2015

Value-Driven Design of non-commercial systems through bargain modeling

Erik Daniel Goetzke
Iowa State University

Recommended Citation

Goetzke, Erik Daniel, "Value-Driven Design of non-commercial systems through bargain modeling" (2015). *Graduate Theses and Dissertations*. 14411.
https://lib.dr.iastate.edu/etd/14411

Value-Driven Design of non-commercial systems through bargain modeling

by

Erik Daniel Goetzke

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:
Christina Bloebaum, Major Professor
Ran Dai
Richard Wlezien

Iowa State University

Ames, Iowa

2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

CFD	Computational Fluid Dynamics
DSM	Design Structure Matrix
FAA	Federal Aviation Administration
FEA	Finite Element Analysis
FY	Fiscal Year
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
LSCES	Large Scale Complex Engineer System
MDO	Multidisciplinary Design Optimization
MOO	Multi-Objective Optimization
SE	Systems Engineering
USD	United States Dollar (Currency)
VDD	Value Driven Design

ACKNOWLEDGEMENTS

I would like to thank my advisor and committee chair, Dr. Christina Bloebaum, as well as my co-advisor, Dr. Bryan Mesmer, for their guidance and support throughout the course of this research. My lab partners, Hanumanthrao Kannan, Benjamin Kwasa, Elliott Tibor, and Christopher White, were also helpful in developing the understandings for the material presented in this document. Thank you.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

ABSTRACT

The design and development of large scale complex engineered systems requires dependence and coordination of thousands of individuals. In practice, this has shown to span industries, encompassing multiple companies and organizations, and force decisions to be driven mainly by customer requirements. One issue in this development process is related to the stakeholders' desires and their ability to effectively communicate their preferences to the design teams. Value-Driven Design is an approach stemming from systems engineering that addresses this issue by directly incorporating the operational context of the system in this communication of preference.

Value-Driven Design is formed on the premise that a design can be created that maximizes the design organization's preference. It is recognized that other preferences, possibly competing, exist as well and will have an influence on the design. This thesis explores how the negotiation of value preferences can be captured in bargaining models to determine the optimal design for the set of negotiators, taking into account conflicting preferences and player impatience. A notional strategic strike aircraft system is used as an example to illustrate the importance of design perspectives in the emerging practice of Value-Driven Design.

CHAPTER 1

INTRODUCTION

Engineers seek ways to improve the design process, targeting both the efficiency and the elegancy of the methods used. Whether it is in model development, optimization sequencing, or more business-oriented aspects such as organizational hierarchy and how to communicate preference [1-5], new ideas are being tested to enhance system design and find optimal systems. This thesis focuses on preference communication in the design process and how it affects the outcome of the final system. Presently, the customer provides requirements and specifications for the contractor to meet [6]. Any system design satisfying those conditions is considered acceptable. However, this is still a system that only satisfies constraints describing what is not desired for the system. The primary preference for the system is operationally focused, either in a business-operations sense or in a mission-completion sense, depending on the type of customer [7]. Examples of operational goals are maximizing profit, minimizing cost, and maximizing mission success. This thesis considers the ways to design systems using primary preferences.

Even with using primary preferences to drive the design process of a system, further considerations need to be made on the relationship between the buyer and the seller. The market structures for many engineered systems are not strictly competitive but lean more towards monopolies, requiring a strong relationship between the contractor and client. For many systems though, there is a clear disconnect in the desires between the two players when each is focused on their individual primary preference.

A military customer, being an entity of its respective government, may desire a new strategic strike aircraft with a high probability of mission success for supporting campaigns over the next 20-30 years. In contrast, the contractor for this program, being a publicly held entity, will have a preference for maximizing their profit when working on the system. These two preferences signify an inconsistency in the characterization of the “optimal” design. One design will have the best design characteristics related to the highest probability of mission success (at likely a steep price), the other will have the best design characteristics related to the highest profit (with likely a probability of mission success that is not the highest). It is important to understand the perspective to take in the design process. This thesis identifies the inconsistencies between perspectives and possible mechanisms to resolve the issue to better enable a design optimization process where all players are satisfied with the outcome.

CHAPTER 2

BACKGROUND

The recognition for an alternative method to communicate design preference has surfaced in the world of Large-Scale, Complex Engineered Systems (LSCES). These systems can be identified by the number of levels of integration required for the system, the amount of technology development needed, and the high cost of completing and maintaining the program. Development may last for over a decade and the costs may surpass the billion dollar mark, some on their way to 100x beyond that [8]! The aerospace industries, including civil, defense, and space, as well as the power and transportation industries are populated by LSCES. The process by which these systems are designed is changing, with a renewed focus on the operational need [5].

Value-Driven Design is an emerging method that better enables the ability to characterize optimal system designs. This is done by directly incorporating the primary preference for the system, the operational need, into the design sequence. This chapter talks about systems engineering, design optimization, and the evolving role of preference communication.

Systems Engineering

Systems engineering (SE) grew as a discipline used to tackle difficult design problems when industry leaders in the second half of the 20th century started to see that combining the best parts did not always produce the best whole. Many of the systems this trend applied to were interdisciplinary and the connections between those disciplines were typically where the troubles would lay. Before the implementation of SE, the system design would be passed from one specialized team to the next. Each team would add their contribution to the project and hand it off to the next team. This process did not allow for the teams to work together, leaving the design to be mostly driven by the first specialized team. Early aircraft programs were driven mostly by the performance team and designed for a single performance attribute, such as range, endurance, or stability, for example. This consecutive design method made it difficult for revisions or cross-discipline compromises to occur as they would prolong development time and increase the cost, even if it would improve another performance attribute. With the addition of SE to the design process, however, these compromises can be anticipated and then incorporated during the early stages of development to enable the specialized design teams to work side by side. This cooperative design method is used with the idea of creating better systems at lower costs.

The SE team will start by identifying what the customer wants and then communicate those ideas to their subsystem engineering teams, typically in the form of system requirements. The subsystem engineers will communicate additional requirements to the groups they rely on and the process continues to the end of the supply chain. This is the "Decomposition and Definition" phase in Figure 1: the classic "V-model" [6, 9] of the system development lifecycle. These requirements specify desires on technical performance attributes of the system but can also relate to economic performance, such as cost. Other requirements may be added too, such as applicable ones from the FAA, IEEE, or ISO. The system engineering teams constrain the design space and then depend on their design engineers to find a solution in the space that remains, called the feasible design space. For large scale complex engineered systems (LSCES) this work flow requires an industry.

Lockheed Martin is the prime contractor for the F-35 Lightning II aircraft, but Northrop Grumman, BAE Systems, and Pratt and Whitney all play prominent roles in the system life cycle as well [10].

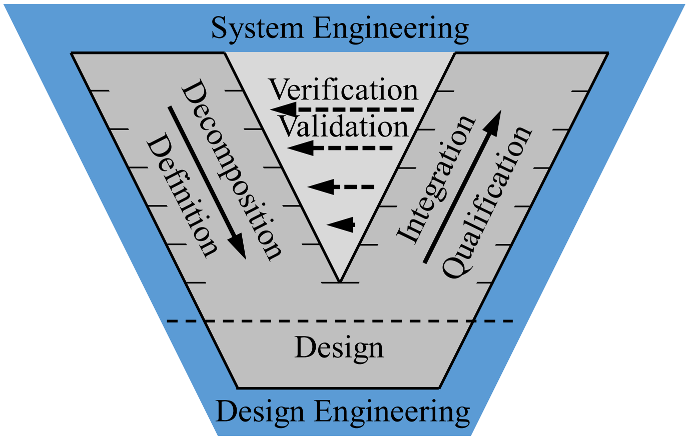


Figure 1. *Systems engineering V-model and the Traditional, Top-Down process*

Systems Engineering focuses on managing the lifecycle of the system. In the Decomposition and Definition phase, this includes managing the design process and determining which configuration best meets the stakeholders' desires. One of the current practices is for engineers to conduct trade-studies to pick which experiments to run and which results are better. This decision tool is needed most when one or more of the requirements are not yet satisfied or are competing. Trade space exploration has extensively been studied, looking at ways to best represent results in multi-dimensional problems [11, 12].

Integration and qualification procedures follow as the sub-systems and components are defined, designed, and developed. These assemblies are verified and validated along the way as well (see Figure 1), to ensure they meet the specifications that were laid out from the beginning. This process continues through all levels of subsystem assemblies to final assembly. Ideally, the final assembly meets the customer's expectations and development is complete, onward to full-scale production. If not, revisions must be made to the design, as is represented in a faded manner in Figure 1. This iteration later in the development lifecycle is costly and is hardly an option for LSCES. The development took many years; it would be cheaper to start a new program than revise the current one. The desires may have to be tailored at this phase so that the program can continue.

Multidisciplinary Design Optimization

Multidisciplinary Design Optimization (MDO) allows for more foresight in the design process. When the design space is complex, the design structure requires some iteration to find the best solution. The fields of Multidisciplinary Design Optimization and Multi-Objective Optimization (MOO) have created several frameworks to allow for this [13-15]. Working mostly in the conceptual and preliminary phases of design, MDO and MOO establish couplings between the different analyses required in the design structure. Figure 2 shows an example layout of the workflow between 4 distinct disciplines, called a design structure matrix (DSM) [16]. For aircraft design, this may include Finite Element Analysis (FEA) to estimate structural loadings and deflections, Computational Fluid Dynamics (CFD) to estimate aerodynamic loadings, stability analysis to estimate control surface sizes and response times, and performance calculations to estimate operational capabilities, to name a few. The academic world and industry alike have done well with finding innovative ways to automate the workflow and reduce overall computation time [4, 17-19]. Detailed design can be completed with MDO as well, but the simulation times may be cost-prohibitive to do on a global scale for LSCES.

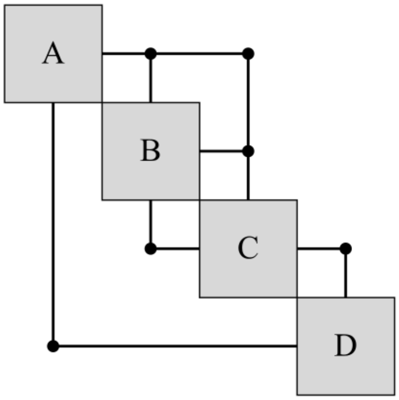


Figure 2*. Example design structure matrix*

To guide the optimization process, an objective function is used in addition to the system requirements. Objective functions may emulate a single performance attribute, such as the operating range of an aircraft, the output of a power plant, or the payload capacity of a rocket. Objective functions may account for multiple performance attributes as well, using a "weighting" scheme to prioritize the attributes. These functions typically do not have a physical meaning due to the mathematical disconnect between the attributes being "weighted" and the resulting inconsistent unit types. This evaluation criterion does enable the optimization process but is usually not something created by the customers. It is created by the engineers as an attempt to mathematically represent intuition and is still open to bias and subjectivity. Which single performance attribute should be chosen? Which collection of attributes should be chosen? How should they be weighted? Is the weighting scheme appropriate?

Preference Communication

Systems engineering uses requirements to communicate preferences on the system design. Multidisciplinary design optimization adds an objective function to this set of criteria so an optimization process can occur. Requirements work to find acceptable solutions, objective functions work to find optimal solutions. Both methods are used with the idea of finding better system designs but have fallen short in one respect: they do not communicate the operational desire. They play with this idea, they flirt with it, but they never go for it.

Value-Driven Design (VDD), an emerging design practice, takes a different approach to influencing system design. The methods and styles of work have not changed but the design preference and the way it is communicated has. VDD uses an economic based objective function to directly communicate the operational desire for the system. The truly optimal system can be found by bringing the operational context of the desired system into the design sequence. These systems will be characterized by their abilities to maximize company profit, surplus value, the probability of operational success, etc., depending on the type of system and the characteristics of the stakeholders.

Requirements Driven Design and Secondary Preferences

The well-established method of designing LSCES is by placing requirements on systems, sub-systems, sub-sub-systems, etc [6]. The group of players influencing the final design of a LSCES can work simultaneously when they are each given requirements to satisfy. Government stakeholders, like the Department of Defense (DoD) or the National Aeronautics and Space Administration (NASA), will hold design competitions to identify the better system between industry leaders' proposed systems. The competition usually starts with a request for proposal (RFP) containing a collection of requirements for the desired system to meet. Figures 3a and 3b show the design spaces when using, and not using, this method to communicate preference. Engineers have a constrained space to work in when using requirements, knowing where to look to find an acceptable solution. A downfall to this process is that all of the designs within the bounded design space are viewed as equivalent, which is highly unlikely when examining the space with primary preferences.

Requirements-driven design is not a deterministic design process. It allows for multiple optimal systems to exist and additional judgment is required to make a final decision on which system to move forward with. Using requirements to communicate design preferences on system attributes has shown to be problematic for the development of LSCES.

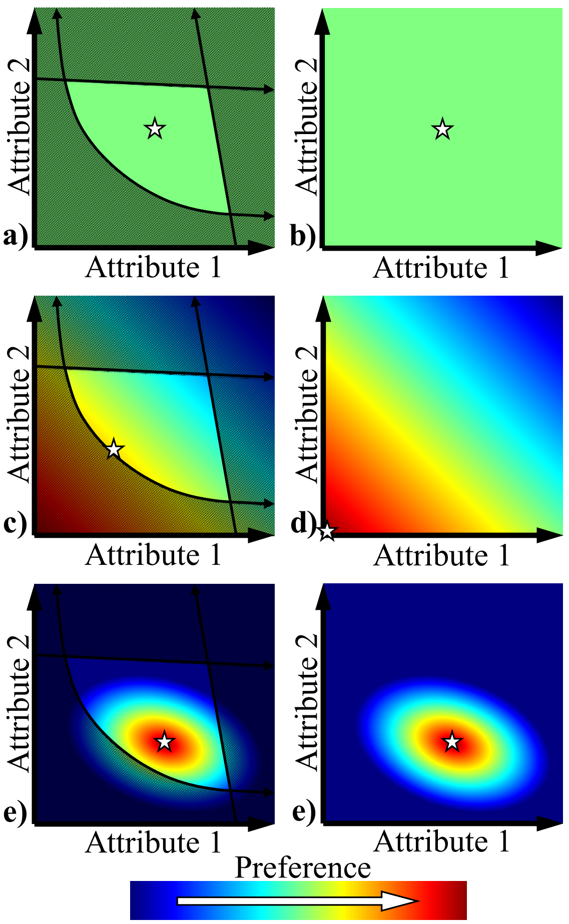


Figure 3. Design spaces showing ways to communicate design preference

While offering a good starting point for engineers, this collection of checklist items has been found to be a culprit for schedule delays and cost overruns. It is estimated that the Department of Defense lost $208 million per day in the 2012 fiscal year from program cancellations and delays [20]. Requirements may overstate specifications, change, and are often competing. In addition, the primary preference for the system is not communicated in the requirements and some intuition is needed for a design team to understand what the customer wants [21]. Figures 3c and 3d demonstrate a simple way to mathematically characterize this intuition. Trade studies can be used to help balance competing attributes and rank order feasible designs. This objective function “weights” the attributes to create a simple linear relationship. While it does present a single solution as optimal, it is only an optimal for a contrived evaluation metric, often leading to designs that are impossible, such as a design with no mass or no cost. Many of the objective functions formed in this manner are unit-less and do not have an orderly basis.

With multiple levels of integration on these complex systems, communicating preference by passing requirements quickly constrains the design space and makes it difficult to find a feasible solution. Adding an objective function helps with decision making but there is a more elegant solution to communicating the preference on the system. Value-Driven Design removes the subjectivity and enables the stakeholders to sidestep the need for many requirements by using a value function to drive the design process.

Value-Driven Design and Primary Preferences

Value-Driven Design (VDD) was started in the early 2000s as an enhancement to traditional design methods, such as Systems Engineering (SE), Multidisciplinary Design Optimization (MDO), and Multi-Objective Optimization (MOO) [5]. By incorporating ideas from Economic Theory in the beginning of the design process, VDD helps to characterize the operational need to be satisfied by the system. Systems operating in commercial markets will be driven by stakeholder desires for profit [22]. Systems operating in non-commercial markets will be driven by stakeholder desires for mission success [23]. The novel idea with VDD is to use an economic based objective function, called a value function, to communicate the operational desire directly to engineers instead of passively by requirements and with simple objective functions. Figures 3e and 3f show example design spaces when using an appropriate value function. Figure 4 shows the basic VDD process, as presented in Paul Collopy's *Value-Driven Design* [5]; only phrasing and

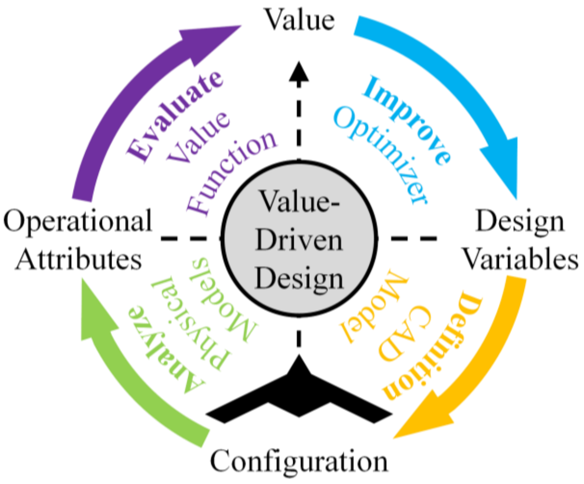


Figure 4. *Value-Driven Design process*

presentation and have been modified. The process is not new to design optimization but is shown for clarity on how the value function is a part of the loop.

Value functions allow for design trade-offs to be internalized in the optimization method. By placing more effort towards characterizing what is truly desired, better systems can be developed. Figure 3f is the ideal design space of VDD; no design constraints and one operationally optimal solution. Value functions, in essence, are more meaningful objective functions and can be used in existing optimization processes. They can also be used to rank-order feasible design alternatives in non-iterative design practices. A lot of effort in the VDD community has been tailored to creating value functions for aerospace systems [7, 22-27]. For commercial system design, the most favored value functions are economic profit, net-present-profit, and surplus value.

Monetary-Based Design

Profit and net-present-profit consider the design from a single company's operational perspective, typically the company doing the design work: "the seller". The difference between revenue and cost is maximized, either in one period with profit or all future periods with net-present-profit, to ensure the seller will stay in business. Revenue sources from this perspective include system sales and future maintenance procedures. Cost sources come from system development and production. The company, or organization, using the system, called "the buyer", is assumed to only affect the revenue stream of the seller, based on the quality of the system [28]. This scenario works best when the seller has a monopoly on the specific market but the buyer does not. When the buyer also has a monopoly on the market, called a bilateral monopoly, surplus value is used instead.

Surplus value is the more appropriate value function to use for most commercial LSCES. These types of systems, by nature, are a part of markets with large barriers to entry and tend to disallow competition [27]. This lack of competition invokes a greater dependency between the buyer and the seller which the concept of surplus value supports. Surplus value considers the collective profit of all primary stakeholders for the system. Cutting out their transactions and assuming they act as one entity, this is the difference between the buyers' revenue streams and the sellers' cost streams. For example, the surplus value of a commercial transport aircraft is the difference between passenger ticket sales over time and the cost to create the aircraft [26]. The surplus is divvied into the individual companies' profits well after conceptual design.

Operational-Based Design

Commercial systems can use the surplus value idea but non-commercial systems cannot. The buyers for these systems are various government entities and they do not have monetary revenue streams from operating these systems. Surplus value would only have the cost of the buyer to consider in this application. Maximizing this value would lead to a system that costs nothing and therefore does not exist.

To circumvent this relationship flaw, cost-plus contracts and fixed cost contracts have been used [29]. The buyer presents their preferences for the system and then enters an agreement with the seller to fully reimburse all costs or sets a price to pay for each unit. Each contract type has its associated risks. The cost-plus contract has the advantage for the seller since they are guaranteed a revenue source as a stated small percentage above the costs to be reimbursed. The disadvantage though is that the buyer could be paying quite a bit more than expected should any schedule delays and demonstration setbacks arise. There is also an incentive to understate costs during design competitions when this contract type is used [30]. The fixed cost contract has the advantage for the buyer in that they know exactly what their cost will be. However, they may not be pleased with the outcome of the system if their willingness to pay is rather low. The next sections further explore ideas in designing for operational abilities.

CHAPTER 3

VALUE FUNCTIONS

This chapter presents different value functions that can be used to design a system with the Value-Driven Design (VDD) design philosophy. Some of the equations and ideas are new; some are shown as a review of past work from the VDD community. Chapter 2 introduced the concept of VDD and how economics can have a larger role in the design process but it must be emphasized how important it is to understand which viewpoint to take when using this process. VDD intends to do to traditional systems engineering (SE) what traditional systems engineering did to design and development. Systems engineering took a step back, so to speak, from single discipline design to coordinate the efforts of multiple disciplines and design teams. It has enabled a small group of individuals to guide the work of many and produce more integrated and complex systems in a well thought-out approach. SE starts and ends with engineers though. It does not allow for other stakeholders to be well integrated in the process. VDD does. VDD takes another step back and actively considers the stakeholders' interest in the system, why they want the system, what role it will fill in their operational needs. This includes the companies, the organizations, and the economics of designing a system. The value function is central to this idea of economic based design [31].

Designing for Monetary Preferences

Commercial systems exist to create a profit for their stakeholders. These systems help provide services but making those services profitable is the primary focus of the stakeholders. The stakeholders would not be able to stay in business if their services were not profitable and it would not be wise to pursue business practices that lose money. Monetary value functions capture this design preference for money. The system can be designed to maximize its profitably as a service or certainly minimize its cost (subject to performance specifications). These economic attributes are important to consider in the lifecycle of a LSCES as they help ensure both the buying and selling companies can stay in business and stay competitive.

Surplus Value (Collective Profit)

Many LSCESs are unique in that their sellers work closely with their buyers to fill custom needs and ensure both parties have a net gain in value. Surplus value works very well to aid in cooperative design of commercial systems. The surplus value, *SV*, is the collective profits of all primary stakeholders in the system, including the seller (designs, develops, produces, possibly maintains) and the buyer (operates, maintains) of the system. The benefit of using surplus value as the evaluation metric is that sub-contractors and other dependent corporations can be included. Equation (1) shows this relationship for an arbitrary number of stakeholders, n_s. The goal is to maximize the difference between revenue from operations and the costs of development and production, removing the middle transactions between the buyer and seller. This coalition is used for design purposes only though, as anti-trust laws would prevent the players in a bilateral monopoly from merging [32]. The individual stakeholder profits are then divvied up later through rationing, bargaining, etc. Past research has explored the use of surplus-related, monetary-based value functions to design systems such as aircraft and gas-turbine engines [22, 25, 26, 33-35].

$$SV = \sum_{i=1}^{n_s} \pi_i \qquad (1)$$

Individual Profit

In commercial markets that see competition, individual companies may wish to solely maintain their perspective when designing a system. The profit for a single company, $\pi_{i,j}$, is the difference between their revenues, $R_{i,j}$, and their costs, $C_{i,j}$, in a given time period, shown in Eq. (2). The index "*i*" is used to refer to the company and the index "*j*" is used to refer to the time period (e.g. a fiscal year) being considered.

$$\pi_{i,j} = R_{i,j} - C_{i,j} \tag{2}$$

Time can be incorporated into the profit accumulation as well by means of net-present-profit, NPP_i. Net-present-profit accounts for the current value of the anticipated future profits, with some manner of discounting, shown in Eq. (3). The discount value, *r*, is used to place emphasis on earlier profit flows and to account for economic inflation. Equation (4) shows a special case of Eq. (3) and one more tailored to LSCES. The profit in the initial period, $\pi_{i,0}$, (e.g. from the acquisition) is different from the profits in future periods, $\pi_{i,j}$, (e.g. from maintenance, overhauls, etc.) in this scenario. The variable *t* represents the number of future time periods to be accounted for.

$$NPP_i = \sum_{j=0}^{t} \frac{\pi_{i,j}}{(1+r)^j} \tag{3}$$

$$NPP_i = \pi_{i,0} + \frac{1}{r}\left(1-(1+r)^{-t}\right)\pi_{i,j} \tag{4}$$

<u>Individual Revenue</u>

The first part of the profit equation is the revenue. A business strategy typically involves minimizing costs and maximizing revenues. Revenue considers the number of products sold and the price they can be sold at. Maximizing this value is good for the business and ensures investment costs during the program life cycle can be covered. Sources of revenue include the initial sale of the items, as well as overhaul and maintenance services later in the lifecycle. Revenue-based value functions can directly relate customer preferences and desires of the system to their willingness to pay for such an item. Market demand models must be captured to identify these consumer desires though. Without such studies, the revenue models will be inaccurate, leading to sub-optimal designs.

These revenue models must also take into account the observation that LSCES do not typically come “off-the-shelf”. The systems are made in low quantities and only for a specific customer or two. With this in mind, it becomes risky to be proactive about making LSCES before buyers have committed. Instead, the seller will usually be under contract to complete the system, common forms being the fixed-cost contract and the cost-plus contract [29]. In the fixed-cost contract, the revenue is stated up front and the seller makes a profit from the surplus in their development and production costs. In the cost-plus contract, the source of revenue is stated to be an agreed upon percentage above the seller’s cost.

Individual Cost

Project cost is an important attribute to manage in the life cycle of a LSCES. One of the most produced jet-propelled fighter aircraft in existence, the F-16 Fighting Falcon, started in a design competition asking for a low-cost aircraft [36]. Among a few other requirements, the United States Air Force’s Lightweight Fighter competition called for a cheaper, easy to maintain aircraft. Today, the F-16 Fighting Falcon is one of the most produced jet-propelled fighter aircraft in existence, having over 4,500 units made. The program has been very successful with aircraft operating in 25 countries [37].

Minimizing cost is the primary goal of many supply chain managers. This goal is seen in the use of simple system designs, lean manufacturing practices, and by purchasing standardized systems to lower the per-unit cost (i.e. taking things "off-the-shelf"). If the costs of a program are too great or grow too much, companies may lose orders, lose investors, or have to recover the losses using other programs in the company. Many newer defense projects have had to decrease the number of orders because of cost growths [38].

The cost of developing, testing, evaluating and producing an aircraft program can be estimated by Eq. set (5). The cost, $C_{i,0}$ (in USD), comes from the perspective of the prime contractor and is decomposed into the airframe program cost [39] and the turbofan engine program cost [40]. Eq. (5) has been modified from the source material to match SI units and have a nomenclature consistent with this document.

$$\begin{gathered} C_{i,0} = C_{Airframe} + C_{Engine} \\ C_{Airframe} = 201 \cdot CPI_{Year} / CPI_{1977} \cdot W_{Emtpy}^{0.798} \cdot V_{Max}^{0.736} \cdot Q^{0.401} \\ C_{Engine} = C_{Engine,Dev} + n_{Eng} \cdot Q \cdot C_{Engine,Prod} \\ C_{Engine,Dev} = 1000 \cdot CPI_{Year} / CPI_{1980} \cdot \left(5.17 \cdot T_{SLS} + 401 \cdot M_{Max} + 70 \cdot n_{Eng} \cdot Q - 526000 \right) \\ C_{Engine,Prod} = 1000 \cdot CPI_{Year} / CPI_{1980} \cdot \left(0.00967 \cdot T_{SLS} + 243 \cdot M_{max} + 1.74 \cdot T_4 - 2230 \right) \end{gathered} \tag{5}$$

The airframe program cost, $C_{Airframe}$ (in USD), is directly related to the empty weight of the aircraft, W_{Empty} (in N), the maximum aircraft velocity, V_{Max} (in m/s), and the quantity of aircraft desired, Q. The engine program cost, C_{Engine} (in USD) is broken down into engine development costs, $C_{Engine,Dev}$ (in USD) and engine production costs, $C_{Engine,Prod}$ (in US$). If the engine is to be taken "off-the-shelf" and does not require development, $C_{Engine,Dev}$ may be ignored. The engine costs are dependent on the sea-level static thrust, T_{SLS} (in N), the maximum inlet Mach number, M_{Max}, and the turbine inlet static temperature, T_4 (in K). These costs are also directly related to the number of engines to be produced, presented here as the product of the number of engines on the aircraft, n_{Eng}, and, again, the quantity of aircraft desired, Q. The cost relations are affected by the year of acquisition as well, due to inflation affecting worker wages, material costs, etc. The inflation can be estimated over a time period with the Consumer Price Index, *CPI* [41].

Additional costs a company may consider in the life cycle of the system are the costs for support and maintenance. These costs are a result of system repairs, overhauls, basic upgrades, and training. For the analyses in this document, the assumption is made that the maintenance costs for the prime contractor, $C_{i,j,MT}$ are 5% of the initial investment cost each year the system is in service, shown in Eq. (6).

$$C_{i,j,MT} = 0.05 \cdot C_{i,0} \tag{6}$$

If the desire of the stakeholder of a for-profit company (i.e. the prime contractor, the seller, etc.) is to minimize cost then Eqs. (5) and (6) would be used as the value function in the early stages of design. This value function would be communicated down to the designers to guide their decision making process. With the attributes captured in the value function, the designers would look to drive the empty weight of the aircraft, the maximum velocity and the quantity produced to minima. Without requirements, this simple value function would result in an aircraft that weighs nothing and does not move; it does not exist. While this plane is unreasonable, it does accomplish the stakeholder's desire of minimizing cost. The cost value function illustrates the importance in properly understanding the desires of the stakeholder, as was illustrated previously in Figure 3. Figure 5 shows the design space when using the minimize cost preference. This figure looks at the airframe cost in a program of 500 aircraft in the 2014 fiscal year (FY). Figure 5 shows how the cost value function drives the designers to an unreasonable airplane configuration.

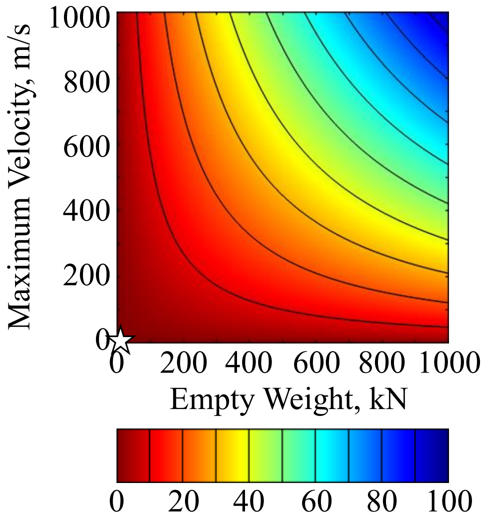


Figure 5. *Design space of airframe cost for a program consisting of 500 aircraft, billion USD in FY 2014*

In these value functions it is important to understand that attributes are typically dependent on one another. Modifying attributes directly can lead to impossible designs. For example, in the design space of Figure 5, it is possible to have a plane of zero empty weight and a high velocity. This is an unrealistic design for an aircraft, resembling a photon more than a Phantom [42], but the simple design space still allows for it. Even if the airframe were negligible, the power plant to produce such a high velocity would cause mass to be present. Value functions help communicate design presence but they require a robust analysis block to help avoid non-physical solutions.

Monetary Value Functions with Requirements

A company operating on its own accord will want to maximize its own profit, or perhaps minimize costs, but may be required to satisfy contractual obligations. The contractual obligations come in the form of system requirements, specifications, and regulations. For example, a new aircraft program could include specifications on the gross take-off weight not exceeding 100,000 pounds and the operating range be 2,000 nautical miles. Less tangible ideas can be included as well, such as the system must be aesthetically pleasing or the system must be user friendly. While adding requirements reduces the allowable design space, and possibly restricts access to the true optimum (recall Figure 3), it is an occurrence that must be accounted for. Equation (7) shows an example of this, presented in the form of an optimization problem for the designer.

$$\begin{aligned} &\max(NPP) \\ &s.t.\ W_{GTO} \leq 100{,}000\, lb_f \\ &\quad Range = 2{,}000\, nmi \end{aligned} \tag{7}$$

Designing for Operational Preferences

The stakeholders of LSCES associated with government entities and nonprofit organizations, will, by definition, not have desires centered on profit. These types of stakeholders are more focused on operational needs instead of economic ones, most likely because a market does not exist to support a revenue stream for the desired system. LSCES in this realm deal with desires concerning research expansion, technology demonstration, and national defense, among others. This section explores different operational-based value functions that may be used to communicate design preference for LSCES in operational roles.

Probability of Operational Success

Several value functions already exist for designing non-commercial systems. The probability of operational success is perhaps the one most attuned to VDD and is derived and shown in Eq. set (8) [7, 23]. The probability of operational success, $p(OS)$, accounts for the number of systems used in the operation, n, and the probability each system will be successful, $p(OS_i)$. The operation fails, $p(OF)$, only if every individual system's operation fails, $p(OF_i)$. We can reasonably assume that the failure of each system's operation is independent of any failures of the other systems. The index "*i*" has no relation to the index "*i*" used for the monetary value functions. Here, it refers to an individual system and does not have a specific relation to any stakeholder. Figure 6 shows this design space for up to 10 systems used in an operation.

$$
\begin{aligned}
p(OF) &= p(OF_1 \cap OF_2 \cap OF_3 \cap \ldots \cap OF_n) \\
p(OF) &= p(OF_1) \cdot p(OF_2 \mid OF_1) \cdot p(OF_3 \mid (OF_2 \cap OF_1) \cdot \ldots \\
&\qquad \cdot p(OF_n \mid (OF_{n-1} \cap \ldots \cap OF_3 \cap OF_2 \cap OF_1)) \\
p(OF) &= p(OF_1) \cdot p(OF_2) \cdot p(OF_3) \cdot \ldots \cdot p(OF_n) \\
p(OF) &= (p(OF_i))^n \\
1 - p(OS) &= (1 - p(OS_i))^n \\
p(OS) &= 1 - (1 - p(OS_i))^n
\end{aligned}
\tag{8}
$$

NASA has had several examples of designing systems for operational success, including the Viking program [43] and the Mars Exploration Rover Mission [44]. In both programs, two rovers were sent to Mars to increase the chance at least one made it to the surface safely and able to work. This value function works well when both the buyer and seller are operationally focused and can disregard costs, profits, or otherwise. Their primary preferences are aligned towards the success of the mission to be performed by the systems. Interdiction missions, evasive maneuvers, and communication networks also have examples of this operational

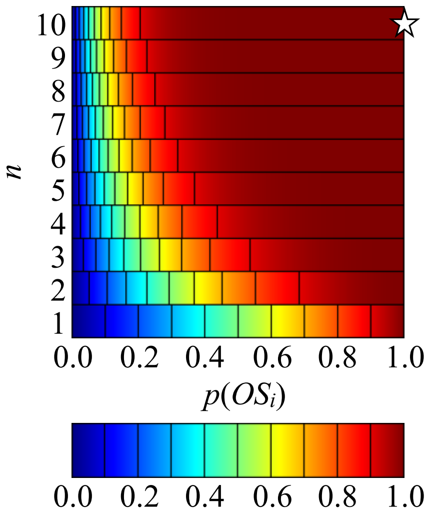


Figure 6. *Design space of the probability of operational success for a group of systems*

focus. Using more systems to accomplish a task can reduce the overall risk. This is performed when the systems have a reduced cost per unit.

Mathematically, the probability an individual system will complete its operation is the intersection of the survivability, *p*(*S*), and the effectiveness of the system, *p*(*E*|*S*), shown in Eq. (9). The design space for this perspective is shown in Figure 7.

$$p\left(OS_i\right) = p\left(S \cap E\right) = p\left(S\right) \cdot p\left(E \mid S\right) \qquad (9)$$

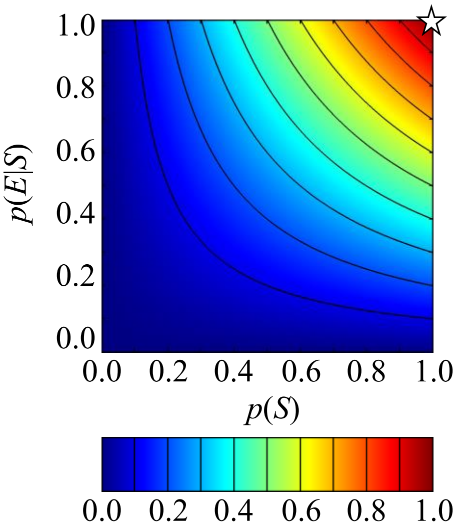


Figure 7. *Design space of the probability of operational success for an individual system*

Probability of Survivability

The probability of survivability is a system attribute that takes into consideration the hazardous environment the system will operate in (such as an active warzone) and the countermeasures needed (e.g. stealth technologies and control system redundancies) to ensure the system will continue to operate as intended. The probability of survivability, *p*(*S*), is the probability the aircraft will not fail (lost, shot down, etc.) when in operation and is shown in Eq. (10). The probability it will fail, *p*(*K*), is the product of the susceptibility, *p*(*H*), and the vulnerability, *p*(*K*|*H*), of the system [45], seen in Eq. (11). This notion can easily apply to other systems, such as spacecraft or watercraft, meant to operate in precarious environments as well.

$$p\left(S\right) = 1 - p\left(K\right) \qquad (10)$$

$$p\left(K\right) = p\left(H \cap K\right) = p\left(H\right) \cdot p\left(K \mid H\right) \qquad (11)$$

The probability of susceptibility is the chance the system will be hit. This attribute can be improved on a military aircraft by reducing the radar cross section, the infrared radiation (from engine exhaust), and the acoustic signature. The aircraft can have a carefully shaped airframe, be specially coated in radar absorbing materials, and have a subsonic flight plan to reduce the susceptibility. Should the system still be hit by an object (e.g. flax, space debris, etc.), the probability of vulnerability is the chance the system will fail if it is hit. Vulnerability can be reduced by having redundancies in system configurations, such as multiple flight control systems or multiple engines, and by spreading critical components throughout the plane [21]. Both components of survivability are important for building a successful system.

Probability of Effectiveness

The probability of effectiveness, $p(E|S)$, is defined as the chance the system can accomplish the operation (given that it can already survive it). For aircraft in military campaigns, the effectiveness relates to the payload size and how much can be carried to complete such operations as interdiction and support missions. For interplanetary spaceflights this ensures the cargo will stay intact from launching until landing between planets.

Probability of Campaign Success

If the systems are meant to operate in more than a single operation, such as in a campaign where there will be multiple operations and missions, the probability of campaign success may be a desire of the stakeholder. The probability of campaign success, $p(CS)$, is the probability all operations are successful. The probability of campaign success is the probability of operational success raised to the m^{th} power, where m is the number of operations to complete in the campaign, shown in Eq. (12). This mathematical form assumes the success of each operation is independent of the other operations.

$$p(CS)=\left(p(OS)\right)^{m} \tag{12}$$

Probability of Lifetime Success

For many systems, such as aircraft carriers and military aircraft, the stakeholders desire the system to be capable of performing multiple campaigns. This leads to a value function that captures the lifetime success of the system. If the systems are intended to have long lifetimes and operate in more than a single campaign then the probability of lifetime success, *p*(*LS*), can be used as the system's value, as shown in Eq. (13). Probability of lifetime success is calculated in similar manner as probability of campaign success.

$$p(LS)=\left(p(CS)\right)^{t} \tag{13}$$

Operational Cost

Stakeholders for operation based systems are still limited by the amount of money they have to spend. While their desires revolve around operations, the costs from performing the operations cannot be overlooked. While the probability of operational success and the functions derived from it stay focused on the buyer (the operator) of the system, variations on this metric do allow for costs to be included.

When cost must be accounted for as well, the cost-per-operation metric has been used, presented in Eq. (14). The cost-per-operation metric, *CPO*, is the ratio of the total cost to the buyer, C_{Buyer}, and the expected number of operations over the lifetime, *s*, of a system. The number of operations flown over the lifetime is estimated by using probabilistic expectations and time value discounting to determine an expected number of operations [46], shown in Eq. (15). Equation (15) introduces the variable *y*, the mean number of years between campaigns.

$$CPO = \frac{C_{Buyer}}{s} \tag{14}$$

$$s = \frac{p(S)}{1 - p(S)} \left(\frac{1 - (p(S))^m}{1 - (p(S))^m (1 - r)^y} \right) \tag{15}$$

A variation on cost-per-operation is the cost-per-kill [47], or more generally, the cost-per-success metric, CPS_i. This metric is the ratio of the cost of the attempt, *CPO*, and the probability the attempt will succeed, $p(OS_i)$, shown in Eq. (16).

$$CPS_i = \frac{CPO}{p(OS_i)} \tag{16}$$

Cost per success can be modified from the individual system attempt to a group perspective, accounting for the optimal number of attempts as well [23]. This can be accomplished with either multiple systems attempting a single time or with a single system attempting multiple times. The group cost-per-success metric, *CPS*, is then the ratio of the cost of a single attempt and the negative natural log of the probability a single attempt will fail, shown in Eq. (17).

$$CPS = \frac{CPO}{-\ln(1 - p(OS_i))} \tag{17}$$

Overall Success

Stakeholders may not have narrow desires such as operational success or campaign success. Many world and military leaders have a desire that can be categorized as overall success. Overall success captures the notion that a conflict is not resolved through excessive military force, as the previously discussed probability of operational success would push towards. Overall success takes into account such ideas as minimal collateral damage, minimal loss of life (on both sides), international politics, and appropriate usage of force to manage a conflict. From a weapon system design standpoint, overall success becomes a balancing act between what is effective and what is appropriate. Weapons that are highly effective could stop a conflict quickly but will bring a large amount of collateral damage as well. This collateral damage may be more than just the immediate physical damage, including global economic repercussions and the initiation of further conflicts and standoffs. On the contrary, weapons that are not effective enough will delay progress in the conflict intervention.

For the conflict itself, there are two primary perspectives to consider when determining the overall success; that of the intervener and that of the target. This goes beyond the buyer-seller relationship of the system, as discussed in the previous sections. The focus is now on the buyer's ability to conduct a successful campaign in the targeted area; the designed system being only a part of that vision. The military intervention must be satisfactory from both perspectives to be considered an overall success. This starts with the intervener's operational successes and discriminatory usage of force and extends to ensuring the violence does not continue or escalate. Thus, overall success, as conflict management, becomes more than just a series of checklists [48] and must be evaluated with a critical eye in policy and politics. A value function of overall success could be captured by measuring the economic stability of the region in need of attention.

CHAPTER 4

COOPERATIVE DESIGN WITH VDD

A relationship between the buyer and the seller of non-commercial LSCES is still needed. As discussed in Chapter 3, value functions are focused on designing either for operational preferences or for monetary preferences. If the seller of the system is publicly or privately held, their goals do not directly translate to the buyer's goals of mission success. A non-government entity selling mission-based systems still has a primary preference for money. Design can be done by using one of the mission-focused value functions and a specific contract type, but an optimal design is not guaranteed, at least not one that is optimal from both perspectives.

The goal of VDD is shifting towards efficiently producing optimal designs from both the buyer's and the seller's perspective. For mission-based LSCES, having a government as the single buyer from a non-government seller quickly becomes a government sponsored monopoly. The aerospace industry has been reduced to a collection of monopolies because of this [27]. From the seller's perspective, this may be more desired than designing systems to maximize profits but the focus of this paper is on the interactions between entities in existing monopolistic markets.

The key to effectively using VDD to design a system comes from understanding the design perspective and the relative power each player has to influence the outcome. Table 1 summarizes the best value function types for the three potential scenarios for LSCESs. These types of systems are analyzed most notably because of the unique markets they create. The design is easier when the perspectives align. When they do not, economic bargaining can be used to balance the two players' desires. This chapter further discusses design perspectives and value functions to gain a better understanding of the power they have when creating systems.

Table 1. *Value Function Summary*

Buyer	Seller	Value Function
Commercial	Commercial	Surplus Value
Government	Government	Probability of Operational Success
Government	Commercial	Bargaining

Economic Bargaining

Embedded in the models of game theory and the ideas of cooperative/competitive decision making are models related to bargaining [49, 50]. In these games, players take turns offering how to divide a resource between each other. The receiving player can either accept the offer (Y) or reject it (N) and propose a counteroffer. Figure 8 shows the basic bargaining model for two players, in extensive form. In this example, the game starts with player 1 making an offer, (x_1, x_2), to which player 2 refuses and makes the counteroffer (y_1, y_2). Each player has a time discount-factor, δ_i, to represent their respective impatience towards repeating the game; 0 represents a completely impatient player, 1 represents a very patient player. The game continues until a player accepts a proposal. Several variations of the game exist but this thesis starts with considering two-player games with infinite horizons, meaning there is not a limit on the number of rounds that can be played. The players have no sense of altruism either and are playing strictly for themselves.

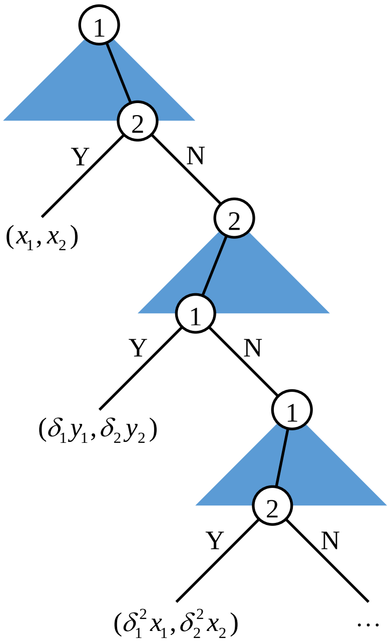

1
2
Y
N
(x_1, x_2)
2
1
Y
N
$(\delta_1 y_1, \delta_2 y_2)$
1
2
Y
N
$(\delta_1^2 x_1, \delta_2^2 x_2)$
…

Figure 8. *Bargain modeling, example extensive form*

The resource being divided has a unit-less size of 1 here. A rational person may think the best combination is to split the resource equally in half, but due to time pressures the split-line skews in favor of the more patient player. The subgame perfect equilibria for this game is shown in Eq. (18) [49]. Player 1 will offer the outcome x^* if he leads and player 2 will offer the outcome y^* if he leads. These combinations give the proposing player the highest payoff while making the receiving player indifferent towards accepting and rejecting the offer. The equilibria solutions are the preferred outcomes in bargaining games as they assure an outcome will quickly be reached and each player has gained something. The equilibria conditions are shown in Eq. (19) [49].

$$\begin{aligned} x^* &= \left(x_1^*, x_2^*\right) = \left(\frac{1-\delta_2}{1-\delta_1\delta_2}, \delta_2\frac{1-\delta_1}{1-\delta_1\delta_2}\right) \\ y^* &= \left(y_1^*, y_2^*\right) = \left(\delta_1\frac{1-\delta_2}{1-\delta_1\delta_2}, \frac{1-\delta_1}{1-\delta_1\delta_2}\right) \end{aligned} \tag{18}$$

$$\begin{aligned} x_2^* &= \delta_2 y_2^* \\ \delta_1 x_1^* &= y_1^* \end{aligned} \tag{19}$$

Bargaining can be used to gather further insight into the buyer-seller relationship of a LSCES. The buyer of a non-commercial LSCES will have a primary preference for operational success and the seller of the system will have a primary preference for maximum profit. Setting aside any contract structures, bargain modeling can be used with VDD to characterize the system with the best compromise between the preferences of both players. They will be bargaining for the attribute split.

Several assumptions need to be stated for this application from theoretical economics to system design. The first assumption is that both players have complete information; both are aware of each other's existence and knowledge set. Further, it is assumed the design space is fully deterministic, at least at the conceptual design phase. The game being played will change if both players are unsure of each other.

Bargaining for Player Profit

First, we consider how bargaining is applied in the traditional sense of players competing for the biggest portion of a resource. This example works intuitively for commercial LSCES. As stated before with LSCES systems being used for commercial applications, the most favored value function to use for evaluation is the idea of surplus value. Surplus value is used to find the best system for the *set* of stakeholders, not any one stakeholder in particular. The buyer will want to maximize his profit – at the probable expense of the seller. The seller will want to maximize his profit – at the probable expense of the buyer. To circumvent this dilemma, surplus value is used to maximize the collective profit of both stakeholders (the terms “stakeholder” and “player” are used interchangeably here). However, each stakeholder is still left to wonder what their individual gain will be from playing a part in this system development and operation. After the optimal system has been determined (via the surplus value metric), the transaction between the players will still have to take place. They will bargain for the price of the system to attempt to maximize their own profit.

Let the buyer be player 1 and the seller be player 2 in this example. They both want to maximize their profit from the available surplus. If given the chance, the buyer will set the price of the system at P_1 and the seller will set the price at P_2. Both price estimates are functions of the system attribute set, A, to have a logical basis for each player’s asking price, shown in Eq. (20).

$$P_1 = P_1(A) \qquad (20)$$
$$P_2 = P_2(A)$$

The buyer's valuation of the system, *V,* is also based on the system attribute set. For a commercial system, this would represent the revenue streams from operating the system (e.g. passenger ticket sales on a transport aircraft, energy sales from a power plant, etc.). The cost of development and production to the seller, *C*, is also a function of the system attribute set, shown in Eq. (21).

$$V = V(A) \qquad (21)$$
$$C = C(A)$$

The surplus value, *SV*, is the difference between the system value to the buyer and the system cost to the seller. Eq. (22) shows this relationship. This is the value the players will bargain over and decide how it will be split up. They will each, in turn, make an offer on how to divide this resource between each other by bargaining for the price.

$$SV = V - C \qquad (22)$$

To expedite the negotiations, a player can offer an equilibria payoff. The payoff equilibria are shown in Eq. (23). If the buyer leads in proposing a system design, his payoff, in equilibrium, will be the difference between his value of the system and the asking price. The seller's payoff will be the difference from the buyer's asking price and the costs of development. These are payoffs, the benefit each player will receive from accepting the offer. The buyer must pay for the system but will receive his perception of its value. The seller must pay for the cost of producing the system but receives revenue to produce a profit.

If the seller leads in proposing the system design, he receives, in equilibrium, the difference from the asking price and the cost of production. The buyer gains a surplus from the difference of his valuation and the asking price of the seller. These strategies are a set of strategies in a much larger set and are used in this section to describe the bargaining of surplus value through price.

$$\begin{aligned} x^* &= \left(x_1^*, x_2^*\right) = \left(V - P_1, P_1 - C\right) \\ y^* &= \left(y_1^*, y_2^*\right) = \left(V - P_2, P_2 - C\right) \end{aligned} \tag{23}$$

The two prices can be solved for using Eqs. (19) and (23), and are shown in Eq. (24). The price each player will try to set is a function of the impatience factors, δ_1 and δ_2, the buyer's value of the system, *V*, and the seller's cost of the system, *C*.

$$\begin{aligned} P_1 &= \frac{\delta_2\left(1-\delta_1\right)V + \left(1-\delta_2\right)C}{1-\delta_1\delta_2} \\ P_2 &= \frac{\left(1-\delta_1\right)V + \delta_1\left(1-\delta_2\right)C}{1-\delta_1\delta_2} \end{aligned} \tag{24}$$

The equilibrium payoffs are shown in Eq. (25), where the prices from Eq. (24) have been substituted into Eq. (23). The payoffs are no longer a function of the asking prices but instead, the characteristics of the players. These two equation sets show the relative bargaining power each player has. A special case is shown in Eq. (26) where the players have the same patience factor, δ. In this special case, the surplus is split almost evenly except that the player receiving the offer has the added impatience factor.

$$\begin{aligned} x^* &= \left(x_1^*, x_2^*\right) = \left(\frac{1-\delta_2}{1-\delta_1\delta_2} SV, \delta_2 \frac{1-\delta_1}{1-\delta_1\delta_2} SV\right) \\ y^* &= \left(y_1^*, y_2^*\right) = \left(\delta_1 \frac{1-\delta_2}{1-\delta_1\delta_2} SV, \frac{1-\delta_1}{1-\delta_1\delta_2} SV\right) \end{aligned} \tag{25}$$

$$x^* = \left(\frac{1}{1+\delta}SV, \frac{\delta}{1+\delta}SV\right)$$
$$y^* = \left(\frac{\delta}{1+\delta}SV, \frac{1}{1+\delta}SV\right) \tag{26}$$

The payoffs are shown more comprehensively in Figure 9. Figures 9a and 9b show the buyer's and seller's payoffs, respectively, from the buyer's equilibrium offer, x^*. Figures 9c and 9d show the buyer's and seller's payoffs, respectively, from the seller's equilibrium offer, y^*. For both players, it is more rewarding to be patient. The offering player will take advantage of the receiving player's desire to end the game quickly and reward themselves with a higher payoff.

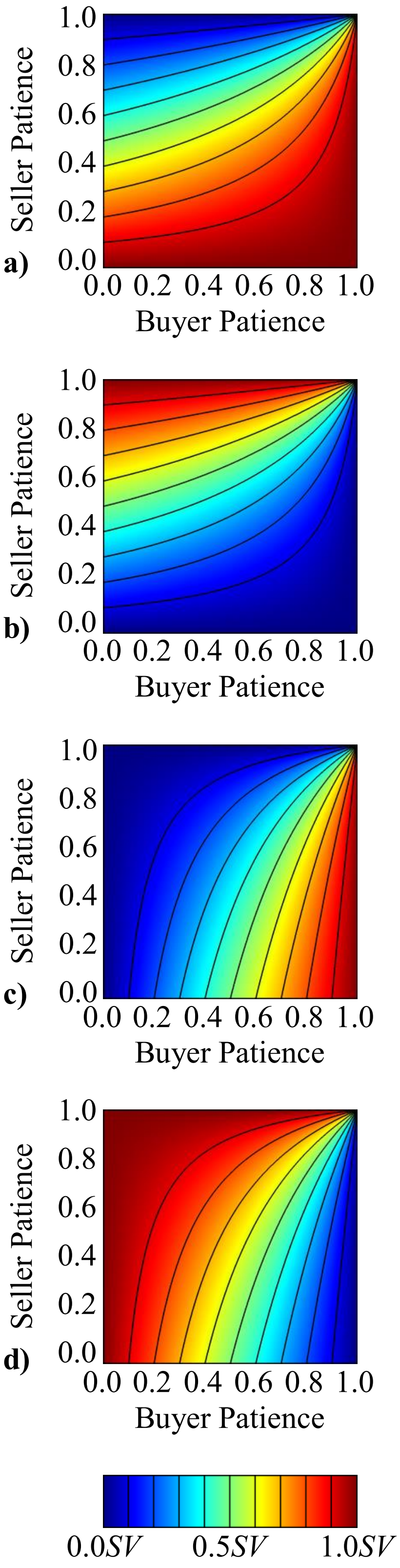


Figure 9. *Payoffs as they relate to player impatience*

Bargaining for Attributes

Bargaining over how to split the surplus value of a system works well for commercial LSCES because both players have monetary preferences. When the players' preferences do not have the same basis though, other variations of the bargaining model can be used. For instance, the two players can directly bargain over the attribute set to use for the system. This final attribute set to use for the design, *EA*, shown in Eq. (27), would start as the lottery between both players' favored attribute sets, A_1 and

A₂, respectively. The buyer favors A_1 since it represents the optimal system for his desire of mission-success (i.e. probability of operational success). The seller favors A_2 since it represents the most profitable system design.

The influence factor, q, would then be solved for from the equilibria payoffs (Eq. (28)) and the equilibria conditions (Eq. (19)) to determine the final attribute set to use for the system design. Equation (29) presents the final equations to solve for. They are non-linear but do allow for an interpolation between the two points in the design space, A_1 and A_2. This final, interpolated point will have benefits for both players even though they are not after the same goals.

$$EA = qA_1 + (1-q)A_2 \tag{27}$$

$$\begin{aligned} x^* &= \left(x_1^*, x_2^*\right) = \left(V(EA) - V(A_0), P(A_1) - C(A_1)\right) \\ y^* &= \left(y_1^*, y_2^*\right) = \left(V(A_2) - V(A_0), P(EA) - C(A_2)\right) \end{aligned} \tag{28}$$

$$\begin{aligned} \delta_1\left(V(EA) - V(A_0)\right) &= P(A_1) - C(A_1) \\ V(A_2) - V(A_0) &= \delta_2\left(P(EA) - C(A_2)\right) \end{aligned} \tag{29}$$

Bargaining for System Price with only Seller Offers

If the system attribute set is not available for bargaining, the players can still bargain over the price of the system. The seller can still attempt to maximize his profit for a given system by bargaining for the price to sell it for. This would move the bargaining into the analysis block of the design process as part of the operational and business analyses. After each design iteration, the seller will try to sell the system configuration at the highest price. The buyer can either accept or decline this system at the selling price. If the price is too high and the system performance too low, he can reject the system and tell the seller to try again.

The design loop will continue to iterate until something can be agreed upon. The seller is trying to maximize his profit here but, with bargaining, has a better way to estimate the price the buyer is willing to pay. Bargaining can be used to enhance the VDD process. VDD already accounts for designing a system for its primary preference - the only addition needed is to consider the interaction between the buyer and seller. Bargaining is a repeated game in competition, but it is through the repetition that a forced cooperation exists. This makes bargaining an ideal method to balance competing preferences and still find an effective solution. The bargaining interactions are the greatest benefit to creating a LSCES as it ensures the best system is made and all stakeholders are happy. Figure 10 shows the VDD process again and presents the addition of bargaining to the evaluation stage. The buyer is focused on the operational attributes of the system and the seller is focused on the value the system will produce for them. Bargaining helps balance their preferences.

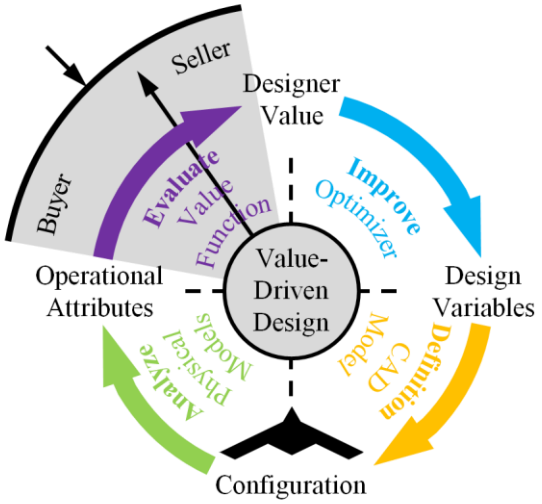


Figure 10. *Value-Driven Design with bargaining*

Role of Bargaining in Design

The knowledge of the bargaining process empowers the designer (seller) to be able to predict the final payoff after a negotiation period. In essence, the bargaining model becomes an additional step in the system analysis stage of the design process (and in the evaluation stage of the VDD cycle). A designer can produce a set of design variables, analyze the attributes associated with those design variables, and then mathematically determined the payoff they would receive from the bargaining game's equilibria. This analysis then enables the designer to perform an optimization using the payoff as the objective function. In this manner, the bargaining model improves the fidelity of the value function (the payoff) by incorporating the impatience of the buyer and seller and their often conflicting value preferences. The bargaining model is related to revenue models which have traditionally tried to predict the buyer's behavior through simple monetary functions [7, 28]. Bargaining models offer improved fidelity to the designer's value calculation which can be easily implemented in traditional system analyses and design optimization processes.

CHAPTER 5

APPLICATION, RESULTS

It is critical that the interactions and preferences of both the buyer and seller be taken into account during the design of a LSCES. This chapter compares the value-based, optimal designs for several of the perspectives shown in Chapter 3 and explores the cooperative ideas discussed in Chapter 4 for a notional strategic strike aircraft program. Through traditional VDD, the non-commercial system can be designed to satisfy either the monetary preferences of the seller or the operational preferences of the buyer. With the addition of bargaining though, a balance between these preferences sets can be achieved to allow for a system that is beneficial to both players.

In this case study, a government entity will buy and a for-profit, commercial entity will develop, produce, and sell the system. The company will go through the conceptual, preliminary, and detailed phases of design to accommodate their customer's acquisition process. At the end of each design phase, a review will take place to check progress and ensure the buyer is pleased with the proposed design and its developmental progress. The buyer is focused on acquiring a system with the highest probability of operational success by their standards (Eq. (8)), but the seller is still focused on a system with the highest anticipated net present profit (Eq. (4) and [7]). The design team can propose any design they like, but if the government entity decides the proposed system does not have enough value or is not in line with their needs, they can opt out of funding further development on the project.

The buyer and the seller each want a system that maximizes their own value, recalling the individualistic assumptions from bargaining theory. Each player is playing this "game" for themselves. The dichotomy in desires can be bridged with bargaining though and allow the designers to determine a system valuable to both parties. This system will not be the most profitable or the most operationally successful, but will effectively tailor both parties' desires for total market control. However, to be clear, this chapter is not a demonstration or advocation of war profiteering techniques; the seller presenting a system that is just beneficial enough to the group of buyers so they sign a check has unethical implications, especially when the systems are intended for use in warfare and with the ability to take lives. This work is solely demonstrating the optimums associated with competing design perspectives and how negotiation tactics can be a part of the engineering process. This chapter focuses on the first stage of the acquisition process: the conceptual design phase.

System Setup

Strategic strike aircraft systems are military aircraft with a narrow range of roles to fill, making them ideal candidates to start with for exploring the VDD of non-commercial systems. These systems have one buyer, one seller, and a single mission plan: cruise out, drop ordinance, cruise back. Strategic strike aircraft do not actively engage in air combat but may be outfitted with evasive mechanisms and special design features. Current aircraft falling into this category include the Rockwell (now Boeing) B-1 Lancer [51], the Northrop Grumman B-2 Spirit [52], and the Boeing B-52 Stratofortress [53]. These aircraft have been designed to carry large payloads and be reusable as interdiction platforms. Many of the current multirole aircraft can fill this role as well, such as the McDonnell Douglas (now Boeing) F-15E Strike Eagle [54], McDonnell Douglas (now Boeing) F/A-18 Hornet [55], Lockheed Martin (still Lockheed Martin) F-22 Raptor [56], and Lockheed Martin (still

Lockheed Martin) F-35 Lightning II [57]. The multirole systems tend to be smaller but, as a squadron, offer effective tactical strike capabilities. Missile systems and other unmanned aerial vehicles can fill this support role too, but are not included in the analysis.

An aircraft design model was developed to create a robust analysis block for use in the design optimization loop [21, 45, 58, 59]. 0 holds the MATLAB codes used to build the model and Fig. 11 shows this model in its design structure matrix form (DSM). The model includes sizing methods for the airframe aerodynamics, system propulsion, and performance specifications. Additionally, economic and operational analyses have been added to allow for the value-based optimization to take place by actively considering the true desires of each stakeholder, which are rooted in these categories. Coupling suspension has also been applied to the DSM to simplify the analysis from more involved methods but is still sufficiently coupled for this case study (the effects of coupling suspension have been investigated previously for satellite systems with favorable insights towards creating simpler, yet effective models [60]). Further, this analysis block has been sequenced so only feed-forward couplings exist.

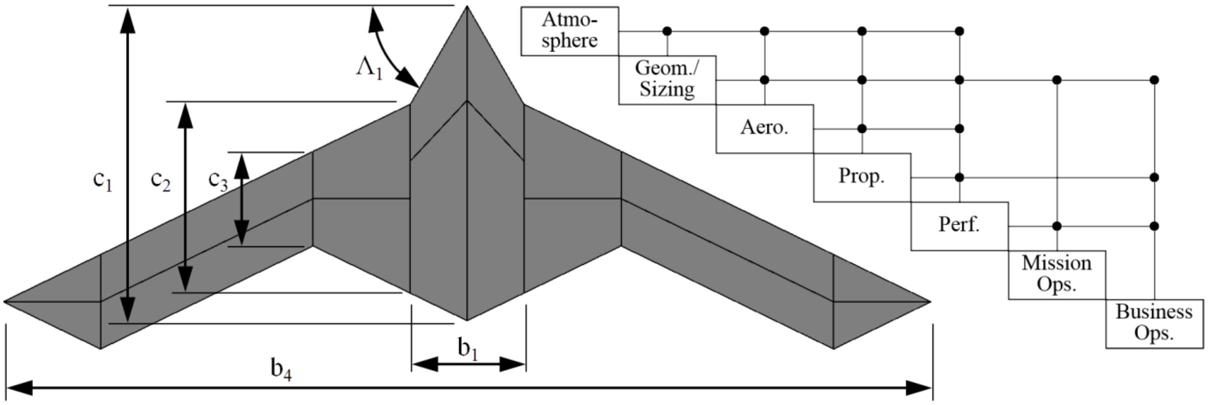


***Figure 11**. Aircraft design variables and design structure matrix*

The generic airframe is also shown in Figure 11 and takes significant inspiration from current stealth technologies and the mission critical aircraft previously mentioned. The airframe has 6 design variables to manipulate in the design sequence, three of which are chord lengths (c_1, c_2, and c_3), two are span lengths (b_1 and b_4), and the last is the leading edge sweep angle of the first wing-body section (Λ_1). The other dimensions are driven by body relations to keep the edges parallel, decreasing the susceptibility of the aircraft by minimizing the directions the reflected radar waves travel [45]. Together, these dimensions and features characterize an arbitrarily shaped blended-wing-body aircraft.

Table 2. *System attributes*

Attribute	Name
W_{GTO}	Gross take-off weight
$W_{Payload}$	Payload weight
Range	Aircraft operating range
M_{Cruise}	Cruising Mach number
C_{Unit}	Unit cost of aircraft
p(S)	Probability of survivability
p(E)	Probability of effectiveness

Table 2 shows the set of attributes used to characterize the aircraft. This attribute set is only able to describes the basic performance of the aircraft but is comprehensive enough to still interface well with both of the stakeholder desires emulated in their value functions. In the following sections, these attributes are displayed next to each of the optimal designs to be discussed; each optimal design being in regard to a different value of a stakeholder.

Seller's Design

The seller of the non-commercial system employs teams of engineers, scientists, analysts, manufacturers, technicians, managers, and more to support the various types of work throughout its lifecycle. The seller is fully aware of the cost of the system and is conscious about the fact that, as a business, they need to turn a profit from the work they do. Under the VDD philosophy, their primary desire is to design a system from a monetary standpoint.

Figure 12 shows the optimal system associated with each of the three different monetary desires the seller may have. Figure 12a shows the system satisfying the seller's potential desire to maximize their revenue from the aircraft program, Figure 12b shows the system satisfying the minimum cost preference, and Figure 12c shows the system that best satisfies the desire to maximize profit. The geometry is simple in each case and the airframes are sized mostly for cruising range. These three systems use the available cargo weight to carry more fuel instead of payload, inadvertently making any one system less effective as a strike weapon. However, the seller may be able to sell more systems in this way, as each system is still survivable, and then be able to reduce the per-unit production cost. Having a large fleet of aircraft to maintain in the coming years will be profitable and is already a common business practice. It displaces competitors' products and with the program continually being active, facilitates system upgrades at lower costs too.

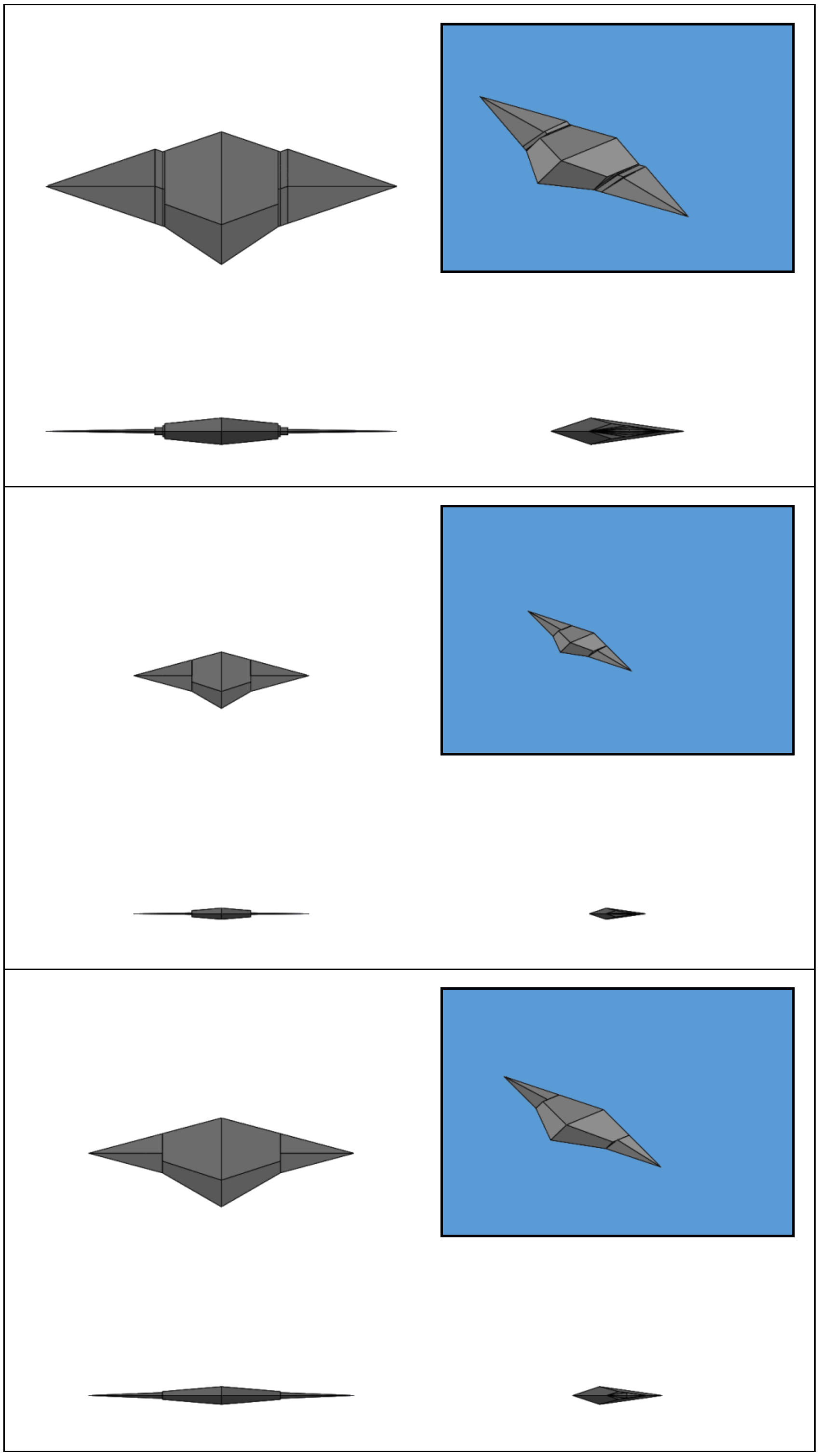
The minimum cost aircraft does not have the capacity to carry munitions in its "optimal" configuration. The maximum revenue aircraft is much bigger, but with its grandiose payload capacity, the seller will not be able to sell as many, increasing the unit cost. Instead, the seller will attempt to market the system that provides the maximum profit. This system falls somewhere between the extremes of the minimum cost and maximum revenue designs. It has a reasonable payload capacity, a long cruising range, and a modest unit cost. The seller anticipates that the (relatively) low price will be attractive to the buyer but the lack of payload capacity will still make it a hard sell as a strike platform.

Buyer’s Design

The buyer is focused on the operational attributes of the system. While cost is a consideration for them, their primary desires are towards the abilities of the system. Figure 13 shows the three optimal designs for the possible operational ideas the seller may desire; the most survivable, the most effective, or the most operationally successful system.

Figure 13a shows the most survivable system. This system has a payload capacity similar to the B-2 [52], but because it is smaller, does not have room for fuel. Without the fuel to make it fly and have an operating range, it will be very safe sitting in the hanger; very survivable behind closed doors. Figure 13b shows the most effective system. This system costs almost 3 times as much as the solely survivable system but does have room for fuel, can fly, and has a much larger payload capacity. These systems push the desires of the operational-based stakeholder to extremes but, like the cost and revenue value functions of the seller, demonstrate how important the perspective of the value function is.

The buyer wants a system with the greatest probability of operational success. Figure 13c shows the system that best satisfies this preference. This aircraft is faster and larger than the design the seller desires most. It also has a higher anticipated price tag due to the technology development costs. The seller anticipates this design will not be as profitable since the technology development will need to be internally funded.



System Attribute	Value
W_{GTO}, kN	500
$W_{Payload}$, kN	70
Range, km	9,000
M_{Cruise}	0.5
C_{Unit}, \$M	100
p(S)	0.93
p(E)	0.33

a) Maximize revenue

System Attribute	Value
W_{GTO}, kN	50
$W_{Payload}$, kN	0
Range, km	8,000
M_{Cruise}	0.5
C_{Unit}, \$M	25
p(S)	0.99
p(E)	0.00

b) Minimize cost

System Attribute	Value
W_{GTO}, kN	200
$W_{Payload}$, kN	20
Range, km	8,000
M_{Cruise}	0.5
C_{Unit}, \$M	50
p(S)	0.97
p(E)	0.13

c) Maximize profit

Figure 12. *Seller's best design (monetary preferences)*

System Attribute	Value
W_{GTO}, kN	400
$W_{Payload}$, kN	210
Range, km	0
M_{Cruise}	0.8
C_{Unit}, $M	300
p(S)	0.99
p(E)	0

a) Survivable

System Attribute	Value
W_{GTO}, kN	1,600
$W_{Payload}$, kN	800
Range, km	1,000
M_{Cruise}	0.95
C_{Unit}, $M	800
p(S)	0.86
p(E)	0.99

b) Effective

System Attribute	Value
W_{GTO}, kN	1,300
$W_{Payload}$, kN	700
Range, km	5,000
M_{Cruise}	0.9
C_{Unit}, $M	500
p(S)	0.92
p(E)	0.98

c) Operational Success

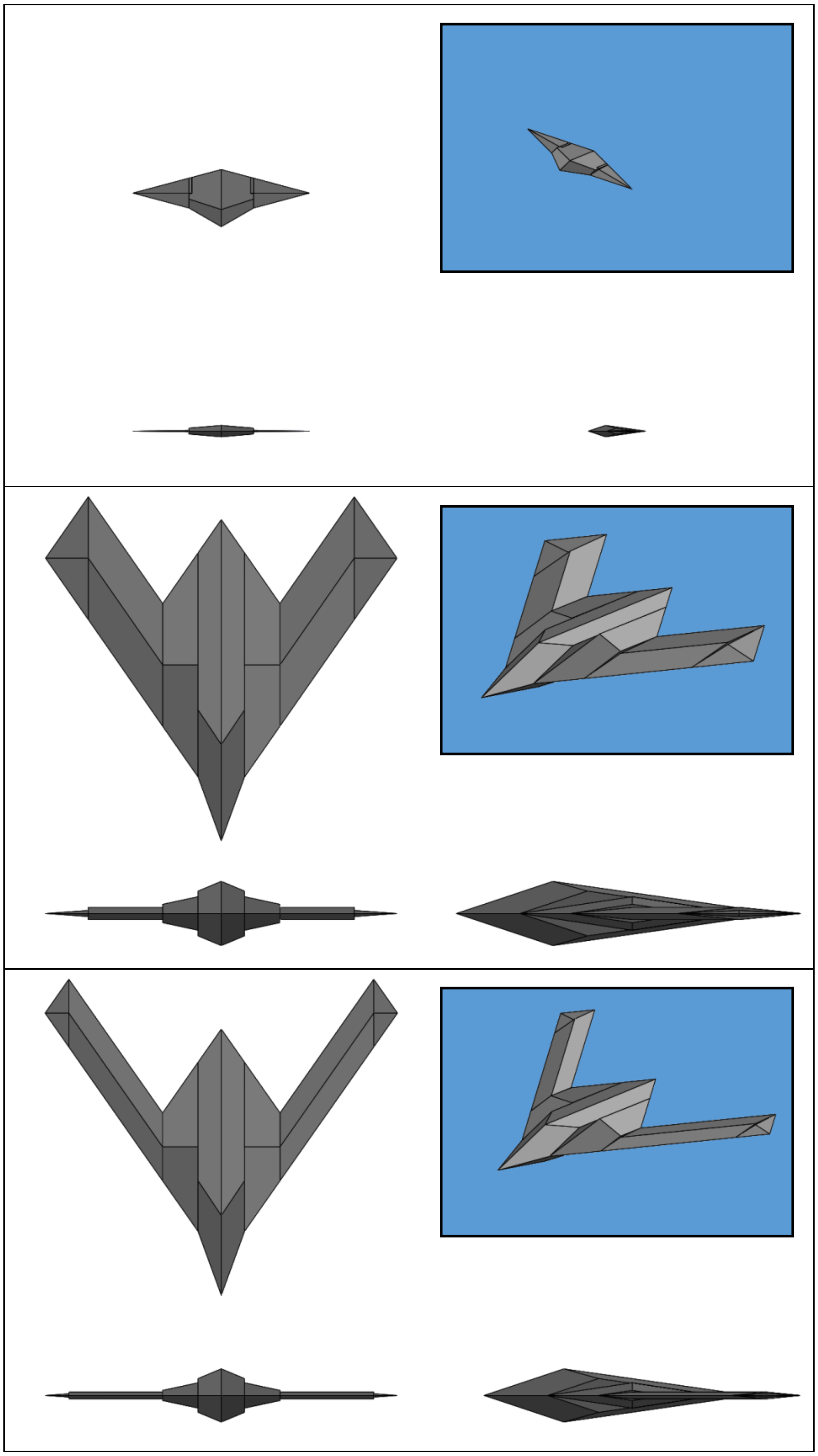


Figure 13. *Buyer's best design (operational preferences)*

Design through Bargaining

The buyer and seller are clearly at odds with each other when designing the system for their primary operational preference. The seller could offer to develop the most profitable system but the buyer would then dismiss the program since it is a system that does not meet their needs. The buyer could make an offer with their idea of the more advanced system, but the seller would not agree to it due to the high development costs and lack of profitability. These notions may seem extreme but, for this case study, are purposefully aligned with classical bargaining theory in which the players are "perfect" and are solely focused on themselves. Future work will use behavioral game theory to understand the extreme scenarios and better understand which player has more power in the game.

A balance must be reached to find the value-based system to move beyond the conceptual design phase and on to preliminary design. This is where bargain modeling can help the design process. Bargain modeling can re-characterize what the best system is by mathematically representing the buyer's and seller's preferences and creating an environment where their trade-offs are based on bargaining factors such as player impatience and the first to offer. Bargaining captures the notion that the seller and buyer want the same thing, increased value, but the manner in which they define the value is typically different.

Figure 14 shows an example of a system that may result due to the bargaining process. It is expected that the agreed upon design will not be the designs represented in Figs. 12 and 13 as they could only result from having a dominant player in the game. A dominant player is someone with total market control and who would take advantage of an extremely impatient player (a player willing to take any offer, so long as the offer does not negatively impact them). In the absence of dominant players, the agreed upon system would be a "balance" between the buyer's preference for operational abilities and the seller's preference for money. Figure 14 is only used as an example of an anticipated system but, with the incorporation of behavioral game theory, future work will look at the effects of player impatience on the game. Figure 14's system displays characteristics from both of the player's preferred designs. The system is survivable, effective, and has a decent cruising range. The unit cost is high, but not prohibitive to the seller. Both players have some benefit when moving forward with this system.

System Attribute	Value
W_{GTO}, kN	1,000
$W_{Payload}$, kN	500
Range, km	6,000
M_{Cruise}	0.5
C_{Unit}, \$M	400
p(S)	0.91
p(E)	0.93

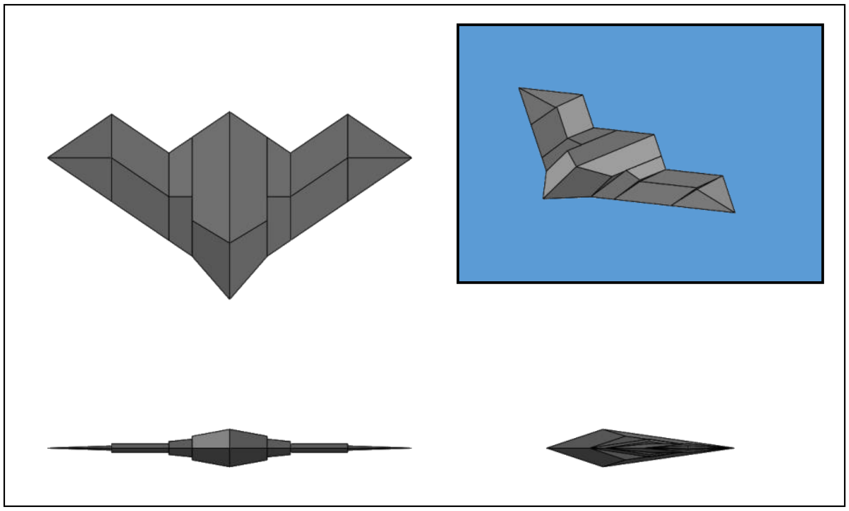


Figure 14. *Design through bargaining*

CHAPTER 6

CONCLUSION

Value-Driven Design stands apart from other system design practices because it enables the design optimization process in an efficient and foresighted manner. Instead of using high-level requirements to communicate preference and constrain the system design space, VDD uses a single value function to find the preferred solution. Value functions are economic-based objective functions that capture the desires of the primary stakeholders and their full need for the system. Value functions can take several perspectives though and it is important to choose the right perspective when committing to building a large scale system. With the cost of LSCESs being characteristically high and continuing to be under-budgeted, more robust and elegant design techniques, such as VDD, are required to build the integrated systems of the future.

Since LSCESs typically only have one buyer and one seller, a collaborative design effort is the most beneficial approach for each stakeholder to consider when defining their relationship between each other. The VDD community has already created value functions for commercial systems that represent this collaborative effort through the idea of surplus value. However, non-commercial systems have yet to be designed in a value-based approach that actively benefits both players. The buyer and seller for non-commercial systems do not have preferences that align, but through the ideas in economic bargaining, a single solution can be determined.

This thesis has shown how differing preferences between a buyer and a seller influence the outcome of the design optimization process and how the gap in desires can be addressed with bargaining techniques. In these scenarios for non-commercial system design, each stakeholder has a value function that fully captures their own desires. Since the buyer's and seller's preferences are not based in the same units (operational-based and monetary-based preferences, respectively), further direction is needed to enable a mutually beneficial optimization. The best solution is not the one satisfying only the buyer or only the seller. The best solution is the one that gives each stakeholder some benefit for participating in the program. Bargain modeling offers a logical way to balance the competing desires of the stakeholders when their values do not directly align.

Applying bargain modeling to Value-Driven Design can be further expanded from the work in this document. This thesis considered a scenario with one buyer, one seller, and a system with one purpose. Future work can look at the effects of competition between multiple sellers and even placing them in multi-round design games, similar to the current acquisition process but in a value-based context. The effects of buyer competition can also be studied wherein each buyer desires slightly different attributes on the design, leading to case studies in multi-role systems. This work can also be expanded to understand how sub-system development can be achieved while maintaining the high-level, system focus. Finally, the effects of repeating games can also be considered with further insight from a historical background to envision future market structures and anticipate the best business practices.

REFERENCES

1. MacCormack, A., C. Baldwin, and J. Rusnak, *Exploring the duality between product and organizational architectures: A test of the "mirroring" hypothesis.* Research Policy, 2012. **41**(8): p. 1309-1324.

2. Mintzberg, H., *Structure in 5's: A Synthesis of the Research on Organization Design.* Management Science, 1980. **26**(3): p. 322-341.

3. Kenway, G.K., G.J. Kennedy, and J.R. Martins, *Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations.* AIAA Journal, 2014. **52**(5): p. 935-951.

4. McCulley, C. and C.L. Bloebaum. *Optimal Sequencing for Complex Engineering Systems Using Genetic Algorithms*. in *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 1994.

5. Collopy, P.D. and P.M. Hollingsworth, *Value-Driven Design.* Journal of Aircraft, 2011. **48**(3): p. 749-759.

6. Buede, D.M., *The Engineering Design of Systems: Models and Methods*. 2nd ed. 2009, Hoboken, New Jersey: John Wiley & Sons, Inc.

7. Goetzke, E.D., C.L. Bloebaum, and B.L. Mesmer, *Profit and Operational-Based Value Functions*, in *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2014, American Institute of Aeronautics and Astronautics: Atlanta, GA.

8. NASA, *ESD Integration: Budget Availability Scenarios*. 2011: spacepolicyonline.com.

9. Forsberg, K. and H. Mooz, *The relationship of system engineering to the project cycle.* 1995.

10. *F-35 Lightning II*. 2014 [cited 2014 November 2]; Available from: http://www.lockheedmartin.com/us/products/f35.html.

11. Miller, S.W., et al., *Toward a value-driven design approach for complex engineered systems using trade space exploration tools*, in *American Society of Mechanical Engineers International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. 2014: Buffalo, New York.

12. Stump, G., et al., *Visual steering commands for trade space exploration: User-guided sampling with example.* Journal of Computing and Information Science in Engineering, 2009. **9**(4): p. 044501.

13. Sobieszczanski-Sobieski, J. and R.T. Haftka, *Multidisciplinary aerospace design optimization: survey of recent developments.* Structural optimization, 1997. **14**(1): p. 1-23.

14. Martins, J.R. and A.B. Lambe, *Multidisciplinary Design Optimization: A Survey of Architectures.* AIAA journal, 2013. **51**(9): p. 2049-2075.

15. Marler, R.T. and J.S. Arora, *Survey of multi-objective optimization methods for engineering.* Structural and Multidisciplinary Optimization, 2004. **26**(6): p. 369-395.

16. Browning, T.R., *Applying the design structure matrix to system decomposition and integration problems: a review and new directions.* Engineering Management, IEEE Transactions on, 2001. **48**(3): p. 292-306.

17. Kenway, G.K. and J.R. Martins, *Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration.* Journal of Aircraft, 2014: p. 1-17.

18. Bowcutt, K.G., et al., *Advancements in Multidisciplinary Design Optimization Applied to Hypersonic Vehicles to Achieve Closure.* American Institute of Aeronautics and Astronautics AIAA-2008-2591, 2008.

19. Leifsson, L., et al., *Multidisciplinary design optimization of blended-wing-body transport aircraft with distributed propulsion.* Aerospace Science and Technology, 2013. **25**(1): p. 16-28.

20. Maddox, I., P.D. Collopy, and P.A. Farrington, *Value-Based Assessment of DoD Acquisitions Programs.* Procedia Computer Science, 2013. **16**: p. 1161-1169.

21. Nicolai, L.M. and G.E. Carichner, *Fundamentals of Aircraft and Airship Design*. AIAA Education Series, ed. J.A. Schetz. Vol. 1. 2010: American Institute of Aeronautics and Astronautics.

22. Collopy, P.D., *Surplus Value in Propulsion System Design Optimization.* AIAA Paper, 1997. **97**(3159).

23. Collopy, P.D., *Value of the Probability of Success.* AIAA Paper, 2008. **7876**: p. 2008.

24. Hollingsworth, P. and D. Patel, *Development of a Surplus Value Parameter for Use in Initial Aircraft Conceptual Design.* Journal of Aerospace Operations, 2012. **1**(4): p. 343-358.

25. Mullan, C., et al., *An Analytical Study of Surplus Value using a Value Driven Design Methodology*. Aviation, Technology Integration and Operations (ATIO). Virginia, USA: American Institute of Aeronautics and Astronautics, 2011.

26. Sutcliffe, P. and P. Hollingsworth. *Development of an Airline Revenue Capability Model for Aircraft Design*. in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. 2010.

27. Collopy, P.D., *Military Technology Pull and the Structure of the Commercial Aircraft Industry*. Journal of Aircraft, 2004. **41**(1): p. 85-94.

28. Kannan, H., C.L. Bloebaum, and B.L. Mesmer, *Incorporation of Value-Driven Design in Multidisciplinary Design Optimization*, in *10th World Congress on Structural and Multidisciplinary Optimization*. 2013: Orlando, FL.

29. *Federal Acquisition Regulation*, in *Part 16 - Types of Contracts*, D.o.D. General Services Administration, National Aeronautics and Space Administration, Editor. 2012, General Services Administration, Department of Defense, National Aeronautics and Space Administration.

30. Harrison, T., *The effects of competition on defense acquisitions*. 2012, DTIC Document: Defense Acquisition University Research Symposium.

31. Collopy, P.D., *Economic-Based Distributed Optimal Design*, in *AIAA Space 2001 - Conference and Exposition*. 2001, AIAA, DFM Consulting: Albuquerque, NM.

32. *Monopolies and Combinations in Restraint of Trade*, in *Title 15 - Commerce and Trade*, U.S.H.o. Representatives, Editor. 2014, Office of the Law Revision Counsel.

33. Collopy, P.D., *Economic-Based Distributed Optimal Design*, in *AIAA Space 2001 - Conference and Exposition*. 2001: Albuquerque, NM.

34. Mesmer, B.L., C.L. Bloebaum, and H. Kannan, *Incorporation of Value-Driven Design in Multidisciplinary Design Optimization*, in *10th World Congress on Structural and Multidisciplinary Optimization*. 2013: Orlando, FL.

35. Cheung, J., et al., *Application of Value-Driven Design to Commercial Aeroengine Systems*. Journal of Aircraft, 2012. **49**(3): p. 688-702.

36. Buckner, J.K., P.W. Hill, and D.B. Benepe, *Aerodynamic design evolution of the YF-16*, in *AIAA 6th Aircraft Design, Flight Test and Operations Meeting*. 1974, American Institute of Aeronautics and Astronautics: Los Angeles, California.

37. *F-16 Fighting Falcon*. Products 2014 [cited 2014 November]; Available from: http://www.lockheedmartin.com/us/products/f16.html.

38. *F-35 Joint Strike Fighter: Current Outlook is Improved, but Long-Term Affordability is a Major Concern*, U.S.G.A. Office, Editor. 2013.

39. Hess, R.W. and H.P. Romanoff, *Aircraft Airframe Cost Estimating Relationships: All Mission Types*. 1987, The RAND Corporation: Santa Monica, CA.

40. Birkler, J.L., J.B. Garfinkle, and K.E. Marks, *Development and Production Cost Estimating Relationships for Aircraft Turbine Engines*. 1982, The RAND Corporation: Santa Monica, CA.

41. *Consumer Price Index*. 2014; Available from: http://www.bls.gov/cpi/.

42. *History: F-4 Phantom II Fighter*. 2014 [cited 2014 November]; Available from: http://www.boeing.com/boeing/history/mdc/phantomII.page.

43. *Viking 1 & 2*. Program & Missions 2014 [cited 2014 5]; Available from: http://mars.nasa.gov/programmissions/missions/past/viking/.

44. *Mars Exploration Rovers*. 2014 [cited 2014 5/2014]; Available from: http://marsrover.nasa.gov/home/index.html.

45. Ball, R.E., *The Fundamentals of Aircraft Combat Survivability Analysis and Design*. Second ed. AIAA Education Series, ed. J.A. Shetz. 2003: American Institute of Aeronautics and Astronautics.

46. Collopy, P., Horton, Randolph, *Value Modeling for Technology Evaluation.* AIAA Paper, 2002. **3622**.

47. Kelly, B.P. and B. Stevens, *Cost effective analysis comparing the small diameter bomb and the joint standoff weapon (A+ Variant)*. 2004, DTIC Document.

48. Rodt, A.P., *Successful Conflict Management by Military Means.* Ethnopolitics, 2012. **11**(4): p. 376-391.

49. Osborne, M.J., *An Introduction to Game Theory*. 2004: Oxford University Press New York.

50. Rubinstein, A., *Perfect equilibrium in a bargaining model.* Econometrica: Journal of the Econometric Society, 1982. **50**(1): p. 97-109.

51. *History: B-1 Lancer Bomber*. 2014 [cited 2014 5/14]; Available from: http://www.boeing.com/boeing/history/bna/b1b.page.

52. *Capabilities: B-2 Spirit Stealth Bomber*. 2014 [cited 2014 5/14]; Available from: http://www.northropgrumman.com/capabilities/b2spiritbomber/pages/default.aspx.

53. *History: B-52 Stratofortress*. 2014 [cited 2014 5/14]; Available from: http://www.boeing.com/boeing/history/boeing/b52.page.

54. *History: F-15 Eagle Tactical Fighter*. 2014 [cited 2014 5/14]; Available from: http://www.boeing.com/boeing/history/mdc/eagle.page.

55. *History: F/A-18 Hornet Fighter*. 2014 [cited 2014 5/14].

56. *Products: F-22 Raptor*. 2014 [cited 2014 5/14]; Available from: http://www.lockheedmartin.com/us/products/f22.html.

57. *F-35 Lightning II: Capabilities: Multi-Mission Capability for Emerging Global Threats*. 2014; Available from: https://www.f35.com/about/capabilities.

58. Anderson, J.D., *Aircraft Performance and Design*. 1999: WCB/McGraw-Hill.

59. Oates, G.C., *Aerothermodynamics of Gas Turbine and Rocket Propulsion*. Third ed. AIAA Education Seriess, ed. J.S. Przemieniecki. 1997: AIAA.

60. Kannan, H., C.L. Bloebaum, and B.L. Mesmer, *Incorporation of Coupling Strength Models in Decomposition Strategies for Value-Based MDO*, in *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2014, American Institute of Aeronautics and Astronautics: Atlanta, GA.

61. *MATLAB*. 2014, The MathWorks, Inc.

62. *SolidWorks*. 2014, Dassault Systèmes SolidWorks Corporation.

APPENDIX

AIRCRAFT DESIGN

MATLAB Codes

The work presented in this document was conducted using codes written with MATLAB [61] and its scripting language. The SolidWorks software [62] was used to render the aircraft images in this document. The following codes take supporting direction from [21, 45, 58] to size and shape an aircraft.

Main Program

```
% ========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
% ========================================================================

h_cruise = 12000;    % m
[~,a_cruise,~,~,~,~]=stdatmo(h_cruise);

W_payload_lim = [0 100000]*9.81;
V_cruise_lim = [0.5,1.5]*a_cruise;
Lambda1_lim = [20 60]*pi/180;
b_lim = [10 20];         % m
cL_lim = [0.5 1];
ct_lim = [2 4];
Q_lim = [90, 100];
x_lim =
[W_payload_lim;V_cruise_lim;Lambda1_lim;b_lim;cL_lim;ct_lim;Q_lim];

p = [h_cruise];
pso_parameters = [1,9,1,0.8,1.1,1.1,0.5,1e-3,1e-3,1400];

[F,x,y,k] = my_pso(@mdo_system,x_lim,11,p,pso_parameters);

W_payload = x(1,:);
V_cruise = x(2,:);
Lambda1 = x(3,:);
b = x(4,:);
cL = x(5,:);
ct = x(6,:);
```

```
CD0 = y(1,:);
c_t = y(2,:);
nE = y(3,:);
range = y(4,:);
W = y(5:7,:);
RCS = y(8,:);
Q = y(9,:);
Cost = y(10,:);
Revenue = y(11,:);
prob_survive = y(12,:);
prob_eff = y(13,:);

[~,a_cruise,~,~,~,~]=stdatmo(h_cruise);
M_cruise = V_cruise./a_cruise;
[x_pos,y_pos,S,Lambda2] = geometry(M_cruise,Lambda1,b,cL,ct);

fprintf('Dimensions\n')
fprintf('  Lambda1: %6.2f deg.\n',Lambda1*180/pi)
fprintf('  Lambda2: %6.2f deg.\n',Lambda2*180/pi)
fprintf('        b: %6.2f m\n', b)
fprintf('      c_L: %6.2f m\n',cL)
fprintf('      c_t: %6.2f m\n',ct)
fprintf('        S: %6.2f m^2\n',S(1,:))
fprintf('       AR: %6.2f\n',b.^2./S(1,:))
fprintf('\n')
fprintf('Aerodynamics\n')
fprintf('      CD0: %6.4f\n',CD0)
fprintf('     TSFC: %8.6f (kg/s)/N\n',c_t)
fprintf('  Engines: %2.0f\n',nE)
fprintf('\n')
fprintf('Weights\n')
fprintf('     W_TO: %6.2f kN\n',W(1,:)/1000)
fprintf('  W_empty: %6.2f kN\n',W(2,:)/1000)
fprintf('   W_fuel: %6.2f kN\n',W(3,:)/1000)
fprintf('   W_bomb: %6.2f kN\n',W_payload/1000)
fprintf('\n')
fprintf('Performance\n')
fprintf(' M_cruise: %4.2f\n',M_cruise)
fprintf('    Range: %8.2f km\n',range/1000)
fprintf('\n')
fprintf('Economics\n')
fprintf(' Quantity: %6f\n',Q)
fprintf('Prog Cost: %6.2f $B\n',Cost/1e9)
fprintf('Unit Cost: %6.2f $M\n',Cost./Q/1e6)
fprintf('Unit Rev.: %6.2f $M\n',Revenue./Q./1e6)
fprintf('\n')
fprintf('Probabilities\n')
fprintf('  Survive: %6.2f\n',prob_survive)
fprintf('Effective: %6.2f\n',prob_eff)
fprintf('\n')

geom_to_sldw(x_pos,y_pos,Lambda1,Lambda2)
```

Particle Swarm Optimization (Custom)

```
function [F,x,y,k] = my_pso(system,x_lim,ny,p,pso_parameters)
%
==========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
==========================================================================
%
==========================================================================
% Particle Swarm Optimization
%   [F,x,y,k] = pso(system,x_lim,ny,p,pso_parameters)
% =============================== INPUTS
==================================
%         system: handle of function name
%                   system = @functionname
%                   [F,y] = functionname(x,y,p,converge_crit_sys)
%          x_lim: limits on design variables
(nx*2)
%                  x_lim(:,1): lower limits
%                  x_lim(:,2): upper limits
%             ny: number of behavior variables in system
%              p: system parameter vector
(1*np)
%                 pso_parameters: [max_F,limit,w,c1,c2,c3,converge_sys,...
%                                  converge_pso,particles]
(1*9)
%                          max_F: maximize F (t/f, 1/0)
%                             Fi: index of F to use as evaluation criteria
%                          limit: use the limits on x (t/f, 1/0)
%                              w: inertial weight (approx 0.8)
%                             c1: weight on personal best difference (1-2)
%                             c2: weight on global best difference (1-2)
%                             c3: weight on neighborhood best difference
(1-2)
%                   converge_pso: pso convergence criteria
%                   converge_sys: system convergence criteria
%                      particles: (optional) number of particles to use
% ============================== OUTPUTS
==================================
%              F: value of best particle
%              x: design variables of best particle
(nx*1)
%              y: behavior variables of best particle
(ny*1)
%              k: number of iterations to converge
%
==========================================================================

    [nx,~] = size(x_lim);       % number of design variables
    max_F = pso_parameters(1);
    Fi = pso_parameters(2);
    limit = pso_parameters(3);
    w = pso_parameters(4);
    c1 = pso_parameters(5);
    c2 = pso_parameters(6);
    c3 = pso_parameters(7);
    converge_crit_sys = pso_parameters(8);
    converge_crit_pso = pso_parameters(9);
    if(length(pso_parameters) > 9)
        particles = pso_parameters(10);
    else
        particles = 10*nx;
    end

    % Populate Design Space and Converge System First Time
    x = x_lim(:,1)*ones(1,particles) + ...
        (x_lim(:,2)-x_lim(:,1))*ones(1,particles).*rand([nx,particles]);
    y = zeros(ny,particles);
    [F,y] = system(x,y,p,converge_crit_sys);

    % Converge Design Space
    F_prev = F(Fi,:)+1;
    if(max_F)
        F_p_best = F(Fi,:)-1;     % personal best value (maximize value)
        update_personal_best = @(f,fp) find(f-fp > 0);
        minmax = @max;
    else
        F_p_best = F(Fi,:)+1;     % personal best value (minimize value)
        update_personal_best = @(f,fp) find(f-fp < 0);
        minmax = @min;
    end
    x_p_best = x;   % personal best position
    v = zeros(nx,particles);
    k = 0;

    while(abs(std(F(Fi,:))/mean(F(Fi,:)))>converge_crit_pso && k < 100)
        % Find personal bests
        s = update_personal_best(F(Fi,:),F_p_best);
        F_p_best(s) = F(Fi,s);
        x_p_best(:,s) = x(:,s);

        % Find global best
        [~,s] = minmax(F_p_best);
        x_g_best = x_p_best(:,s);

        % Find neighborhood best
        if(false)
            % non-dimensionalize distance by limits on design space
            particle_distance = zeros(particles,particles);
            for i=1:nx;
                particle_distance = particle_distance + ...
                    ((ones(particles,1)*x(i,:) - x(i,:)'...
                    *ones(1,particles))/(x_lim(i,2)-x_lim(i,1))).^2;
            end
            particle_distance =
+(~logical(floor((particle_distance.^0.5)/(0.1*nx^0.5))));
            % indicies with "1" are within 10% distance
            particle_distance(particle_distance == 0) = NaN;
            [~,s] = minmax(F'*ones(1,particles).*particle_distance,[],1);
            x_n_best = x(:,s);
        else
            x_n_best = x;
        end
        % Update x vectors
        v = w*v +(ones(nx,1)*rand(1,particles)).*(c1*(x_p_best
-x))+...

(ones(nx,1)*rand(1,particles)).*(c2*(x_g_best*ones(1,particles)-x))+...
                 (ones(nx,1)*rand(1,particles)).*(c3*(x_n_best
-x));
        x = x+v;
        if(limit)  % Check for particles outside limits
            for i=1:nx;
                s = find(x(i,:) < x_lim(i,1));
                x(:,s) = x(:,s) - v(:,s).*abs(ones(nx,1)*((x_lim(i,1)-
x(i,s))./v(i,s)));
                s = find(x(i,:) > x_lim(i,2));
                x(:,s) = x(:,s) - v(:,s).*abs(ones(nx,1)*((x_lim(i,2)-
x(i,s))./v(i,s)));
            end
        end

        % Converge system with new x
        F_prev = F(Fi,:);
        [F,y] = system(x,y,p,converge_crit_sys);
        k = k+1;
        fprintf('.')
        if(mod(k,10) == 0)
            fprintf('\n')
            v = zeros(nx, particles);
        end
    end
    fprintf('\n')
    [~,s] = minmax(F_p_best);
    x = x(:,s);
    y = y(:,s);
    F = F(Fi,s);
end
```

Aircraft Analysis Block (DSM)

```
function [F,y] = mdo_system(x,y,p,converge_crit_sys)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
% Finds values that fit the x values provided
% ================================= INPUTS
================================
% x: design variables
7xn
%    x(1,:) = payload weight
%    x(2,:) = cruise velocity
%    x(3,:) = Lambda1
%    x(4,:) = b
%    x(5,:) = cL
%    x(6,:) = ct
%    x(7,:) = quantity produced
% y: behavior variables
%    [empty]
% p: parameters
1xn
%    p(1,:) = h_cruise: cruising altitude
% converge_crit_sys: system convergence criteria
%    [empty]
% ================================ OUTPUTS
================================
% F: Value/Objective function
9xn
%    F( 1,:) = range
%    F( 2,:) = gross take-off weight, N
%    F( 3,:) = designer cost, USD
%    F( 4,:) = designer revenue, USD
%    F( 5,:) = designer profit, USD
%    F( 6,:) = probability of survivability
%    F( 7,:) = probability of effectiveness
%    F( 8,:) = probability of operational success
%    F( 9,:) = bargained design
% y: behavior variables
13xn
%    y( 1,:) = profile drag coefficient
%    y( 2,:) = thrust specific fuel consumption
%    y( 3,:) = number of engines
%    y( 4,:) = range
%    y( 5,:) = gross take-off weight, N
%    y( 6,:) = empty weight, N
%    y( 7,:) = fuel weight, N
%    y( 8,:) = radar cross section
%    y( 9,:) = quantity of aircraft produced
%    y(10,:) = designer revenue, USD
%    y(11,:) = designer profit, USD
%    y(12,:) = probability of survivability
%    y(13,:) = probability of effectiveness
%
=========================================================================

    W_payload = x(1,:);       % N
    v_cruise = x(2,:);        % m/s
    Lambda1 = x(3,:);
    b = x(4,:);
    cL = x(5,:);
    ct = x(6,:);
    Q = x(7,:);

    h_cruise = p(1);    % m

    t(1) = 2;    % m
    t(2) = 0.5;  % m
    % Atmospheric properties
    [rho_cruise,a_cruise,~,~,~,dvisc]=stdatmo(h_cruise);
    M_cruise = v_cruise./a_cruise;

    [x_pos,y_pos,S,Lambda2,c_bar] = geometry(M_cruise,Lambda1,b,cL,ct);
    Re = rho_cruise.*v_cruise.*c_bar./dvisc;
    [W_fuel] = sizing(S,t,W_payload);
    [CD0,K] = aerodynamics(M_cruise,Re,2*S(1,:),S(1,:),b,Lambda2,1);
% aerodynamics
    [c_t,nE,T_max] = propulsion(b);
% engine
    [range, W_GTO, W_Empty, W_Fuel] = performance(c_t, CD0, K, S(1,:), ...
        rho_cruise, M_cruise, W_payload, W_fuel);    % weight estimate
(range estimation)
    [RCS,prob_survive,prob_eff] = operations(x_pos, y_pos,
S(1,:),[t(1),t(2)], ...
        W_payload, range);
    [Profit_0,Revenue_0,Cost_0] =
economics(W_Empty,v_cruise,Q,2013,T_max,nE,...
        range,W_payload,prob_survive);

    y =
[CD0;c_t;nE;range;W_GTO;W_Empty;W_Fuel;RCS;Q;Cost_0;Revenue_0;prob_survive
;prob_eff];

    F(1,:) = range;
    F(2,:) = W_GTO;
    F(3,:) = Cost_0;
    F(4,:) = Revenue_0;
    F(5,:) = Profit_0;
    F(6,:) = prob_survive;
    F(7,:) = prob_eff;
    F(8,:) = prob_survive.*prob_eff;
    % F(9,:) = 0.5*F(5,:) + 0.5*F(8,:)*1e11;
    F(9,:) = 0.5*F(5,:) + 0.5*F(8,:)*3e11;

    figure(1)
    plot(x_pos,y_pos)
end
```

1976 Standard Atmosphere

```
function [rho,a,T,p,kvisc,dvisc,Re_MC]=stdatmo(h,dT)
%
==========================================================================
% 1976 Standand Atmosphere model
% ================================= INPUTS
================================
%            H: Altitude, m
%           dT: Temp. offset, C,K
% ================================ OUTPUTS
================================
%          rho: Density, kg/m^3
%            a: Speed of sound, m/s
%            T: Temperature, K
%            P: Pressure, Pa
%           nu: Kinematic viscosity, m^2/s
%           mu: Dynamic viscosity, m^2/s
%        Re_MC: Reynold's Number/(Mach Number*Characteristic length)
%
=========================================================================

    if (nargin < 2)
        dT = 0;
    end

    % Constants
    R=287.05287;    %N-m/kg-K; value from ESDU 77022
    % R=287.0531;   %N-m/kg-K; value used by MATLAB aerospace toolbox
ATMOSISA
    gamma=1.4;
    g0=9.80665;     %m/sec^2
    Bs = 1.458e-6;  %N-s/m2 K1/2
    S = 110.4;      %K

    T = zeros(size(h));
    p = T;
    h_base=[0, 11, 20, 32, 47, 51, 71, 86]*1000;    % m
    hmax = 90000;

    n1=(h<=h_base(2));
    n2=(h<=h_base(3) & h>h_base(2));
    n3=(h<=h_base(4) & h>h_base(3));
    n4=(h<=h_base(5) & h>h_base(4));
    n5=(h<=h_base(6) & h>h_base(5));
    n6=(h<=h_base(7) & h>h_base(6));
    n7=(h<=h_base(8) & h>h_base(7));
    n8=(h<=hmax & h>h_base(8));
    n9=(h>hmax);

    T_base = 288.15;    % K
    p_base = 101325;    % Pa
    i = 1;

    % Troposphere
    if any(n1(:))
        [T(n1),p(n1)] = gradient_layer(h(n1),h_base(i),T_base,p_base,-
0.0065,g0,R);
    end
    [T_base,p_base] = gradient_layer(h_base(i+1),h_base(i),T_base,p_base,-
0.0065,g0,R);
    i = i+1;

    % Tropopause
    if any(n2(:))
        [T(n2),p(n2)] =
isothermal_layer(h(n2),h_base(i),T_base,p_base,g0,R);
    end
    [T_base,p_base] =
isothermal_layer(h_base(i+1),h_base(i),T_base,p_base,g0,R);
    i = i+1;
    % Stratosphere 1
    if any(n3(:))
        [T(n3),p(n3)] = 
gradient_layer(h(n3),h_base(i),T_base,p_base,0.001,g0,R);
    end
    [T_base,p_base] = 
gradient_layer(h_base(i+1),h_base(i),T_base,p_base,0.001,g0,R);
    i = i+1;

    % Stratosphere 2
    if any(n4(:))
        [T(n4),p(n4)] = 
gradient_layer(h(n4),h_base(i),T_base,p_base,0.0028,g0,R);
    end
    [T_base,p_base] = 
gradient_layer(h_base(i+1),h_base(i),T_base,p_base,0.0028,g0,R);
    i = i+1;

    % Stratopause
    if any(n5(:))
        [T(n5),p(n5)] = 
isothermal_layer(h(n5),h_base(i),T_base,p_base,g0,R);
    end
    [T_base,p_base] = 
isothermal_layer(h_base(i+1),h_base(i),T_base,p_base,g0,R);
    i = i+1;

    % Mesosphere 1
    if any(n6(:))
        [T(n6),p(n6)] = gradient_layer(h(n6),h_base(i),T_base,p_base,-
0.0028,g0,R);
    end
    [T_base,p_base] = gradient_layer(h_base(i+1),h_base(i),T_base,p_base,-
0.0028,g0,R);
    i = i+1;

    % Mesosphere 2
    if any(n7(:))
        [T(n7),p(n7)] = gradient_layer(h(n7),h_base(i),T_base,p_base,-
0.002,g0,R);
    end
    [T_base,p_base] = gradient_layer(h_base(i+1),h_base(i),T_base,p_base,-
0.002,g0,R);
    i = i+1;

    % Mesopause
    if any(n8(:))
        [T(n8),p(n8)] = 
isothermal_layer(h(n8),h_base(i),T_base,p_base,g0,R);
    end
    [T_base,p_base] = 
isothermal_layer(h_base(i+1),h_base(i),T_base,p_base,g0,R);
    i = i+1;

    if any(n9(:))
        T(n9)=NaN;
        p(n9)=NaN;
    end

    T = T + dT;

    rho = p./T/R;
    a = sqrt(gamma * R * T);     % m/s
    kvisc = (Bs * T.^1.5 ./ (T + S)) ./ rho; % m2/s

    mu0 = 1.827e-5; % Pa*s
    T0 = 291.15;    % K
    C = 120;        % K
    dvisc = mu0*(T0+C)./(T+C).*(T./T0).^1.5;
    Re_MC = rho./(dvisc.*a);
end

function [T,p] = gradient_layer(h,h_base,T_base,p_base,K,g0,R)
    T = T_base + K*(h-h_base);
    p = p_base*(T./T_base).^(-g0/(K*R));
end

function [T,p] = isothermal_layer(h,h_base,T_base,p_base,g0,R)
    T = T_base;
    p = p_base*exp(-g0./(R*T).*(h-h_base));
end
```

Geometry

```
function [x,y,S,Lambda2,c_bar] = geometry(M,Lambda1,b,c_L,c_t)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
% ================================= INPUTS
================================
%       b: span
1xn
%      bp: span prime
1xn
%     c_L: shortest distance from wing edge to nose (projected)
1xn
%     c_t: tip chord of wing (wing width)
1xn
% Lambda1: Leading edge sweep angle of body
1xn
% Lambda2: Leading edge sweep angle of wing
1xn
% ================================ OUTPUTS
================================
%       x: x-position of points of interest
11xn
%       y: y-position of points of interest
11xn
```

```
%        S: planform area (internal area of polygon)
1xn
%
===========================================================================


    [~,nn(1)] = size(M);
    [~,nn(2)] = size(Lambda1);
    [~,nn(3)] = size(b);
    [~,nn(4)] = size(c_L);
    [~,nn(5)] = size(c_t);
    n = max(nn);
    if(nn(1) == 1)
        M = M*ones(1,n);
    end
    if(nn(2) == 1)
        Lambda1 = Lambda1*ones(1,n);
    end
    if(nn(3) == 1)
        b = b*ones(1,n);
    end
    if(nn(4) == 1)
        c_L = c_L*ones(1,n);
    end
    if(nn(5) == 1)
        c_t = c_t*ones(1,n);
    end

    Lambda2 = Lambda1+15*pi/180;
    bp = b./3;

    [~,n] = size(b);

    m = 11;
    x = zeros(m,n);
    y = x;

    s = c_t > (b-bp).*cos(Lambda2);
    if(sum(s) > 0)
        cts = (b-bp).*cos(Lambda2);
        c_t(s) = cts(s);
    end

    x(1,:) = 0;
    y(1,:) = 0;
    x(2,:) = x(1,:) + c_L.*cos(Lambda1)./sin(Lambda2-Lambda1);
    y(2,:) = y(1,:) + c_L.*sin(Lambda1)./sin(Lambda2-Lambda1);
    x(3,:) = x(2,:) + (b/2-y(2,:))./tan(Lambda2);
    y(3,:) = y(1,:) + b/2;
    x(4,:) = x(3,:) + c_t./(2*sin(Lambda2));
    y(4,:) = y(3,:) - c_t./(2*cos(Lambda2));
    y(5,:) = (bp/2);
    x(5,:) = x(4,:) - (y(4,:)-bp/2)./tan(Lambda2);
    x(6,:) = x(5,:) + (bp/2)./tan(Lambda2);
    y(6,:) = y(1,:);

    x(7,:) = x(6,:)-abs(x(6,:)-x(5,:));
    y(7,:) = y(6,:)-abs(y(6,:)-y(5,:));
    x(8,:) = x(7,:)+abs(x(5,:)-x(4,:));
    y(8,:) = y(7,:)-abs(y(5,:)-y(4,:));
    x(9,:) = x(8,:)-abs(x(4,:)-x(3,:));
    y(9,:) = y(8,:)-abs(y(4,:)-y(3,:));
    x(10,:) = x(9,:)-abs(x(3,:)-x(2,:));
    y(10,:) = y(9,:)+abs(y(3,:)-y(2,:));
    x(11,:) = x(1,:);
    y(11,:) = y(1,:);

    S = polygon_area(x,y);
    S(3,:) = 2*polygon_area(x(2:5,:),y(2:5,:));     % Wing
    S(2,:) = S(1,:)-S(3,:);                         % Body


    c_bar = S(1,:)./b;
end

function A = polygon_area(x,y)
    [m,~] = size(x);
    A = x(m,:).*y(1,:) - x(1,:).*y(m,:);
    for i=1:(m-1);
        A = A + x(i,:).*y(i+1,:) - x(i+1,:).*y(i,:);
    end
    A = abs(A)/2;
end
```

Sizing

```
function [W_fuel] = sizing(S,t,W_payload)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
    rho_payload = 1372;     % kg/m^3 if the Mark 82 was rectangular
    rho_fuel = 810;         % kg/m^3

    Total_Volume = S(2,:)*t(1) + S(3,:)*t(2);
    r = 0.4*(1-exp(-Total_Volume/50));    % estimate 35 % of aircraft
volume is fuel and payload
    %r = 0.4;
    Fuel_and_Payload_Volume = Total_Volume.*r;
    W_fuel = (Fuel_and_Payload_Volume-
W_payload/9.81/rho_payload)*rho_fuel*9.81;
    W_fuel(W_fuel < 0) = 0;
end
```

Aerodynamics

```
function [CD0,K] = aerodynamics(M,Re,Swet,Sref,b,lambda_tc,tc)
```

```
% the following function is based on observed trends
    CD00 = 0.01;
    CD0 = CD00 *  (1+(0.5+atan(50*(M-1-
0.1*sin(lambda_tc)))./pi).*cos(lambda_tc).^2);

    S = Sref;
    AR = b.^2./S;
    %e = 1-0.04*AR;      % from Fig G.9 - experimental
    e = exp(-AR./18);       % similar to line above to continue forever
    K = 1./(pi*e.*AR);
end
```

Propulsion

```
function [c_t, nE, T_max] = propulsion(b)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
%
=========================================================================
%     b:
%
=========================================================================
%   c_t: Thrust Specific Fuel Consumption
%
=========================================================================

    [~,n] = size(b);
    x = 4*(b/60);
    x = 4*ones(1,n);
    c_t = 0.00006*x;    % (kg/s)/N, from F110 (approx 0.00005892 (kg/s)/N
    nE = min([ceil(4*b/60);4*ones(1,n)]);
    T_max = 135000;     % N, from F110
end
```

Performance

```
function [range, W_GTO, W_Empty, W_Fuel] = performance(c_t, CD0, K, S, ...
    rho_cruise, M_cruise, W_payload, W_fuel)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
    % find range from fuel weight
    [~,n] = size(W_fuel);
    del = 1e-6;
    del_v = 1 + [-1;0;1]*del;
    range = del_v*ones(1,n)*10000000;    % guess 1,000 km
    W_abbrev = ones(3,1)*W_fuel + 20;
    k = 0;
    while(max(abs(W_abbrev(3,:) - W_fuel)) > 10 && k<30)
        W_fuel_m = weight(c_t, CD0, K, S, rho_cruise, M_cruise, W_payload,
range(1,:));
        [W_abbrev,W] = weight(c_t, CD0, K, S, rho_cruise, M_cruise,
W_payload, range(2,:));
        W_fuel_p = weight(c_t, CD0, K, S, rho_cruise, M_cruise, W_payload,
range(3,:));
        range = del_v*(range(2,:) + (range(3,:) - range(1,:))./...
            (W_fuel_p(3,:) - W_fuel_m(3,:)).*(W_fuel - W_abbrev(3,:)));
        k = k+1;
    end

    range = range(2,:);
    range(range<0) = 0;
    W_GTO = W(1,:);
    W_Empty = W(2,:);
    W_Fuel = W(3,:);
end

function [W_abbrev, W] = weight(c_t, CD0, K, S, rho_cruise, M_cruise,
W_payload, range, endurance)
% ================================= INPUTS
================================
%        c_t: thrust specific fuel consumption
(1xn)
%        CD0: Profile drag coefficient
(1xn)
%          K: Lift induced drag coefficient
(1xn)
%          S: Planform area, m^2
(1xn)
% rho_cruise: air density at cruising altitude, kg/m^3
(1xn)
%   M_cruise: Cruising Mach number
(1xn)
%  W_payload: Payload Weight, N
(1xn)
%      range: Aircraft range
(1xn)
% ================================ OUTPUTS
================================
%   W_abbrev: Weight matrix, abbreviated, N
(3xn)
%               W(1,:) Gross takeoff weight
%               W(2,:) Empty weight
%               W(3,:) Fuel weight
%          W: Weight matrix, N
(mxn)
%               W(1,:) is gross take-off weight
%               W(m,:) is empty weight
%
========================================================================
% let's just keep everything in SI units
% Chapter 5 of Fundamentals of Aircraft and Airship Design

    [~,n] = size(CD0);

    reserve = 0.05;     % 5% of fuel is on reserve
    trapped = 0.01;     % 1% of fuel is trapped in fuel lines

    mission = [0;1;2;4;2;3;6];
    m = length(mission);
    M = zeros(m,n);
    M(2,:) = M_cruise;
    M(3,:) = M(2,:);
    M(4,:) = M(3,:);
    M(5,:) = M(4,:);

    W = zeros(m,n);
    Wp = W;
    Wm = W;

    del = 1e-6;

    W_TO = 2000000; % guess
    W(1,:) = W_TO;
    Wp(1,:) = W(1,:)*(1+del);
    Wm(1,:) = W(1,:)*(1-del);
    W_empty_e = W_TO+2*del;
    k = 0;
    while(max(abs(W(m,:)-W_empty_e)) > 1 && k<30)
        [W]  = mission_weights(W
,mission,c_t,CD0,K,S,M,rho_cruise,W_payload,...
            range/2,reserve,trapped);
        [Wp] =
mission_weights(Wp,mission,c_t,CD0,K,S,M,rho_cruise,W_payload,...
            range/2,reserve,trapped);
        [Wm] =
mission_weights(Wm,mission,c_t,CD0,K,S,M,rho_cruise,W_payload,...
            range/2,reserve,trapped);
        W_empty_e = 0.986*W(1,:).^0.947;     % in N, 0.911.*W(1).^0.947 in
lbf;
        W(1,:) = W(1,:) + (Wp(1,:) - Wm(1,:))./(Wp(m,:) -
Wm(m,:)).*(W_empty_e-W(m,:));
        Wp(1,:) = W(1,:)*(1+del);
        Wm(1,:) = W(1,:)*(1-del);
        k = k+1;
    end
    W = real(W);

    W_abbrev(1,:) = W(1,:);   % gross take-off weight
    W_abbrev(2,:) = W(m,:);   % empty weight
    W_abbrev(3,:) = W(1,:) - W(m,:) - W_payload;
end

function [W] = mission_weights(W,flight_plan,c_t,CD0,K,S,M,rho_cruise,...
    W_payload,range_operating,reserve,trapped)
% ================================= INPUTS
================================
%          W: Gross takeoff weight
(1xn)
%flight_plan: Flight plan vector
(mx2)
%               first column provides flight plan identification number
%               second column provides useful information
%               third column provides useful information
%               1. Acceleration: Mach start, Mach end
%               2. Cruise:
%        c_t: thrust specific fuel consumption
(1xn)
%        CD0: Profile drag coefficient
(1xn)
%          K: Lift induced drag coefficient
(1xn)
%          S: Planform area, m^2
(1xn)
% rho_cruise: air density at cruising altitude, kg/m^3
(1xn)
%   M_cruise: Cruising Mach number
(1xn)
%  W_payload: Payload Weight, N
(1xn)
%      range: Aircraft range
(1xn)
% ================================ OUTPUTS
================================
%   W_abbrev: Weight matrix, abbreviated, N
(3xn)
%               W(1,:) Gross takeoff weight
%               W(2,:) Empty weight
%               W(3,:) Fuel weight
%          W: Weight matrix, N
(mxn)
%               W(1,:) is gross take-off weight
%               W(m,:) is empty weight
%
=========================================================================
    [m,~] = size(W);
    for i=2:m;
        switch flight_plan(i)
            case 1      % Acceleration (increase in kinetic or potential
energy)
                % use for take-off, climbing, and accelerating speed,
about 0.97 - 0.975
                W(i,:) = (1 - 0.0278.*M(i,:) - 0.0088.*M(i,:).^2) ./ ...
                    (1 - 0.0278.*M(i-1,:) - 0.0088.*M(i-1,:).^2) .* W(i-
1,:);
            case 2      % Cruise (requires cruise distance)
                W(i,:) = range_breguet_jet_i(W(i-
1,:),range_operating,c_t,...
                    rho_cruise,S,K,CD0);
            case 3      % Loiter
                endurance_loiter = 30*60;
                W(i,:) = endurance_jet_i(W(i-
1,:),c_t,CD0,K,endurance_loiter);
            case 4      % Drop payload (requires payload weight)
```

```
                W(i,:) = W(i-1,:) - W_payload;
            case 5      % Combat
                W(i,:) = W(i-1,:);
            case 6      % Land
                % this landing stage incorporates fuel reserves and
trapped fuel estimation
                W(i,:) = W(1,:) - W_payload - ...
                    (1+reserve+trapped)*(W(1,:) - W_payload - W(i-1,:));
        end
    end
end

function [W1] = range_breguet_jet_i(W0,R,c_t,rho,S,K,CD0)
% maximum range of a jet (flying at max(V*L/D)), assuming it flies at a
% constant altitude (constant air density)
    W1 = (W0.^0.5 - (R.*c_t.*(2.*rho.*S).^0.5.*(3*K.*CD0.^3).^0.25)/3).^2;
end

function [W1] = endurance_jet_i(W0,c_t,CD0,K,E)
% maximum endurance of a jet (flying at max(L/D))
    W1 = W0.*exp(-2*E.*c_t.*(K.*CD0).^0.5);
end
```

Operations

```
function [RCS,prob_survive,prob_eff] = operations(x_pos, y_pos, S, t,
W_payload, range)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
    [RCS,prob_survive] = survive(x_pos,y_pos,S,t);
    [prob_eff] = effective(W_payload, range);

end

function [prob_eff] = effective(W_payload, range)
    prob_eff = (1-exp(-W_payload./175000)).*(1-exp(-range/11000));
end

function [sigma,prob_survive] = survive(x,y,S,t)
    A1 = t(1)*((x(5,:)-x(1,:)).^2 + (y(5,:)-y(1,:)).^2).^0.5 - ...
         t(2)*((x(5,:)-x(2,:)).^2 + (y(5,:)-y(2,:)).^2).^0.5;
    A2 = t(2)*((x(3,:)-x(2,:)).^2 + (y(3,:)-y(2,:)).^2).^0.5;

    s = A1>A2;
    if(sum(s) == length(A1))
        A = A1;
    elseif(sum(s) == 0)
        A = A2;
    else
        A(s) = A1(s);
        A(~s) = A2(~s);
    end
    lambda = 1;     % m  0.3 GHz frequency - Long-Range Detection radar
    sigma = 4*pi*A.^2./lambda.^2;
    prob_susceptible = 1-exp(-sigma/10);
    prob_killed = exp(-S./1000);
    prob_survive = prob_susceptible.*prob_killed;
end
```

Economic Performance

```
function [Profit_0,Revenue_0,Cost_0] =
economics(W_empty,V_max,Q,Year,T_max,nE,range,W_payload,prob_survive)
%
=========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
=========================================================================
    Cost_0 = lifecycle_cost(W_empty,V_max,Q,Year,T_max,nE);
    Revenue_0 =
0.0075*(range/1000).^2.*W_payload.^0.5.*prob_survive.^6.*Q;
    Profit_0 = Revenue_0-Cost_0;

end

function [C] = lifecycle_cost(W_empty,V_max,Q,Year,T_max,nE)
% ================================= INPUTS
================================
% W_empty: Empty weight of aircraft
1xn
%   V_max: Maximum speed of aircraft
1xn
%       Q: Quantity of aircraft to produce
1xn
%    Year: Year of production and development
1xn
%   T_max: Maximum engine thrust
1xn
%      nE: Number of engines per aircraft
1xn
% ================================ OUTPUTS
================================
%       C: Cost of production and development
1xn
%
=========================================================================

    [HR_Tooling, HR_Engineering, HR_Manufacturing, HR_QualityControl] =
hourly_rate(Year);
    [H_Tooling, H_Engineering, H_Manufacturing, H_QualityControl] =
hourly(W_empty,V_max,Q);

    C_Tooling = H_Tooling*HR_Tooling;
    C_Engineering = H_Engineering*HR_Engineering;
    C_Manufacturing = H_Manufacturing*HR_Manufacturing;
    C_QualityControl = H_QualityControl*HR_QualityControl;

    inflation = cpi(1977,13,Year,13);

    C_DevelopmentSupport = 25.1*inflation*W_empty.^0.63.*V_max.^1.3;
    C_FlightTestOperations =
687*inflation*W_empty.^0.325.*V_max.^0.822.*Q.^1.21;
    C_ManufacturingMatAndEquip =
6.08*inflation*W_empty.^0.921.*V_max.^0.621.*Q.^0.799;

    %T_SLS = ;
    %M_max = ;
    %T_R = ;
    %C_ProductionEngine = 2306*(0.043*T_SLS+243.3*M_max+0.969*T_R-2228);
    C_ProductionEngine = 436*T_max.^0.8356.*Q.*nE;

    C = C_Engineering + ...
        C_DevelopmentSupport + ...
        C_FlightTestOperations + ...
        C_Tooling + ...
        C_Manufacturing + ...
        C_QualityControl + ...
        C_ManufacturingMatAndEquip + ...
        C_ProductionEngine;
end

function [HR_Tooling, HR_Engineering, HR_Manufacturing, HR_QualityControl]
= hourly_rate(Year)
    HR_Tooling = 2.883*Year-5666;
    HR_Engineering = 2.576*Year-5058;
    HR_Manufacturing = 2.316*Year-4552;
    HR_QualityControl = 2.6*Year-5112;
end

function [H_Tooling, H_Engineering, H_Manufacturing, H_QualityControl] =
hourly(W,S,Q)
    H_Tooling = 5.99*W.^0.777.*S.^0.696.*Q.^0.263;
    H_Engineering = 4.86*W.^0.777.*S.^0.894.*Q.^0.163;
    H_Manufacturing = 7.37*W.^0.82.*S.^0.484.*Q.^0.641;
    H_QualityControl = 0.13*H_Manufacturing;
end

function c = cpi(Y0, M0, Y1, M1)
%
========================================================================
% Copyright: Erik Goetzke, Iowa State University, Department of Aerospace
% Engineering
% Last Modified: 12/8/2014
%
========================================================================
% ================================ INPUTS
================================
% Y0: base year
% M0: base month (1-12 for month selection, outside this range for yearly
average)
```

```
% Y1: current year
% M1: base month (1-12 for month selection, outside this range for yearly
average)
%     ex: Y0 = 1986
%         M0 = 2
%         Y1 = 2012
%         M1 = 13
% =============================== OUTPUTS
=================================
% c: cpi1/cpi0
%     ex: cpi0 = 109.3
%         cpi1 = 229.594
%         c = 2.101
%
=========================================================================
    c0 = cpi_lookup(Y0,M0);
    if(nargin > 2)
        c1 = cpi_lookup(Y1,M1);
        c = c1/c0;
    else
        c = c0;
    end
end
```

note: The function “`cpi_lookup`” is not included here due to its extensive size and it only repeating what is found online from the U.S. Burearu of Labor Statistics.

Aircraft Visualization

```
function geom_to_sldw(x_pos,y_pos,Lambda1,Lambda2)
    c1 = x_pos(6);
    b1 = y_pos(2)*2;

    c2 = x_pos(6) - y_pos(2)*(1/tan(Lambda1) + 1/tan(Lambda2));
    b2 = 2*(y_pos(5)-y_pos(2));

    c3 = c2 - b2/2*(1/tan(Lambda2) + 1/tan(Lambda2));
    b3 = 2*(y_pos(4)-y_pos(5));

    fprintf('Geometry for SolidWorks Model\n')
    fprintf('  c1: %6.2f m\n',c1)
    fprintf('  c2: %6.2f m\n',c2);
    fprintf('  c3: %6.2f m\n',c3)
    fprintf('  b1: %6.2f m\n',b1)
    fprintf('  b3: %6.2f m\n',b3)
    fprintf(' Lam: %6.2f deg\n',90-Lambda1*180/pi)
end
```

note: This function is used to support a SolidWorks part file that generates the configurations shown in this document.