST model. Propellant estimates assume an orbit maintenance ΔV as given by Eq. (2), although in the cases of TIMED and Jason-2 this number is very small due to the high altitudes of the orbits. For Jason-2, an additional 120 m/s of ΔV is included as indicated by Ref. 25. For the purposes of this comparison, no design margin is included in any budget (i.e., mass, power, propellant, or cost). The remaining inputs into GT-FAST are summarized in Table 7.

Table 8 summarizes the comparison between several actual metrics from the Jason-2 mission and their calculated counterparts in the GT-FAST model. Note that wet and dry masses agree very well (within 2.2% and 3.2%, respectively), and average power also agrees quite well for this first-order model (within 14.9%). A significant discrepancy exists in terms of cost, but this may be partially explained by substantial cost overruns and schedule slippage encountered in the Jason-2 project (to the extent that a major new instrument, the Wide Swath Ocean Altimeter, was entirely descoped in 2005).[28] An earlier 2005 Jason-2 cost estimate of $250 to $300 million[28] is much closer to GT-FAST's estimate of $250 million. Finally, although the actual Jason-2 spacecraft launched on a Delta II 7320-10,[24] GT-FAST selects the smaller and less costly Taurus 2210 launch vehicle. It deserves note that a modified Peacekeeper missile (with a smaller payload capacity than the Delta II) was considered for Jason-2 after an offer from the Department of Defense Space Test Program but was not selected because of certification and risk concerns.[28]

Table 9 is identical in format to Table 8 and summarizes the comparison between actual metrics from the TIMED mission and their calculated counterparts in the GT-FAST model. Again, wet and dry masses agree quite well, and average power is acceptable given this first-order model. In this case, cost is also very accurate (within 8.4%). In this case again, GT-FAST selects the smaller Taurus 2210 instead of the Delta II 7920-10. However, it should be noted that the Taurus 2210 could not have launched both TIMED and Jason-1 (as was done in reality); if 500 kg is manually added to the required launch capacity in GT-FAST, the model correctly predicts that a Delta II is required.

Table 7. Inputs into GT-FAST for Jason-2 and TIMED spacecraft models.[25, 27]

Spacecraft	Payload		Orbit		Pointing Requirement (deg.)	Mission Duration (years)
	Mass (kg)	Power (W)	Altitude (km)	Inclination (deg.)		
Jason-2	111.0	145.0	1336.0	66.0	0.1	5.0
TIMED	162.0	174.0	625.0	74.1	0.5	2.0

Table 8. Comparison between Actual and GT-FAST Predictions of Key Metrics for Jason-2.[24]

Spacecraft	Dry Mass (kg)	Wet Mass (kg)	Average Power (W)	Program Cost ($FY08M)	Launch Vehicle
Actual Jason-2	462.0	490.3	468.9	424.4	Delta II 7320-10
Predicted Jason-2	447.3	479.6	538.7	249.5	Taurus 2210
Prediction Error	- 3.2%	- 2.2%	+ 14.9%	- 41.2%	

‡‡‡‡ The exclusion of the 24/7 communications unit and high-bandwidth downlink unit required slight changes to GT-FAST's internal logic since, as noted in Section II.A.2, these components are normally required within a cluster to pass internal consistency checks.

Table 9. Comparison between Actual and GT-FAST Predictions of Key Metrics for TIMED.[26,27,29]

Spacecraft	Dry Mass (kg)	Wet Mass (kg)	Average Power (W)	Program Cost ($FY08M)	Launch Vehicle
Actual TIMED	592.0	592.0	401.7	239.5	Delta II 7920-10 (with Jason-1)
Predicted TIMED	521.6	523.5	498.7	219.4	Taurus 2210
Prediction Error	- 11.9%	- 11.6%	+ 24.2%	- 8.4%	

V. Conclusion

In summary, this paper has presented the internal mechanics and application of the Georgia Tech F6 Architecture Synthesis Tool (GT-FAST), a point design tool for rapid sizing and synthesis of fractionated satellite architectures. The manner in which fractionated designs are specified, including both discrete and continuous-variable inputs, was discussed, including the matrix representations of the launch manifest and placement of fractionated components in Figs. 4 and 5. Next described were the methods, models, and assumptions used in estimating elements of mass, power, and cost. The final section included sample outputs from GT-FAST for a notional fractionated architecture, demonstrated an example of the GT-FAST trade study capability, and presented a partial validation of the GT-FAST outputs against the currently-operational Jason-2 and TIMED satellites.

One important note to make is that the implementation of GT-FAST shown throughout this paper has been directed toward analysis of a DARPA F6 demonstrator intended for a circular low or medium Earth orbit. However, there is little that precludes GT-FAST from being modified for other fractionated spacecraft applications. In fact, it has already been adapted in one instance for analysis of a geosynchronous communications satellite. Existing subsystem mass, power, and cost models are interchangeable with other application-specific models a user may prefer, and launch vehicle capacity and cost models can also be easily updated. Additionally, the framework provided by the matrices in Figs. 4 and 5 makes the use of other fractionatable components (i.e., other than SSRs, MDPs, high-bandwidth downlinks, etc.) relatively simple to implement with minimal changes to internal logic. For example, earlier implementations of GT-FAST included power subsystem fractionation options through microwave power beaming hardware.

A wealth of possibilities exists for future expansion of GT-FAST. Currently, GT-FAST can size architectures consisting of up to nine fractionatable components, and future analyses may require the consideration of more components. This poses no problem to the current architecture of the GT-FAST point design tool, although it does present challenges in evaluating the resultant very large fractionated architecture trade space, as addressed in Ref. 5.

Additional future work on GT-FAST includes updates to the default launch vehicle database to include the most recent available launch vehicle performance and cost data for foreign, developmental, and proven domestic launch options (for more details, see Ref. 21). A more complete consideration of launch vehicle reliability may also be also worth consideration in future implementations of GT-FAST, and parametric cost models for fractionated spacecraft (as opposed to traditional monolithic spacecraft) would also be useful in future implementations. Also, future versions of GT-FAST might include options to size spacecraft based on discrete “parts kits” rather than based on “rubberized” parametric scaling relationships used in the present implementation. Consideration may also be given to a faster-running version of GT-FAST in MATLAB rather than the current (but more flexible) Excel platform.

Finally, a useful route for future work is the development of a comprehensive approach to defining and standardizing performance metrics for fractionated architectures. Currently, the primary metrics output by GT-FAST are mass, power, and cost. Other metrics specifying vehicle or payload performance characteristics are allowed to be user-specified and user-programmed, but it would be desirable for a standard set of such metrics to be hard-coded into GT-FAST and available to every user. A discussion of the selection of some of these metrics (and how they are traded against each other) is provided in Ref. 5.

The authors believe that GT-FAST holds significant potential for future analyses of fractionated spacecraft and represents a critical piece of any framework aimed at permitting value-informed decisions for such architectures. The rapid analysis enabled by this tool becomes particularly useful when coupled with trade-space exploration strategies such as in Ref. 5, and the expandability and adaptability of the tool permit its use for potentially a wide variety of fractionated spacecraft applications. It is hoped that GT-FAST and the ideas it represents find broad use with engineers and decision-makers considering fractionated systems in the future.

Acknowledgments

For their help in locating information on the Jason-2 and TIMED spacecraft, the authors would like to thank Philippe Landiech at CNES, David Kusnierkiewicz at APL, Alan Buis and Stephen Cole at NASA, and John Christian at the University of Texas at Austin. Thanks are also due to James Wertz from Microcosm Space Mission Engineering and Kathy Tomajko from Georgia Tech. The first author would also like to acknowledge the support of the Department of Defense through the National Defense Science and Engineering Graduate (NDSEG) Fellowship.

References

[1]"Broad Agency Announcement: System F6," DARPA Tactical Technology Office, BAA07-31, 16 July 2007, URL: http://www.darpa.mil/TTO/solicit/BAA07-31/F6_BAA_Final_07-16-07.doc [cited 21 Nov. 2008].

[2]Brown, O. and Eremenko, P., "Application of Value-Centric Design to Space Architectures: The Case of Fractionated Spacecraft." AIAA 2008-7869. AIAA Space 2008 Conference and Exposition, San Diego, 9-11 Sept. 2008.

[3]Hitch, C.J., *Decision-Making for Defense,* University of California Press, Berkeley, 1965, Chap. 3.

[4]LoBosco, D.M., Cameron, G.E., Golding, R.A., and Wong, T.M., "The Pleiades Fractionated Space System Architecture and the Future of National Security Space." AIAA 2008-7687. AIAA Space 2008 Conference and Exposition, San Diego, 9-11 Sept. 2008.

[5]Lafleur, J.M. and Saleh, J.H., "Exploring the F6 Fractionated Spacecraft Trade Space with GT-FAST." AIAA 2009-6802. AIAA Space 2009 Conference and Exposition, Pasadena, 14-17 Sept. 2009.

[6]McCullers, L.A., "Aircraft Configuration Optimization Including Optimized Flight Profiles, Multidisciplinary Analysis and Optimization, Part I." NASA CP-2327, 1984.

[7]O'Neil, D.A. and Mankins, J.C., "The Advanced Technology Lifecycle Analysis System (ATLAS)." IAC-04-IAA.3.6.3.01. 55th International Astronautical Congress, Vancouver, 4-8 Oct. 2004.

[8]Thompson, R.W., "System Architecture Modeling for Technology Portfolio Management using ATLAS." IEEE Aerospace Conference, Big Sky, 4-11 March 2006.

[9]Kipp, D.M., Dec, J.A., Wells, G.W., and Braun, R.D., "Development of a Planetary Entry Systems Synthesis Tool and Conceptual Design and Analysis." 3rd International Planetary Probe Workshop, Athens, 27 June – 1 July 2005.

[10]Alemany, K., Wells, G., Theisinger, J., Clark, I., and Braun, R., "Mars Entry, Descent, and Landing Parametric Sizing and Design Space Visualization Trades." AIAA 2006-6022. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, 21-24 Aug. 2006.

[11]Komar, D.R., Hoffman, J., Olds, A., and Seal, M., "Framework for the Parametric System Modeling of Space Exploration Architectures." AIAA 2008-7845. AIAA Space 2008 Conference and Exposition, San Diego, 9-11 Sept. 2008.

[12]Crocker, A.M., Charania, A.C., and Olds, J.R., "An Introduction to the ROSETTA Modeling Process for Advanced Space Transportation Technology Investment." AIAA 2001-4625. AIAA Space 2001 Conference and Exposition, Albuquerque, 28-30 Aug. 2001.

[13]Myers, R.H. and Montgomery, D.C., *Response Surface Methodology*, Wiley, New York, 2002.

[14]Mathieu, C. and Weigel, A.L., "Assessing the Fractionated Spacecraft Concept." AIAA 2006-7212. Space 2006, 19-21 Sept. 2006.

[15]Larson, W.J., and Wertz, J.R. (ed.), *Space Mission Analysis and Design,* 3rd ed., Microcosm Press and Kluwer Academic Publishers, El Segundo, 1999.

[16]Mahr, E.M., "Small Satellite Cost Model 2007 (SSCM07) User's Manual." ATR-2007(8617)-5. The Aerospace Corporation, 30 Sept. 2007.

[17]NASA Goddard Space Flight Center, *National Space Science Data Center Spacecraft Query* [online database]. URL: http://nssdc.gsfc.nasa.gov/nmc/SpacecraftQuery.jsp [13 Feb. 2009].

[18]Martin, D.H., Anderson, P.R., and Bartamian, L., *Communication Satellites*, 5th ed., The Aerospace Press, El Segundo, 2007.

[19]Mortenson, M.E., *Geometric Modeling*, John Wiley & Sons, New York, 1985.

[20]NASA Johnson Space Center, Mission Operations Cost Model, Online Software, 25 May 2007, URL: http://cost.jsc.nasa.gov/MOCM.html [8 May 2009].

[21]Fleming, E.S., Lafleur, J.M., and Saleh, J.H., "Response Surface Equations for Expendable Launch Vehicle Payload Capability." AIAA 2009-6656. AIAA 2009 Conference and Exposition, Pasadena, 14-17 Sept. 2009.

[22]Isakowitz, S.J., Hopkins, J.P., and Hopkins, J.B., *International Reference Guide to Space Launch Systems*, 3rd ed., AIAA, Reston, 1999.

[23]Brown, O., "Common Launch Vehicle Data for Econometric Analysis." DARPA/TTO Memorandum, 10 April 2008.

[24]Cole, S., Buis, A., et. al., "Ocean Surface Topography Mission / Jason 2 Launch." Press Kit, NASA, June 2008. URL: http://www.nasa.gov/pdf/228260main_jason-2.pdf [10 May 2009].

[25]Terrenoire, P., Douillet, F., Carucci, A., et. al., "Jason-2: Consolidating and Enlarging the Jason-1 Performance in Ocean Altimetry." IAC-03-B.2.07. 54th International Astronautical Congress, Bremen, 29 Sept. – 3 Oct. 2003.

[26]Brown, D., Neal, N., and Marren, K., "TIMED: Thermosphere, Ionosphere, Mesosphere, Energetics and Dynamics." Mission Guide, NASA and Johns Hopkins University Applied Physics Laboratory, December 2006. URL: http://www.timed.jhuapl.edu/WWW/common/content/pdfs/missionGuide.pdf [10 May 2009].
[27]Kusnierkiewicz, D.Y., "An Overview of the TIMED Spacecraft." *Johns Hopkins APL Technical Digest*, Vol. 24, No. 2, 2003, pp. 150-155.
[28]de Selding, P.B., "Data Continuity May Be Affected by Delays in Jason-2 Development." *Space News*, 31 May 2005. URL: http://www.space.com/spacenews/archive05/Jason_053005.html [10 May 2009].
[29]Yee, J., Cameron, G.E., and Kusnierkiewicz, D.Y., "An Overview of TIMED." SPIE Conference on Optical Spectroscopic Techniques and Instrumentation for Atmospheric and Space Research III, Denver, July 1999, SPIE Vol. 3756, pp. 244-254.