A Property-Based System Design Method with Application to a Targeting System for Small UAVs

by

Brian E. Mihok

B.S. Aerospace Engineering Sciences, University of Colorado (2004)
B.S. Computer Science, University of Colorado (2004)
Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2006

Author ....
Department of Aeronautics and Astronautics
May 26, 2006

Certified by...........
Jeff Miller
Charles Stark Draper Laboratory
Thesis Supervisor

Certified by...........
Brent Appleby
Lecturer in Aeronautics and CSDL Technical Supervisor
Thesis Supervisor

Accepted by ....
Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

A Property-Based System Design Method with Application to a Targeting System for Small UAVs

by

Brian E. Mihok

Submitted to the Department of Aeronautics and Astronautics
on May 26, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The aim of system design is to define an optimal integration of components for the achievement of an overarching objective. As a result, engineering systems often cannot be designed with the disciplines meeting in isolation, but instead require collaboration for a synergism of goals, especially in aviation-based systems where tradeoffs are inherent to the design. Furthermore, defense-related projects require a strict acquisition process that requires companies to submit proposals for contracts. The system design method proposed here is geared towards the proposal stage of design and is aimed at enabling objective, informed design decisions. As such, the method uses the system's properties in a utility function-based evaluation to determine the best alternative. Towards these ends, the method defines criteria critical to the system's evaluation and functions to translate the system's properties related to these criteria into scores. The system's properties are derived from relationships with the components properties and between the components and their environment. As a result, the method translates component properties into system properties, which are then turned into scores. A utility function is used to create a total system utility for the alternative, which serves as the basis for comparison. A Python-based tool was written to facilitate the method, encapsulating the process in a high-level, easily configurable script. The method was demonstrated on the design of a targeting system for small UAVs. Three targeting methods were considered: assuming a flat Earth, using DTED data, and using range data. The evaluation revealed a descending utility order of DTED, Flat Earth, and Range based upon the system's stated requirements. While the Range method produced the most accurate results by far, its unit cost was well beyond the allocated budget, as was its power. DTED data was found to be a beneficial addition to small UAVs. In the evaluation, the method was able to elucidate the key information required to shape the design and thus showed promise.

Thesis Supervisor: Jeff Miller
Title: Charles Stark Draper Laboratory

Thesis Supervisor: Brent Appleby
Title: Lecturer in Aeronautics and CSDL Technical Supervisor

Acknowledgments

First and foremost, I would like to give thanks to God for blessing me with such a wonderful life. I don't deserve even a fraction of what I've been given. As for this thesis, Soli Deo Gloria.

To my wonderful family, I offer my eternal thanks, though the words are not enough. Nobody has ever had a more supportive family. You are the reason that I am here.

To MeLea, the best thing that happened to me while in Boston, thank you for your love and support during such a stressful time. If this is what life is like when we're both in school and stressed out of our minds, I can't wait to experience life with you outside of school!

To Jeff, thank you for many hours and many meetings of collaboration. Perhaps I should have entitled the system design method "Miller's Method!" I'm endless impressed with your Python skills and your ability to get things to work. I enjoyed working with you very much. I hope I didn't jade you of the thesis process to the point were I will be your first and only student. Here's to your advising many more successful theses.

To Brent, one of the primary reasons I came to MIT, thank you for your wisdom and for always finding time in your busy schedule. Thank you for giving all us fellows the freedom to pursue what we wanted and the direction to help us get there.

And of course, to the boys, whom I'll address in alphabetical order by last name (so don't whine about the order...though we all know I always preferred Drew ☺).

To Drew, the most honorable man I've ever met and the one non-Mihok more than any who has taught me about what it means to personify a Christian lifestyle not through lectures, but simply through actions. I am a different person for having met you.

To Jon, three out of six classes together and I wouldn't have it any other way. I wish you all the best in your new life with Kara and pray that the Lord blesses you guys. By the way, you're crazy getting married and turning in a thesis within two days of each other, but it just shows how gracefully you can navigate life.

To Tom "Princess" Krenzke, my Tyler Durden, thanks for probably two pages, single-spaced worth of inside jokes, to the point where we don't really need to even speak to each other any more, at least not in a why that other people can understand. Thanks for being like me and keeping me sane. Here's to countless conversations over Kung Pow Chicken at Mulan, a #41 at Cafe Kiraz, a Desmond Burger at Asgard's, or any other the eating establishments we regularly attended.

To Pete, a man who is as quick with a joke as with an explanation on any topic that I came across, I really enjoyed getting to know you. I appreciate the encouraging words you always freely gave. I admire how you could work when you had to but then could joke when it was time. My work environment will be less enjoyable without your presence.

To Stephen, thank you for your Latex proficiency and your friendship. There's few people I've met that are as naturally talented as you. I enjoyed our adventures with Tom and Drew and will think of them when I think back on my time here at

MIT. Best of luck at Draper and try not to take over for Brent too quickly.

Finally, I would be remiss if I didn't thank Louis Lopez and Mike Terry, without whom 6.555 and 6.825, respectively, would have been so much harder for me, if not impossible. I enjoyed working with you guys so much that the work was almost enjoyable.

Contents

1 Introduction **15**
- 1.1 Background . . . 15
  - 1.1.1 Systems and System Engineering . . . 15
  - 1.1.2 Aviation Systems . . . 18
  - 1.1.3 Problem Motivation . . . 24
- 1.2 Current Methods . . . 27
  - 1.2.1 Matrix-Like Evaluation . . . 28
  - 1.2.2 Multi-Disciplinary Optimization (MDO) . . . 31
  - 1.2.3 Other Methods . . . 39
- 1.3 Problem Statement . . . 51
- 1.4 Thesis Outline . . . 53

2 System Design Approach **55**
- 2.1 Assumptions . . . 55
- 2.2 Methodology . . . 56
  - 2.2.1 Requirements Analysis . . . 57
  - 2.2.2 Select Design Criteria . . . 59
  - 2.2.3 Establish Relationships between Physical Properties and Scores 61
  - 2.2.4 Define the Alternatives . . . 62
  - 2.2.5 Define Subcomponents . . . 64
  - 2.2.6 Establish Relationships Amongst Properties . . . 65
  - 2.2.7 Problem Tradeoffs . . . 66
  - 2.2.8 Optimize Criteria . . . 68
- 2.3 Deliverables . . . 68

3 Application Background **71**
- 3.1 Application Motivation . . . 71
- 3.2 Major Current Problems with Small UAVs . . . 72
- 3.3 Application Problem Statement . . . 73
- 3.4 General Small UAV Requirements . . . 73

4 Small UAV Targeting System Analysis **79**
- 4.1 Requirements Analysis . . . 79
  - 4.1.1 Methodology . . . 79
  - 4.1.2 Application . . . 81

4.2 Select Design Criteria . . . 91
  4.2.1 Methodology . . . 91
  4.2.2 Application . . . 93
4.3 Establish Relationships between Physical Properties and Scores . . . 95
  4.3.1 Methodology . . . 95
  4.3.2 Application . . . 103
4.4 Define Alternatives . . . 113
  4.4.1 Methodology . . . 113
  4.4.2 Application . . . 118
4.5 Define Subcomponents . . . 128
  4.5.1 Methodology . . . 128
  4.5.2 Application . . . 130
4.6 Establish Relationships Amongst Properties . . . 135
  4.6.1 Methodology . . . 135
  4.6.2 Application . . . 139
4.7 Problem Tradeoffs . . . 148
  4.7.1 Methodology . . . 148
  4.7.2 Application . . . 154
4.8 Optimize Criteria . . . 175
  4.8.1 Methodology . . . 175
  4.8.2 Application . . . 176

5 Conclusions and Future Work **195**
5.1 System Design Method Conclusions . . . 195
5.2 Targeting System Conclusions . . . 196
5.3 Future Work . . . 197

A Product Specification Sheets **199**
A.1 Optical Cameras . . . 199
A.2 Altimeters . . . 207
A.3 Autopilots . . . 210
A.4 Laser Rangefinders . . . 220

B Complete Results of Simulation **225**

List of Figures

1-1 Growth and diversification of requirements for aerospace vehicles (Taken from [18]). 19
1-2 Technological activities and interactions within the system life-cycle process (Taken from [14]). 21
1-3 System acquisition process activities and interactions over the life cycle (Taken from [14]). 22
1-4 Generic life cycle cost versus time (Taken from [14]). 25
1-5 Life cycle cost versus time for a specific project (Taken from [18]). . . . 26
1-6 Traditional design freedom and knowledge vs time, along with the relative influence of each design consideration at the various stages (Taken from [18]). 26
1-7 Revisiting the design freedom and knowledge vs time plot, but this time with the goal for an improved system designed graphed as well (Taken from [18]). 27
1-8 Flowchart of the matrix-like evaluation method. 28
1-9 Aspect ratio of an aircraft given three constraints (Taken from [25]). . 32
1-10 Diagram of MDO framework (Taken from [26]). 33
1-11 Range of design objectives for MDO (Taken from [25]). 34
1-12 Example of the pareto front in MDO (Taken from [25]). 34
1-13 Example of a curve used in a graphical optimization (Taken from [25]). 35
1-14 A sample problem statement, labeled with its components (Taken from [25]). 36
1-15 Sample formulation of a design problem in NLP (Taken from [26]). . . 36
1-16 Example of processor performance vs cost in 1996 (Taken from [16]). 40
1-17 Example of a scatter plot used in the cost effectiveness comparison of alternatives (Taken from [31]). 45
1-18 OPN design process and tools (Taken from [17]). 51
1-19 OPN modeling and metrics (Taken from [17]). 52
1-20 Defense acquisitions process (Taken from [23]). 53

2-1 Flowchart of the proposed system design method. 56
2-2 Classification of different navigation methods. 63
2-3 Components of a navigation solution and how to obtain the components. 64

3-1 A picture of AeroVironement's Raven and DragonEye, used by the Army and Marine Corps, respectively (Taken from [1]). 74

4-1 Tree of the customer's motivations for small UAVs. 82
4-2 Scenario of a vehicle detecting a man-sized object. 90
4-3 The visual cone from the aircraft to the soldier. 91
4-4 Hierarchy of discrete bound options. 96
4-5 Fitting a polynomial to the data using Matlab's *polyfit* function on the provided example for the first four orders. 99
4-6 Fitting a cubic spline to the data using Matlab's *spline* function. . . . 101
4-7 An adjustment to the original spline by adding the point $(0.4, 0.2)$ . . 102
4-8 The continuous function that defines the translation of the mass of the components into a score. 105
4-9 The continuous function that defines the translation of the volume of the components into a score. 107
4-10 The continuous function that defines the translation of the power consumption of the components into a score. 108
4-11 The continuous function that defines the translation of the unit cost of the components into a score. 110
4-12 The continuous function that defines the translation of the unit cost of the components into a score. 111
4-13 The continuous function that defines the translation of the robustness of the solution into a score. 112
4-14 The continuous function that defines the translation of the latency of the solution into a score. 114
4-15 Hierarchy of targeting solutions 119
4-16 Top-view (XY view) of an encounter. ψ is the vehicle's heading. . . . 120
4-17 Example of Flat Earth method. The range (hypotenuse) and horizontal leg are in the xy direction of the camera's LOS vector, which is the red vector (heading vector) in Figure 4-16. 121
4-18 Example of changing terrain while using Flat Earth method. Again, red vectors along the heading vector, as in Figure 4-16. 122
4-19 Example of DTED providing the shape of a given terrain at different levels of fidelity. 123
4-20 Example of how multiple looks at a target can converge to a single location (Taken from [20]) 125
4-21 Example of the DPPDB method (Taken from [20]) 126
4-22 Example of a target and some local drop points required for determining the target's location (Taken from [20]) 127
4-23 Partial And/Or decomposition of a computer. 131
4-24 The geometry of a Flat Earth targeting encounter revisited, with the appropriate dimensions labeled with the corresponding terms to match the equations in the text. 142
4-25 Example of an encounter where the target's altitude is above the vehicle's reference altitude. 144
4-26 Sample sensitivity analysis of how error in height affects TLE. 150
4-27 Example of Class of Components and Individual Components Trades 152

4-28 Sources of error for a camera-based targeting system with a relative-to-self method 155
4-29 Sensitivity analysis for the pitch error. 159
4-30 Demonstration of why the substantial difference exists between the methods for pitch error. 160
4-31 Sensitivity analysis for the yaw error. 162
4-32 Sensitivity analysis for the yaw error graphed with a polar plot. . . . 164
4-33 Sensitivity analysis for the yaw error over all possible values graphed in a polar plot. 165
4-34 Sensitivity analysis for the height error. 166
4-35 Demonstration of the geometry of the height error. 168
4-36 Sensitivity analysis for the XY position error. 169
4-37 Sensitivity analysis for the range error. 171
4-38 Sensitivity analysis for the target altitude error. 172
4-39 Demonstration of the geometry of the target altitude error. 173
4-40 Flow chart of the Python scripts used to enumerate and evaluate the possible alternatives. 178
4-41 Flow chart of the Python scripts used to enumerate and evaluate the possible alternatives. 179
4-42 Basic shape of the probability of the target altitude as used in the robustness evaluation. 183
4-43 Plot of the system's TLE vs. its total mass. 187
4-44 Plot of the system's TLE vs. its total power. 188
4-45 Plot of the system's TLE vs. its total size. 189
4-46 Plot of the system's TLE vs. its total unit cost. 190
4-47 Plot of the system's total mass vs. its total unit cost. 191
4-48 Plot of the system's total power vs. its total unit cost. 192
4-49 Plot of the system's total size vs. its total unit cost. 193

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

1.1 An example of a matrix-like evaluation 29
1.2 SWOT Matrix and associated strategies (Taken from [8]). 48
1.3 OPN language primitives (Taken from [17]). 50

3.1 Evolution of CEP over time (Taken from [29]) 76

4.1 Sample small UAV targeting requirements 88
4.2 The number of resolvable light-dark cycles required across a target's critical dimension for various discrimination tasks (Taken from [30]) . 89
4.3 Standard target sizes as defined by the Army's Night Vision Lab (Taken from [30]) 90
4.4 Camera models selected with their associated properties. 136
4.5 Autopilot models selected with their associated properties. 136
4.6 Altimeter models selected with their associated properties. 136
4.7 Laser Rangefinder models selected with their associated properties. . 136
4.8 Error level taken from the components listed in Tables 4.5 - 4.7 . . . 148
4.9 Upper bounds of the error for both the threshold and objective levels evaluated with the target altitude on the reference line and a 2° downhill slope
(UB = Upper Bound). 174
4.10 Upper bounds of the error for both the threshold and objective levels evaluated with the target altitude $50ft$ above the reference line and a 2° downhill slope (UB = Upper Bound, DNM = does not ever meet the requirement). 175
4.11 Table of the percent contribution to the total cost of each of the design criteria. 179
4.12 Average error and standard deviation for the three methods. 184

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Background

1.1.1 Systems and System Engineering

System Definition and Elements

A system is:

- "An assemblage or combination of elements or parts forming a complex or unitary whole, such as a river system or transportation system
- Any assemblage or set of correlated members, such as a system of currency
- An ordered and comprehensive assemblage of facts, principles, or doctrines in a particular field of knowledge or thought, such as a system of philosophy
- A coordinated body of methods or a complex scheme or plan of procedure, such as a system of organization an management, or any regular or special method or plan of procedure, such as a system of marking, numbering, or measuring [14]."

While this definition is broad enough to encapsulate everything from a transportation system to a system of measurement, not every collection of items, facts, methods, or procedures is a system [14]. For example, while the items in a room have relationships with each other, they lack the unity, functional relationship, and useful purpose that is required to be a system [14].

The elements of a system are *components*, *attributes*, and *relationships* [14]. Components are the operating parts of a system, consisting of *input*, *process*, and *output* [14]. Attributes are the properties or discernible manifestations of the components. Relationships are the links between the components and the attributes [14]. Using this terminology, the definition of a system is restated as a set of interrelated components working together toward some common objective or purpose, with the following properties [14]:

1. The properties and behavior of each component of the set has an effect on the properties and behavior of the set as a whole.
2. The properties and behavior of each component of the set depends on the properties and behavior of at least one other component in the set.
3. Each possible subset of components has the two components listed previously; the component cannot be divided into independent subsets.

These properties elucidate the difficulty of finding optimal designs given the combinatorially increasing nature of system alternatives as the system grows in complexity.

Subsystems

The properties established in [14] ensure that the system has some characteristic or behavior that is unique from its individual components. That is, the properties ensure that the system is more than the sum of its component parts [14]. However, any component of a system may themselves be systems, termed *subsystems*, and every system may itself be a component of a larger system, thereby creating a hierarchy of systems. An illustrative example is an air transportation system, which has subsystems that include the aircraft, terminals, ground support equipment, and controls [14]. The system and subsystem further can be decomposed to have components of equipment, people, and software. However, this designation of system, subsystem, and components is relative, because the system at one level in the hierarchy could be considered a subsystem or component of another.

System Boundaries and Constraints

In addition to its components and hierarchy, an important part of a system formalization is its boundaries. Everything outside the boundaries of a system is considered to be the *environment* [14]. The material, energy, and/or information that passes from the environment to the system is *input*, the material, energy, and/or information that passes from the system to the environment is *output*, and the combination of the two is the system's *throughput* [14]. From before, a system is roughly all the components, attributes, and relationships needed to accomplish an objective. The objective provides the purpose for which all the components, attributes, and relationships are organized while constraints on the system limit its operation and define the boundary within which it is intended to operate [14]. Continuing down the hierarchy, the system then places boundaries and constraints on its subsystems.

The systems viewpoint, as described by [14], looks at the system from the top down rather than from the bottom up. They originally treat the system as a black box and define how it interacts with its environment. Then, they decompose the black box into smaller black boxes in the form of subsystems that work together to achieve a common objective. The decomposition continues until no component remains undivided that could be considered a system.

With the system makeup and interaction with the environment defined, the classification of systems will be discussed next. The classification of systems will then allow for the definition of a subset of systems on which the remainder of thesis will focus. Also, the classification will allow for the definition and examination of principles used in the creation of systems.

Classification of Systems

The classification of systems is done for the human convenience and to illustrate the various dichotomies that exist in systems. The first such dichotomy is related to the origin of the systems, with *natural systems* being those that came into being through natural processes and *human-made systems* being those created by humans [14]. An example of a natural system is the Nile River basin ecosystem and with the Aswan High Dam serving as a human-made system example.

The second classification scheme is whether the system is a *physical system* or a *conceptual system* [14]. Physical systems are those that are manifested in physical form whereas conceptual systems have symbols represent the attributes of components [14]. An example of a physical system is the F/A-22 Raptor, whereas conceptual system example is the flight control software onboard the Raptor.

The third classification scheme proposed by [14] is whether the system is *static* or *dynamic*. Static systems have no activity, such as a bridge, whereas dynamic systems combine components with activity [14]. An example of a dynamic system is a school, as it combines a building, students, books, and curricula.

Finally, a system can be either *open* or *closed* [14]. A closed system is one that interacts minimally with its environment, such as a chemical reaction reaching equilibrium in a closed vessel [14]. By contrast, an open system allows information, energy, and matter to cross its boundaries, such as a plant or a business unit [14].

System Engineering Characteristics

With a system defined, bounded, and classified, the process of engineering a system will now be examined. Despite the maturity of the system engineering field, a widely accepted definition of system engineering does not exist. Still, the definitions do have common characteristics. Four such common characteristics are [14]:

- a top-down approach that views the system as a whole.
- a life-cycle orientation that addresses all phases to include system design and development, production and/or construction, distribution, operation, maintenance and support, retirement, phase-out, and disposal.
- a better and more complete effort is required regarding the initial definition of system requirements, relating to specific design criteria, and the follow-on analysis effort to ensure the effectiveness of early decision making in the design process
- an interdisciplinary or team approach throughout the system design and development process to ensure that all design objectives are addressed in an effective and efficient manner.

Summary and Transition

Given the interdisciplinary, complex nature of modern systems, systems design is often a difficult task. However, the hierarchial formalization in this section lends itself to computer aided modeling at a selectable level of detail. As will be seen later, some methods approach this problem at a high level and, as a result, tend to be arbitrary in nature. Others encapsulate all levels of detail in a rigorous analysis and thus tend to be difficult to use and time-consuming to set up. A need exists for a middle ground between these two extremes. The method proposed in this thesis is aimed at filling this need. It defines the components of the system, establishes their attributes, and then defines how these translate into the attributes of the system. The system's attributes are then used in a cost function-based evaluation. While the system design method proposed later in the thesis can most likely be applied to any human-made system, it was created to be best used on a human-made, physical, dynamic, open system.

1.1.2 Aviation Systems

The specific example selected to demonstrate the method was soldier-portable Unmanned Aerial Vehicles (UAVs). UAVs can be classified in the broader aviation systems class. While the specifics of small UAVs will be discussed in greater depth later both later in this chapter and in the subsequent chapters, the next few sections will discuss characteristics of aviation systems in general. The discussion will then narrow in on defense-related UAVs.

A Brief History of Aviation Design

In order to best understand the current characteristic of an aviation design, it is important to understand the evolution of the aviation design process from its inception. According to Schrage et al. [18], during the pioneering years of aviation, the designer was one of the central figures in the creation of the aircraft. He/she served as the jack-of-all-trades that was responsible for not only the design, but also as the main resource in aerodynamics, structures, materials, propulsion, manufacturing, and was usually the test pilot [18]. The knowledge necessary to design an aircraft was very practical, reality-based information and was capable of being stored in the minds of some of the titans of the day, such as the Wright Brothers, Glen L. Martin, Breguet, DeHavailland, Fokker, Heinkel, and Sikorsky [18].

The early 1930s saw the rise of specialization in the major disciplines within aviation [18]. Evaluation in wind tunnel tests in aerodynamics, thin shell analysis in structures, processing and forming techniques in production, and thermodynamic efficiencies in propulsion all began to contribute to the design of an aircraft. This specialization made it nearly impossible for one person to stay abreast of all the latest developments in each field and as a result, the lead designer evolved into what would today be called the systems engineer [18].

The late 1950s brought about a gradual change where the importance and prestige of analytical specialists soared [18]. This was partly due to the impetus given to

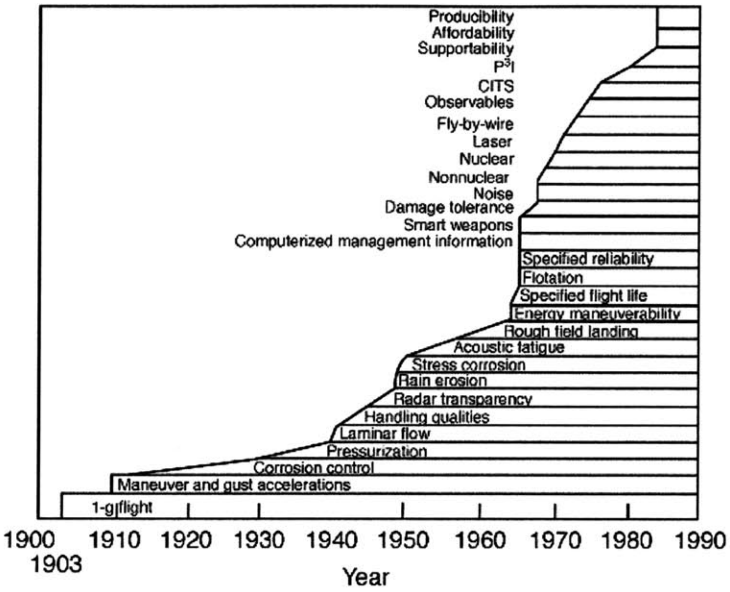


Figure 1-1: Growth and diversification of requirements for aerospace vehicles (Taken from [18]).

missiles, rockets, and spacecraft that were all one-of-a-kind, single-use systems that used a new set of design guidelines and partly due to the demands of the military, who were striving for maximum performance [18]. Sputnik, the Apollo program, and the American-Soviet space race had a large part in the transfer of prestige. As a result, the best minds were attracted to research and development in order to expand the limits of scientific knowledge and push for ever increasing performance, resulting in the design of an aircraft becoming more of the implementation of novel ideas produced by others than being a titan, like the early days, or systems engineer, like in the previous era [18].

The 1970's, marked by a major aviation slump, brought about two major changes in aircraft design. First, computer-aided design was developed, freeing the designer from some of the most monotonous, menial portions of the design process [18]. Secondly, the procurement policy of the military experienced a major overhaul [18]. Instead of always trying to achieve the maximum performance, the new policies desired a balance amongst goals that included life-cycle cost, reliability, mantainability, vulnerability, and others in addition to performance [18]. This trend manifested itself in the diversification of design requirements for advanced aeronautical vehicles, as shown in Figure 1-1. Schrage et al. [18] state that the experience of the 1960s had shown that for military aircraft, the cost of the final increment of performance usually was excessive in terms of other characteristics. They also state that the airlines realized that the entire system must be optimized when meticulous cost accounting showed possible savings due to improved reliability and maintainability.

The emphasis on requirements balanced between performance, life-cycle cost, and the various -ilities (maintainability, vulnerability, reliability, etc.) continues through until today. Current requirements documents for aviation vehicles, whether they be inhabited or uninhabited, all contain requirements for the availability of the system, the mean time to repair, and other non-performance-related criteria. An example of this balanced requirements-based approach in action is the F-35 Joint Strike Fighter (JSF). The Air Force, Marines, and Navy were each looking to develop the next generation of aircraft to replace their aging systems (F-15, F-16 for Air Force, F-14 for the Navy, AV-8 Harrier for the Marines). Instead of each developing their own aircraft, the three services requested one parent aircraft that could be modified for the specific needs of the three services. In doing so, the services acknowledged that while the performance of each variant might not be optimal, the benefits of combining the vehicle upgrades, such as the amount of money saved and having common parts to all aircraft, would supersede the diminished performance in terms of importance. This point was reinforced at the announcement of the winner of the contract at an October 26, 2001 press conference when Jim Roche, Secretary of the Air Force at the time, said, "The Lockheed Martin team is the winner on a best-value basis [27]."

With the evolution of the aviation design process established and the current focus of aviation system briefly mentioned, the typical design process used to design aviation systems will be examined.

Typical Design Process

According to Schrage et al [18], design is a hierarchial, evolutionary process depicted as phases from conceptual to preliminary to detail design and then manufacturing and production. Blanchard and Fabrycky represent this hierarchial structure graphically, as shown in Figures 1-2 and 1-3.

Aircraft design synthesis and optimization of a conceptual system is typically based on achieving a fuel balance and a minimum weight configuration through parametric variation of a few critical design parameter, such as wing loading and aspect ratio [18]. Since aerodynamics and propulsion are the critical disciplines to achieving fuel balance and vehicle performance, they are emphasized during the conceptual stage of design [18]. As the initial configuration is frozen and the aircraft progresses to the preliminary design phase, hardware design considerations begin to dominate and the role of the structures team becomes dominant [18]. When the aircraft moves into the detailed design phase and flight-worthiness is a primary concern, the role of the controls discipline increases in order to improve the overall handling and flight dynamics [18].

Discipline Interaction

Inherently, aerospace vehicles are engineering systems whose performance depends on parts designed under many different disciplines and whose behavior are governed by a large set of coupled equations [18]. These coupled equations are typically divided into major disciplines, such as the ones mentioned in the previous section, so that experts can address their specialties. In doing so, coupling between divisions is either retained or neglected based upon the design team's judgement of what is

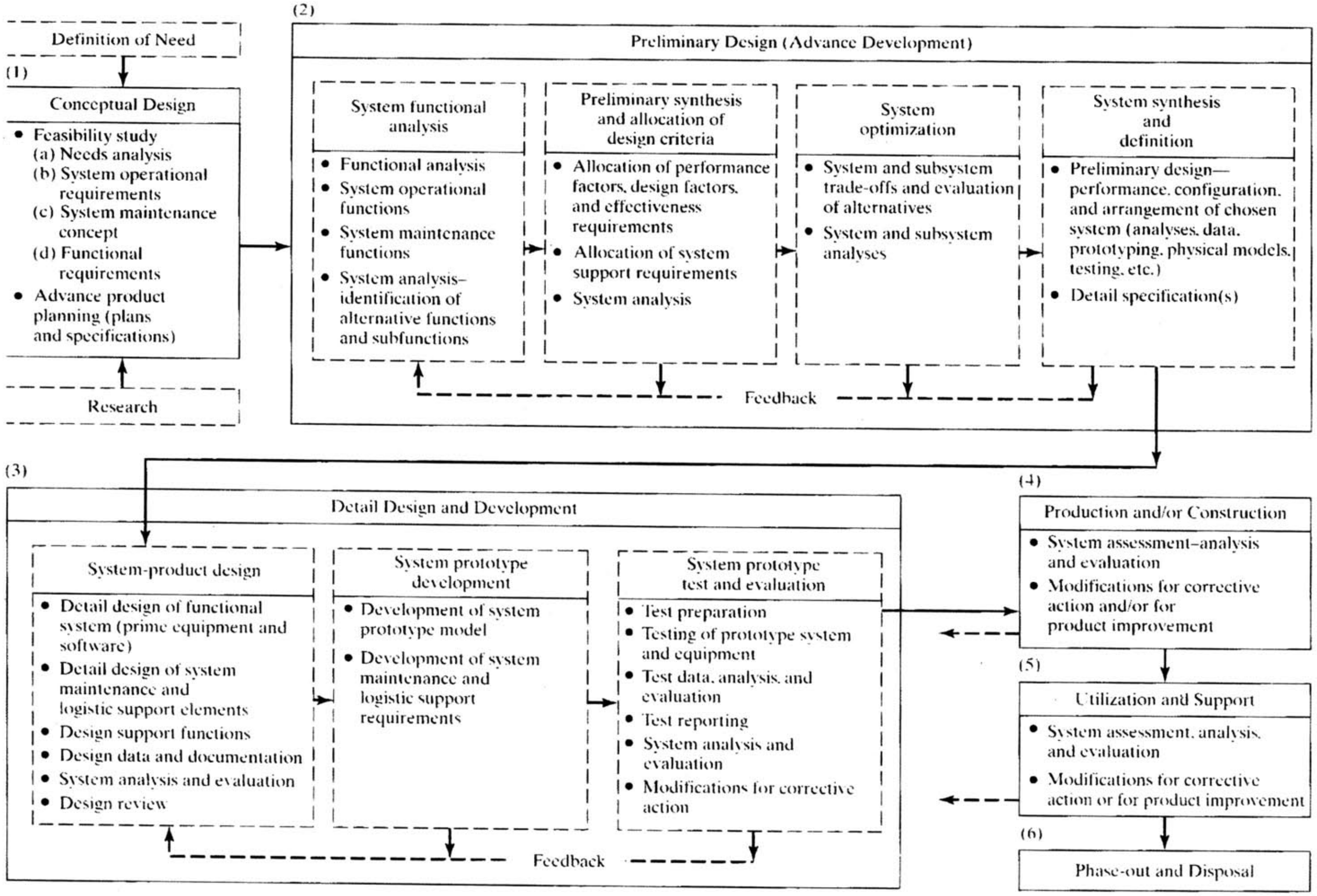


Figure 1-2: Technological activities and interactions within the system life-cycle process (Taken from [14]).

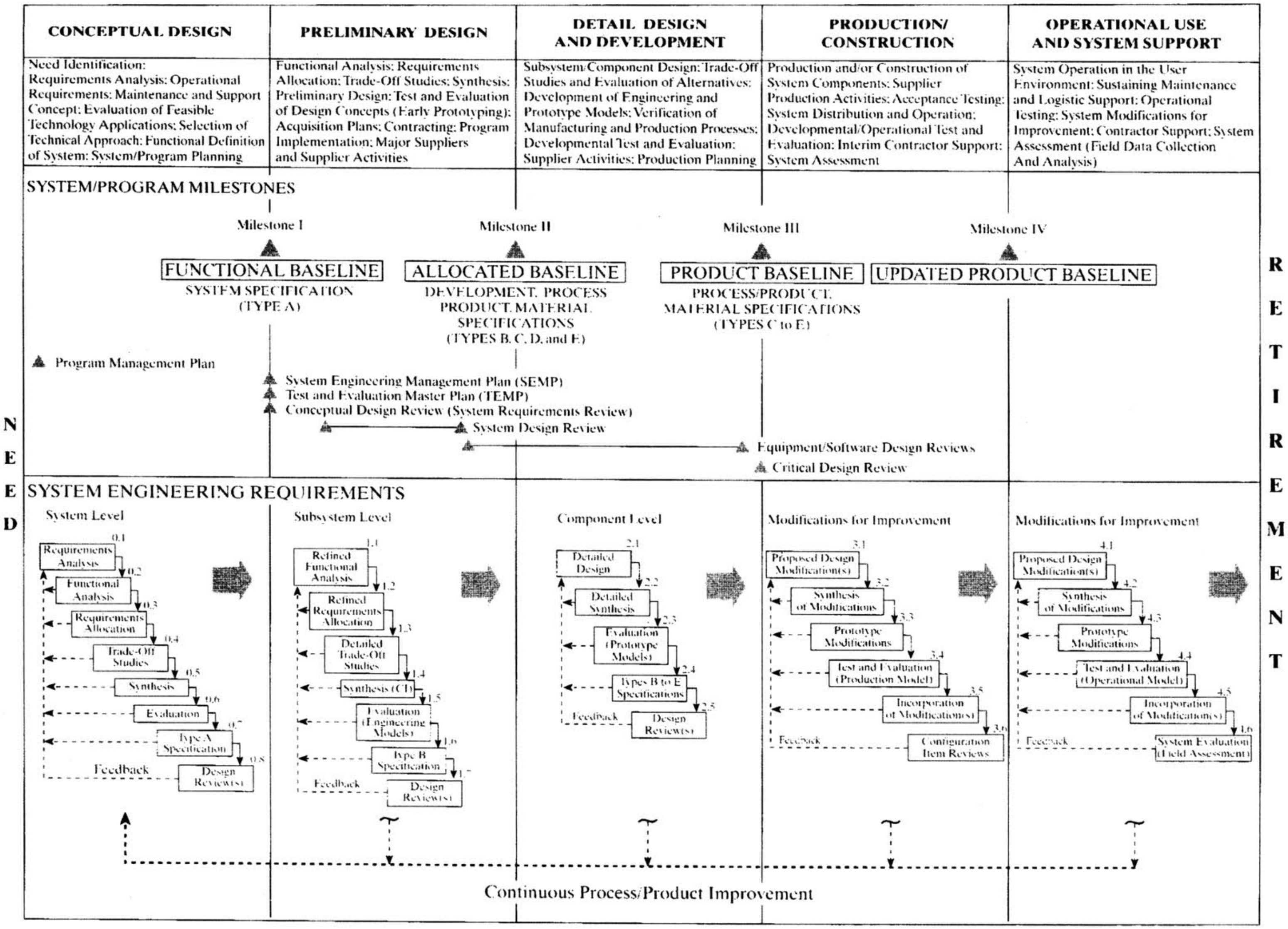


Figure 1-3: System acquisition process activities and interactions over the life cycle (Taken from [14]).

important, the assumptions made about the vehicle or mission, or their skill level in each particular area. The reduction in coupling reduces the design burden but at the sacrifice of fidelity. In general, as the complexity of the vehicle increases, the more attention needs to be paid to the coupling [18]. If the coupling mentioned above were not present, a superposition of components created by separate teams could be combined. However, since the coupling is strong, an interdisciplinary approach to design is required. This matches the previous definition of a system where the properties and the behaviors of each component (and the overall system) depended upon all of the other components.

An example of the inherently conflicting tradeoffs in aviation design and the interplay of multiple disciplines can be seen through the desire to increase a vehicle's range. One possible solution is to increase the aspect ratio of the wing. However, this leads to a greater amount of induced drag, thereby requiring a bigger engine in order to reach the same maximum speed requirement. The increase in the engine size requires that the plane be stronger to handle the extra thrust. Between the increased engine size and the stronger material, the aircraft becomes heavier. The increased weight has a detrimental affect on the range of the aircraft, so the result of the increase in aspect ratio made to increase range may actually have an opposite effect than intended.

Defense-Related UAVs

The Aviation Systems section has thus far examined the evolution of the design of these systems, as well as their typical design process and the interaction of disciplines within the design. The discussion will now narrow from aviation systems in general to specifically those that are defense related and uninhabited.

In addition to typical aviation requirements, defense-related UAV systems also have stringent requirements placed upon them in certain unique areas, including the system's operational environment and ease of use. In order to be considered a reliable, functional tool for soldiers to use in battle, the system needs to be able to function anywhere that a soldier is deployed. Since soldiers are deployed in the worst conditions on Earth, the UAVs that support them are also subjected to these extreme conditions. Nowhere is this more true than for the systems in support of special forces troops. The operational environment could be a combination of blistering heat, severe cold, below sea level, on top of mountains, in precipitation, in high winds, or any other force of nature. The system could be required to be submerged in water for multiple hours and then operate just like normal. However, simply functioning is not enough. The battlefield provides soldiers with a tremendous amount of information and events to process, a number of tasks to perform, and very little time in which to do it all. The system needs to be easy to use, requiring a minimum amount of the soldier's attention while still providing a maximum amount of functionality.

To these ends, defense-related UAVs are becoming increasingly complex and are being asked to perform more and more tasks. Examining both the current and desired high-level functionality of the navigation subsystem is illustrative of the desired increase in complexity and functionality. Current navigation methods range from completely teleoperated flight to following preprogrammed GPS waypoints, with limited amounts of freedom mixed in. However, the future of UAV navigation appears to hold a push towards full flight and trajectory automation. By increasing the level of autonomy of the vehicle, the soldier could concentrate on other activities and not have to solely be dedicated to flying the vehicle. It is also foreseeable that the aircraft could be required to schedule tasks requested from the ground in a fashion similar to what is attempted by current interplanetary probes. The increased level of autonomy in flight control, trajectory planning, and task scheduling will lead to the ability to control multiple vehicles with the same ground equipment.

Proposals In addition to supplementing the requirements of civilian aviation projects with the needs of the soldiers and the battlefield, defense-related UAVs also have the added feature of being proposal-driven. In this environment, the customer (usually one of the branches of the military, DARPA, Special Forces Command, or the Department of Defense) produces a request for proposals, along with an associated requirements document describing what the system needs to do. Companies in the aerospace defense industry then typically have a short time to produce a proposal for their solution to the particular system requirements. These proposals are text documents describing the company's solution and are typically limited to a set number of pages. In order to create a competitive proposal, companies need to motivate why their proposal is the best of all the possible alternatives in a concise description. They typically do this by giving a high-level description of the solution with as much technical analysis as the limited time frame will support. This analysis typically has pieces that range from well-established methods and numbers, to methods and numbers based upon educated assumptions, ultimately to methods and numbers that are essentially a best "engineering guess." The success of proposals can literally be the difference between life and death of companies or business units of large corporations.

1.1.3 Problem Motivation

As has been shown in the previous sections, large engineering systems are interdisciplinary and complex. This is especially true of aviation-related systems. The design process of aviation systems has evolved to the point where performance is not the only consideration, but instead life-cycle cost, maintainability, reliability, and other factors are considered important as well. Using the traditional evolutionary, hierarchial design approach for aviation systems, the early stages of design have a significant impact on the final cost and performance of the resulting vehicle. This is seen generically in Figure 1-4, and again specifically for a missile system created by Boeing in Figure 1-5. Figure 1-4 shows that by the end of the conceptual design stage, decisions have been made that dictate over 50% of the final design. Furthermore, Figure 1-5 shows a commitment of 70% of the final dollars spent on the project occurred by the end of the conceptual design. Similarly, Figure 1-6, when compared to Figure 1-4, shows that [14] and [18] also closely agree about how both design freedom and knowledge about the design vary as the project progresses. Intuitively, both figures show that

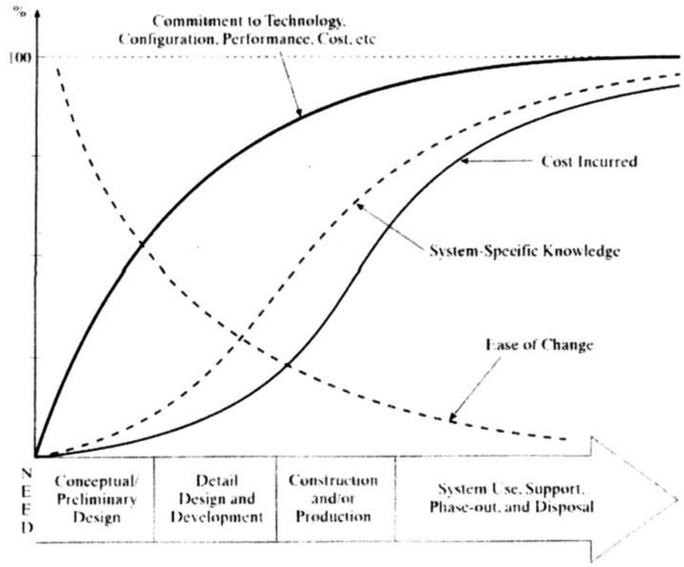


Figure 1-4: Generic life cycle cost versus time (Taken from [14]).

as the project progresses, more is known about the design and as decisions are being made, there is less freedom to make changes.

Figure 1-6 also shows the influence of each the major design consideration at the various stages of the projects. This is done by having horizontal bars whose lengths correspond to the level of importance of a particular design consideration for a given stage. The design criteria are labeled to the right of the bars. For example, the first row of bars correspond to aerodynamics, the second to propulsion, etc. As discussed earlier, aerodynamics and propulsion are the most important disciplines during the conceptual design. Once the configuration is frozen and the project progresses to the preliminary design phase, hardware begins to dominate and structural considerations increase in importance. Once the system has matured to the detailed design phase, the flight handling becomes important, leading to an increase in the importance of controls. Also, the manufacturing, cost, and similar such disciplines become much more significant.

The problem with the traditional aviation system design method is that the reduction in freedom in the later stages of development severely limits what can be done within the realms of controls, manufacturing, cost, and other criteria. If a major problem is detected in one of these areas in the detailed design phase, the cost to make a change in terms of money and time is quite significant. An optimal system design would thus take into account all the major disciplines with equal weighting during the conceptual design when the cost to make a change to the design is insignificant. Schrage et al. [18] propose the adjusted model shown in Figure 1-7. When contrasted against Figure 1-6, the model in Figure 1-7 has a longer preliminary design phase and all the major disciplines other than cost are equally weighted in this stage, thereby making the knowledge about the design available sooner.

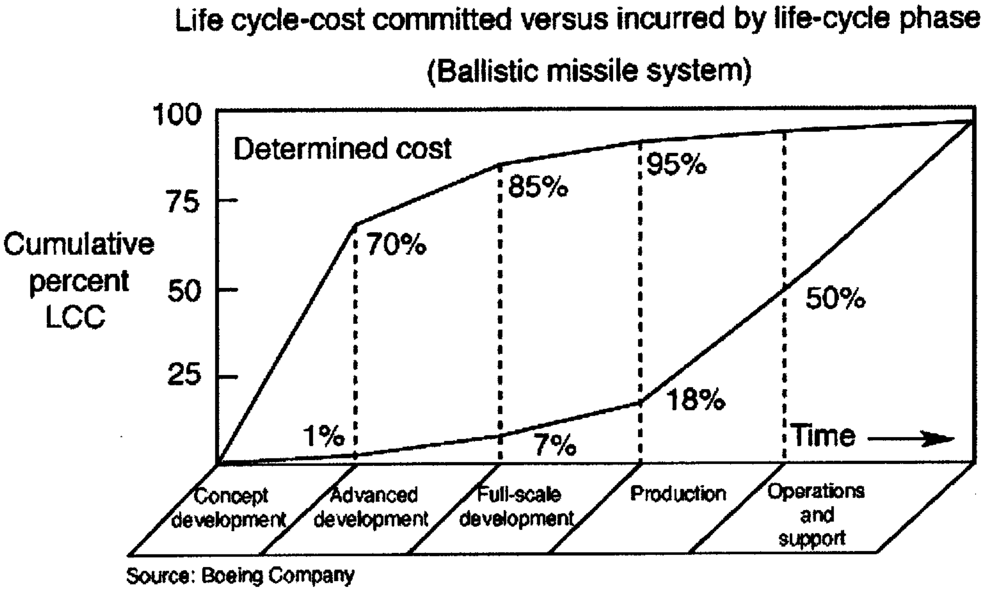


Figure 1-5: Life cycle cost versus time for a specific project (Taken from [18]).

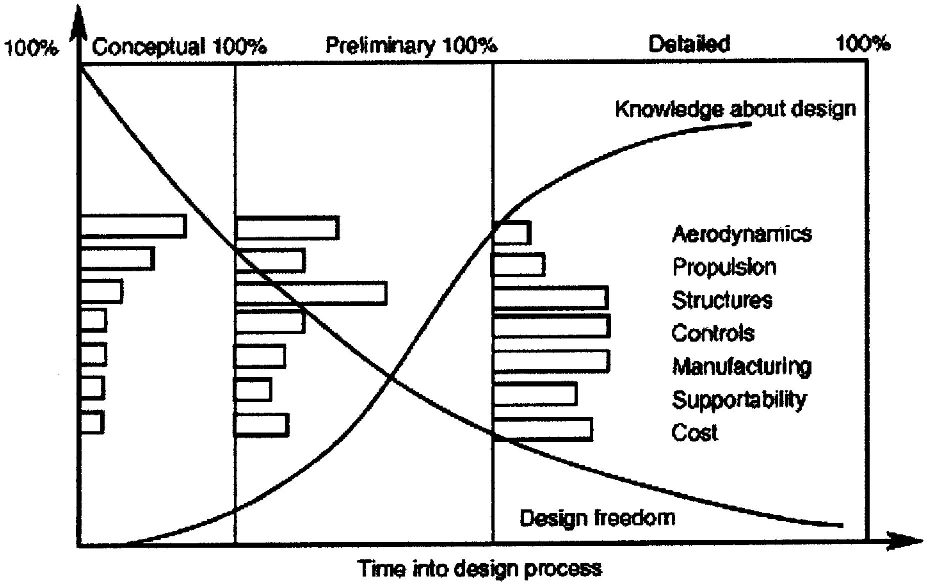


Figure 1-6: Traditional design freedom and knowledge vs time, along with the relative influence of each design consideration at the various stages (Taken from [18]).

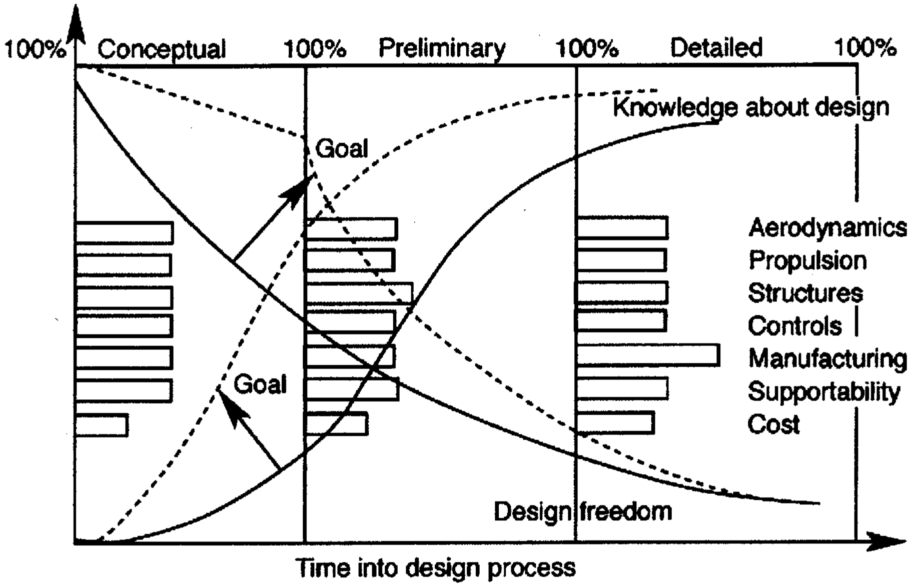


Figure 1-7: Revisiting the design freedom and knowledge vs time plot, but this time with the goal for an improved system designed graphed as well (Taken from [18]).

Additionally, defense-related aviation systems are proposal-driven. While the proposed model could shift the paradigm of how aviation systems are designed, it would make sense to simultaneously develop methods to aide in the creation of successful proposals. Doing one without the other makes little sense. If the design method were improved but proposals were not addressed, a corporation would have a lower likelihood of actually getting the contract to implement the new design method. Similarly, if the proposal creation was improved without altering the underlying system design, the result would be a suboptimal design and the company would be less likely to win contracts in the future. Thus, a need exists for an aviation system design method that incorporates knowledge of all disciplines at the beginning so as to optimize the design while also lending itself well to the proposal-driven nature of the defense-related aerospace industry. Good proposals have objective design decisions based upon sound engineering principles, as well as the details that justify these decisions. The method therefore needs to facilitate the objective decisions and provide the designers with the necessary details so that the decisions can be made.

1.2 Current Methods

With the nature of defense-related aviation system design and acquisition established, the next section will discuss methods that are typically used by engineers and program managers when attempting to design such a system. After the discussion of the current methods, the chapter will conclude with the problem statement for the thesis and an outline of the remainder of the text.

1.2.1 Matrix-Like Evaluation

The first method considered will be referred to as a matrix-like evaluation. In this method, the user defines criteria by which to rate the system. These criteria are typically related to the requirements. Examples could include monetary cost, technology readiness level, and mass. The criteria are next assigned weights. The weights serve to quantify the relative importance of the evaluating criteria to each other. For instance, if a system has four criteria but one is significantly more important than the other three, then the weights could be 5, 1, 1, and 1 for the four criteria, with the 5 being assigned to the most important item. In doing so, the evaluation is essentially divided into eight parts, with the most important criteria holding 5 of the eight parts (62.5%) and the other criteria holding 12.5% each. Once the weights are defined, scores are assigned to the criteria for each solution. The scores are based upon an arbitrary numeric scale such as 1 to 10. The assigned scores are then multiplied by the weights to give the composite score for each possible solution. The best alternative is then the solution that has the highest composite score. In essence, this method is simply a weighted summation of the scores of the individual criteria. A flowchart of the matrix-like evaluation process is shown in Figure 1-8.

Example

This method is best seen through an example and is best visualized through a spreadsheet. For an example, let's consider a scenario where a corporation is working on a system and they need a solution for one of the components of the system. Their options are either to buy a commercial off the shelf (COTS) product or to make it themselves. The corporation decides to evaluate the options based upon five criteria: (1) the ultimate monetary cost of the component, (2) the time to completion, (3) the amount of control they have over the design of the component and its interface with other devices, (4) the amount of support they would receive if something went wrong with the component, and (5) the performance of the component. Of these five criteria, time to completion and control over the design and interface are considered significantly more important than the others. Of the remaining three criteria, performance is much more important than the other two, followed by cost, and then finally support. As a result, the weights assigned are 10 to time and control, 5 to performance, 3 to cost, and 1 to support.

Next comes the assignment of values to the criteria. Let's first consider the option of buying the COTS product. The rating scale used for all of the criteria will be 1 to 10, with 1 being the worst score and 10 being a perfect score. Let's assume that the

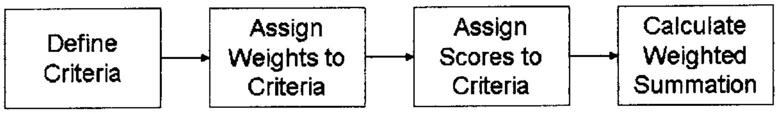


Figure 1-8: Flowchart of the matrix-like evaluation method.

COTS product is expensive. As a result, the buying option receives a 2. However, since it is immediately available, the score for time to complete is a 10. The fact that the component is a finished product for sale means that the corporation has no input into the design or the interface, so the control criteria receives a 1. Next, let's assume the maker of the COTS product provides excellent technical support, thereby producing a score of 9 for support. Finally, let's say the component has a performance score of 8. These numbers are shown in the first row of Table 1.1.

Table 1.1: An example of a matrix-like evaluation

	Cost	**Time**	**Control**	**Support**	**Performance**
Buy	2	10	1	9	8
Make	7	1	10	1	7

Next is the evaluation for the corporation's option of making the component themselves. Let's assume the corporation believes that they could build the component at a fairly inexpensive price, yielding an evaluation of 7 for cost. However, it would take a considerable amount of time, giving a score of 1. Of course, since the corporation would be designing the component from scratch, they would have complete control of its design and interface with other parts, so the control is a 10. Yet, if something goes wrong, there is nobody to turn to for help, making the support criteria a 1. Finally, let's assume that the performance score would be a 7. These numbers are shown in the second row of Table 1.1.

The final step is to calculate the composite score for each option. This simply requires multiplying the individual scores by their associated weights and then summing the results. For the buying option, this yields an evaluation of $2 * 3 + 10 * 10 + 1 * 10 + 9 * 1 + 8 * 5 = 165$. Similarly, for the option of building their own component, the composite score becomes $7 * 3 + 1 * 10 + 10 * 10 + 1 * 1 + 7 * 5 = 167$. From this calculation, the option with the highest score is the option to make their own component. As a result, according to this evaluation, the corporation should build their own component.

Pros and Cons

Pros The two main benefits of this method of analysis are how quick and easy it is to perform. In simple cases, the entire analysis, from creation of the criteria to the calculation of the composite scores can take a matter of minutes. Another advantage is how well it can be represented in a spreadsheet. Most people in the business and engineering world are familiar with Microsoft Excel or Matlab, either of which could very easily be utilized to perform this method. Using these tools to implement the method give a nearly ubiquitous access to the evaluation. This representation also provides for a visualization of the data in a concise format. This allows for comparison of the methods on each individual criteria. For instance, a quick glance at the table shows which method is has the best score for cost. The ability to represent the data visually is quite useful for proposals and briefings. Also, though not represented in this example, the method is useful to justify methods that are clear-cut winners.

A final benefit of the method is that it is easily adaptable with changing conditions. For instance, if the corporation no longer was concerned about time, the weight for time could be changed to 1 and the composite scores could be recalculated. This allows corporations to consider different scenarios. In this case, they could consider short-term vs. long-term considerations. In other cases, if technology readiness level was a criteria, corporations could consider a scenario were a product could be immediately available along with one where development would be required. Once the scores for the individual criteria are assigned, many different evaluations can be performed through simply changing the weights.

Cons For it's advantages, the matrix-like evaluation has some severe limitations. The assignment of the scores for the individual criteria tend to be arbitrary. Typically, if something is "bad," it is assigned to the low end of the range and if something is "good," it is assigned to the high end of the range. However, there is no true definition of what it means to score a particular number. Because of this, people tend to score the various alternatives relative to each other and not on an absolute scale. For instance, a person might have arbitrarily decided that the performance of the COTS product was an 8 and then established that the performance would be "slightly worse" for the component if the corporation made it, thereby giving the make option a 7. However, what they are really saying is that the make option's performance is -1 as compared to the buy option. While this ensures some sanity within a given criteria, it may adversely affect the overall results because, due to the arbitrary nature of the scoring, the numbers are not standardized across the columns. That is, an 8 in the performance category does not necessarily correspond to an 8 in the time category.

Even if great care is taken to provide solid reasoning to the scores, the method does not capture the interaction between the criteria. In the above example, it is most likely the case that the more money and the more time the corporation spends developing their own component, the greater the performance of the final product. Thus, as the score for performance increases, the scores for cost and time go down. However, there is no way to extract or quantify that relationship from the matrix-like method. All that is shown are the numbers assigned to the criteria and all other details are obfuscated.

Furthermore, people tend to change numbers based upon how the evaluation comes out to better fit what they think should happen. In this case, the method is used to justify their own thoughts more than to evaluate the actual solution.

A further limiting factor is the use of discrete ranges. In the case of close evaluations, the error due to the discrete nature of the scores could potentially make a difference. Suppose in the above example, the performance of the COTS product actually should have been 8.4. With this change, the the composite scores come out to be exactly even for the two alternatives. However, very few people would ever say 8.4, but would in fact give a score of either 8 or 9. Depending on the score they chose, the answer becomes different.

Summary

In the matrix-like evaluation method, the user first defines criteria by which to rate the alternatives. Next weights are assigned to each criteria to establish its relative importance. Each alternative then receives a score for the individual criterium. Finally, the individual scores are multiplied by the weights and the results are summed into a composite score. The alternative with the highest score is the best and should be selected. Advantages of this method are that it is easy, quick, can be performed in ubiquitously used tools, allows for easy visualization, and is easily adaptable to different circumstances. The disadvantageous include the arbitrary nature of the scores, the lack of standardization of numbers across criteria, the lack of capturing interrelationships of the criteria and obfuscation of all details other than the scores of the criteria, and the errors associated with discrete number ranges. The method is good at justifying using a method that is clearly better than others or performing a coarse analysis used to down-select between alternatives, but is not good at evaluating close decisions.

1.2.2 Multi-Disciplinary Optimization (MDO)

The next method of system-level design considered is Multi-Disciplinary Optimization (MDO). According to [25], MDO is a methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena using integrated analyses to account for interactions amongst disciplines.

In MDO, the emphasis is on the multidisciplinary nature of the design process for complex engineering system. As discussed before, disciplinary specialists strive toward improvement of the objectives and satisfaction of constraints defined by their own discipline. In doing so, they generate side effects that other disciplines have to absorb. In Section 1.1.2, this was shown through the example of attempting to increasing the range of an aircraft by increasing the aspect ratio. The discussion that followed showed that the increase in aspect ratio had an effect on the weight of aircraft and the result was that the range may have either be increased or decreased in the process. Figure 1-9 revisits this example, this time also considering the constraints of flutter, rate of climb, and takeoff length.

The three constraints are shown as the labeled colored lines in the figure. In order to satisfy any given constraint, the system must be located completely under the line. Therefore, in under to satisfy all three constraints, the system must be under all three lines. The figure shows that an optimum aspect ratio could not be obtained by viewing the constraints separately from each other, further establishing the need for a multidisciplinary approach. However, looking beyond the figure, one realizes that many such trades have to be considered simultaneously in the design of an aircraft and they have to be resolved not only to end up with a net positive impact on the parameter being varied, but they also have to be solved without violating the constraints imposed by each of the participating disciplines [18]. Schrage, et al. state that the challenge of design that MDO is trying to solve is "How to decide what to

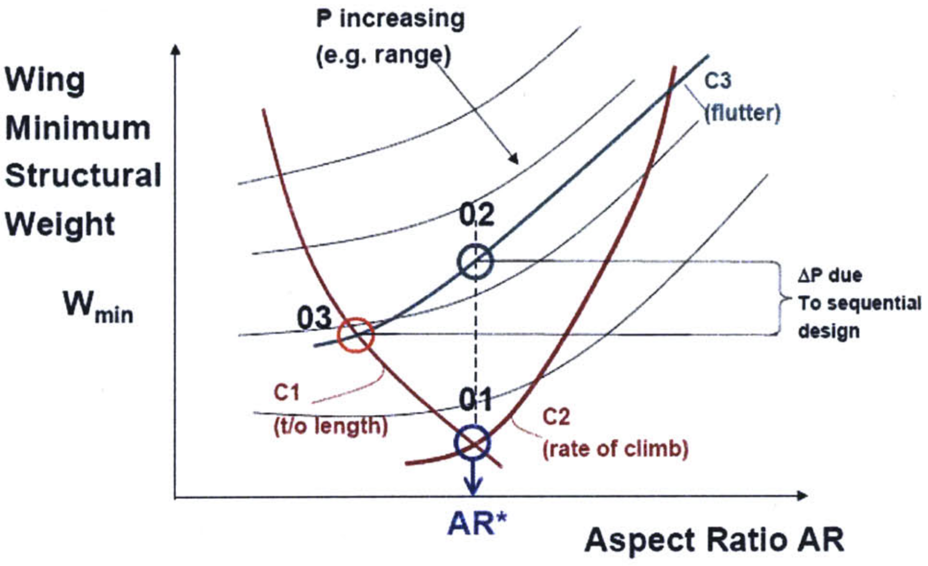


Figure 1-9: Aspect ratio of an aircraft given three constraints (Taken from [25]).

change and to what extent to change it when everything influences everything else" [18]. They further say that two characteristics of an integrated design process, such as MDO, are:

1. any new information originated anywhere (in any discipline) in the design organization is communicated promptly to all recipients to whom it matters and
2. when a change of any design variable is proposed, the effect of that change on the system as a whole, on it parts, and on all the disciplines are evaluated expeditiously and used to guide the system

The framework MDO uses to achieve this communication and interdependence of the design pieces will be examined next, followed by a discussion of optimization and other aspects of MDO. The challenges of MDO will next be examined and the section will close with the pros and cons of MDO.

MDO Framework

A typical design process in MDO involves the following nine step approach [25]:

1. Define overall system requirements
2. Define design vector x, objective J, and constraints
3. System decomposition into modules
4. Modeling of physics via governing equations at module level - model execution in isolation

5. Model integration into an overall system simulation
6. Benchmarking of model with respect to a known system from past experience, if available
7. Design space exploration (DoE) to find sensitive and important design variable x_i
8. Formal optimization to find $min J(x)$
9. Post-optimality analysis to explore sensitivity and tradeoffs: sensitivity analysis, approximate methods, isoperformance, include uncertainty

This process is shown graphically in Figure 1-10.

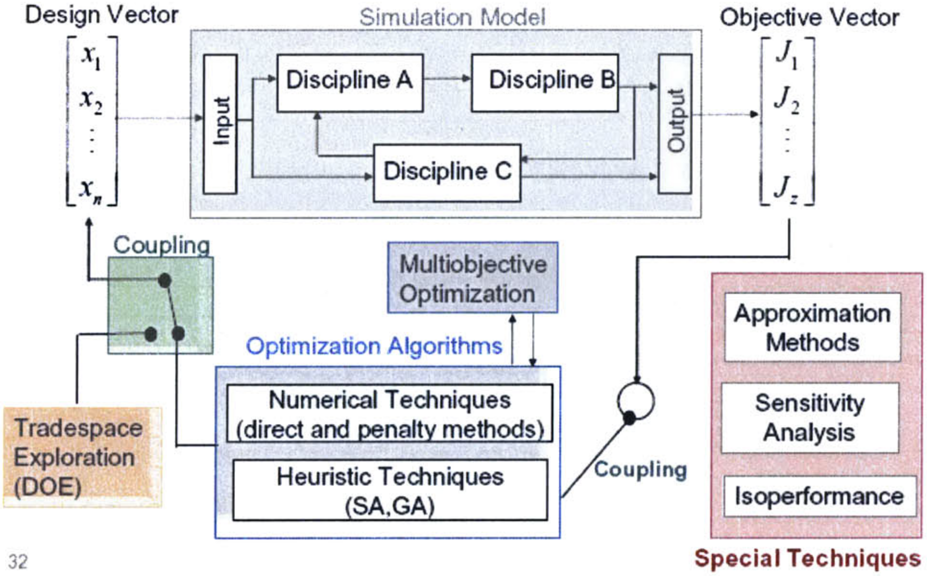


Figure 1-10: Diagram of MDO framework (Taken from [26]).

However, the process is typically not this clean. A common issue occurs when the optimizer returns a mathematically valid but physically unreasonable solution due to an error in the problem setup. Another common problem arises when the optimizer is unable to find a solution that satisfies all of the constraints. When this is the case, one or more constraints need to either be modified or removed. When either of these two modeling inaccuracies occurs, then the problem is adjusted and then the process is tried again. Thus, instead of the clean, linear approach of theory, the method is much more of an iterative debugging process that is highly interactive until a reasonable model is achieved [25].

While the O of MDO would suggest that it is used to only find optimal solutions, MDO is also sometimes used to find a solution that is merely feasible. This is especially true if a large number of constraints exist. Figure 1-11 shows the entire range of design objectives for MDO, ranging from feasibility to pareto. Pareto is in reference to the pareto front, which derives its name from Vilfredo Pareto, creator of the theory of Pareto efficiency. If a system is Pareto efficient, no component can be made better off without another being made worse off [3]. Figure 1-12 shows a graphical representation of the pareto front taken from an example in [25].

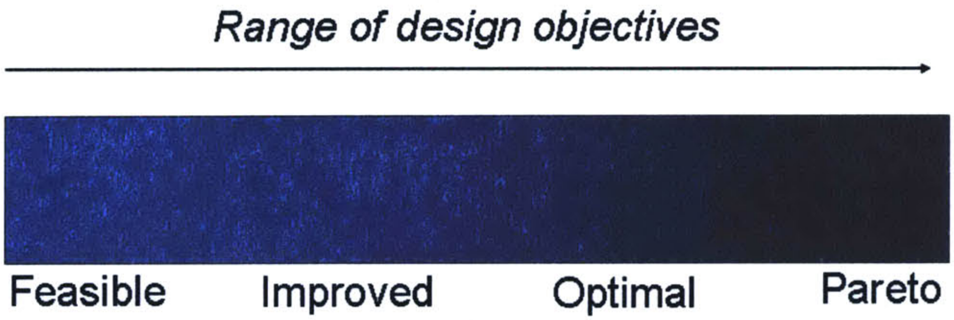


Figure 1-11: Range of design objectives for MDO (Taken from [25]).

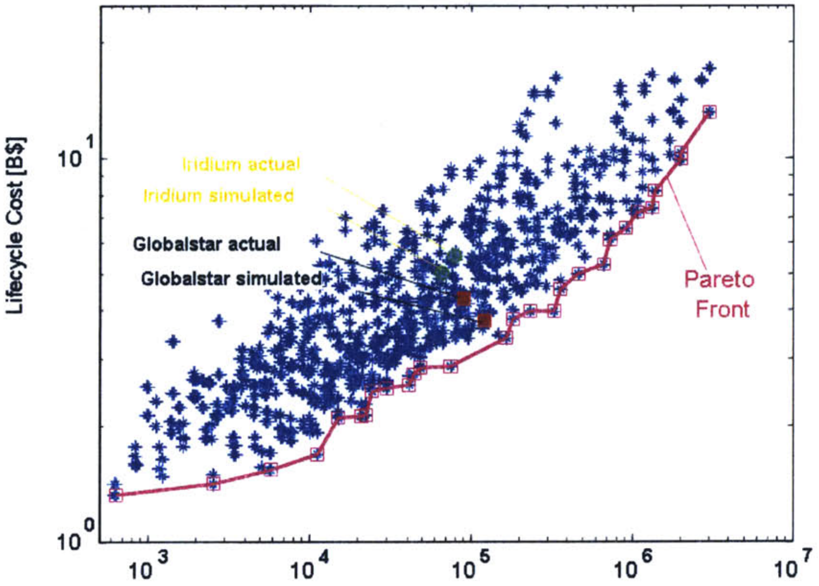


Figure 1-12: Example of the pareto front in MDO (Taken from [25]).

Many of the terms and tasks in the nine step list above of a typical MDO approach to a problem have not yet been introduced. The next two sections discuss the important features of the MDO framework in detail, starting with the optimization and then continuing on with the other features.

Optimization and Objective Functions

In the past, optimization was done using graphical based methods. Here, two variables are selected and put on orthogonal axes in a base plane. Then, a curved surface is plotted above this plane representing some user-defined measure of performance that is a mathematical function of the two variables. This measure of performance is called an *objective function.* The performance can then be either minimized or maximized, depending on the desired result, by following the graph. An example of a curve representing an objective function is shown in Figure 1-13.

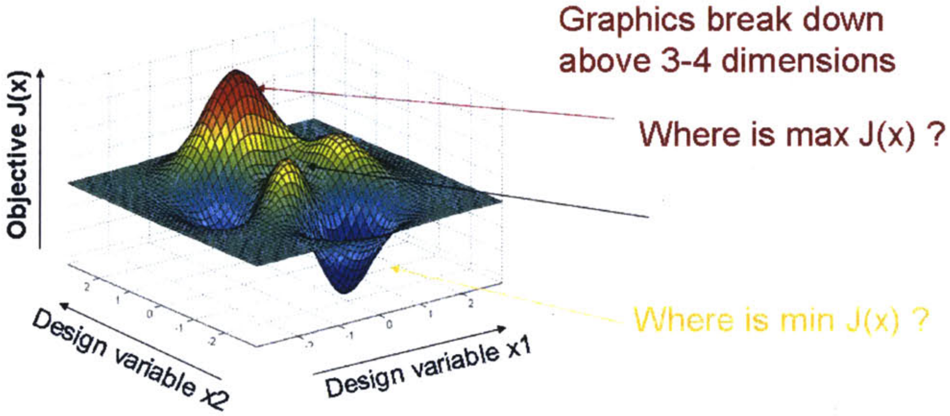
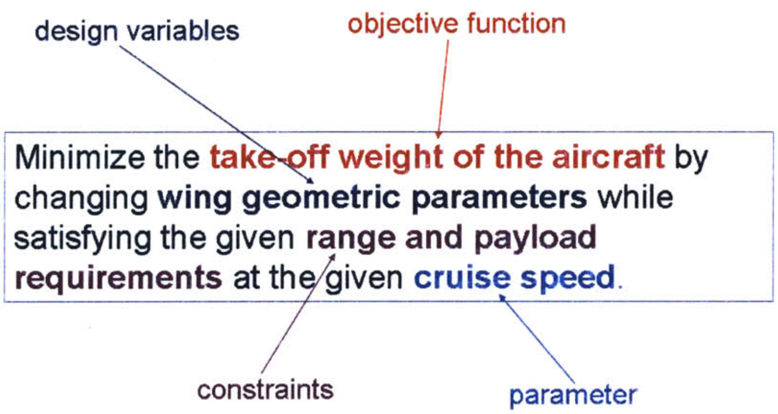


Figure 1-13: Example of a curve used in a graphical optimization (Taken from [25]).

As the Figure 1-13, problems exist with graphical-based optimization algorithms. The most prevalent problem is local extrema. Most algorithms are susceptible to getting trapped in local extrema and as a result, returning incorrect minima and maxima. Also, inherent to graphical-base methods is that they only work for problems of very reduced dimensionality. One common way of dealing with this for problems of increased dimensionality is to use two variables at a time and optimize them. This is done for all the combinations of variables until a pareto-like compromise is reached. The problem is that this is a difficult balance to achieve using a brute force method and often is not the same as a global optimum for the entire system. Plus, for $n > 3$, there begins to be a combinatorial dimensionality issue and the design space cannot be completely computed in polynomial time [25].

The same principle used in the creation of the objective function in the graphical method can be used for general design. Any design can be defined by a vector in multidimensional space where each design variable represents a different dimension [18]. The objective function may express cost, weight, range, aerodynamic or propulsive efficiency, return on investment, or any combination of parameters. A sample problem statement, appropriately labeled with the various elements, is shown in Figure 1-14.

The objective function is subject to functional constraints that are governed by given relationships between variables and parameters and to upper and lower bounds of variables [18]. The side constraints define the permissible part of the curved surface where the optimum value has to be found [18]. Thus, the quantitative side of the design problem may be formulated in formal notation as a problem of Nonlinear Mathematical Programming (NLP) [18]. A sample formulation is shown in Figure



Minimize the **take-off weight of the aircraft** by changing **wing geometric parameters** while satisfying the given **range and payload requirements** at the given **cruise speed**.

Figure 1-14: A sample problem statement, labeled with its components (Taken from [25]).

1-15. As shown in the figure, x is a vector of the design variables, $x_{i,LB}$ is the lower bound of the i^{th} variable, $x_{i,UB}$ is the upper bound of the i^{th} variable, p is a vector of constant parameters, J is an objective function, g is a vector of inequality constraints, and h is a vector of equality constraints.

$$
\begin{aligned}
&\min \mathbf{J}(\mathbf{x},\mathbf{p}) \\
&\text{s.t.}\ \ \mathbf{g}(\mathbf{x},\mathbf{p}) \leq 0 \\
&\qquad \mathbf{h}(\mathbf{x},\mathbf{p}) = 0 \\
&x_{i,LB} \leq x_i \leq x_{i,UB} \quad (i = 1,\dots,n)
\end{aligned}
\qquad
\text{where}\quad
\begin{aligned}
\mathbf{J} &= \left[J_1(\mathbf{x}) \ \cdots \ J_z(\mathbf{x})\right]^T \\
\mathbf{x} &= \left[x_1 \ \cdots \ x_i \ \cdots \ x_n\right]^T \\
\mathbf{g} &= \left[g_1(\mathbf{x}) \cdots g_{m_1}(\mathbf{x})\right]^T \\
\mathbf{h} &= \left[h_1(\mathbf{x}) \cdots h_{m_2}(\mathbf{x})\right]^T
\end{aligned}
$$

Figure 1-15: Sample formulation of a design problem in NLP (Taken from [26]).

It is important to be explicit about the dimensionality of the components. The objective function, $J(x,p)$, could be a single function or contain many functions, as shown in Figure 1-15. This corresponds to "single-objective" objective functions and "multiobjective" objective functions, respectively. Similarly, $g(x,p) \leq 0$ is a vector of inequality constraints and can thus contain any number. The same can be said for the equality constraints of $h(x,p) = 0$. The multiobjective relation of $J(x,p)$ may be as general as admitting all J_i's on equal footing and rendering J a vector, or as specific as a weighted sum of the J_i's, reducing J to a scalar [18]. As a result, specifying $J(x,p)$ defines the desired balance of the various objectives J_i. Applying this to the design problem at hand, the multiobjective formulation represents a translation of the customer's ranked requirements and goals into a mathematical statement of the design problem [18].

The idea of formulating a design problem in rigorous, mathematical terms is a key component of MDO. In contrast to graphical methods, MDO mathematically traces a path in the design space from the initial design toward improved designs with respect to the objective function and does it operating on a large number of variables and functions simultaneously [18]. This feat is not possible for humans and the path taken is not biased by intuition or experience, both definite pluses of a machine-driven design approach. However, the visibility of the reasons for the design decisions corresponding to the twists and turns of the search path remain obscured inside a black box. Post optimality and parameter sensitivity analysis can provide much information that can raise the confidence of the designer and allow him/her to assess the impact of changes to the original problem formulation on the outcome [18]. For instance, if the p values contained in the equations shown in Figure 1-15 vary in an uncertainty range, it may be practical to optimize the design for the most probable P first [18]. Subsequently, a range of near-optimum designs may be approximated by extrapolation in the neighborhood of the nominal design using the derivatives of the optimal J and x [18]. Many different optimization methods have been used in conjunction with MDO, including gradient-based methods such as Newton's method and sequential quadratic programming, population-based methods such as genetic algorithms and particle swarm, and other methods such as random search and simulated annealing.

Other Aspects of MDO

As mentioned before, Figure 1-10 and the preceding list show the basic framework of the MDO design process. The previous section detailed the optimization involved in step 8 and the post-optimization sensitivity analysis of step 9, as well as gave examples of the design variables and constraints list in the other steps. This section will very briefly touch on some of the remaining aspects of MDO, both those stated directly in the step-by-step process and other general characteristics not explicitly stated.

MDO is not intended to be a completely automated, push-button type of design methodology, but instead one that incorporates the strengths of both humans and computers. It has a qualitative side dominated by human inventiveness, creativity, and intuitive understanding of the many complex real-world constraints upon the system. The other side is quantitative, concerned with numerical answers to the questions that arise on the qualitative side. This approach is consistent with the creative characteristics of the human brain and the efficiency, discipline, and infallible memory of the computer [18]. Since the continual concern about the "what if" questions is what creative design is all about, having a capability to answer such questions expeditiously and comprehensively will constitute a quantum jump in the design process effectiveness and efficiency [18]. MDO attempts to provide this capability through the synergy of human ideas and discrete thinking with computer evaluation and combinatorial capabilities.

The first two steps of the MDO process are to define the overall system requirements and define the design vector, objective function, and constraints. The elements of the latter step were described in the optimization section. The design vector, objective function, and constraints are usually logical extensions of the requirements, which are usually provided by the customer in the form of a requirements document. A sample was provide in the optimization section and further detail can be obtained from literature.

The third step of the MDO design process involves decomposing the system into modules. It is generally agreed that the challenge posed by the quantitative side of an advanced aircraft design needs decomposition that breaks the large, intractable problem into smaller subproblems while maintaining the couplings among the subproblems [18]. The decomposition approach stems from the realization that the analysis and sensitivity analysis that generate data needed by optimization algorithms may easily account for more than 90% of the total computation time [18]. Numerous decomposition schemes have been proposed in literature and undoubtedly more will be developed in the future. Two very broad classes of decomposition are hierarchical based and non-hierarchical decomposition. More information about the specifics of decomposition in MDO can be found in the subject's extensive literature.

Proper modeling techniques are beyond the scope of this discussion. However, at the time of writing, modeling is usually performed in Matlab, with legacy Fortran code used when necessary. However, even with the legacy Fortran simulations, there is a push to move towards Matlab and Simulink in order to update the technology. If speed is of greater importance, then simulations are implemented and performed using C/C++ code. While the computation time is faster, the implementation time is usually longer with C/C++ than with Matlab.

Once the model of the system is created, it needs to be validated to ensure that the answers are correct. *Benchmarking* is the process of validating a simulation by comparing the predicted response against reality. In a simulation-based design process such as MDO, benchmarking is of utmost importance and is required in order to ensure that fidelity is being achieved. From the list of steps, benchmarking occurs after the models are integrated into the system simulation and is followed by the design space exploration. This makes sense in the logical flow of the MDO design since the models have to be verified before they are used to explore the design space, but can't be used to do so until the design defines what the system is supposed to do, what its boundaries are, and models of the system's components are created and integrated.

It should also be noted that parallel processing is sometimes mentioned as a tool that can be employed by MDO to speed up the computation required by the models. However, parallel processing assumes that a problem can be broken into large independent pieces that can be computed on separate processors [18]. A noticeable speed up is not achieved simply by using a multiprocessor computer for executing a method that originated in a serial computer environment. In order to extract full computational potential from a multiprocessor computer, new algorithms need to be employed that take advantage of the parallel environment. An extensive literature exists on distributed computing algorithms. While a significant speed up is possible, it comes at the cost of complexity.

Pros, Cons, and Additional Challenges

With the framework of MDO established, discussion now turns to the benefits, downsides, and major challenges in using MDO. The primary benefit of MDO is that it is a systematic, logical design procedure that handles a wide variety of design variable and constraints and is not biased by intuition or experience. As a result, it produces an optimized, integrated system as opposed to a system of optimized components. Also, de Weck asserts that MDO reduces the amount of design time [25].

One major challenge and potential downside of MDO lies in the trade between fidelity and complexity of the simulation model. In general, there is a tradeoff between achieving the highest fidelity simulation possible and the complexity of the model. Since computational time and numerical problems grow rapidly with the number of design variables, fidelity is sacrificed to obtain models with short computation times or that are of a realistically manageable size. However, if too much fidelity is removed through the reduction of design variables and constraints, then the model returns erroneous conclusions and is rendered useless.

Complexity is not only an issue with the number of variables, but also with the interaction of the various pieces of the simulation. The tedium of coupling variables and results from disciplinary models is such that MDO engineers can spend 50–80% of their time doing data transfer [26]. Additionally, the user interface with the models is often very unfriendly and, as a result, changing problem parameters can be difficult. Also, many MDO systems degenerate into a very specialized tools that are only valid for one problem and can only be used by a handful of select people intimately involved in the design. Furthermore, the use of MDO in design is limited to the range of projects that are applicable to the analysis methods of MDO. Not all projects have well-defined numerical constraints and objective functions to optimize. Finally, creating even a simple MDO analysis tends to be time consuming. While de Weck's assertion that MDO reduces the design time might be true once the system has reached the level of full commitment to the project, at the proposal stage of design, MDO might prove too time consuming to implement.

In addition to overcoming the above-listed hardships, MDO needs to make significant progress on allowing for creativity and intuition while still leveraging rigorous, quantitative tools in the process. Progress is also required in the areas of incorporating higher-level upstream system architecture aspects early in the design and data visualization in multiple dimensions [26].

1.2.3 Other Methods

The next several sections describe additional methods in an attempt to give a more complete picture of the system design methods used in practice. Of course, this is not an exhaustive list, but instead should be considered a sampling of the overall space of system design.

Cost As an Independent Variable (CAIV)

The purpose of the Cost As an Independent Variable (CAIV) system of management is to provide the customer with a highly capable system that is affordable over its life cycle [16]. This is done by trading off performance with cost and schedule to set aggressive, yet realistic objectives for the system. It is embraced by the Department of Defense (DOD) as a key acquisition consideration.

Motivation To demonstrate why this trade off is necessary, the historical example of selecting a processor for a project in April of 1996 will be considered. The price of the processor varied with a single measure of performance, the clock rate in MHz. The resulting graph of performance versus cost is shown in Figure 1-16.

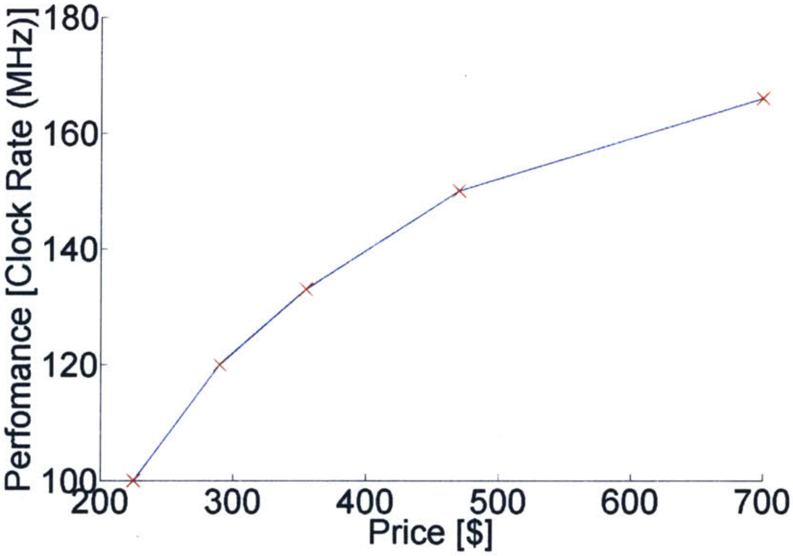


Figure 1-16: Example of processor performance vs cost in 1996 (Taken from [16]).

Figure 1-16 shows that the last 11% of performance (150 MHz to 166 MHz) leads to a 45% increase in cost ($469 to $678). Conrow states that in many cases, striving for the last 3% of performance that can be obtained at a certain time can lead to a 20 to 30% increase in the cost [16]. In addition to the precipitous rise in cost, a development phase design near the upper limit of achievable performance often introduces considerable risk to the project, not only in terms of cost, but also in schedule.

As stated earlier, while the emphasis of military projects has shifted away from the strictly performance-driven metric of the 1960s to include more of a balanced emphasis on cost, reliability, maintainability, and other areas, the requirements that exist could still unnecessarily dictate a type of design or drive cost. Using the processor example, consider the possibility of the requirement being set at 166 MHz, but really only 150 MHz were needed to perform the specified task. The result would be a substantial price increase with no added benefit to the project other than meeting a requirement that was not well specified. While being a simple example, Figure 1-16 is illustrative of how drastically unregulated requirements can drive cost.

In CAIV, cost-performance trades are done to establish curves similar to that of Figure 1-16. The results are then used to set the performance and cost requirements for the system. These are usually done with a range for key requirements, with a threshold value that must be met and an objective value that is the desired level.

This process transforms cost into much more of an independent variable, as the performance can be established to match a budgetary constraint. By treating cost as an independent variable through the use of cost-performance trades, cost objectives are balanced against mission needs, taking into account existing technology, maturation of new technologies, and anticipated process improvements in both DOD and industry. In doing so, CAIV reduces the dominance of performance-driven design aspects and helps managers to recognize the risk level. It also helps the requirements community set performance objectives.

Management However, CAIV is still useful after the requirements have been set. In much the same way that CAIV can be used to set requirements, CAIV can also be used to trade amongst requirements. That is, by easing one requirement, gain could be achieved in another. Reduction or even elimination of a requirement could also serve to mitigate risk. Inherent to both the establishing and adjusting of the stated requirements of a system is a constant cooperation between the customer and the proposing organization. The system is no longer a static, rigid entity that must be blindly created, but a dynamic, fluid structure created by all those involved, whether that be the customer or the proposing company. This is an attempt to eliminate the effect of requirements with unintended consequences, such as the F-111's requirement to fly supersonic at sea level (an artifact of a failed attempt to cooperate with the Navy).

In its proper implementation, CAIV is a twofold process. First, CAIV is a planning activity that establishes and adjusts the program cost objectives through the use of cost-performance analyses and trades and second, it is used to help execute a program in a way meet these cost objectives [16]. As stated earlier, the cost-performance (and even schedule) trades shape the requirements and the proposed design approaches based upon cost effectiveness. Once a specific design is selected, cost objectives should be allocated for specific system elements and cost items. As the project approaches production and funding constraints are better known, the cost effectiveness focus is modified to affordability considerations, focusing instead on the alternatives that are practical given the project's budget [16]. The proposals that are thus deemed best according to the CAIV method are those that have the best benefits to cost ratios with proper risk mitigation coupled into the design.

Pros and Cons CAIV is blessed by DOD and thus gains a great deal of credibly from the endorsement. It allows for a dynamic system that will meet the system requirements while also being affordable. However, CAIV says nothing about how to actually realize the systems. It is not a design methodology as much as it is an acquisition methodology. However, since it is used by the DOD for acquisitions, its principles must be followed and supplemented by the design methodology used by a proposing company.

Analysis of Alternatives (AoA)

According to the DoD-sponsored Defense Acquisition Handbook [31], an Analysis of Alternatives (AoA) is an analytical comparison of the operational effectiveness, suitability, and life cycle cost of alternatives that satisfy established capability needs. Since the DoD requires an AoA at milestone decision points for major defense acquisition programs, the Defense Acquisition Handbook has an in-depth description of how to conduct an AoA. The quality of the description, coupled with the fact that the DoD controls a great deal of the money spent in the defense industry, makes the Defense Acquisition Handbook ([31])a valuable source of information for AoA. Most of the information in this section, even if not explicitly cited, came from this source.

The initial application of AoA on a project investigates various conceptual solutions with the goal of identifying the most promising alternatives. This analysis is useful in the Concept Refinement Phase of the project. An AoA is performed again at Milestone B in order to justify the rationale for formal initiation of the acquisition program [31]. Thus, AoA helps elucidate which concepts to focus on and helps justify the allocation of funds, depending upon the stage of the project.

AoA Plan The Defense Acquisition Handbook states that a major step to a successful AoA is the creation and coordination of a well-considered analysis plan, including a roadmap of how the analysis will proceed and who is responsible for doing what. The following is the recommended outline for an AoA report and will serve as the key to the rest of the discussion about AoA:

- Introduction
  - Background
  - Purpose
  - Scope
- Ground Rules
  - Scenario
  - Threats
  - Environment
  - Constraints and Assumptions
- Alternatives
  - Description of Alternatives
  - Nonviable Alternatives

    - Operations Concepts
    - Support Concepts
- Determination of Effectiveness Measures
    - Mission Tasks
    - Measures of Effectiveness
    - Measures of Performance
- Effectiveness Analysis
    - Effectiveness Methodology
    - Models, Simulations, and Data
    - Effectiveness Sensitivity Analysis
- Cost Analysis
    - Lifecycle Cost Methodology
    - Models and Data
    - Cost Sensitivity and/or Risk Analysis
- Cost-Effectiveness Comparison
    - Cost-Effectiveness Methodology
    - Displays or Presentation Formats
    - Criteria for Screening Alternatives
- Organization and Management
    - Study Team/Organization
    - AoA Review Process
    - Schedule

The introduction section of the AoA plan describes the developments that led to the AoA, including any preceding analysis. It should also reference applicable documents, such as the capability needs document and any AoA guidance. Finally, it should identify the level of detail of the study and the breadth and depth of the analysis required to support the specific milestone decision [31].

The ground rules sections details the scenarios, threats, assumed physical environment, constraints, and additional assumptions. The scenarios are typically derived from defense planning scenarios and are augmented by more detailed intelligence products including targeting information and enemy and friendly orders of battle [31].

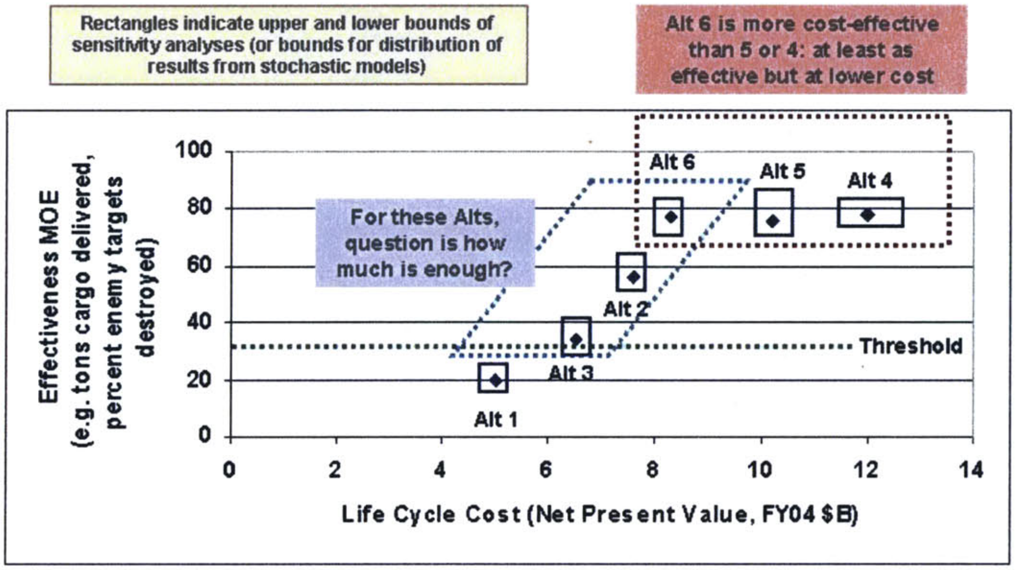
The third section of the AoA plan deals with the range of alternatives addressed in the analysis. The range should be kept to a manageable number, with a greater danger being including too many alternatives for the given resources [31]. The number of alternatives can be limited by avoiding similar but slightly different alternatives and the early elimination of alternatives with properties outside acceptable ranges, such as a life cycle cost that is too high. Many studies employ a baseline case that retains one or more existing system to represent a benchmark of current capabilities [31]. An additional alternative is then typically based upon major upgrades and/or service-life extensions to the benchmark. The operations concept sub-bullet refers to the details of the peacetime, contingency, and wartime employment of the alternative within projected military units or organizations [31]. The support concepts describe the plans from system training, maintenance, and other logistics support [31].

In the determination of effectiveness measures section of the AoA, the mission tasks, measures of effectiveness, and measures of performance sub-bullets are all metrics used to measure the military worth of each alternative. Military worth is fundamentally the ability to perform mission tasks, which are derived from the identified capability needs [31]. Mission tasks are usually expressed in terms of general tasks to be performed to correct the gaps in needed capabilities, such as communicating in a jammed environment, and should not be state in solution-specific language. Measures of effectiveness provide the details that allow each alternative's proficiency in performing the mission task to be quantified [31]. Finally, a measure of performance is a quantitative measure of a system characteristic, such as range or weapons load, chosen to enable calculation of one or more measures of effectiveness [31]. They are typically linked to parameters in the capability needs document.

The fifth section of the AoA plan as described in the Defence Acquisitions Handbook details the effectiveness analysis, which is based upon the military worth established in the previous section, the assumed scenarios and threats, and the nature of the selected alternatives. The levels of effectiveness analysis can be characterized by the numbers and types of alternative and threat elements being modeled. A typical classification would consist of four levels: (1) system performance, based on analyses of individual components of each alternative or threat system, (2) engagement, based on analyses of the interaction of a single alternative and a single threat system, and possibly the interactions of a few alternative systems with a few threat systems, (3) mission, based on assessments of how well alternative systems perform military missions in the context of many-on-many engagements, and (4) campaign, based on how well alternative systems contribute to the overall military campaign, often in a joint context [31]. Most AoAs involve analyses at different levels, where the outputs of the more specialized analysis are used as inputs to more aggregate analyses. At each level, establishing the effectiveness methodology often involves the identification of suitable models, other analytic techniques, and data. The measures of effectiveness established in the previous section should serve as the basis for this identification. Sensitivity analyses are also important to address. Along these lines, it is important to also point out what the critical assumptions are that drive the results of the analysis and show how the results change with variations in these assumptions [31].

The cost analysis section of the AoA plan describes the approach to the life cycle cost analysis. It is considered to be on par with the effectiveness analysis in terms of performance. When the costs of the alternatives have significantly different time periods or distributions, appropriate discounting methods should be used to calculate the life cycle of each alternative [31]. The life cycle cost analysis is a major effort that requires the attention of experienced, professional cost analysts.

The penultimate section of the AoA analysis and the last analytical section deals with the planned approach for the cost-effectiveness comparisons. The difficulty of this comparison is that typically all the alternatives have both different costs and different levels of effectiveness. As a result, the selection is not as simple as picking the alternative with the greatest effectiveness given equal cost or lowest cost given equal effectiveness. A common technique used to placate this difficulty is the use of a scatter plot of effectiveness versus cost. On example of such a plot taken from [31] is shown in Figure 1-17.


Rectangles indicate upper and lower bounds of sensitivity analyses (or bounds for distribution of results from stochastic models)

Alt 6 is more cost-effective than 5 or 4: at least as effective but at lower cost

For these Alts, question is how much is enough?

Figure 1-17: Example of a scatter plot used in the cost effectiveness comparison of alternatives (Taken from [31]).

The final section of the AoA plan details the organization and management of the study. Typically the study team is composed of a diverse mix of military, civilian, and contractor personnel. The program office may provide assistance or data to the team, but the AoA should not be the responsibility of the program manager and the study team should not reside in the program office [31]. In certain cases, the AoA may be assigned to a federally funded research and development center. The AoA study team is usually organized along functional lines into panels, with a chair for each panel [31]. Typical panel divisions might be threats and scenarios, technology and alternatives, operations and support concepts, effectiveness analysis, and cost analysis [31]. The effectiveness panel usually is the central committee and integrates the work of the other panels. The organizational section also needs to describe the oversight and review process for the AoA.

Pros and Cons Like CAIV, AoA is blessed by the DoD. As a result, it is required on major defense acquisition programs. It is a well-established methodology that has been applied to a variety of projects and has proven to be successful. A great deal of resources have been applied to implementing and improving the AoA process and future projects benefit from this investment and experience. However, drawbacks exist with AoA. The formal AoA approach requires a considerable amount of resources, including time, money, and expertise. It is a much better fit as part of a major program that either is funded or is guaranteed to be funded as part of the program's formal design structure. The resource requirement, especially in terms of time, makes AoA a poor fit for a proposal environment before a contract has been awarded.

Strengths, Weaknesses, Opportunities, Threats (SWOT)

The SWOT framework, which stands for Strengths, Weaknesses, Opportunities, and Threats, was first described in the late 1960's by Edmund P. Learned, C. Rolan Christiansen, Kenneth Andrews, and William D. Guth in *Business Policy, Text and Cases* [9]. SWOT assesses the internal and external factors that affect an organization. The internal factors are be classified as strengths or weaknesses and the external factors are be classified as opportunities or threats.

Definitions In defining strengths, [8] and [9] focus on creating a competitive advantage, with [8] saying that strengths are the resources and capabilities that can be used to serve as a basis for developing a competitive advantage. [13] gives perhaps an overly simplistic definition of strengths as what an organization can do. Examples of strengths include patents, strong brand names, good reputation amongst customers, cost advantages due to proprietary information, innovative products or services, business location, or any other aspect that adds value to a company's product or service [8], [10].

The definition of weaknesses given by the sources tended to be the logical negative of the definition given for strengths, including [9]'s factors that hinder a competitive advantage or [13]'s things an organization can't do. However, [8] points out that an absence of a strength could be viewed as a weakness and that weaknesses could also come as the flip side of strengths. To demonstrate this, [8] gives the example that a firm's large manufacturing capacity might be seen as a strength that competitors do not share, but that it might prevent them from acting quickly to changes in the strategic environment. Other weaknesses include the logical negative of most of the strengths listed above (a lack of patents, poor brand names, etc), a lack of access to the best natural resources, a lack of access to key distribution channels, poor quality goods, or any other aspect that subtracts value to a company's product or service [8], [10]. Danca points out that strengths and weaknesses exist not only internal to an organization, but also within key relationships between a firm and its customers [13]. He continues by stating that the role of the internal portion of SWOT is to determine where resources are available or lacking in order to identify strengths and weaknesses so that marketing managers can then attempt to match the strengths with the opportunities and thereby create new capabilities while also trying to develop a strategy to minimize weaknesses.

NetMBA defines an opportunity as the chance to introduce a new product or service that can generate superior returns [9]. However, this is too limited in scope to be wholly accurate. Opportunities could also exist to offer an existing product or service in a new market. For instance, situations could exist where corporations could enter into a new country that just had an embargo lifted or market to a country that is just becoming developed and previously did not have the technology for a service or product. [9] does correctly point out, however, that opportunities can arise when changes occur in the external environment, often times related to customers, competitors, market trends, suppliers, partners, social changes, new technology, the economic environment, or the political and regulatory environment. Threats stem from the same changes in the external factors, but instead act to hinder a business or a person. Going back to [13]'s overly simplistic definitions, threats are potentially unfavorable conditions for an organization.

Also, it should be noted that some sources used TOWS interchangeably with SWOT while [10] says that TOWS looks at negative factors first so that they can be transformed into positive factors. Either way, it appears that any difference is mainly in semantics and only possibly involves a change in connotation.

Goals The various sources state the general purpose of SWOT differently. [9] says that SWOT is used for generating strategic alternatives from situational analysis and is generally used in marketing plans. This theme is echoed in [13]'s statement that SWOT provides direction and serves as the basis for marketing plans. [8] latches on to the strategy portion saying that SWOT is instrumental in strategic formulation and selection while [28] says that SWOT is a tool for market analysis and can also be applied to career planning. [10] takes a slightly different approach in saying that SWOT is a tool for auditing an organization and its environment.

These various statements can be used to say that SWOT is a planning tool that helps the user identify strengths, weaknesses, opportunities, and threats so that they can leverage strengths, correct weaknesses, capitalize on opportunities, and deter threats [9]. It can be used to distinguish an organization from competition and allow it to compete successfully in a market [21]. SWOT results could further show that a corporation should not necessarily pursue the most lucrative opportunity, but instead should develop a competitive advantage by finding a fit between the firm's strengths and upcoming opportunities. It could also prepare itself for other upcoming opportunities by overcome weaknesses [8].

Some of the classes of strategies potentially developed from SWOT can be seen in Table 1.2. These strategies are all taken from [8]. S-O strategies are those that pursue opportunities that are a good fit with the entity's strength. By contrast, W-O strategies overcome weaknesses to pursue opportunities. S-T strategies identify ways that the entity can use its strengths to reduce its vulnerabilities to external threats and W-T strategies establish a defensive plan to prevent the entity's weaknesses from making it highly susceptible to external threats.

Table 1.2: SWOT Matrix and associated strategies (Taken from [8]).

	Strengths	**Weaknesses**
Opportunities	S-O Strategies	W-O Strategies
Threats	S-T Strategies	W-T Strategies

Pros and Cons The primary benefit of SWOT is that it is a very simple framework that can be done with little more than interviews and surveys. A further benefit is that it reduces a large quantity of situational factors into a more manageable profile [9]. One of the primary detractions of SWOT is that it is very subjective and arbitrary [10], [9]. Two people rarely come up with the same SWOT analysis for the same situation, making the factors subjective, and a classification of the same factor be a strength to one person and a weakness to another, thereby making the classification seem arbitrary. This leads to the complaint that SWOT tends to oversimplify factors by classifying them into categories in which they may not fit [9]. More important than superficial classification of these factors is the firm's awareness of them and its development of a strategic plan to use them to its advantage [9].

Furthermore, SWOT is by design a method for analyzing internal and external factors to a situation of interest for a business organization or a person so that a strategic plan can be created to best further the entity's objective while also protecting itself. It helps motivate way an entity should or should not act a certain way. It in no way tells them how to accomplish the stated actions and is thus not much help in a design process. Nevertheless, it is a valuable high-level tool for motivating why an organization should allocate resources to a project or why it should enter into a certain proposal.

Tips In order to mitigate some of the subjectivity of SWOT, the sources offered tips on how to properly conduct the analysis. However, there is not a consensus on some of the details. For instance, [10] says the analysis should always be conducted relative to the competition (i.e., better than or worse than the competition) while [13] says that SWOT must be customer-focused to gain maximum benefit, since a strength is really only meaningful when it is useful in satisfying the needs of the customer. This makes sense with his overly simplistic definition of a strength as what an organization can do because clearly the activity of the organization must satisfy a need of the customer's in order to be considered useful. However, modifying the definition of a strength to include the customer's perspective could eliminate this discrepancy. Furthermore, [10] says that being specific is good, but then encourages users to keep the analysis short and simple. Making the analysis specific while keeping it short and simple appears to be contradictory objectives.

However, most of the sources are in agreement that in order for any fidelity to be present in the analysis, the user must be realistic about strengths and weakness. [9] encourages users to seek input from the entire spectrum of stakeholders, potentially including employees, suppliers, customers, strategic partners, and others. Some encourage a question-based approach where the user answers predetermined questions to spur thoughts. Finally, [10] encourages users to distinguish between where an organization is today and where it could be in the future.

Feasibility Study

A feasibility study is designed to establish the key issues that will lead to the success or failure of a project or business [24]. It ultimately aims to answer the question of whether or not a proposed solution could possibly meet the needs of a given problem or opportunity. A feasibility study can be conducted on an entire class of solutions or for a particular instance of a solution. An example of the former would be whether or not it is possible to create a UAV under 10 g that provides video while an example of the latter would be whether or not the Raven UAV could be modified to meet the targeting requirements of the FCS program. It is a first order analysis that aims to provide enough information to answer the question of feasibility without providing all the details on how the functionality would be accomplished.

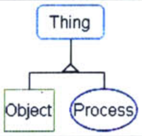
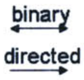
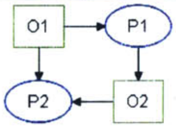
The benefit of a feasibility study is that it mitigates a great deal of the risk associated with a new or unproven solution by allowing for a reduced commitment before full investment is required. However, several major problems exist with feasibility studies. Like CAIV, a feasibility study is not necessarily a design methodology, but instead is a practice used to make the most of resources. While great in theory, it sometimes proves difficult to provide enough detail to know if an idea is feasible without implementing it fully. There is a trade between the amount of detail put into the study (and therefore its fidelity) and the amount of resources (time, money, etc) required to conduct the study. An assumed risk of conducting an exploratory evaluation is that unforseen details could arise in the actual implementation that prevent the realization of a system that was initially thought to be feasible. Thus, the risk of false positives, saying that a solution is feasible when it is not, is related to the level of depth of the analysis.

On the opposite end of the spectrum, the risk of false negatives, saying that a possibility is not feasible when it actually is, is tied to the breadth and quality of the ideas that are evaluated. A famous example of this is Samuel Pierpont Langley's evaluation of the feasibility of powered flight after the aircraft he designed with a $50,000 grant from the War Deparment crashed into the Potomac River on December 8, 1903 [7]. Langley said that it would be years before powered flight would be achieved by mankind, would require a great deal more investment, and would require an improvement in technology. Nine days after his machine crashed, Orville and Wilbur Wright achieved such a feat at Kitty Hawk [12]. Langley's analysis was based solely on his design and did not encompass the entire body of knowledge and state of technology of the day.

Object Process Network (OPN)

The final evaluation method discussed is the Object Process Network (OPN). OPN is a current research and development project being developed through a collaboration between Massachusetts Institute of Technology (MIT) and the Charles Stark Draper Laboratory. OPN is a metalanguage that allows system designers to model and evaluate systems in a graph-based, visual environment. It provides an extensible vocabulary and computing rules to simulate complex systems with various types of architectural metrics and interactive modes [17]. It does this using three primitives, which are shown in Table 1.3.

Table 1.3: OPN language primitives (Taken from [17]).

Linguistic Primitives	*Thing*	*Relationship*	*Graph*
Property	A unique name within the context of its residing Graph	A set of directly related things	A set of things and their corresponding set of relationships
Classes In OPN	Object, Process	Binary and Directed Relationship	Object-Process Network
Embedded operators	Set and get name	Evaluate (return true/false)	Add, Remove (Things and their relationships)
OPN Examples			

The system designers provide domain expertise to model the desired systems while OPN's graphical user interface provides the view to navigate and edit the system models [17]. OPN's execution engine controls the execution processes by following the application-specific language scheme [17]. A flow of the design process using the OPN tools is shown in Figure 1-18.

In OPN, structured equations such as iterative calculations and closed-form formulas are embedded in processes. The equations are encoded in Jython syntax, which is the Python programming language on a Java virtual machine. Doing so allows OPN to leverage the features of a high-level, high performance scripting interpreter while also allowing users to call arbitrary functions written in Java through this interface.

OPN allows both manual architecting to computer-assisted architecting. This range is shown in Figure 1-19. The top oval in the figure, architectural option modeling, prescribes the space of feasible options [17]. The architects and stakeholders can use OPN's modeling tool to present their policies and rules in computable forms. Architectural instances can then be automatically generated given feasibility rules and policies, as well as the variability specified in the Option Space Model [17]. Finally, architectural metrics are computed for each architectural instance by using discrete transformation rules, probabilistic inference, and customized algorithms.

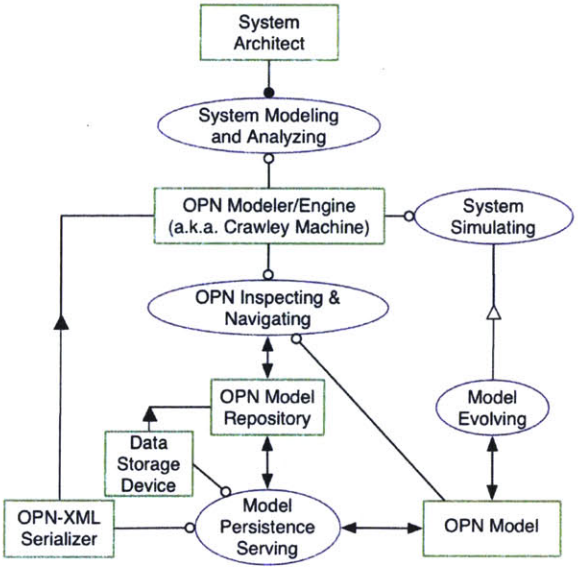


Figure 1-18: OPN design process and tools (Taken from [17]).

Pros and Cons A major benefit of the OPN methodology is that it allows for analysis at any level, whether it is rigorous low-level analysis or big picture, high-level analysis. It also uses some of the latest techniques in computer science and utilizes computers as more than just calculators. However, since the tool is currently under development, it is not ready for public use. Even when it does become available, it will not garner the same amount of trust as some of the other methods that have been widely accepted by industry. This stems from the fact that while some methods have had a great deal of resources poured into them, especially those endorsed by the DoD, and have been tested on numerous projects, OPN will still be untested. If it is successfully demonstrated to work on a project of nontrivial stature, industrial recognition and use will follow.

1.3 Problem Statement

> **Develop a method that captures the interrelationships of multidisciplinary components, provides a high-level analysis of the technical details of the system, and ensures that all of the requirements are met while also allowing for the design to be condensed into a proposal time frame. It would also be of great benefit if this same methodology could then be used in the remaining stages of the project to provide some of the information required by the DoD in their approved methods, such as CAIV and AoA.**

The previous discussion has shown that aerospace defense systems have stringent, demanding requirements that span both technical and non-technical disciplines. Avi-

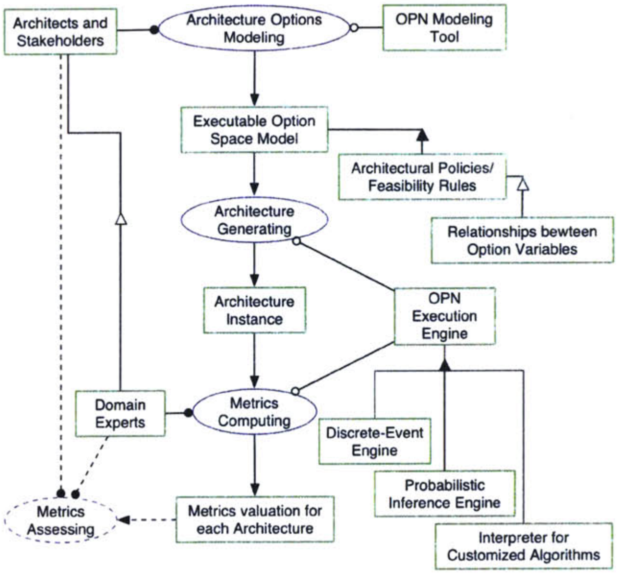


Figure 1-19: OPN modeling and metrics (Taken from [17]).

ation design has evolved from a performance-driven process to one where performance is balanced with life cycle cost, maintainability, reliability, and other goals. As a result, no one person is typically an expert in all the fields that are required for the design of the system. However, as was shown above, the components cannot be designed in isolation from each other by separate people because of the relationships between the components and the competing requirements. Furthermore, the defense industry is inherently proposal driven, thereby requiring interested parties to present their solutions to the customer in a high-level fashion that is detailed enough to establish a competitive advantage yet concise enough to fit within the time restrictions of a proposal cycle.

Figure 1-20 shows the defense acquisition process as described by the Department of Defense's Systems Management College [23]. Though it is not shown, the proposal cycle precedes the timeline. The timeline shown in the figure is more applicable when a project has been awarded or decided upon ahead of time. None of the current methods adequately addresses this subsection of the timeline. The methods examined that are applicable are the matrix-like evaluation, SWOT, feasibility studies, and OPN. However, as was shown, the matrix-like evaluation is arbitrary, obfuscates all detail of the design, and does not take into account the interactions of the components. SWOT deals with the strategic side of the business, answering the question of which projects to pursue, but not how to pursue them. Since it only answers what a company should do and not how to do it, SWOT would thus not be helpful in winning proposals and thus would not be useful in aviation design. Feasibility studies are useful to identify the major issues involved in a project and can provide analysis on one or more alternatives. However, the time required in order to receive an accurate answer of whether an alternative is truly feasible might be prohibitive in

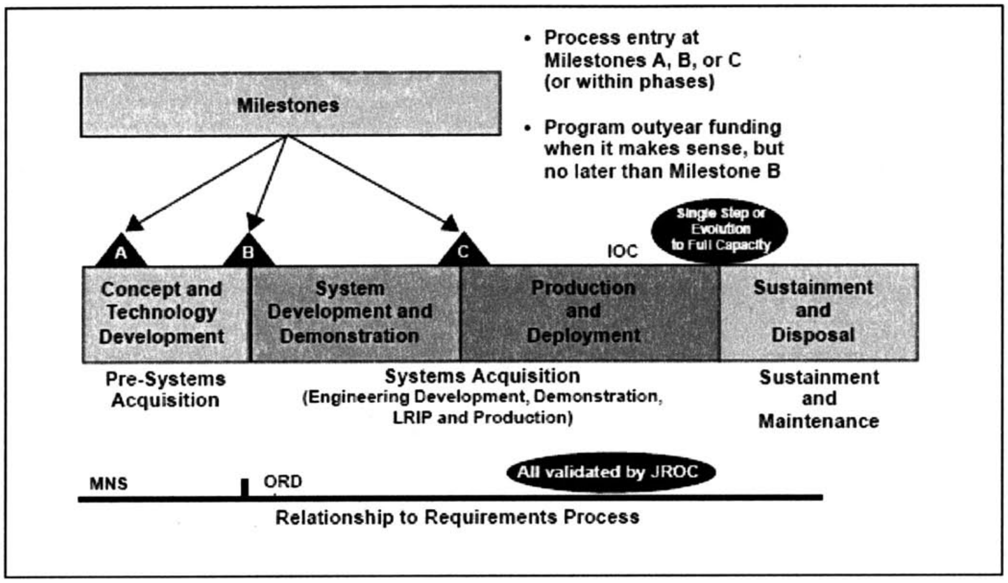


Figure 1-20: Defense acquisitions process (Taken from [23]).

a proposal environment. Finally, while OPN shows promise, it is still not available for widespread use.

The other methods examined, MDO, CAIV, and AoA, are useful design methodologies, but are more applicable to either a different location in the timeline or fulfill a different niche than a high-level proposal design. MDO could be used in a proposal environment, but usually the time required to get even a simplified version of an MDO analysis would be prohibitive for a proposal cycle. MDO is a much better option for a longer time scale, such as when a contract has already been awarded and preliminary design begins or the project is major enough to have a long proposal time frame. Similarly, CAIV and AoA are useful methodologies and provide valuable input into the design of a system. However, they are much more geared towards managing existing projects and providing assurance to the DoD or other sponsors that the project is meeting the non-performance related goals and that the best choice is being made at major decision points. Thus, a gap exists in system design methodologies in support of the proposal stage of a project.

1.4 Thesis Outline

The remainder of the thesis is aimed at addressing the above problem. Chapter 2 of the thesis will give a detailed description of the proposed system design method. Chapter 3 will provide a background for the application selected to demonstrate the method, which is the targeting subsystem of small UAVs while Chapter 4 will provide the details of this demonstration. Finally, Chapter 5 will contain the concluding thoughts of the thesis, including lessons learned from developing the method, findings related to the selected application topic, and possible future work.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

System Design Approach

This chapter will discuss the new system design method intended to assist primarily in the initial or proposal phase of a project. It will begin by stating the assumptions used in developing the method, which will establish the prerequisites for using the method. Then, a high-level description of the method will be discussed, followed by a detailed description of each step in the process. The chapter will then conclude by elucidating the output of the methodology by discussing the deliverables.

2.1 Assumptions

- The designers already have decided that they are going to work on a certain project
- The designers already have requirements for the system, but the requirements are considered flexible

The first assumption eliminates the need for strategic planning prior to deciding where to allocate resources. As will be discussed later, the method still supports strategic planning, but the strategic element is a byproduct of the analysis done towards a certain goal on a selected product. SWOT and other similar methods analyze internal and external factors and decide the best resource allocation strategy to maximize the group's objectives. By contrast, the proposed design method performs analysis on a certain project and then allows for strategic planning within the context of that project, such as addressing the typical "what if" questions of design or if further related work should be pursued.

In a similar vane to assuming that no strategic planning is required in order to decide where to allocate resources, the design methodology also assumes that the requirements for the project have already been established, thereby eliminating the need to create the requirements from scratch. It is important to note that use of "requirements" includes two types of requirements, customer-given requirements and internally-specified requirements. Customer-given requirements could come in the form of a formal requirements document, an Operational Requirements Document

(ORD), or some other form of communication originating from the customer formally or informally stating what the system is required/desired to do.

It is further assumed that the designers have requirements internal to their group. This could include which specific subset of the system they are trying to design to, budget, schedule, personnel assignment, or anything else that is required to manage the design. As a result, the group knows the problem being addressed by their design and the constraints within which the design needs to fit.

While the second assumption dictates that the requirements are already established, it also necessitates that the requirements contain some fluidity. As was briefly discussed in the CAIV description, the requirements are such that they are not a rigid, static set of rules, but instead could be traded with other requirements or other solutions that fit the customer need. A similar statement could be made about the problem being addressed. While the designers already have knowledge of which portion of the system their design will address, their problem statement should contain flexibility. This knowledge without rigidity eliminates the need for the method to contain a problem definition section.

2.2 Methodology

Fundamentally, the methodology rates the various alternatives based upon how the intrinsic properties of its components and the alternative's aggregate performance compares to a user-defined objective function that is based upon selected design criteria. This statement encapsulates many of the steps of the design process, each of which will be described below in greater detail. A flowchart of the entire method is shown in Figure 2-1.

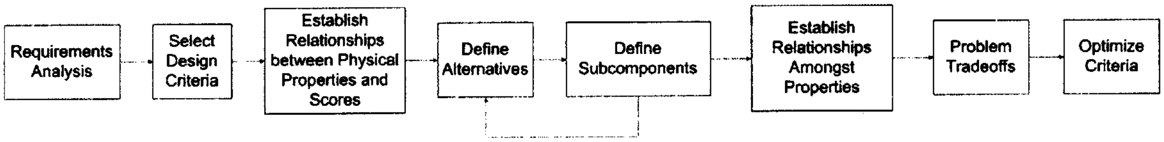


Figure 2-1: Flowchart of the proposed system design method.

The first two steps, requirements analysis and select design criteria, are aimed at setting the specifications to which the system will be designed and evaluated. Next, mathematical functions or bounds are established to objectively translate the system's properties into scores. At this point, the table is set for the design of the actual system. As a result, potential solutions brainstormed in the fourth step. Then, for each alternative, the equations governing the interaction of the properties with each other and the environment are derived. These last two steps are done iteratively in conjunction with the subsystem decomposition to ultimately model the system to the desired level of specificity. Once the equations and models are establish, pertinent trades are performed to adequately explore the design space. Finally, the design is optimized. Each of these steps will be individually discussed in greater detail in the following sections.

2.2.1 Requirements Analysis

The goal of the requirements analysis is to answer two primary questions:

1. What does the customer truly want?
2. What constraints exist on the system, both stated and implied?

These two questions will be addressed individually in the next two sections.

What Does the Customer Truly Want?

To answer the first question, the designers must look beyond what is stated and try to deduce the customer's true intent. The customer's true desires can be obfuscated by any of the following factors:

- Solution-specific requirements
- The customer might not know what they want
- The customer might be unaware of what is possible and as a result, establishes requirements detrimental to an innovation

Solution-specific requirements are those that contain information that dictate using or give unfair advantage to a specific type of solution without it being necessary. An example of this is a requirement for a software protocol to be secure and to be written in C++ when the customer is only really interested in the security. The inclusion of C++ is an implementation-specific detail that eliminates the use of Java, Python, Perl, or any other programming language. As a result, instead of judging the alternatives based upon the provided functionality, security in this case, they are also judged based upon an artificial criterion. The danger of the inclusion of C++ or any solution-specific information in requirements is that it limits the potential solutions to a specific subset on an arbitrary basis, which could lead to a suboptimal solution being chosen unnecessarily.

Another reason why the customer's true desire for a system might not be known is that the customer might not know what they want. This might be the case when the customer is experiencing a new type of problem that needs to be solved, but they are unsure how to do so. However, instead of just describing the characteristics of the problem to be solved, the requirements may be their interpretation of what is needed to solve the problem. Whereas the first type of issue had superfluous information that limited the potential design space, this type of issue is invalid information that does not truly address the issue of interest.

The final factor that can obfuscate a customer's true desires for the system is the fact that the customer might not be aware of the possibilities that exist for a system and thus write the requirements to unnecessarily bias against a novel approach to a problem. Where the previous issue dealt with invalid information due to a new problem being addressed, this deals with information that hinders a new method to an old problem. This issue is similar to the first. However, the first issue deals with requirements containing information that explicitly mentions or physically dictates a specific implementation, thereby requiring that the implementation method be used, whereas this issue deals with requirements that establish objectives that cater to a certain class of solutions.

An example is that the customer could set a requirement for a system to fly under a tree canopy, avoiding collision with trees, with the real goal behind the requirement being that system be able to track and identify objects underneath the canopy. However, instead of flying under the canopy and directly imaging the target, a system could fly above the canopy and image the target with techniques such as small wavelength lidar. If this is the case, the system accomplishes the customer's true objective to be able to track and identify objects underneath a tree canopy, but it is not able to fly underneath the canopy and avoid obstacles as the quoted requirement actually says. While the quoted requirement might be a subsidiary of another requirement that deals directly with tracking an object underneath a canopy, the presence of the quoted requirement is detrimental to solutions that do not require flying underneath the canopy to accomplish the desired task. While never directly stated, the quoted requirement assumes a certain class of solutions to the desired functionality and thus hinders solutions that are not of the assumed nature.

The preceding four paragraphs described scenarios where requirements could be improperly written so as to bias the evaluation of alternatives. When requirements are written properly, they state the functionality that is required and any necessary bounds without mention of how to realize the functionality. They also contain only information that pertains to the problem at hand, not the customer's interpretation of how to address the problem. Finally, good requirements establish metrics and objectives that are independent of basic approach. However, since requirements provided by customers are not always good, sound requirements, designers must be able to identify the above issues and be able to extract the customer's true motivation from the incomplete, biased requirements they are given. Doing so allows them to create the a design that best meets the customer's needs, which they can then use to leverage a change in the stated requirements.

What constraints exist on the system, both stated and implied?

The second major issue addressed in the requirements analysis is to establish the stated and implied constraints placed upon the system by the requirements. Here, "constraints" and "requirements" are used interchangeably, as the requirements act to constrain the system. The stated requirements are easy to identify and simply need to be extracted from the communication with the customers. The implied requirements on the system are not quite so easy. An implied or derived requirement is one that is not directly stated, but is levied on a system in order to meet a stated requirement or performance metric. For example, as will be seen later, requirements could exist for a system to detect a target at a certain altitude and slant range during the day using an onboard optical camera. The physics of the scenario and the number of pixels required to be able to detect a target act together to dictate a minimum resolution and field of view of the camera. While the requirements never mention the resolution and field of view of the camera, lower bounds can be derived from the existing requirements. These lower bounds are derived requirements for the camera subsystem.

While not explicitly stated, derived requirements are just as important to identify as the stated requirements. As was discussed above, if the derived requirements create an unnecessary advantage to a particular solution, identifying the derived requirements and presenting them juxtaposed with a viable alternative to the customer gives the designers leverage to change the requirements. It is also important to note that finding derived requirements is only needed on the subset of the requirements pertaining to the problem being addressed. Thus, if a design team is working on the targeting subsystem, they do not necessarily need to discover the derived requirements of the data encryption algorithms used in communication.

Requirements Analysis Wrap-Up

The major goals of the requirements analysis task are to establish what the customer truly wants and to identify all the constraints, explicit and implicit, on the system of interest. The true desires of the customer could be hidden in the stated requirements by solution-specific language, the customer not knowing exactly what they want, or the custom not knowing all the available solutions. The designer must identify the customers intent in the presence of this obfuscation and then work with them to develop requirements that more accurately reflect their desires. Also, the designers must analyze the communication with the customer and extract all pertinent derived requirements. These are constraints placed upon the system in order to meet stated requirements or performance metrics. At the conclusion of this step, the designers should know what the customer wants, along with the constraints and performance required of the system.

2.2.2 Select Design Criteria

With the customer's motives established and the system constraints and performance levels set, the next step in the design is to select from all the considerations that go into the system, the criteria by which the system will be evaluated and thus the criteria to which the system will be designed. A small set of examples of design criteria for aviation systems include weight, size, power, and robustness. Selecting the design criteria includes both identifying the criteria and establishing the performance level or numerical bounds on the criteria. As was shown in the CAIV method before, the levels are best specified as a range, such as threshold and objective levels. Many times, these numbers may come directly from the customer, but other times, the numbers may be set by derived requirements.

Several different considerations go into the selection of design criteria. Design criteria might come from any of the following:

- Requirements that dictate the design
- Requirements that are especially important to the customer

- Requirements that are fundamental to the purpose of the system
- The subset of requirements that pertain to the given subsystem being addressed
- Requirements that are hard to achieve

Requirements that dictate the design of a system are important to take into account in the design criteria. These are the requirements that fundamentally change course of a design. Some examples in aviation systems include the ability to hover, fly supersonically, or to achieve stealth. Since these requirements have such a large impact on the design, their influence should also be present in the design criteria. The design criteria could measure their ability to achieve the desired functionality.

Another important consideration in the design criteria are the requirements deemed important by the customer. This makes logical sense since the point of the design criteria is to serve as a basis for evaluating the various alternatives, a process which should be anchored in what the customer desires. The system that best provides the desired functionality requested by the customer should be the one that evaluates to the highest rating. Thus, the design criteria should reflect the customer's priorities. At this point in the design methodology, the relative weighting of the design criteria have not been established, but it suffices to say at this point that the criteria most important to the customer should be included while the least important considerations might be excluded.

Additional design consideration must be given to the properties that are inherent to the nature of the system that the customer is requesting. For instance, if the customer desires a cheap, low-power processor, cost and power should be important design criteria, even if the majority of the requirements deal with the processor's performance. While the inclusion of these criteria is seemingly obvious, it is important not to leave them out of the evaluation.

The design criteria might also simply be the subset of criteria that apply to the problem being addressed. This might be the case if the designers are working on a specific subsystem or component of a larger subsystem. For example, if the designers are working on the communications subsystem of a UAV, they probably do not care what the target location error is for the targeting subsystem. Instead, they are concerned with the requirements specific to their chosen problem and how decisions they make affect overall system parameters, such as weight, power, size, and cost.

A final input to the design criteria might be the relative difficulty of achieving a particular requirement. This could affect the criteria either positively or negatively. If the project is risk-adverse and the requirement is not particularly important, difficult requirements might be left out. However, if the requirement is important, then the difficult task must be given strong consideration and might end up driving the design.

Summary of Selecting Design Requirements

The primary purpose of this step is to select the criteria to which the system will be designed and by which the system will be evaluated. These criteria could be requirements that dictate the design, requirements the customer feels are very important, properties that are inherent to the system desired by the customer, requirements that are hard to achieve, or simply the subset of requirements that deal with the area of interest within a system definition. This step of the design process closes when the criteria used to evaluate the system's design are identified and bounds are attached to the criteria where appropriate.

2.2.3 Establish Relationships between Physical Properties and Scores

At this point in the design process, the customer's needs have been established, the system's constraints and performance levels have been defined, and the criteria used to direct and evaluate the design have been selected with the appropriate numerical bounds and performance levels attached. The next step in the design process is to define how the properties of the system translate into scores in the evaluation metric.

The evaluation metric that will be used is very similar to the one that is used in the matrix-like evaluation. As was discussed in the previous chapter, in the matrix-like evaluation, the criteria are defined and scores are assigned to each criteria for each alternative. One of the problems given for this method was that the scoring was subjective and was often altered to grant a higher evaluation to an alternative of choice. In the new methodology, scores are still assigned to each of the design criteria, but instead of subjective scoring, relationships are created to translate from the system's properties to the scores. These relationships usually take the form of discrete bounds or continuous functions that take a property as input and produce a score as output. For example, if the criteria was weight and discrete bounds were being used, the function might look like the following:

$0.0 \leq x < 0.1lb \Rightarrow 5$
$0.1 \leq x < 0.2lb \Rightarrow 4$
$0.2 \leq x < 0.3lb \Rightarrow 3$
$0.3 \leq x < 0.4lb \Rightarrow 2$
$0.4 \leq x < 0.5lb \Rightarrow 1$

In this case, anything equal to or above $0.5lb$ would be considered outside the acceptable region and the alternative would fail to meet the requirement. Numbers such as 0.5 that serve as the cutoff would be taken from the requirements established in the first two steps. Intermediate numbers would then be established based upon the possible range of values. The bounds do not have to be equally spaced, nor do they have to encapsulate the entire space of possible values. Further implementation details will be provided in Chapter 4. Likewise, a continuous relationship might be a mathematical function such as $y = x^2$ that takes in an input property and returns an output score.

These relationships are defined for each of the criteria selected in the previous step. It is important to note that this is done before the alternatives have been defined. Doing so prevents the translation from properties to scores from being biased by favoritism for a given alternative. Instead, it is aimed at producing an objective evaluation based upon how the system's properties relate to the stated and implied requirements deemed important through the selection of the design criteria. In doing so, not only is the bias eliminated for a particular alternative, but the relations also provide a foundation for the scores. Instead of arbitrarily assigning a value to the first alternative and then assigning all the subsequent scores relative to the first, the relationships lend legitimacy to all of the scores. A definition is created for each score in each criteria. Thus, this step concludes when all of the criteria have relationships defined that objectively translate the underlying properties on which the criteria are based into scores.

2.2.4 Define the Alternatives

The next step of the design process is to define possible solutions to the problem being addressed. A great many methods exist on how to create a list of alternatives from scratch. Three prominent methods used in the design of engineering systems are brainstorming, systematic decomposition of the problem, and past experience. Each of these will be examined in succession.

Brainstorming is a creative method whose goal is to come up with as many solutions to a problem as possible. While certainly grounded in reality, brainstorming encourages atypical ideas that attempt to break down assumptions about the best way to approach a problem. As a result, brainstorming has the potential to produce a wide range of alternatives that have vastly different approaches. To encourage the creative process, typically no criticism of ideas is allowed during the session. A brainstorming session is typically given a time limit prior to beginning and possibilities are recorded as they are suggested, with an emphasis on getting as many thoughts out as possible in the given time frame.

Another possible method to creating a list of possible solutions to a problem is through the use of systematic decomposition. Two types of decomposition could be used:

- Decomposition of the alternatives into different classes of approaches
- Decomposition of the desired functionality and then list ways to accomplish the tasks

The fundamental difference between the two approaches above is that the first decomposes the alternatives into a hierarchy while the second itemizes the final product and addresses how to achieve each result.

Figure 2-2 provides an example of a systematic decomposition of the alternatives for navigation into different classes of approaches. As the figure shows, the top levels of the tree form the different classes of navigation approaches. These include absolute and relative. In absolute navigation, the position is given in terms of coordinates in space/on Earth independent of other objects while relative gives the position in terms of an object's location in relation to another object whose location is known. From there, the specificity is added to the classes to begin to distinguish them into specific solutions. For instance, the next level of classification is active or passive, where passive means that the navigator does not radiate navigation signals and active means the navigator is radiating navigation signals. After this level, some specific solutions start to develop, such as using a laser rangefinder or celestial navigation. While Figure 2-2 is not necessarily a complete hierarchy of navigation it serves to illustrate the method.

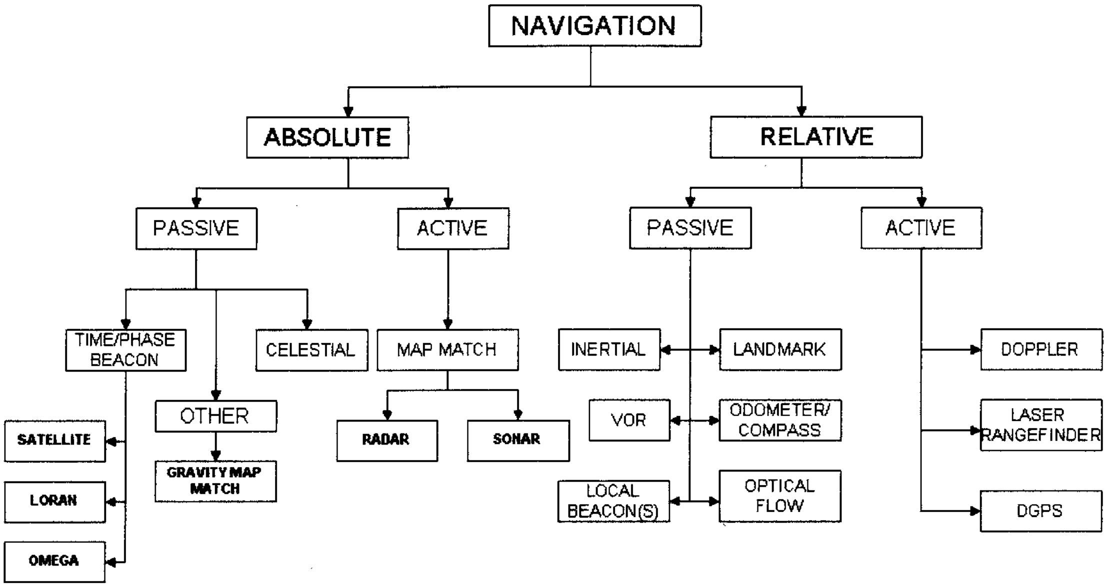


Figure 2-2: Classification of different navigation methods.

The other approach of systematic decomposition involves the itemization of the components of the desired functionality and then listing different ways of arriving at these components. This is illustrated in Figure 2-3. According to Figure 2-3, the components of a navigation solution are position, change in position, orientation, and change in orientation. A list of different ways to obtain these components is listed below the components in the figure. In order to obtain a navigation solution, a selection must be made for each of the components. An advantage of this approach is that it identifies combinations that might not otherwise be conceived.

The final method for creating the list of alternatives that will be discussed is basing the list on past experience. In this method, the designers start with ways that others have used to accomplish the task in the past. For instance, in targeting, a subset of previously uses techniques includes the use of a laser designator, a laser pointer, and visualizing the target with a camera. The legacy solutions can then be supplemented with next generations solutions, derivative solutions, or entirely new solutions. The traditional solutions act to start the thought process about how to approach the problem and then spur other methods from there.

Once a list of potential solutions has been created by any of the three methods above or another method not discussed, the designers need to down-select from all the possibilities listed to the alternatives that are to be evaluated as part of the design trades. This process is influenced by a large number of factors, including, but not limited to the feasibility of the proposed solution, the customer's desired

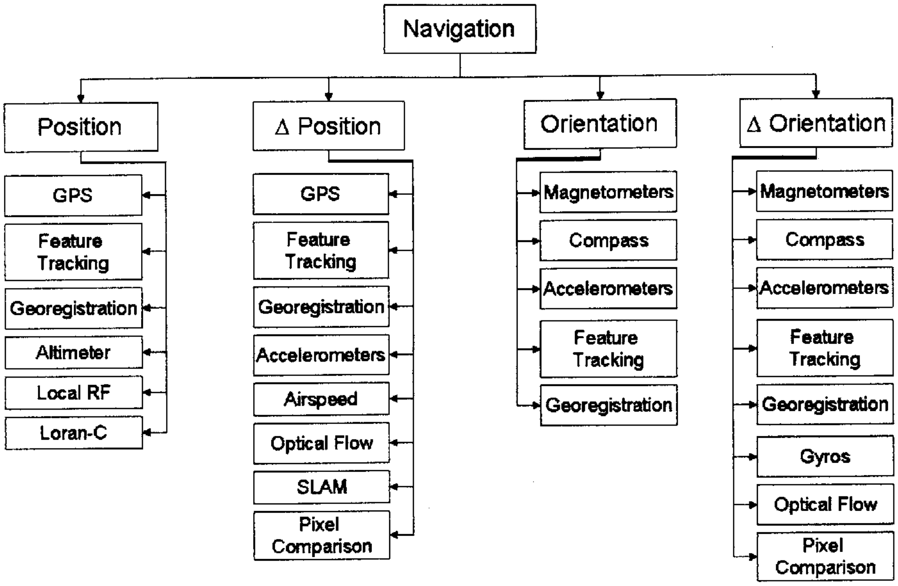


Figure 2-3: Components of a navigation solution and how to obtain the components.

functionality, the technical maturity of the solution, the amount of acceptable risk, the amount of time and money available, and the skill set and experience of the workers on the project. This down-selection is not an evaluation of which choice is optimum, but instead is a reduction of the list to a more manageable size so that the method evaluation can be performed within the given timeframe and with the given resources.

Thus, the list creation and subsequent down-selection should yield a list of the most promising alternatives given the explicit and derived requirements. This final list should be of a size that allows the design team to evaluate each alternative on the list as possible solutions to the problem addressed by the design.

2.2.5 Define Subcomponents

With the alternatives defined and down-selected to the ones of greatest potential, the next step in the design process is to define the subcomponents. As shown in Figure 2-1, this can also be seen as an iterative process with defining the alternatives. Thus, instead of having a linear flow, the method allows for MDO-like decomposition and modeling. This iterative process allows the designers to decompose the problem into a hierarchy of subsystems and components, as was described in Chapter 1. In the first iteration, the alternatives for the problems are defined on the system level. The next iteration then decomposes each alternative into its subsystems and components. At the leaves of a full-depth hierarchial tree, the decomposition yields an enumeration of individual components that could be used for a specific type of method. For instance, if the problem of interest were navigation, for the specific solution of GPS + IMU, the leaves of the hierarchical tree would enumerate the different model GPSs and IMUs that could be used (i.e., for GPS, Trimble TA-12S, Garmin GNC 300XL, etc).

The decomposition of the system should continue until it is one level below the level of specificity at which the designers want to evaluate. This allows them to uses the properties of the components comprising the alternatives that they are evaluating, leading to an objective evaluation of the alternatives. Thus, this step concludes when the alternatives have been decomposed to include one level below the depth at which they will be evaluated.

2.2.6 Establish Relationships Amongst Properties

As stated in the previous section, the properties of the alternatives can be used to evaluate the alternatives in conjunction with the design criteria defined above. The next step aims to establish how the properties can be determined. This often includes defining any governing equations of the properties for each of the alternatives remaining on the list or coming up with measurement metrics for the properties that are not defined by a specific set of equations.

In terms of defining the governing equations, a wide variety can exist in the complexity of these equations. The relationships could be as simple as summing the component contribution to the whole or as complex as requiring a model of part or all of the system. An example of a component-wise summation is the calculation of the system's total weight. Each component of the system contributes its individual weight so that the total system weight is just the sum of the weight of all the components. An example on the more complex end of the scale is the modeling required for such properties as target location error. In this example, the property depends on many variables, both intrinsic to the system and within the environment. Depending on the targeting method used, the target location error could rely on the amount of error present in several components in the system, their speed in providing an answer, their resolution, and many other properties. The target location error also could depend on environmental factors such as the vehicle's altitude, the slant range to target, and many more. As a result, the target location error of a particular method depends on numerous properties of the components of the system, such as their accuracy and resolution, as well as numerous environmental factors.

The properties that are not governed by equations typically require simulation, prototyping, or past experience in order to obtain an estimate of the property. An example of a property that is not governed by equations is the amount of time taken to run a section of software. This property might be determined by running a section of the code and then extrapolating the total time. Another method would be to run existing code that is similar in nature and time how long it takes to run.

It is important to note that on the complex analysis such as would be required to model target location error or the estimation of the time taken to run a complex algorithm that has yet to be written, the level of fidelity of the measurement needs to be traded with the time required to conduct the analysis. In the case of the target location error, the model could be as simple as a few equations defining the physics and geometry of the vehicle to the ground or as complex as a full-blown simulation that takes into account every factor and performs error analysis. While the latter produces more accurate results, it takes longer to develop and thus might not fit into the timeline of the design, especially considering the early stage of the project.

Also, within a system, different fidelity models can be used for different properties. Perhaps target location error is an important system property, so a full simulation is created. However, perhaps communication range is a lower priority and as a result, a very simple relationship is defined. By the end of this step, the relationship between all of the properties are defined, either through equations, modeling, statistical analysis on sample runs, or any other method used.

2.2.7 Problem Tradeoffs

The next section of the design process performs the trades necessary to supply the information required to make the design decisions. The previous statement implies the trades provide the information that allow designers to make decisions, and this is indeed the case. Since the trades are used to justify the decisions, the design is a reflection of the trades performed. Given the impact of the trades, the trade space must be thoroughly analyzed to see how best to investigate the important issues of the design. Identifying the important issues of the design was the goal of defining the design criteria, and thus the design criteria are a logical place to begin when deciding which trades to perform. Typically, at least three different types of analysis exist within the design criteria. These include sources of error analysis, options analysis, and future scenario analysis. Each of these will be examined below.

Sources of Error Analysis

Analysis of the sources of error is not a traditional trade in the sense of making a decision between a variety of options or how to balance many competing factors, but instead tends to supply information needed in other trades. To this end, a common goal of performing this type of analysis is to identify all the sources of error and then to establish the each source's bound necessary in order to meet the system's requirements. This is typically done by performing sensitivity analysis.

Sensitivity analysis is designed to establish how much of a change in output is derived from a known change to a given parameter. By holding all other sources of error constant and varying only one source, the designers can establish the upper bound of error tolerable by the system for the source of interest. This upper bound can then be used to establish derived requirements for the system. Sensitivity analysis will be discussed in greater detail in Chapter 4. It is also important to note that for a particular source of error, different upper bounds can be established for different methods, as each alternative might have a different sensitivity to the source of error.

Options Analysis

For any new derived requirements established by the sources of error analysis or for any preexisting requirements, the designers need to evaluate the best way to meet the constraints. In the case of new, unforseen requirements, this might involve another iteration of defining alternatives. It could also involve trades examining the different options that the designers have. In general, the options analysis family of trades try to give the designer information about how the system would be affected if various choices were made. It allows them to vary parameters in a controlled way such that the difference in performance can be attributed only to the choices being made, thereby permitting valid conclusions to be drawn about the effects of the choices. It is a broad class of trades and is thus hard to generalize. For conceptual systems, such trades might involve examining different implementation methodologies. For example, in a software project, an options trade might examine the benefits and detractions of using a particular programming language. However, as was stated in Chapter 1, the design methodology was primarily developed for human-made, physical, dynamic, open systems, so the primary focus will be on physical systems.

For physical systems, one of the most common types of trades in the family of options analysis is component analysis. This group of analyses includes class of component analysis and individual component analysis. In a class of components analysis, the designers first create classes of a particular type of component, usually based upon an important property. An example might be cost, where the designers could divide the available components into low, medium, and high cost classes. Then, the designers would take sample components from each class and use their properties in each method to see how the system would perform. Thus, instead of picking a particular component, the designers pick a class of components to use in a design. By using this form of analysis, the designers reduce the amount of work they have to do in investigating different components.

Similar to the class of components analysis is the individual components analysis. Instead of dividing the components into classes, the individual component analysis deals with all the available components. Each component is analyzed for its effect on the system's performance. This method is more thorough, but also more time consuming. As with the sensitivity analysis, both the class of component and individual component trades will be examined in greater detail in Chapter 4.

Future Scenario Analysis

The final type of analysis that will be discussed is the future scenario analysis. In this type of analysis, some key parameter(s) in the evaluation is(are) modified to evaluate how the system would be affected. In a component-based, physical system, a common type of future scenario analysis is future innovation analysis. In future innovation analysis, designers generally change a property of interest to be that of some future hypothetical value. The change usually reflects an improvement in the property (hence the innovation in the name). For instance, the designers might be interested in seeing the effect on the system if a component could achieve the same level of performance while being half of the weight. However, negative changes could also be investigated, such as the effect if a component doubled in price due to scarcity in supply.

Future scenario analyses are generally interested in answering the "what if" questions of system design. As with the error analysis, future scenario analysis is more an informative type of analysis than a decision analysis. If desired, however, it could also certainly aid in component selection. For instance, if the designers have reason to believe the customer will want to make a requirement more stringent in the future, they could perform a future scenario analysis with the future requirement level. In such a scenario, the alternative that evaluates the highest with the current requirement might fail on the future requirement. Thus, if an alternative exists that gives comparable performance to the best alternative on the current scenario and significantly outperforms all others on the future scenario, the designers might decide to use the alternative that performs better on the future scenario in their current design. Also, future scenario analysis can also be used to help justify development of a new product. Designers could show the impact on a system by achieving some target properties and thus use the improved performance to solicit funding.

2.2.8 Optimize Criteria

The last step in the design process is to optimize the design given the constraints and the goals. Like was shown in Figure 1-11 in the MDO discussion, while optimization is the goal and is the basis for which the step is constructed, in actuality, all different levels of analysis could be performed. The simplest "optimization" analysis is just to enumerate the possibilities, define an objective function, score the alternatives based upon there properties as defined above, plug the scores into the objective function, and then pick the highest result. In this case, the objective function would be a mathematical function of the scores for each of the criteria and can be seen as being much the same as matrix-like evaluation. The objective function would indeed be a weighted summation of the scores for the criteria. The difference between the methods is that the new method would be objectively scored based upon the system's components and would have the trades to justify the scores provided while the matrix-like evaluation would simply have the scores subjectively assigned by the designers. On the other extreme, the designers could do a rigorous NLP optimization as shown in Chapter 1 in the MDO discussion. This would formalize the constraints and goals in equality and inequality statements and then would use NLP techniques to converge to an answer.

2.3 Deliverables

When taken to fruition, the system design method answers the question about which alternative ranks the highest given the user-defined objective function. This can help the designer select which alternative is to choose amongst the options being considered. However, unlike the matrix-like evaluation method, it also provides details about how this conclusion was reached. It allows the designers to see trends in the data as derived from the trade studies conducted. As a result, nothing is hidden in the design and selection process.

In addition, the designers now have a flexible model of the system that can enumerate instances of alternatives. As was shown, this opens up options for evaluating future possibilities by considering components that have properties that allow for future innovation. This ability allows designers to consider the "what if" questions of design and allows. The method could also be used to have strategic planning by attempting to deduce the yield of future projects. Another source of flexibility is that the design methodology allows for different objective functions to be defined based upon the criteria that are used. The different objective functions allow for different evaluation for different perceived scenarios. For example, on objective function could punish technology that needs a longer gestation period while another could reward such a property. The former scenario would be an objective function for a project that needs a solution quickly while the latter would be used for a research project whose purpose is to develop new technology. This flexibility is useful in developing a business strategy.

Concretely, the system forces designers to select design criteria that are important to consider for the system and brainstorm alternatives for the system. Once the most promising alternatives are identified, decomposition of the alternatives into their components occurs and the pertinent properties of the components are identified. Then, in addition to all the useful trade data mentioned, an objective function is created. All of these objects or pieces of data are critical to the design and forcing the designers to identify them will help considerably on any proposal effort. If a proposal is not involved, it will at the very least aid in their decision making process.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Application Background

This chapter introduces the general application of man-portable UAVs, hereafter referred to as small UAVs. It will first describe the motivation for examining small UAVs and then look at sample requirements for a small UAV system. The information in this section will be provided through the examination of two small UAV programs, the Army's Future Combat Systems (FCS) Class I, the smallest of the four UAV classes, and the Special Operation Command's Rucksack Portable UAV (RPUAV). The chapter will then examine the major problems that current small UAVs are experience and will then conclude with a statement of the precise problem to which the previous chapter's system design method will be applied.

3.1 Application Motivation

Recent operations in Iraq and Afghanistan has shown the value of UAVs in combat. Two of the vehicles that have been the most successful and have received the most press for their performance have been the RQ-1 Predator and the RQ-4 Global Hawk. Both aircraft have takeoff weights in the thousands of pounds and a wingspan of over 39 ft, with the Global Hawk being considerably bigger than the Predator. The Predator has the advantage of being equipped with Hellfire missiles, allowing it to engage the enemy instead of simply providing reconnaissance. Given the planes' success in combat, UAVs are starting to become universally recognized as a valuable military asset.

However, for soldiers on the ground, getting access to Predator- and Global Hawk-sized vehicles is quite difficult. Due to their relative scarcity, these UAVs tend to operate on a special need basis only and require pre-approval in order to fly a certain mission. Only under extremely urgent circumstances would a Predator or Global Hawk be dispatched due to a unit's request or spontaneously rerouted while in flight. As a result, individual platoons cannot currently rely on having Class IV size UAV support for their every day operations.

The Army does not think that the individual soldier's need for UAV support can be feasibly fulfilled by the Global Hawk and Predator due to, amongst other reasons, logistics and economics. Since the UAVs are so large, there is no way for the platoon to transport it with them in combat. Plus, the aircraft require runways to takeoff and land, require specialized maintenance, and require a specialize ground control station, all of which the platoon could not provide. In addition, the cost of the Class IV vehicles is prohibitive to supporting individual platoons.

Instead, the Army and other branches see small UAVs filling the role of providing individual soldiers with UAV support. According to the Army's FCS specifications, small UAVs will be part of the future equipment for every platoon. The support that the UAVs will provide include reconnaissance, scouting (what's over the next hill), and targeting for precision engagement. The current philosophy for small UAVs appears to be to make them cheaply so that if the are lost, it is not a large issue to replace them. This certainly would not be the case for a Predator or Global Hawk, nor will it clearly ever be the case for soldiers.

3.2 Major Current Problems with Small UAVs

While the current small UAVs fielded in Iraq and Afghanistan have provided a valuable service, they have also experienced some major problems or have major deficiencies in their capabilities. One major problem in current small UAVs is that they are losing their GPS lock when operating in "urban canyons." Urban canyons is the military term for an urban setting amongst buildings that form a canyon-like view of the environment. In such settings, the obstacles either block the GPS signals so that the receiver cannot acquire the necessary number of satellites to obtain a position lock or the multipath caused by the obstacles produces a position solution that is inaccurate. Since GPS is the primary source of navigation for most small UAVs, the operator has a choice of either pulling the vehicle up and out of the urban canyon so that it can reacquire the requisite number of satellites or continue flying with either no position solution or one that is inaccurate. Both scenarios are not desirable. If the vehicle pulls out of the urban canyon, the small UAV's mission will most likely be lost. If the operator chooses to continue on without a proper position solution, the vehicle is at a greater danger of striking an object or the operator is required to fly solely with the aid of the camera and dead reckoning. This is a much more labor intensive task and would most likely not meet the self location error requirement.

Another problem that current small UAVs are having is related to the recovery method for the vehicles. The current method of landing the vehicles is to bring them in for a controlled crash, using a belly skid method to bring the vehicle to a stop. The energy of the vehicle is dissipated as the vehicle's are designed to break apart into their constituent pieces upon impact. In theory, this allows the operator to simply pick up the pieces, put them back together, and relaunch the vehicle if necessary. However, in practice, the vehicles are incurring more damage than anticipated. As a result, maintenance costs are higher than desired and availability is not as high as is desired. Additionally, while belly skidding saves the weight and drag of having landing gear, it affects where components might be placed. For instance, since the vehicle is hitting on its belly, the cameras cannot be placed on the bottom of the fuselage for fear of damage. However, this is the natural place for the camera in order to allow the vehicle to perform surveillance. A definite need exists for an automatic landing system. This would save maintenance cost, increase availability, allow the vehicles to be constructed with less durable and thus lighter material, and would allow freedom in the placement of components.

An example of a major deficiency in current small UAVs is that the vehicles lack targeting capabilities. In their current states, small UAVs are essentially flying cameras with navigational awareness and potentially a specialized sensor suite. As implied in the previous paragraph, the navigation is typically done using a GPS receiver. The vehicle then usually has an optical camera and a night vision camera. However, the vehicles often lack anything that aides in targeting. They do not have laser pointers or laser designators, but instead the limited capability they do have is simply in ascertaining what the camera is pointed at and deducing where that field of view is located on the Earth. Within the vision-based targeting, intelligent algorithms such as object detection and tracking are typically not employed to give a solution. The current wind conditions are not even provided, which would be crucial to know for accurately engaging the target. All told, little is done to provide a targeting solution other than an equivalent to the dead reckoning in navigation.

3.3 Application Problem Statement

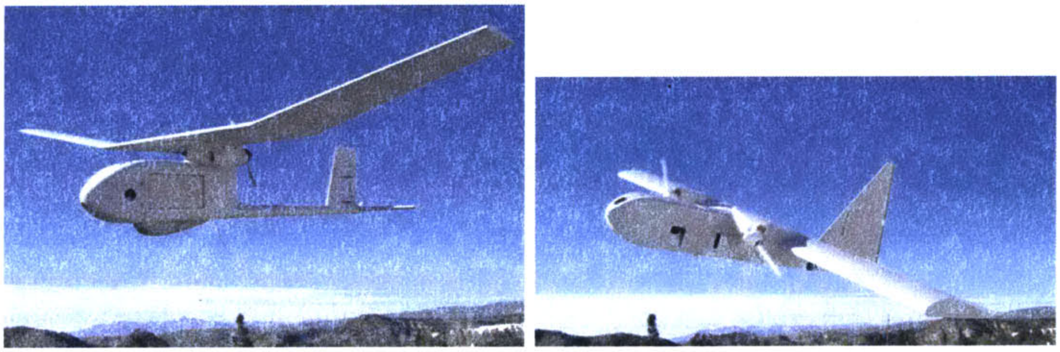
The application to which the system design methodology will be applied is the targeting subsystem of small UAVs. The design methodology will be applied to consider several different targeting alternatives that small UAVs could use. The sample requirements provided in this chapter will serve as the requirements to which the targeting system will be designed. The alternatives will not only take into account the targeting related requirements, but instead all of the requirements to ensure that the targeting alternatives are in compliance with the system specifications. The details of the system design application to the targeting subsystem of small UAVs will be discussed in the next chapter.

3.4 General Small UAV Requirements

In order to simulate the design of a small UAV targeting system, sample requirements are given below. The actual requirements for almost all aviation-based military systems, small UAVs included, contain sensitive information that is either classified or, at the very least, restricted from public release. As a result, the actual requirements for the Army FCS and the SOCOM RPUAV will not be included. Instead, general sample requirements will be given that cover the major themes of the projects without divulging sensitive information. The current small UAVs will also be used to help set the appropriate levels for the requirements using information about the vehicles that is taken solely from the public domain. While the sample requirements are not as detailed or all-encompassing as the actual ones, they accurately capture the major design drivers for the project. If an actual design were to be done, the detailed requirements would be used. The method would not need to change, however. Also, it should be noted that the requirements will be given as a range such that a threshold value must be met and an objective value will serve as the goal for the requirement. This idea was introduced in Chapter 1 during the discussion of the CAIV method.

As will be discussed in greater detail during Chapter 4, four major design criteria for any aviation-based project are weight, size, power, and performance. In addition, cost is a critical dimension to almost every project. The requirements for each of these areas will be discussed in the subsequent paragraphs.

First, the weight of the targeting system will be examined. The current small UAVs that are most widely used are the Raven and the DragonEye, both made by AeroVironment. Pictures of the two vehicles are shown in Figure 3-1. The Raven is the vehicle of choice for the Army and the DragonEye is the vehicle used by the Marine Corps. According to Aerovironment's website, the DragonEye has a total weight of $5.9lb$ and the Raven has a weight of $4.2lb$ [1]. From this data, the weight of the overall vehicle is assumed to be within the $4-6lb$ range. Typically, a majority of the weight of the overall vehicle is taken up by the batteries and the structure. As a result, the absolute maximum the targeting system could weigh and still have the vehicle achieve the desired weight is $2lb$, with the desired weight being $0.5lb$, which is just over that of a typical Raven payload of $0.45lb$.



(a) Raven (b) DragonEye

Figure 3-1: A picture of AeroVironement's Raven and DragonEye, used by the Army and Marine Corps, respectively (Taken from [1]).

In this case, size refers to the volume taken up by the targeting system. The targeting system needs to fit in the fuselage of the vehicle, so quantifying the requirement for the size is, in essence, determining the volume available in the fuselage for the targeting system. Again, the Raven and the DragonEye are used to assist in this process. According to AeroVironment, both have a length around 3 ft, tip to tail [1]. However, as can be seen from Figure 3-1, in terms of storable volume, the fuselage of the Raven extends only to the trailing edge of the wing will the fuselage of the DragonEye extends all the way to the tail. The abbreviated thick fuselage of the Raven is connected to the tail by a significantly thinner rod, which is thin enough to prevent the storage of components. As a result, based upon a cursory examination of the pictures of the vehicles, it would appear that the Raven has less storage space in the fuselage than does the DragonEye. As a result, the estimated volume of the Raven's fuselage will be used as the threshold value for the size requirement. According to [11], the diameter of the Raven's fuselage is $0.3ft$. It was assumed that the width and height of the fuselage were roughly the same. By examining pictures of the Raven both on AeroVironment's website and on the web in general, it appeared that the wide portion of the fuselage accounted for approximately 40% of the length of the vehicle. Since the length of the Raven was stated as $1.1m$ ($3.609ft$), the length of the fuselage in which components could be stored is approximately $3.609ft * 0.40 = 1.4436ft$. Thus, the approximate volume of the Raven's fuselage is $0.3ft * 0.3ft * 1.4436ft$, or $0.1299ft^3$, which is $3679.038cm^3$. This serves as the threshold value, with the objective being $1000cm^3$.

The Raven can serve again as the basis for the power and cost requirements. According to [6], the cost of an individual Raven is \$35000. As a result, the maximum amount of money to be spent on the targeting system would be \$10000, with a preference to be under \$5000. The power is harder to estimate due to the unavailability of the exact power consumption of the Raven during a typical flight. However, it is estimated that a vehicle of the Raven's size would need $50W$ during a flight, with the significant majority going towards powering the motor. If 10% of the power is used for powering the onboard equipment, then the threshold for power would be at $5W$.

The final criteria to be considered is performance. The fundamental evaluation of the targeting system's performance is how accurately it provides the target's location. This can be quantified by determining the magnitude of the error in the location returned by the system as compared to the target's actual position. This quantity will be referred to from now on as the target location error, or TLE. A definition of the total system error is [15]:

$$TSE = \sqrt{TLE^2 + (VEE * T)^2 + DE^2} \tag{3.1}$$

where TSE is the total system error, VEE is the velocity estimation error, T is time, and DE is the delivery error. For a stationary target, the above equation simplifies to:

$$TSE = \sqrt{TLE^2 + DE^2} \tag{3.2}$$

General Cannon defines delivery error as the accuracy with which the projectile or missile can be delivered to a desired aimpoint. A common measure of this error is the circular error probable (CEP), which is the radius of the tightest circle centered at the target at which 50% of the projectiles fall within the circle and 50% of the projectiles fall outside the circle. In order to be able to set the magnitude of the TLE requirement, both terms in Equation 3.2 need to be investigated.

The delivery error of bombs has evolved over time. In [22], Mok references a study of the British Royal Air Force's strategic bombing of Germany during World War II that said of the bomber crews that had thought they had hit their targets, only one third of the bombs had actually landed within five miles of the target. He continues to say that the Norden bombsight was perhaps the most accurate of its time, yet it still had a CEP of over a kilometer [22]. This led to the Allied Forces deploying fleets of bombers to drop hundreds of bombs in order to destroy the necessary targets. Mok states even though the technologies of bombsights improved over time, the CEP during the Korean War was still over $300m$, falling to around $100m$ by the Vietnam War [22]. He makes the statement that the improvements in CEP could be attributed to improvements in the aircraft and that even by the end of the Vietnam War, the United States had very few precision guided munitions in its arsenal [22]. The ones they did have were used for air-to-air combat, such as the Sidewinder and Sparrow, as opposed to air-to-ground capabilities [22]. Mok states that since the nuclear weapons were the driving force behind military policy of the time, precision was neglected in favor of megatons of power [22]. The troubled that the US had in precision striking during the early phases of the Vietnam War, however, led to the development of laser-guided bombs (LGBs) [22]. During the interim time between the end of the Vietnam War and the start of the Persian Gulf War, the Maverick and Tomahawk cruise missile were also developed. As a result, while the precision guided munitions accounted for only 4.3% of the tonnage of munitions dropped during the first Persian Gulf War, they accounted for 75% of the serious damage to Iraqi targets [22]. By the time of the conflict in Kosovo, the CEP of the precision-guided munitions had fallen to within the blast radius of the their payloads [22].

The evolution of precision targeting can also be seen by looking at Table 3.1. Table 3.1 looks at the case of trying to hit, with a hit probability of 90 per cent, a target measuring 60 x 100 feet using 2,000 pound unguided bombs dropped from medium altitude. It displays the number of bombs required to accomplish the task, the number of aircraft necessary to deliver the bombs, and the CEP of the munitions.

Table 3.1: Evolution of CEP over time (Taken from [29])

Conflict	Number of Bombs	Number of Aircraft	CEP (in ft)
WWII	9,070	3,024	3,300
Korea	1,100	550	1,000
Vietnam	176	44	400

The final look at the evolution of CEP involves examining the classes of different munitions currently available, with a look into the future for the desired CEPs of the next generation of munitions. Three classes of munitions are unguided, guided, and precision. These three classes have, or are desired to have, CEPs of 50+ m, 10 m, and 1 m, respectively.

With the CEPs established for the different classes of munitions, the discussion turns back to Equation 3.2, substituting the CEP for guided munitions into the equation for the delivery error. In this case the equation becomes:

$$TSE = \sqrt{TLE^2 + 10^2} = \sqrt{TLE^2 + 100} \tag{3.3}$$

The last variable to define the total error of the system is the TLE. If the TLE was approximately 50 m, then the equation would evaluate to:

$$TSE = \sqrt{50^2 + 100} = \sqrt{2500 + 100} = 50.99m \tag{3.4}$$

However, if the TLE was approximately 1 m, then the equation would evaluate to:

$$TSE = \sqrt{1^2 + 100} = \sqrt{1 + 100} = 10.04m \tag{3.5}$$

In each case, Equation 3.2 is dominated by the term that is significantly larger than the other. This simple analysis shows that the benefit of using munitions with smaller CEP is not realized unless the object can be targeted with comparable error. As a result, the TLE is desired to be at least as small as the CEP of the munition that it is used with in a strike, with the desire to be to make it as small as possible. Thus, for the targeting system being developed, the TLE will have a threshold of 50 m (164.04 ft), with a desired value of 10 m (32.81 ft). These are taken from the unguided and guided munitions CEPs.

The final bit of requirements that need to be defined are the flight parameters that affect the properties of the targeting system. One such parameter is the altitude of the vehicle. The small UAV usually have an operating altitude of no more than 1000 ft, with the typical altitude being around 500 ft above ground level (AGL). Also, the aircraft have Electrical/Optical (EO) and Infrared (IR) camera onboard that serve as the primary payloads. These camera will serve as the means by which to do targeting. The cameras can either be fixed at an certain angle onboard the aircraft or be attached to a gimbal system that allows it to move independent of the vehicle. It will be assumed that the camera is fixed at a 65.38° angle from the vertical (looking straight down). The height and camera angle thus determine the slant range from the vehicle to the target, which is $1200ft$. This is found by defining a right triangle from the aircraft to the target, with the height serving as the vertical leg, the camera angle from vertical serving as the angle from the vertical to the hypotenuse, and the range-to-target (range) serving as the hypotenuse. This geometry will be examined in greater depth in Chapter 4.

One final requirement that is common to small UAVs is that the system must be able to detect targets at the given slant range during the day and night. The nighttime restriction is usually relaxed in comparison to the daytime. Typical targets of interest usually include man-sized objects and military equipment such as tanks and trucks.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Small UAV Targeting System Analysis

This chapter applies the new system design methodology to the specific application of a targeting subsystem for a small UAV. The system design methodology was introduced in general terms in Chapter 2 and the requirements for the small UAV system were defined in Chapter 3. This chapter will get into the specifics of how to perform the different steps of the design process (the methodology) and then demonstrate it actually being applied to the targeting subsystem (the application). In terms of organization, each step in the design process is given its own section, with subsections for the methodology and application. Lower level sections are created as needed.

4.1 Requirements Analysis

4.1.1 Methodology

As stated in Chapter 2, the first step in the design methodology is to analyze the requirements. Section 2.2.1 stated that in the requirements analysis section, designers need to answer the questions of what does the customer truly want and what are the constraints, stated and implied, on the system. While these questions are stated separately, they have a considerable amount of overlap and can be addressed in either order, or even together.

A crucial part of trying to ascertain the customer's true motivation is to look for improperly written requirements. As stated in Section 2.2.1, properly written requirements state the desired functionality and any necessary bounds without mentioning how to realize the functionality. They also contain only the information that pertains to the problem, not the customer's interpretation of how to address the problem. Also, if any performance metrics are established, they are also stated independently of the approach. By contrast, Section 2.2.1 listed three common reasons why a requirement might be improperly written. The three reasons were that the requirement contained solution-specific information, the customer might not know what they want, and the customer might be unaware of the possibilities for the solution. Section 2.2.1 contained examples of each type of reason. The designers of the new system need to be able to identify the biased requirements in order to discuss them with the customer in the hopes of getting them changed.

Once the improperly written requirements are identified and "fixed," the designer's attention turns to all the proper requirements to ascertain what the customer desired. Typically, the requirements range from high-level objectives to detailed performance metrics of a specific components. Also, redundant requirements or requirements that do not apply to the specific part of the problem being addressed may be present. This is especially true if the system being designed is a subsystem of a larger system. Given the potential existence of superfluous information, the requirements must be trimmed down to only the ones pertaining to the task at hand. Also, if not already done, the designers need to organize the requirements by functionality, subsystem, or some other logical means. If applicable, the designers should identify the requirement's priority level, when the required functionality should appear, or any other pertinent information. Spreadsheets are typically an excellent way of achieving this organization.

Once the requirements are organized, the designers should identify which ones are high-level objectives versus specific performance metrics. They should look for commonalities in the motivation behind the required functions. For instance, if the system being designed is an online order form for a company and requirements exist for a secure interface with the server and for the user's sensitive information not to be displayed on the screen, it appears that the customer wants to protect users from identity theft. While this motivation is not directly stated, it can be inferred from the nature of the requirements and the environment of online purchases. Of course, not all implied motivations are as clear cut.

When several such criteria have been identified, the designers can look to combine these criteria into even higher level motivation. By repeating this process, a motivation hierarchy can be created for a system, with the stated and derived requirements serving as the leaves. This motivational hierarchy can serve as a tool to use with the customer to address improperly written requirements. Designers can show the customer what is written, what the purposed changes are, and show that the new requirement still fits within what the customer actually wants.

The final piece of the requirements analysis is to identify any derived requirements. Chapter 2 described an implied or derived requirement as one that is not directly stated, but is levied on a system in order to meet a stated requirement or performance metric. Often times, the derived requirements are solution specific and are, as a result, not ubiquitous. For example, requirements that bound the error to a solution do not necessarily contain generalizable derived requirements since the amount of error might depend on how the solution is obtained. However, if the error is a result of a specific component and all solutions must contain that component, then a generalizable derived requirement can be produced. Chapter 2 discussed the example of a requirement for detecting an object at a certain distance away setting constraints on the camera used. In this case, the detection was to be done with a camera and thus the constraints on the camera could be done at this stage. Solution-specific derived requirements are set after the solutions have been defined and any governing equations have been established. However, the non-solution-specific requirements can be defined at this step. Identifying them is a matter of searching through the requirements and analyzing how the requirements are to be met. If there is only one way, then the designers need to establish how the requirement constrains the system and note any bounds that are not clearly stated in the requirement.

It should also be noted that, as stated above, the tasks needed to be completed in this step of the design process are often done in conjunction with each other and are only separated here to present the discussion in a coherent form.

4.1.2 Application

As stated above, the main task of the requirements analysis step is to answer the two primary questions of what does the customer truly want and what constraints exist on the system, both stated and implied. These two questions will be answered relative to the targeting system for small UAVs in the subsequent sections.

What Does the Customer Truly Want?

In order to understand the customers motivation for the targeting system on small UAVs, it is informative to start by motivating the project as a whole. This will provide background on what the system is supposed to do and thus help show how the targeting system fits into the big picture. Once the system-wide motivation is established, the analysis will turn more to the specific targeting system.

Small UAV System Figure 4-1 shows the hierarchy of the customer's motivations as extracted from the actual detailed project requirements. The tree was created such that it is organized horizontally, placing the root of the tree on the left. Also, it should be noted that each of the stated requirements for the project would be leaves in the tree, thereby placing them on the far right. For example, the requirement stating that the system must weigh less than a specified weight and be man-portable would be a descendant of the "Small, Light" node. However, as in Chapter 3, sensitive nature of the requirements prevent them from being included. Also, some of the requirements could fit under multiple categories. For instance, the endurance requirement could fit in at least the "Vehicle Specs" and the "Endurance" nodes. Finally, it should be noted that the figure is aimed around the unit using the small UAVs, termed the small unit. As discussed in Chapter 3, the military sees small UAVs filling the role of supporting individual platoons or SOCOM teams. As a result, it is natural to focus on how the UAVs affect the small units and why the customers desire what they do.

Figure 4-1 shows that at the highest level, the customer wants to add value to the unit using the small UAVs while minimizing the burden required to operate it. The value associated with a military asset can fall into one of two categories, the ability to perform a desired task or the ability to help an entity survive. These two categories thus serve as the subdivisions of the "Adds Value to Small Unit" node, with the ability to perform a desired task being labeled "Enhances Effectiveness" and the ability to help an entity survive being labeled "Enhances Survivability."

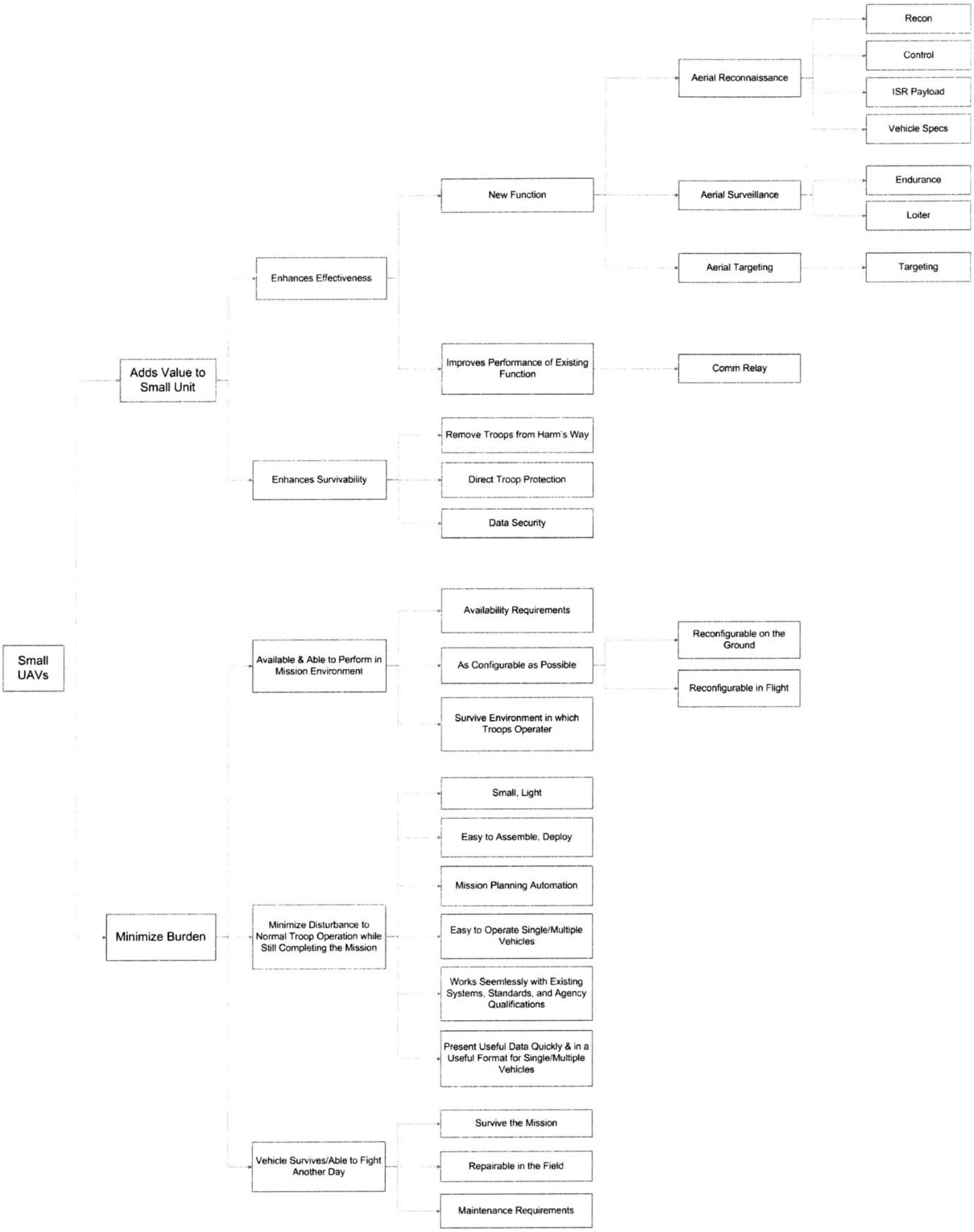


Figure 4-1: Tree of the customer's motivations for small UAVs.

The desired tasks that the small UAV are asked to perform can be divided into either new functions that the unit could not previously perform or improving upon the existing tasks that the unit is already asked to do. This is shown in Figure 4-1 as the "New Function" and "Improves Performance of Existing Function" subnodes under the "Enhances Effectiveness" node. The new functions are, as the name suggests, aimed at stretching the capabilities of the small unit beyond its previous bounds. The new tasks that are stated in the requirements include aerial reconnaissance, aerial surveillance, and aerial targeting, with the subnodes of each of these being the classes of stated requirements that support these tasks.

Each of the three new tasks provides an airborne capability to what the unit is usually tasked to do on the ground. Reconnaissance involves gathering intelligence about an area or object. It could be seen as answering the question of what is over the next hill or similar such questions. Aerial reconnaissance deviates from ground-based reconnaissance in speed, flexibility, and standoff distance. Aerial reconnaissance can get to an area at faster speeds and is not hindered by terrain, thus allowing it to move quicker once there. The flexibility of terrain navigation also combines with the birds-eye view of the situation to allow the vehicle a less obstructed view of the world. Plus, should a threat arise, the vehicle has a greater distance between itself and the threat, allowing it a better chance of escaping. Thus, aerial reconnaissance is seen as different enough from ground-based reconnaissance to be placed into the new function category.

This same line of thinking can also be applied to the other two functions. Surveillance refers to the task of persistently observing an object of interest. The advantage of aerial surveillance can be seen in the civilian application of a police chace. In order to track a subject while minimizing the risk to the public and to the officers involved, many police departments of major cities use helicopters to track a suspect rather than cars or motorcycles. The bird's eye view allows for greater ease of tracking and makes it much harder for the subject to get away. For targeting, the flexibility also applies, as well as the standoff distance.

The existing function that the UAV could improve upon is providing a communications relay. The system provides a communication relay when the system connects communications between two or more groups of troops or equipment that are geographically separated. Much like a cell phone tower transmits calls, the system can act to transmit a signal between parties. The added elevation of flight can greatly increase the range of transmission and reduce the impact of elevated terrain.

In general, the military sees autonomous machines taking over tasks that are dull, dirty, and dangerous. Dull functions are those that are highly repetitious and require little thought or skill, such as airborne refueling. Dangerous missions could include the type of unknown over-the-next-hill surveillance usually provided by scouts. Dirty tasks are usually ones that involve lethality, such as targeting. The tasks mentioned above could be considered specific instances of the three d's (dull, dirty, and dangerous), with more of these types of tasks to be added in the future.

The small UAVs sole addition to the small unit, however, is not just in accomplishing tasks. As mentioned above, it also enhances the small unit's ability to survive. The foremost way that it does this is by removing the troops from harm's way. As was done in Chapter 3, the Raven and DragonEye can give an estimate of the range of small UAVs. According to AeroVironment's website, the DragonEye has an range of 5 km and the Raven has a range of 10 km, with the desire to increase these figures in the future [1]. By having an operating radius in the tens of kilometers, future small UAVs can perform the dangerous and dirty tasks while the troops are back at a distance. The reconnaissance capabilities of the vehicle allow the unit to be aware of their surroundings and provide intelligence that will make them safer by decreasing the chance of an ambush. The reconnaissance capability will increase the information the unit has of the battle, better allowing them to formulate a winning strategy. Also, on the dirty tasks such as targeting, the troops are back away from the area of action. Without the UAV, the unit would have to be present providing either coordinates or lasing the target themselves, thereby putting them in the midst of a battle or at the very least, a dangerous situation. However, with the UAV, the troops could be at a safer distance from the action while the UAV is accomplishing the same functionality.

Another way that the customer requires the small UAV to protect the troops is through direct troop protection. This includes requirements that are explicitly designed to look out for the safety of the small unit. An example of this is a requirement for the system to have a sensor onboard that detects the presence of harmful biological chemicals in the environment. The requirement does not have the system perform an action, but instead it provides information that aids in keeping the troops safe.

The final way that the small UAV protects the troops is by having data security. This class of requirements is aimed at preventing the UAV from being used against the troops. If the enemy could decrypt the information that the UAV is sending, the potential exists that they could find out information about the unit that would make it less safe or that they could manipulate the environment to have the UAV provide false information. It was stated earlier that one of the advantages of using the UAVs is that they provide additional information about the unit's surroundings or the battle, thereby allowing the unit to formulate a better plan for success and survival. However, if the information is accessible by the enemy, then they could taint the information to fit best with their strategy of attacking the unit. In doing so, the benefit of the UAVs would be for naught. The data security class of requirements is aimed by the customers to prevent this scenario.

As stated above, the two main objectives of small UAVs is to add value to the small unit while minimizing the burden required to operate the vehicle. The preceding several paragraphs described in depth the details of what the customer wanted in terms of adding value to the unit. The next several paragraphs will detail what the customer desires in terms of minimizing the operational burden. The three subcategories of minimizing the burden on the unit are "Available and Able to Perform in the Mission Environment," "Minimize Disturbance to Normal Troop Operation while Still Completing the Mission," and "Vehicle Survive/Able to Fight Another Day." Each of these subsection while be discussed in the subsequent paragraphs.

The first way that the customer desires for the small UAVs to minimize the burden to the unit is to be available and able to perform in the mission environment. Part of this desire from the customer is self-explanatory. In order for the vehicle to be of any use to the unit, let alone an improvement over their current methods, the vehicle must be available to use. Examples of unavailability would be if after each mission it became unusable or if the cost of the system or its maintenance was prohibitive to small units possessing their own system. Assuming that the unit does have a system, it must be able to operate anywhere that the unit is deployed without making the soldiers doing anything special. If the system only functions in certain environments, then it just becomes extra weight that the unit must lug around. The less obvious subnode of this section is that the customer desires the system to be as configurable as possible. They call for the vehicle to be both configurable on the ground and in flight. On the ground, the customers desire for the vehicle to be as modular as possible. In doing so, the unit would be able to switch out sensors and onboard equipment depending upon the mission at hand. Thus, the payload can be changed to meet the mission needs, allowing the same vehicle to be used in a variety of different ways. While in flight, the configurability of the vehicle switches from payload modularity to that of task rescheduling and flight path alteration. The customer desires the ability to alter navigational waypoints midflight as the operator sees fit. The configurability of the vehicle both in the ground and while in flight not only allows the system to accomplish more, but it also prevents the unit from needing multiple different systems to accomplish the varied tasks, thereby also minimizing the burden on the small unit.

The second subcategory under the minimize the burden node is that the customer desires the vehicle to minimize the disturbance to the normal troop operation while still completing the mission. This whole category revolves around making the soldier's life as easy as possible when it comes to the small UAV. This includes being small and light, thereby being as easy as can be to carry around. Then, when the system is deployed or retrieved after a mission, the customer wants the system to be easy to assemble or disassemble. The customer clearly is envisioning the unit potentially using the system while in the heat of a battle or on the move, thereby not having much time or attention to dedicate to the vehicle. With this scenario in mind, they desire to have a system that is able to be assembled/disassembled on the move in a very short time and require no special tools or effort from the unit.

This theme continues with the mission planning and ease of operation. The customers desire that the preprogramming of the mission plan before the vehicle is deployed is intuitive, feature-rich, flexible, thorough, and easy on the operator. Also, the customer strongly desires that the aircraft be easy to use. They want a flexible system that does not require multiple soldiers dedicated solely to maintaining the vehicle's flight, but instead they want the vehicle to do most of the work while needing only cursory input from the operator. This is especially seen by the fact that the customers want the interface to be easy enough to control multiple vehicles, which could clearly not be possible if the vehicle required significant input during flight.

The same ease on the operator trend continues with the presentation of the information received from the vehicle on the ground station. The customer desires that it be presented in a way that gives the users what they need without much effort, including the ability to easily select between data from vehicles that are not even the same class of vehicles. The customer thus is interested in seamlessly integrating the pieces of the battlefield to give the unit all the information possible at a given time. Continuing with the theme of seamlessly integrating the battlefield, the customers want the system to work with existing systems currently being used by the military. The customer thus wants the system to display data from multiple types of vehicles, have similar power sources, be interoperable with existing encryption and video standards, and require only existing tools for maintenance. The customer truly sees the vehicle as fitting into the existing military system requiring as little extra effort as possible and even bridging some of the gaps that currently do exist between equipment.

Finally, the last way that the customer desires for the vehicle to not be a burden to the unit is to have it survive the battle and to still be able to fight another day. This can be divided into surviving the mission and then being able to be repaired so that the vehicle can be used on subsequent missions. The latter half can be divided into being repairable in the field and also some general maintenance requirements. The customer is making the point in these requirements that it is not simply enough for the system to fly a mission well and then be lost, or to make it back but then have to be total overhauled before it can be used again. Instead the customer desires to have a system that can repeatedly fly successful missions and be a valuable asset to the unit without needing to see the inside of an electronics shop every time.

The last several paragraphs have discussed in detail what the customer truly desires from the small UAV. Fundamentally, they want a system that adds value to the unit operating the UAV while also minimizing the burden of operation. The value added to the unit comes either in the form of enhancing the effectiveness of the unit by allowing it to perform a desired task or by enhancing the unit's survivability. The customer also wants the system to be as unobtrusive to the unit as possible. This desire has broad reaching implications on the system, from the modularity of the payload to the mean time to repair to how the data from the vehicle are viewed. At their root, however, all these requirements stem from the same principle: the customer wants the system to free the soldiers to perform other tasks simultaneously, not tie them to a system that requires constant attention. The system in essence becomes an additional asset for the unit, performing its task while the unit is also working. As a result, not only does the system accomplish tasks the unit could not previously perform, but it also allows them to physically do more at once than they previously could. The customer also sees the system fitting into the existing infrastructure of the military, using parts, tools, and standards that already exist instead of requiring the wheel to be reinvented. Thus, the overall impression the customer gives regarding the small UAV system is that it should add functionality while making as small a ripple as possible in the existing infrastructure and tactics of the military.

Targeting With the overall system motivated, the discussion now turns specifically to the targeting system. Much of what the customer wants can best be viewed within the framework established for the overall system as the same underlying motivations exist. The customer wants to enhance the unit's effectiveness while minimizing the burden.

As mentioned before, the current method of targeting for the small unit involves either directly lasing a target or calling back coordinates to the friendly forces doing the attacking. Either situation has some very undesirable aspects to it. The foremost undesirability is the risk to the unit. The very nature of targeting inherently deals with lethality. As a result, if the unit is targeting an object of interest, by definition they are near something deemed hostile enough to merit being destroyed. It can then be implied that the soldiers' discovery by the enemy would be met with hostility. Even if the soldiers are not detected, once the attack takes place, any other enemy in the area will certainly be under heightened alert, putting the soldiers at even greater danger. Since the soldiers are still in the area, they are not removed from danger after their objective is met. Plus, in order to target the object, not only must they be close enough to see the object, but they must also be free from obstructions that prevent maintaining a line of sight on the target. Thus, if the object attempts to hide from the soldiers by moving behind an obstacle, the soldiers must displace from their position to maintain targeting, thereby putting them at further risk.

The customer desires to reduce the risk to the soldiers and the ability of the object to hide from being targeted by using aerial reconnaissance. The operational radius of the UAVs can be used to remove the troops from harms way. Instead of being within line of sight and in direct contact of the object, the unit can be back at a distance while either the system targets the object automatically or the soldiers teleoperate the system. In addition to increasing the distance from the soldiers and the target zone, the small UAV has the advantage of an aerial view of the situation, making it harder for objects to hide and easier for the system to reacquire them if they do.

In terms of the actual targeting, the customer desires accurately engaging the typical high-value military targets. The requirements include the need to be able to detect enemy combatants and military equipment, such as tanks and trucks. Plus, the potential target list is increased if the small UAVs are used to support precision engagements. The support for precision engagement could come from algorithmic solutions or directly from a laser designator or laser pointer, if the equipment is present.

All told, the customer sees small UAVs targeting combining the potential for the most precise high tech engagements currently available with the advantage of an aerial view and troops safely off at a distance. Thus, they get the most precise targeting possible using a robust medium and do not need to risk its most valuable assets, that being its soldiers. As a result, the customer sees a tremendous upside to small UAV targeting.

What Constraints Exist on the System, Both Stated and Implied?

The last section answered the first major question addressed by the requirements analysis, namely what does the customer truly want. The discussion now turns toward answering the second major question of what constraints exist on the system, both stated and implied. The stated requirements will be presented first, followed by a look at the implied requirements.

Stated Requirements The stated requirements for the targeting system take root in the overall requirements for the small UAV system listed previously in Chapter 3.

The methodology section above mentioned that the initial requirements listing might contain redundant requirements or requirements that do not apply to the specific part of the problem being addressed, especially if the system being designed is a subsystem of a larger system. This would certainly be the case for the targeting system if all the requirements for a given small UAV project such as the FCS Class I or RPUAV was used. However, since Chapter 3 contains a striped down version of the requirements only pertaining to the targeting system, there are no superfluous requirements. The methodology section also stated that it is often helpful to organize the requirements by functionality, subsystem, or some other logical means while also identifying the requirement's priority level, when the required functionality should appear, or any other pertinent information. Again, these suggestions would serve useful provided with many pages of requirements defined over the entire timeline of the project and covering the gamut of customer desires, from absolutely critical to ancillary benefits. However, since Chapter 3 contains only the bare essential requirements for the targeting system, a variety of the suggestions are not applicable. For readability sake, the requirements from Chapter 3 are put into spreadsheet form in Table 4.1.

Table 4.1: Sample small UAV targeting requirements

Requirement	**T/O**
The targeting system shall weigh no more than 2 lb.	T
The targeting system shall weigh no more than 0.5 lb.	O
The targeting system shall take up less than $3680cm^3$ of space and fit into a space approximately $0.3ft$ x $0.3ft$ x $1.4436ft$ in dimension.	T
The targeting system shall take up less than $1000cm^3$ of space and fit into a space approximately $0.3ft$ x $0.3ft$ x $1.4436ft$ in dimension.	O
The targeting system shall consume less than $5W$ of power during a typical mission	T
The targeting system shall consume less than $2W$ of power during a typical mission	O
The targeting system shall cost less than \$10,000.	T
The targeting system shall cost less than \$5,000.	O
The targeting system shall have a TLE of less than $50m$.	T
The targeting system shall have a TLE of less than $10m$.	O
The small UAV shall fly at an altitude of $500ft$ AGL.	T
The small UAV shall contain an E/O and IR camera and the E/O camera will be fixed on the vehicle at an angle of 65° from vertical	T
The small UAV will be able to detect standard military vehicles such as a tank or a truck at its operating altitude	T
The small UAV will be able to detect a man-sized object at its operating altitude	O

Table 4.2: The number of resolvable light-dark cycles required across a target's critical dimension for various discrimination tasks (Taken from [30])

Target (Broadside)	**Detection**	**Recognition**	**Identification**
Truck	0.90	4.50	8.00
M-48 Tank	0.70	3.50	7.00
Stalin Tank	0.75	3.30	6.00
Centurion Tank	0.75	3.50	6.00
Half-Truck	1.00	4.00	5.00
Jeep	1.20	4.50	5.50
Command Car	1.20	4.30	5.50
Solider (Standing)	1.50	3.80	8.00
105 Howitzer	1.00	4.80	6.00
Average	1.00 ± 0.25	4.00 ± 0.35	6.40 ± 1.50

Derived Requirements The second half of defining what constraints are placed upon the system is identifying the derived requirements. The last two requirements listed in Table 4.1 deal with the system's ability to detect targets. According to [30], target acquisition is generally concerned with the detection of points of interest and their subsequent recognition and identification. These criteria were first quantified in the 1950s by John Johnson, a scientist at the United States Army Night Vision Lab. Johnson used image intensifier equipment to measure the ability of volunteer observers to identify scale model targets under various conditions [2]. His experiments produced the first empirical data on perceptual thresholds that was expressed in terms of line pairs or cycles [2], where one white bar and one black bar equate to a cycle [30]. At the first Night Vision Image Intensifier Symposium in October in 1958, Johnson presented his findings in a paper entitled Analysis of Image Forming Systems, which described both image and frequency domain approaches to analyzing the ability of observers to perform visual tasks using image intensifier technology [2]. Later referred to as Johnson's Criteria, Johnson's findings were such an important breakthrough that they became the de facto industrial standard.

As mentioned above, target acquisition typically involves detection, recognition, and identification. In colloquial terms, detection allows the observer to say "There's something out there," recognition allows them to say "It's a tank," and identification allows them to say "It's a T72 tank" [30]. The original Johnson's Criteria accepted values for these tasks are included in Table 4.2 for nine common military targets.

Johnson's Criteria can be used to determine the necessary sensor resolution for any visual acquisition task at a given range. In order to do this, the target size needs to be established. Table 4.3 contains the standard target sizes for a man and tank as defined by the Army's Night Vision Lab.

At the time of this writing, it was not clear how to translate the line cycles in Johnson's criteria to the number of pixels in current optical cameras. [19] states that Johnson's Criteria requires three overlapping pixels for the detection of a six foot

Table 4.3: Standard target sizes as defined by the Army's Night Vision Lab (Taken from [30])

Description	Width	Height	Critical Dimension
Tank	2.3 m	2.3 m	2.3 m
Man	0.5 m	1.8 m	1.0 m

tall man. While this seems low and could not be verified, it was used for a baseline analysis. Future searching in the literature would be required to either validate this number or produce a number of greater fidelity.

Table 4.1 states that the E/O camera needs to be able to detect a man-sized object at its operating altitude, a scenario roughly shown in Figure 4-2. As stated in Chapter 3, the flight parameters dictate that the vehicle has a slant range of approximately $1200ft$ to the target. Table 4.3 states that a man-sized object is 0.5 m width and 1.8 m tall. If the man had three pixels overlaid on him, then the size of the pixel would have to be $1.8m/3pixel = 0.6m/pixel$.

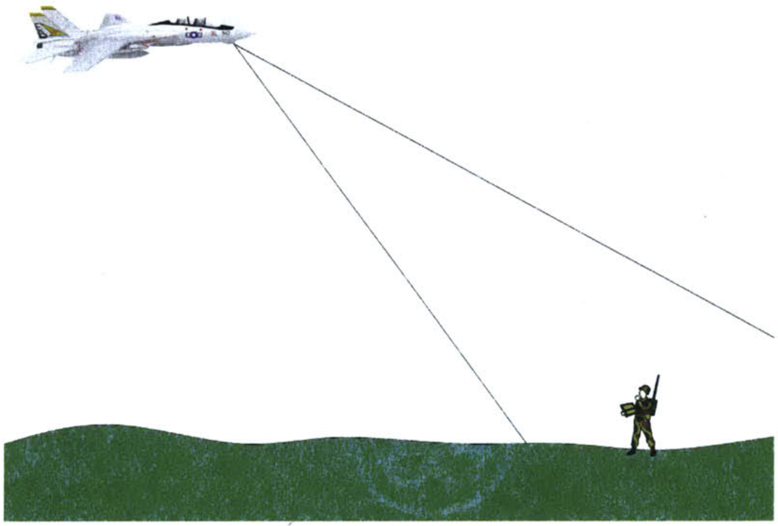


Figure 4-2: Scenario of a vehicle detecting a man-sized object.

Figure 4-3 shows a picture of just the camera's field of view, turned horizontally, with the enemy soldier in the visual cone. The figure shows that the soldier is overlaid by the requisite minimum of 3 pixels required for detection. The angle between the top of the pixel on top of the soldiers head to the bottom of the pixel at the base of the soldier's feet is shown in the figure to be an angle α. Similarly, the entire angular width of the cone is denoted as an angle θ. Also, the total number of pixels in the vertical direction (along the soldiers height) is denoted as n. Given these variables, the constraint that the man-sized object must have at least three pixels can be shown

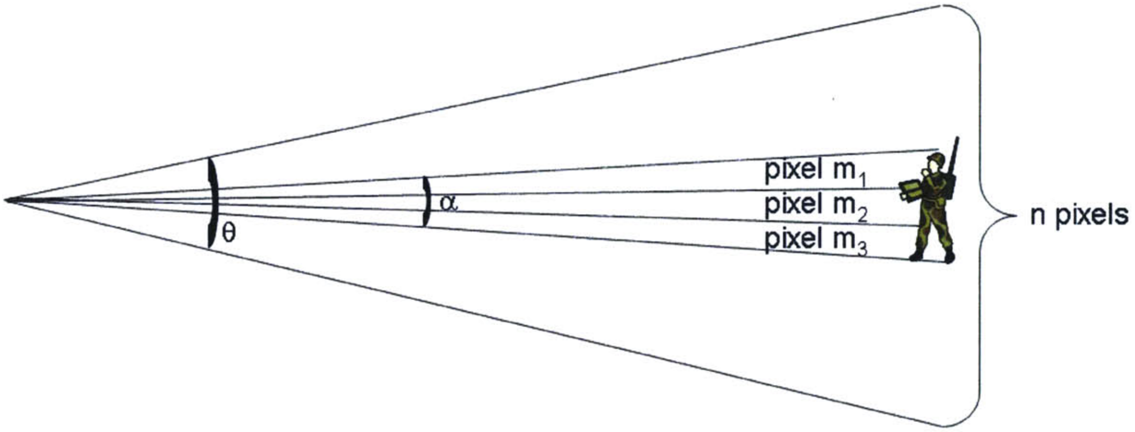


Figure 4-3: The visual cone from the aircraft to the soldier.

as $\frac{\alpha}{\theta} \geq \frac{3}{n}$, which can be simplified to $\frac{n\alpha}{\theta} \geq 3$.

The above equation serves as an additional constraint on the camera. The new constraint still gives the designers a considerable amount of freedom. $\frac{\alpha}{\theta}$ is the percentage of the camera's view that a 1.8 m tall object takes up at a slant range of 1200 ft. This fraction depends on the zoom of the camera. Also, the designers can adjust n, which translates into the camera's resolution. The equation shows that for a high resolution camera (n is big), then the man-sized object has to take up less of the camera's field of view. However, as the resolution decreases (n decreases), $\frac{\alpha}{\theta}$ must increase. This could happen by either α increasing, perhaps due to zoom, or by changing the field of view, θ, of the camera. This shows that the designers still have the freedom to play with the camera's properties, but they are bound by the constraint.

The analysis of detecting a man-sized object at the specified flight parameters illustrates the process of determining a derived requirement. While this is the only derived requirement in the provided application, if all the requirements for the actual programs were examined, many more could be ascertained. The evaluation method would not change for these derived requirements, as the example above was illustrative of the process.

4.2 Select Design Criteria

4.2.1 Methodology

The essence of the select the design criteria step in the design process is to establish the criteria by which the system will be evaluated and thus, the criteria to which the system will be designed. In selecting the design criteria, it is often helpful to work backwards by thinking of the cost function first as opposed to trying to define the criteria based upon an in-depth analysis of each individual requirement. Doing so helps the designers to think of the big picture of the project instead of getting bogged down in the details. The whole point of the cost function is to quantitatively and objectively assign worth to a design. Inherently, the items in the cost function are the criteria deemed crucial in assessing the worth of the system, with the items receiving the most weight being those that are most valuable. Section 2.2.2 stated that the design criteria for a system usually stemmed from one of five items. They were:

- Requirements that dictate the design
- Requirements that are especially important to the customer
- Requirements that are fundamental to the purpose of the system
- The subset of requirements that pertain to the given subsystem being addressed
- Requirements that are hard to achieve

Section 2.2.2 contained an explanation for each of these bullets.

The second and third bullets above are the two that are aimed at ascertaining and addressing the big picture questions of the project. The second bullet strives to answer the question of how the customer sees the system. That is, if all the frills were stripped from the system, what would the customer consider the absolute bare minimum functionality that would indicate an improvement from the current state. Another potential way of thinking about this is to consider how the customer would describe the system if they were only given a short paragraph, including what features they would include in the description.

The third bullet, requirements that are fundamental to the purpose of the system, certainly overlaps some with the previous bullet, as both are trying to ascertain the fundamental aspects of the system. However, this item deals less with the directly stated features, but instead tries to address some of the implied foundations. This type of concept was seen in the previous section when discussing the customer's underlying motivation for writing requirements. Specifically, the example of writing requirements that really were attempting to prevent identity theft of a web-based system was mentioned. In addition to this implied motivation uncovering, this bullet also aims to uncover the criteria that are fundamentally attached to a given discipline. For example, most every satellite must deal with thermal and vibrational issues, roller coasters are inherently concerned with safety, and artificial hearts are concerned about the life-cycle of the mechanical parts. These types of criteria are thus the criteria that are not specific to a given instance of a type of project, but are instead specific to the class of project itself. They are criteria that are identifiable with any project in a given field. They may be obvious and stated clearly in the requirements or may be assumed and omitted. However, they are no less important.

The remaining three bullets then get into more of the specifics of the system than the big picture of the project. The first bullet deals with the requirements that change the course of the design. If a requirement has such a profound impact on the design and it is intended by the customer to do so (that is, the requirement is properly written and the customer understands its impact), then that weight should be reflected in the design criteria. The fourth bullet, the subset of requirements that pertain to the given subsystem being addressed, allows for situations where all the requirements might not necessarily be applicable to the system being designed. As a result, the bullet allows the designers to select the design criteria from the applicable set of requirements. This would have been the case if all of the requirements from one of the actual small UAV projects had been used. Finally, the last bullet of potentially including requirements that are hard to achieve allows for the designers to have the design criteria reflect their effort. If the majority of the time is spent on a select few requirements, then they might need to be included in the design criteria.

Since the criteria identified form the foundation of the cost function used to evaluate and optimize the design, they must be able to have quantifiable metrics attached to them. Generic statements such as "The system must perform (a given functionality) well" do not work as design criteria. Also, while it is true that the big picture of the system should be considered first, the cost function, and thus the design criteria, can be as specific as desired for the problem addressed. If the designers see fit to create a design criteria that is specific to a given requirement, they can certainly do so. In the end, the design criteria should be an accurate reflection of what is important for a given system. If that involves a very specific requirement, then it should be included.

4.2.2 Application

Applying the above criteria to the targeting subsystem of small UAVs yields design criteria of weight, size, power, unit cost, accuracy, robustness, and latency. Each of these will be discussed individually in the subsequent paragraphs.

Weight is a fundamental concern for nearly every aerospace application, whether aviation or space-related. For aircraft, the weight translates into how much lift is required in order to achieve flight. The lift required then dictates the geometry of the vehicle and the propulsion system used, which has implications on the size of the plane and the material strength required to house the propulsive system. Small additions in payload weight could require an increase in the size of the propulsive unit or a change in the geometry of the wings (increase in aspect ratio, more surface area, etc). These changes further increase the weight, thereby translating a small change in weight due to payload to become larger. Plus, additional work goes into the redesign of the system. As a result, weight is a design criteria for nearly every aviation system.

Much of the same argument can be applied to the size of a system. For aircraft, size translates into weight/lift or into drag, thereby becoming a consideration for the aerodynamics. For subsystems of a designed vehicle, size becomes a matter of physically fitting into the structure where space is usually a precious commodity. With all other considerations equal, the lighter and smaller the component in aviation the better.

Power is also a critical concern on any aviation project. For small UAVs, power required translates directly into the size of the batteries. On most small UAVs, the batteries are a substantial portion of the weight and take up a non-trivial amount of space, thereby being directly connected to the previous two criteria. Additionally, the batteries are also the major driving component in the vehicle's endurance. These considerations are usually iterative in nature. An initial estimate is made of the size of batteries required by the system. Then, the components are selected given the functional requirements of the vehicle. The voltage and current required for the system are then set once the components have been finalized. The batteries are selected based upon the voltage and current required, as well as the necessary endurance. The final battery size is then compared to the volume and lift capability of the vehicle. If the system is not compatible, either the vehicle itself is changed or the components are changed. This whole cycle stemmed from the fact that power is clearly required for all of the components to function and the batteries are such major contributors to the weight of the vehicle. Given these considerations, power is a necessary design criterion.

As with almost any endeavor, engineering-related or not, money is a critical concern. Reality dictates that projects have a finite amount of budget and that the systems must finish under some absolute threshold, with an objective budget usually also present. In engineering systems, a number of different types of monetary costs can be calculated, including, amongst others, life-cycle cost, unit cost, and production cost. At least one of these, if not all, is critical for all but the luckiest project managers. For simplicity's sake, unit cost was chosen as the design criteria to represent monetary concerns.

All of the previous criteria had nothing to do per se with the targeting system, but more with the fundamental concerns of aviation systems. The next several criteria turn specifically to the targeting concerns. One of the foremost concerns is the accuracy of the position that is returned by the system. This is perhaps the foremost requirement for a system whose ultimate purpose is lethality. In a best-case scenario for an inaccurate targeting system, a strike fails and the target gets away unharmed. In a worse-case scenario, innocent or friendly lives are lost in addition to the target getting away. Due to the importance of the accuracy of the solution returned by the targeting system in a human, military, and even political sense, accuracy was included in the design criteria.

Accuracy is certainly not the only important criteria for targeting. Robustness is another key consideration. Robustness deals with the system's ability to produce an answer in a variety of scenarios and mission parameters, including ones that may be unfavorable to the system's functionality. A system that can only produce a solution under a certain limited set of circumstances is clearly not as useful to a soldier as a system that can ubiquitously provide a solution. As a result, robustness is an important measuring stick of the worth of a system and should thus be included in the design criteria.

The final design criteria used to assess the alternatives is latency. Latency deals with the amount of time required in order to produce the solution after the information has been collected. In a combat situation, the less time required to produce a solution on the location of a target the better. The amount of delay could be the difference between a successful engagement. For instance, if a solution is perfect in all conditions, but comes 30 seconds after the imagery was collected, this would probably be detrimental to the success of engaging any fast moving target. Due to the variation in success of an encounter based upon the delay of the solution, latency was included in the design criteria.

The three targeting-specific criteria were thus designed to help assess the success of engaging the target whose position is returned. Accuracy and latency deal with the fidelity and validity of the information provided, respectively, while the robustness deals with the solution's reliability. Without any of the three characteristics, the system would not be dependable for the operators.

4.3 Establish Relationships between Physical Properties and Scores

4.3.1 Methodology

The last section described the design criteria by which the system would be evaluated. It was stated that thinking of the ultimate cost function that would be used to numerically perform the evaluation was often beneficial to help flesh out the design criteria. The next step attaches the bounds to the design criteria that allows for the system to be objectively evaluated.

Section 2.2.3 compared the process of establishing relationships between the physical properties and scores to how scores were assigned in the matrix-like evaluation method. In the matrix-like evaluation, scores were not based upon anything concrete, but were instead were usually done relative to the first alternative that was scored. In order to combat the subjectivity of the score assignment, it was proposed that concrete bounds should be assigned to the design criteria that translated the systems actual properties into scores. Section 2.2.3 stated that the bounds either took the form of discrete bounds or continuous functions, giving an example of both. The subsequent paragraphs will discuss different options that designers have in terms of implementing these bounds.

Discrete Bounds

The first option for the translation of the properties to scores is the use of discrete bounds. Figure 4-4 gives a hierarchy of options available for discrete bounds. All of the options in the figure rest on two assumptions:

- An entire region of values is covered.
- The bounds are set up such that the lowest region is first, followed by successively higher values.

These two assumptions are seen in the discrete bound example in Section 2.2.3. In the example, the entire region between $0lb$ and $0.5lb$ is covered. Also, the bounds are in succeeding order, with $0.0lb$ to $0.1lb$ coming first, followed by $0.1lb$ to $0.2lb$, coming next, with the remaining bounds following in increasing numerical order.

The first choice available in Figure 4-4 is ascending vs. descending. This refers to the value of the scores relative to the bounds. In the example in Section 2.2.3, the

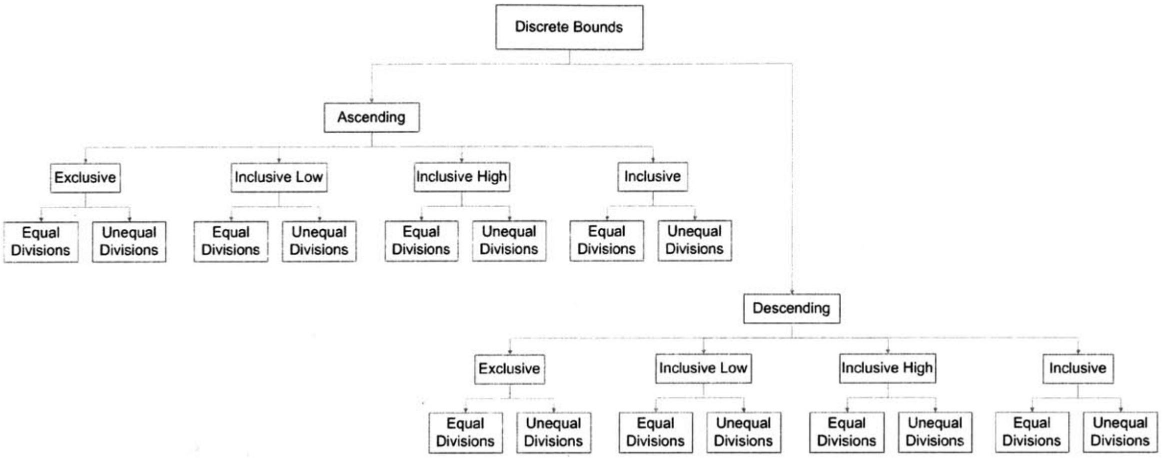


Figure 4-4: Hierarchy of discrete bound options.

bounds were descending because first region, $0.0lb$ to $0.1lb$, had the highest score of 5 while the last region, $0.4lb$ to $0.5lb$, had the lowest score of 1. If the scores were reversed such that the first region had the lowest score and the last region had the highest score, then the bounds would be ascending. This label is the reason for the second assumption. If the bounds were not in an order, then a general statement could not be made about the scores. Thus, the ascending vs. descending distinction refers to the scores relative to the set bounds. Ascending bounds are used for properties that need to be maximized, such as a measure of percent correct, while descending bounds are used for properties that need to be minimized, such as the cost of the system.

The next choice to be made about the discrete bounds is about the degree to which the bounds are inclusive. The choices are exclusive, inclusive low, inclusive high, and inclusive. This is simply a choice of whether a region takes one of four general equations:

- $lb < x < ub$
- $lb \leq x < ub$
- $lb < x \leq ub$
- $lb \leq x \leq ub$

where lb is the lower bound and ub is the upper bound. The only difference between the four general equations is the presence and the location of the $<$ and $\leq$ signs. In the first bullet, both the lower and upper bounds are excluded from the region, making it the region referred to in Figure 4-4 as exclusive. The next two bulleted equations have only one bound included. In the second bullet, the lower bound is included and in the third bullet, the upper bound is included while the lower bound is excluded. The two cases are referred to as inclusive low and inclusive high, respectively. Finally, the last region includes both bounds, earning it the label of inclusive. It should be noted that it is up to the designers to define the regions in such a way that there is not a conflict on the bounds. That is, the upper bound is included in one region, then the lower bound of the next region cannot be included, as the same number would thus be defined to be two different scores. The example in Section 2.2.3 solved this problem by having all the bounds be inclusive low, thereby leaving all the bounding numbers unconflicted.

The last decision to make is whether the regions are equally space or not. In essence, this decision is how to divide the selected region up into the different subregions. All of the subregions could be equal in size or the could have different sizes. This is up to the designer to decide and is problem-specific. The example shown in Section 2.2.3 has equal sizes for each of the subregions.

The final note on discrete bounds is in regards to any property that falls outside of the selected region. If the property is beyond the highest score (lower than the lower bound in descending or greater than the upper bound in ascending), then the property is simply given the highest score possible. Conversely, if the property is beyond the lowest score (lower than the lowest bound in ascending or greater than the biggest bound in descending), then the property is judged to be in violation of the defined constraint and a score of $-\infty$ is assigned to the property. A $-\infty$ is used because in a weighted summation, the composite score will then always be $-\infty$, thereby indicating that the system did not meet all the specified constraints and needs to be adjusted accordingly.

Continuous Functions

The other option for the translation of the properties to scores is the use of continuous functions. As with discrete bounds, several options exist within continuous functions. The first is to define a mathematical function to translate the scores directly. In Section 2.2.3, an example of $y = x^2$ was given. Any designer-defined function would do. In the case of mathematical functions, the property would be the input and the score would be the output.

Another option is to define a number of points and then fit a function to these points. Many techniques exist towards this end, including cubic splines, polynomial fitting, and regression analysis. These techniques are greatly simplified by the use of mathematically-based programs such as Matlab. The use of Matlab to create cubic splines and polynomials to fit data will be discussed in the subsequent paragraphs.

Polynomial Fitting The first technique of fitting a function to data points as implemented in Matlab is polynomial fitting. The Matlab function for performing this task is *polyfit*. This function takes three parameters as input, the independent variable, the dependent variable, and the order of the polynomial desired to fit to the data. The independent and dependent variables combine to form the ordered pairs of the points. *polyval* returns the coefficients of the polynomial to the order that the user specified. The polynomial is found using a least squares method.

The *polyval* function in Matlab is best shown through an example. The discrete bounds case will be modified so that the ordered pairs are $(0.0, 1.0)$, $(0.1, 0.8)$, $(0.2, 0.5)$, $(0.5, 0.0)$, and $(0.7, -10.0)$. In this example, a weight of $0lb$ is still the highest score and $0.5lb$ is the highest weight that will not result in a penalty. If the independent variable is called x and the dependent variable is y, then the above would be encoded in Matlab as:

```
x = [0.0, 0.1, 0.2, 0.5, 0.7];
y = [1.0, 0.8, 0.5, 0.0, -10.0];
```

The designers would then simply need to select which order polynomial to fit to the data. If they wanted to fit a fourth-order polynomial to the data, the Matlab function call would be:

```
p = polyfit(x, y, 4)
```

The variable p would then hold the coefficients for the fourth-order polynomial such that the first number is the coefficient for x^4, the second number is the coefficient for x^3, and so on. As a result, if p was $[-261.9048, 226.1905, -54.5238, 1.4524, 1.0000]$, this would translate to an equation of $y = -261.9048x^4 + 226.1905x^3 - 54.5238x^2 + 1.4524x + 1.0000$.

When using *polyfit* in this way, it is usually best to graph the equation to ensure that it behaves the way that was expected. The Matlab function *polyval* can be used to do this. *polyval* takes two parameters, the polynomial coefficients and values at which to evaluate the polynomial. In order to get an adequate idea of the behavior of the function, a significant number of points need to be used. The Matlab function *linspace* can be used to do this. *linspace* divides a region into a specified number of points. For example, if the designers wanted to divide the region between 1 and 10 into 1000 parts, they could do:

```
xGraph = linspace(1, 10, 1000);
```

With the points defined at which to evaluate the function, *polyval* can then be called by:

```
yGraph = polyval(p, xGraph);
```

Matlab can then be used to plot the results. The following text is a Matlab script that was used to encode the example and then plot the polynomial fits for the first four orders. The resulting plot is shown in Figure 4-5.

```
clear all, close all, clc
set(0, 'DefaultAxesFontSize', 18)

x = [0.0, 0.1, 0.2, 0.5, 0.7];
y = [1.0, 0.8, 0.5, 0.0, -10.0];

figure

for i = 1:4
   p = polyfit(x, y, i)
```

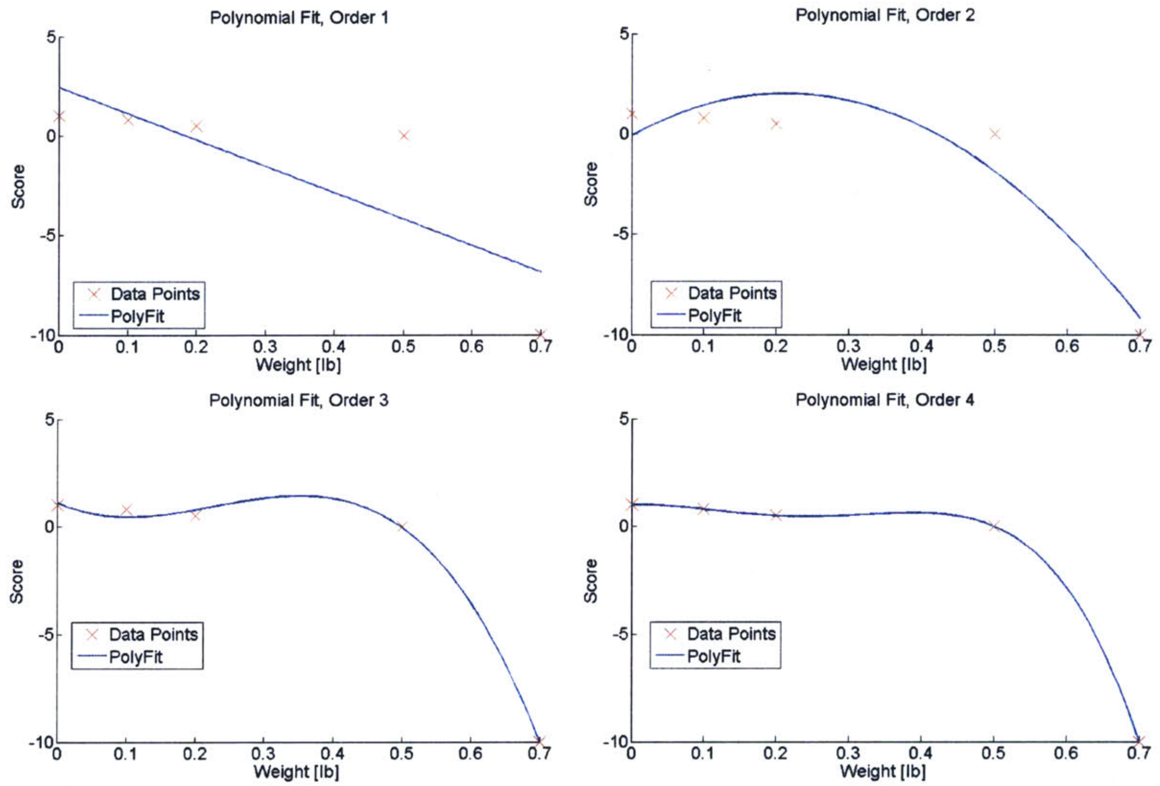


Figure 4-5: Fitting a polynomial to the data using Matlab's *polyfit* function on the provided example for the first four orders.

```
    xGraph = linspace(x(1), x(end), 100);
    yGraph = polyval(p, xGraph);

    subplot(2, 2, i), hold on, plot(x, y, ’rx’, ’MarkerSize’, 18)
    subplot(2, 2, i), plot(xGraph, yGraph, ’LineWidth’, 2)
    titleStr = sprintf(’Polynomial Fit, Order %s’, num2str(i));
    title(titleStr);
    xlabel(’Weight [lb]’)
    ylabel(’Score’)
    legend(’Data Points’, ’PolyFit’, 0)
    axis([0.0 0.7 -10 5])
end
```

Figure 4-5 shows why it is a good idea to graph the resulting polynomials. The basic shape of just the X's in the figure show the basic intent of the designers. It appears that they desire a monotonic decrease in the score from a weight of $0lb$ to $0.5lb$, followed by a precipitous drop in score from $0.5lb$ to $0.7lb$. However, the second- and third-order polynomials do not realize this desire. The second order increases from 0 to 0.22 and then drops from there, while the third-order polynomial increases in the 0.1 to 0.35 range. If these functions were used, then heavier systems would be rated higher than lighter systems, which is counter to they designers intentions. The fourth-order polynomial is closest to the intent of the designers, with a near-monotonic nature and the precipitous drop after $0.5lb$. However, even this polynomial increases slightly in the 0.31 to 0.39 range.

It is also important to note that a new feature is present in continuous functions that was not present in discrete bounds. This feature is a region beyond the threshold weight set in the requirement where the system is docked points for being in violation of the requirement, but is not necessarily given a score of $-\infty$. In the example above, as was stated in Section 2.2.3, the threshold weight was set at $0.5lb$, making it the greatest upper bound in the discrete bounds case. Anything above $0.5lb$ is thus in violation of the requirement. However, there might be situations where greater than $0.5lb$ would be acceptable if the system performed much better in another area. By putting in a region where the system score is penalized but not eliminated from consideration, the designers allow for the requirements trading that was discussed in Chapter 1. While this requirement would not be met, the overall system might still be an improvement. This also eliminates situations where the system is just barely outside of the requirement, such as $0.5005lb$, but is assigned a score of $-\infty$. While the region allows for increased flexibility in the design, it should also be noted that the further the system gets away from the threshold value, the more it is punished. As stated above, the fourth-order polynomial captures the true intent the best in that the scores fall off very quickly after the threshold value, thereby encouraging the system to be as close to the specified value as possible. Since the translation is a polynomial, it it defined for any value of x, but as it gets further from the threshold, the score approaches $-\infty$.

Cubic Spline The next continuous function option in Matlab is the cubic spline, which is done through Matlab's *spline* function. According to the Matlab documentation, the *spline* function provides the piecewise polynomial form of the cubic spline interpolant to the data values Y at the data sites X. This means that the function finds a cubic function the connects the two data points and meets the criteria for the overall spline.

This method is very similar to that of the polynomial fitting of the data in terms of use. As before, it takes the independent and dependent variables as input. However, since the function inherently uses a cubic spline function, there is no input for the order as there was before. Also, similar to before, the function returns the coefficients that define the spline. However, since the function is piecewise and cubic, there are four coefficients for each segment of the spline. The way that Matlab does this is through the use of a struct. A matrix of coefficients is labeled as *coefs*. Each row of the matrix contains the coefficients for the cubic function that is valid for the corresponding segment, where the segments are numbered from left to right. As a result, the segment that is the most negative has the coefficients in the first row, the next segment is in the second row, and so on. Also, the struct contains a vector labeled *breaks*, which is simply the x coordinates passed into the function.

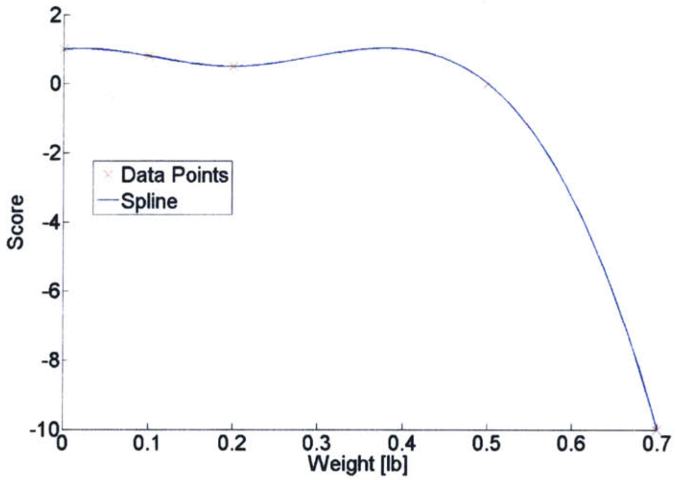


Figure 4-6: Fitting a cubic spline to the data using Matlab's *spline* function.

As before, visualizing the function in a graph is necessary to ensure the proper behavior of the function. *linspace* can be used again to generate the x values. Instead of using *polyval* to evaluate the function, *ppval* is used. *ppval* handles the piecewise nature of the polynomial. As a result, the code required to generate the graph transforms to the code below:

```
clear all, close all, clc
set(0, 'DefaultAxesFontSize', 18)

x = [0.0, 0.1, 0.2, 0.5, 0.7];
y = [1.0, 0.8, 0.5, 0.0, -10.0];
splineStruct = spline(x, y);
xGraph = linspace(x(1), x(end), 100);
yGraph = ppval(splineStruct, xGraph)

figure, hold on
plot(x, y, 'rx', 'MarkerSize', 18)
plot(xGraph, yGraph, 'LineWidth', 2)
xlabel('Weight [lb]')
ylabel('Scores')
legend('Data Points', 'Spline', 0)
```

This code segment yields the graph shown in Figure 4-6.

As Figure 4-6 shows, the cubic spline suffers from the same problem as the second- and third-order polynomials seen in the previous section. Instead of having a monotonic decrease to $0.5lb$, the spline has a large increase from $0.2lb$ to $0.38lb$. However, since the designers cannot change the order, the only degree of freedom

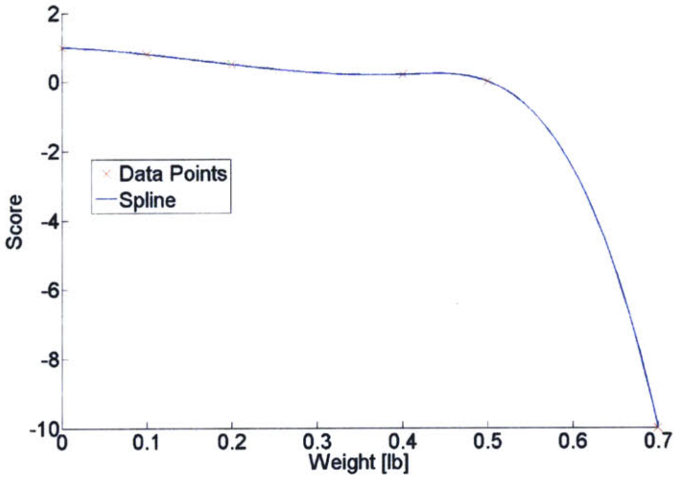


Figure 4-7: An adjustment to the original spline by adding the point $(0.4, 0.2)$

they have is changing the data points to which the spline is fit. The points can be changed in three ways: the location of points (the x values), the values of the points (the y values), or the number of the points. The first two changes are self-explanatory. In terms of the number of points, the designers could either add points or subtract points from the data. To demonstrate the effect that adding a point could have, the point $(0.4, 0.2)$ was added to the graph and the spline was recalculated. The resulting graph is shown in Figure 4-7. The cubic spline returned using the added data point was much closer to the designers intention. The addition of the point took out the rise in scores while still keeping the good parts of the previous spline.

Finally, it should be noted that the cubic spline has the same advantage as the polynomial fits of the extra region beyond the threshold where the score can be penalized without being disregarded. However, since the cubic spline is only defined piecewise between the points, at the end of the extra section, the spline is no longer defined. As a result, the scores would then drop off to $-\infty$ like in the discrete bound case. By contrast, as was previously mentioned, the polynomial fit to the data was defined for all values of x. The cubic spline, however, still offers an advantage over the discrete bounds, though, because it does have the penalty region. Even if negative scores were added to the discrete bound case, it would not capture the precipitous drop-off in scores like the spline or the polynomial fits can.

Normalization of the Scores

Regardless of whether discrete bounds or continuous functions are used for the score translation, the scores should be normalized so that they are all on the same scale. That is, the positive numbers should all vary between the same numbers, such as 0 to 1 or 0 to 10. By normalizing the translation schemes, the requirements all contribute equally to the cost function before the weights are applied. As a result, the weights determine the actual contribution to the overall composite score. If the normalization was not done, then the weights would not represent the contributions accurately. For instance, if a scale for one criteria was 1 to 10 and another was 1 to 100 and both criteria received the top scores, then the first criteria would contribute 10 and the second criteria would contribute 100. Then, if the weights were set such that the first criteria had a weight of 10 and the second had a weight of one, the actual contribution to the composite would be 100 for both criteria. However, the weights state that the first should have been 10 times as important. By normalizing the scores, the contributions are equalized, making the weights accurate. In the vernacular, normalization allows the designers to compare apples to apples instead of apples to oranges.

4.3.2 Application

The previous step in the process established seven criteria by which to evaluate the various designs of the targeting system. They were weight/mass, size, power, cost, accuracy, robustness, and latency. As stated at the beginning of this step's methodology section, the aim of this step is to attach bounds to each of the design criterion in order to allow for the objective evaluation of the system. The next several pages contain the resulting bounds for each criterion. Before the functions are presented, the next paragraph will discuss their similarities.

Each of the bounds were created in the same fashion. Each was a continuous function that used the Matlab function *polyfit* as was described in detail in the previous section. The bounds were all monotonically decreasing in nature, having quantities that needed to be minimized. This meant that the scores were highest when the properties were closest to zero. The bounds have a nearly flat plateau region at the beginning that stretches from zero up until the point of the objective level. After the objective level for the criteria, the function experiences a steady decrease to a score of zero, which occurs at the threshold value. In the post-threshold region, the bounds retained the use of the negative score area, as witnessed by the precipitous drop that occurs in all the functions after the threshold. As a result, the scores become increasingly negative the further the value gets in the positive direction from the threshold value. Additionally, all of the bounds were normalized such that the positive scores were between 0 and 1. The negative region was formed by having an extra data point beyond the threshold that was given a score of -10. Finally, in order to form the curves, the order of the polynomial was manipulated, as were the points to which the polynomial was fit. The most useful technique in getting the curve to follow a certain shape was to add data points and then increase the order of the fitting polynomial. in the interest of space, only the final data points, the equations generated, and the resulting plots will be shown instead of a step-by-step derivation of all the plots. In each case, the starting data points were at (0, 1), (obj, 1), (threshold, 0), and (threshold + δ, -10), where the value of δ depended upon the threshold value. The points and order were manipulated to from there to get a properly shaped curve.

Mass

As Table 4.1 shows, the mass requirement stated that the targeting system must be under $2lb$ (907.2g), with a preference to be under $0.5lb$ (226.8g). Nine ordered pairs were used to fit the polynomial. The x and y values were:

```
massX = [0.0, 50, 100, 250, 400.0, 600.0, 800.0, 907.0, 1200];
massY = [1, 1, 1, .85, .6, 0.35, 0.13, 0, -10];
```

A seventh-order polynomial was fit to the points. The resulting equation was: $y = -1.7366 * 10^{-19}x^7 + 5.655 * 10^{-16}x^6 - 7.2504 * 10^{-13}x^5 + 4.5651 * 10^{-10}x^4 - 1.3957 * 10^{-7}x^3 + 1.5261 * 10^{-5}x^2 - 0.00050785x + 1.0004$. The graph of this equation in the region over the specified x values is shown in the first part of Figure 4-8.

The first part of Figure 4-8 shows the basic shape of the bound that was described above. The first part is best fit at demonstrating the near monotonic decrease of the bound and the precipitous fall off after the threshold region. It is important to note that the equation is not simply defined in the shown region of x, but is valid for all values of x. As a result, the graph would continue to fall off after 1200. It is up to the designers to ensure that no negative values of mass are passed into the function, as this would be a clear violation of the laws of physics even though it is mathematically possible with the function.

The second part of the second of Figure 4-8 shows two zoomed in pictures of the bound function. The first spans over the positive scores while the second graph is zoomed in around the seemingly flat first portion of the graph. The view of the positive score region gives a much better idea of how the function behaves. The graph shows how the function begins to decrease at a fairly steady rate after hovering around the same value initial. The plot also shows a glimpse of the curve turning even further downward after the threshold value.

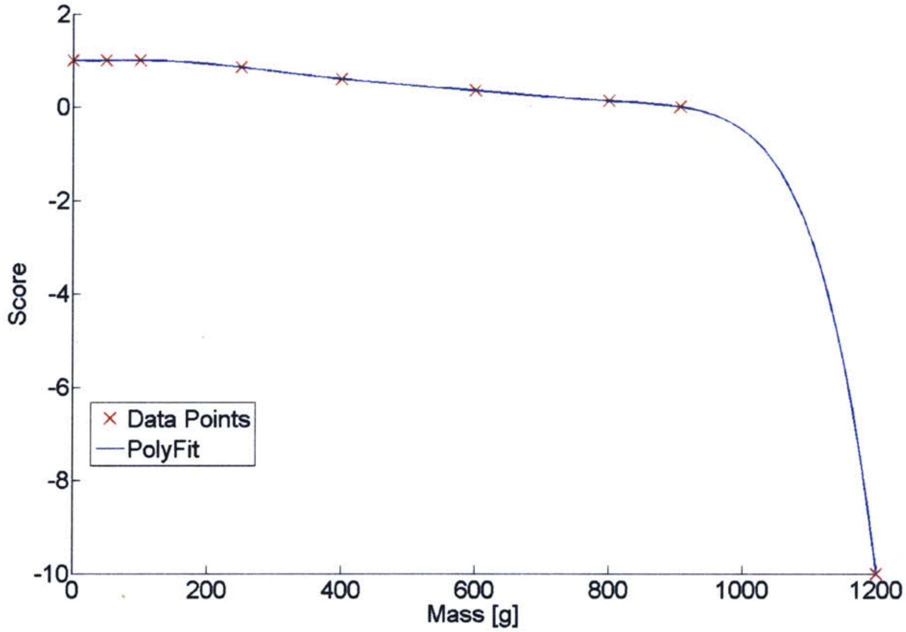
The zoomed in view around the top shows the cost of fitting the polynomial to the data. Since the polynomial being fit is a seventh-order polynomial, it cannot simply be exactly horizontal at the beginning of the x region and still fit all of the data points. As a result, it oscillates a bit before it enters into the steady decrease. While this is contrary to the desire of the function, it should be noted that the amplitude of the oscillation is only about 0.005, which is half of one percent. Thus, the resulting error is quite small and is worth allowing the rest of the curve to fit the data better.

The result of this process is a seventh-order equation that takes the system's mass as input and returns a score. The remaining design criteria follow the same basic pattern as was described above. As a result, the discussion on the remaining criteria will be limited.

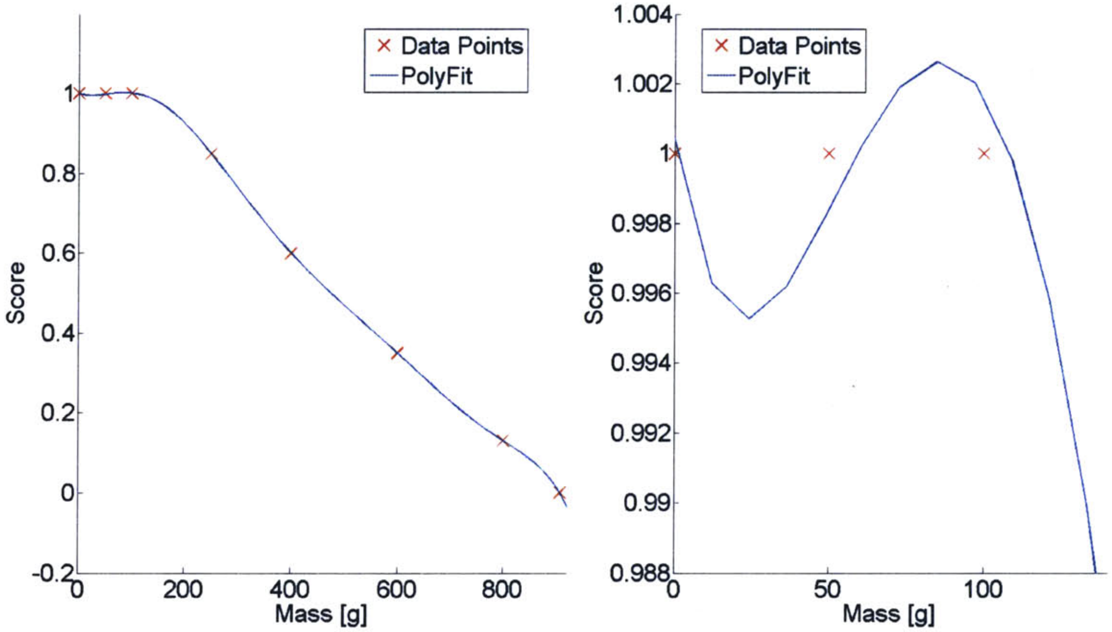
Size

Table 4.1 states that the volume of the targeting system must be under $3680cm^3$, with a preference to be under $1000cm^3$. It must also fit into a $0.3ft$ x $0.3ft$ x $1.4436ft$ space. As a result, the vectors used to generate the bounds were:

```
sizeX = [0.0, 150, 300.0, 600.0, 800, 1000.0, 1500.0, 2400.0, 3000.0,
```



(a) Mass bound



(b) Mass bound zoomed in on specific regions

Figure 4-8: The continuous function that defines the translation of the mass of the components into a score.

```
3315.0, 3670.0, 4500.0];
sizeY = [1.0, 1.0, 1.0, 1.0, 0.98, .95, 0.75, 0.4, 0.2, 0.115, 0.0,
-10.0];
```

An eighth-order polynomial was fit to the data points. The equation for the bound was $-1.1168*10^{-26}x^8+1.5935*10^{-22}x^7-9.1046*10^{-19}x^6+2.6367*10^{-15}x^5-4.0123*10^{-12}x^4+2.9984*10^{-9}x^3-1.039*10^{-6}x^2+0.00012834x+0.99895$. The graph of the equation is shown in the first part of Figure 4-9.

As was the case with the mass bound, the polynomial hovers around the top score up until reaching the objective value, after which it displays a steady fall decline until a score of zero at the threshold value. Again, there is a slight rise initially in the curve, but the amplitude is less than 0.005, or half of one percent of the evaluation.

Power

The next design criteria considered was the power requirements. The threshold value for the power was $5W$, while the objective value was $2W$. The power criteria presented less of a problem to generate the polynomial. Only a sixth-order polynomial was required. The data points that were used were:

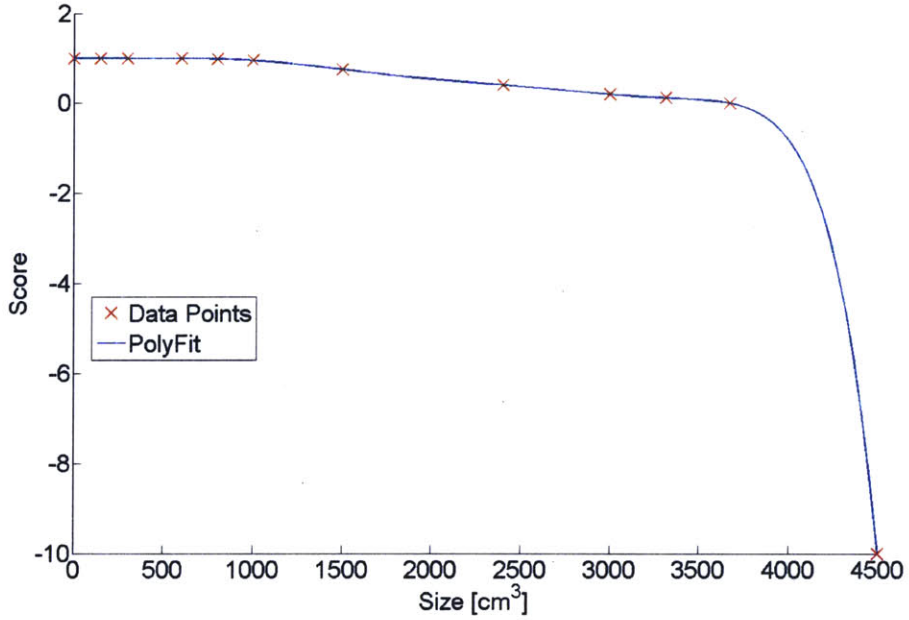
```
powerX = [0.0, 0.6, 1.5, 3.0, 4.0, 5, 7];
powerY = [1, 1, 0.95, 0.60, .3, 0, -10];
```

The resulting equation was $-0.0013595x^6+0.017568^{-15}x^5--0.078233x^4+0.14037x^3-0.14214x^2+0.049481x+1$. The graph of the polynomial is shown in Figure 4-10. The same observations from the previous bounds still apply; the polynomial has a roughly flat portion until the threshold, steadily decreases to the threshold value, which has a score of zero, and then precipitously drops from there. The initial part exhibits a small increase of less than 0.005.

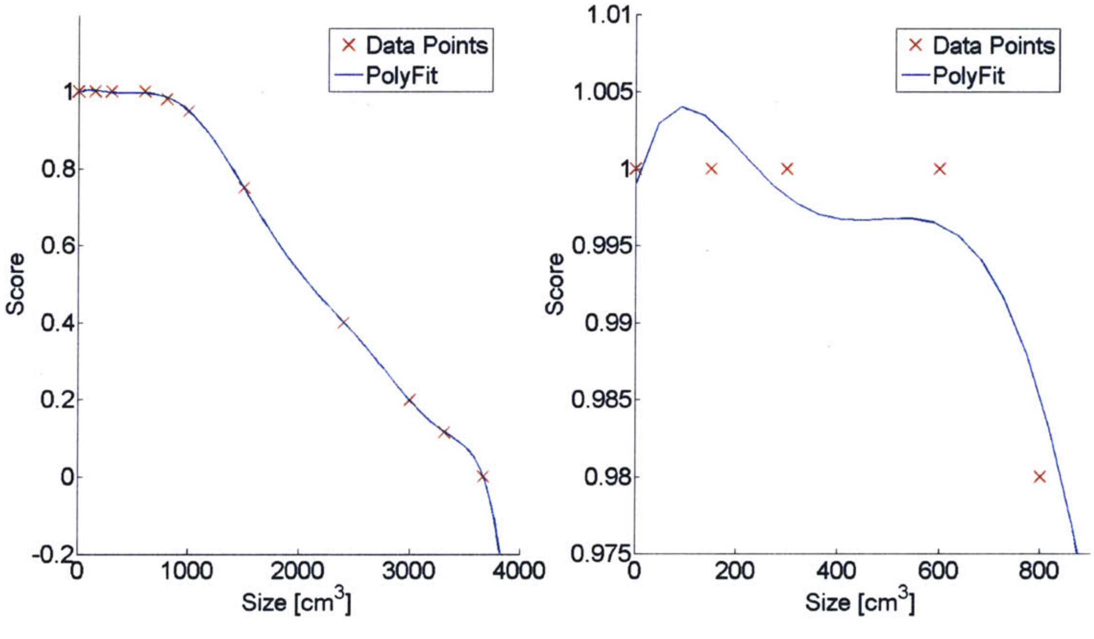
Unit Cost

Unit cost was next in line for consideration. The unit cost could be no more than \$10,000.00, with a preference to be under \$5,000. Since the objective value was such a high percentage of the threshold value (50%), fitting the polynomial was difficult. If the same flat region was desired, an 11-order polynomial was required, with oscillations that were approximately 1-1.5% of the total value. However, doing so would offer no difference in scores for over half of the x values in the region of interest. As a result, it was decided to have a gradually sloping region up until the objective, followed by a increased sloped region until the threshold. Doing so eliminated the initial oscillations entirely and allowed for a reduction in order to a ninth-order polynomial. The data points used were:

```
unitCostX = [0.0, 250, 500, 1000.0, 1500.0, 2000.0, 2500.0, 4000.0,
5000.0, 7000.0, 8500.0, 9200.0, 10000.0, 12000.0];
unitCostY = [1, 0.995, 0.99, 0.98, 0.97, 0.96, .95, .90, .85, 0.6,
0.3, 0.15, 0, -10];
```

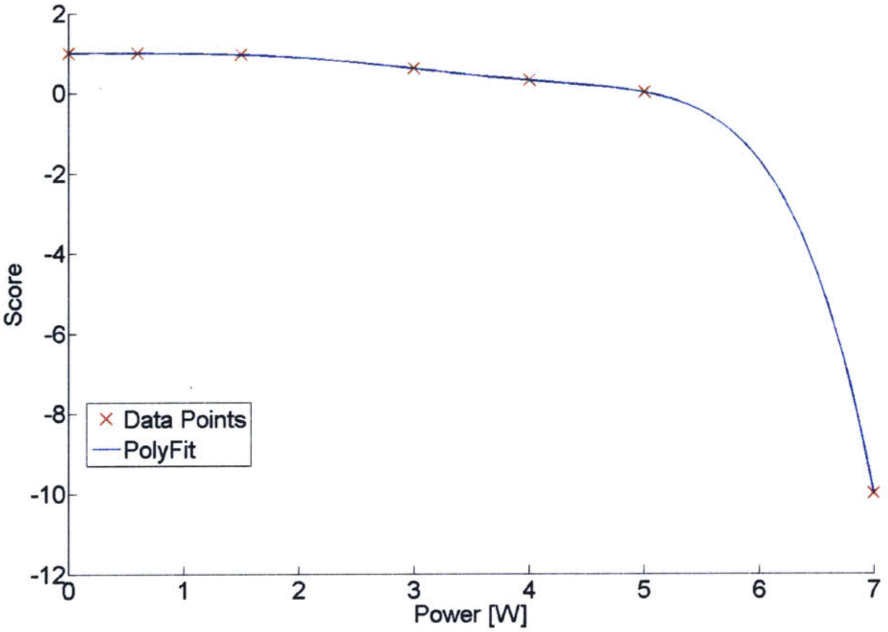


(a) Size bound

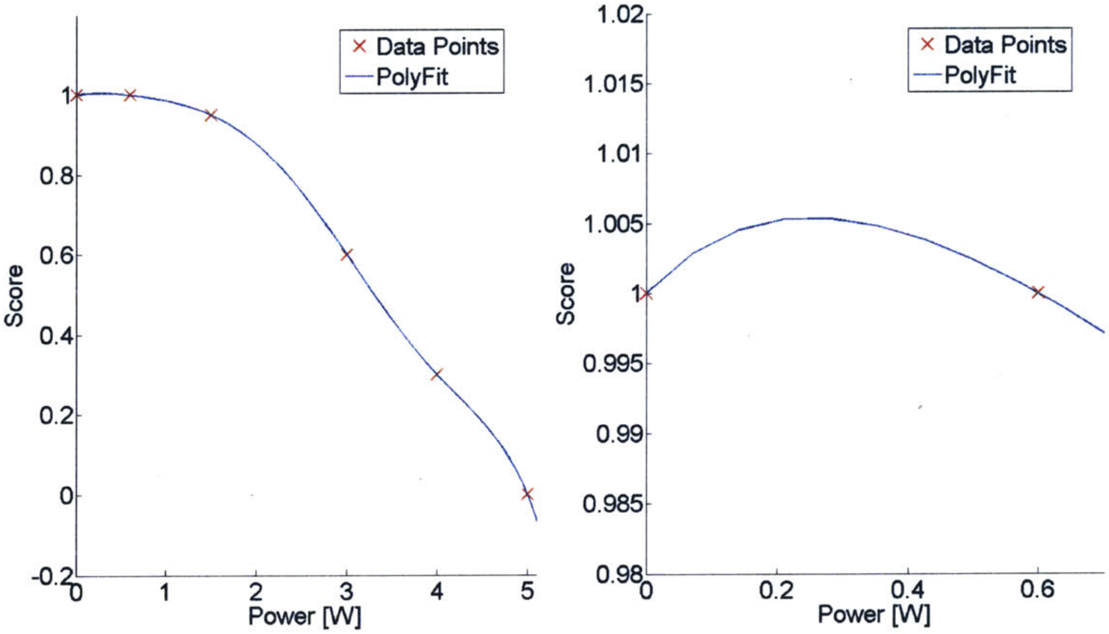


(b) Size bound zoomed in on specific regions

Figure 4-9: The continuous function that defines the translation of the volume of the components into a score.



(a) Power bound



(b) Power bound zoomed in on specific regions

Figure 4-10: The continuous function that defines the translation of the power consumption of the components into a score.

The resulting polynomial was $-7.8425*10^{-34}x^9+3.3928*10^{-29}x^8-6.0918*10^{-25}x^7+5.8730*10^{-21}x^6-3.2799*10^{-17}x^5+1.0647*10^{-13}x^4-1.9184*10^{-10}x^3+1.7011*10^{-7}x^2-7.5339*10^{-5}x+1.0022$. The corresponding graph is shown in Figure 4-11. The figure shows the gradual decrease at the beginning, with the change in slope occurring around the objective value. The zoomed in view over the first half of the region shows that no overshoot occurred.

Accuracy

The accuracy threshold was set at $50m$, or $164.042ft$, with an objective to be less than $10m$, or $32.8084ft$. A ninth-order polynomial was fit to the following data points:

```
accuracyX = [0.0, 8.0, 16.0, 32.0, 60, 90.0, 120.0, 140.0, 153.0,
165.0, 200.0];
accuracyY = [1.0, 0.98, 0.96, 0.90, 0.7, .5, .3, 0.16, 0.08, 0,
-10];
```

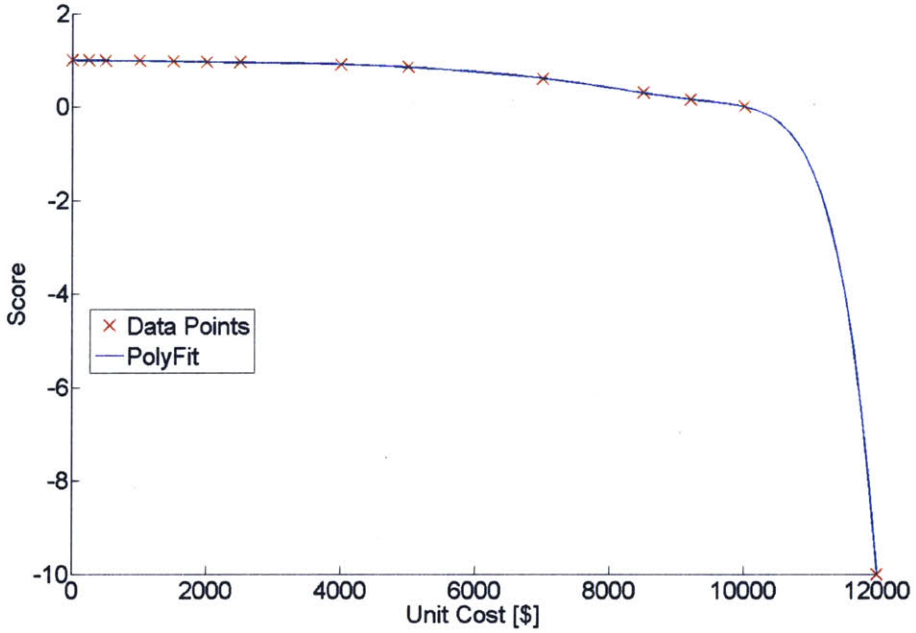
The resulting equation was $-7.1178*10^{-18}x^9+5.2122*10^{-15}x^8-1.6049*10^{-12}x^7+2.7041*10^{-10}x^6-2.7136*10^{-8}x^5+1.6456*10^{-6}x^4-5.7393*10^{-5}x^3+0.00095006x^2-0.0079026x+1.0011$. The graph of this equation is shown in Figure 4-12. As with the unit cost bound, the function experiences a monotonic decrease in the associated region with no initial oscillations above 1. The gradual drop at the beginning turns into a greater decrease after the objective value. The precipitous drop is also present following the threshold value.

Robustness

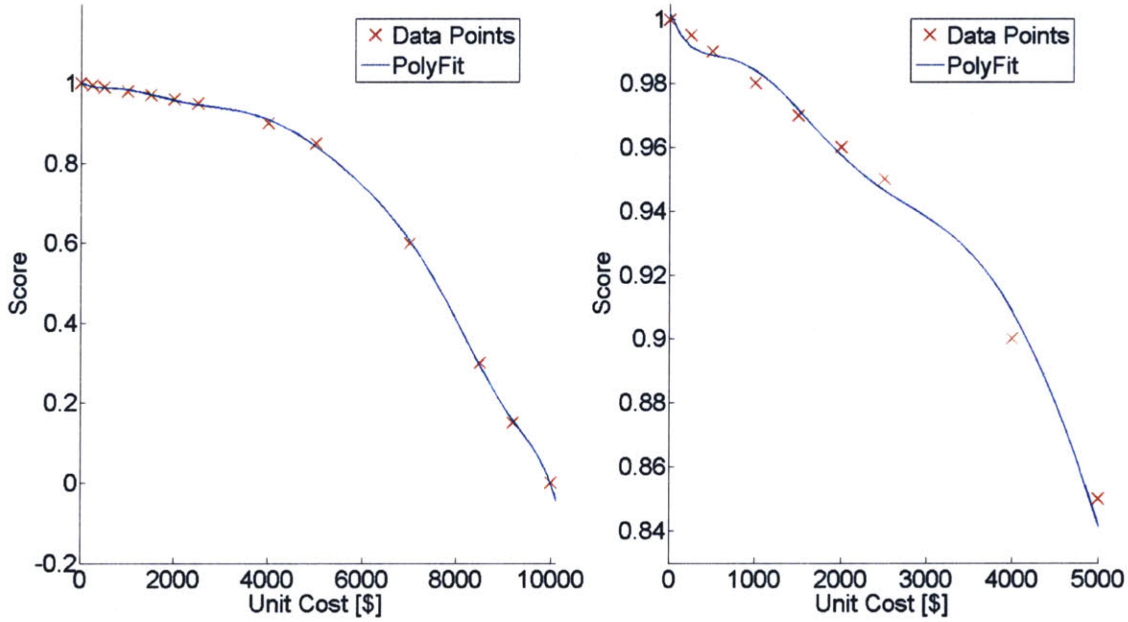
The penultimate criteria considered was robustness. Unlike the previous criteria, there is not a clear physical definition of robustness. As a result, a measurement for robustness needed to be created before the bound could be set. As will be described later, a risk analysis was performed for determining the robustness of the system. In it, different error levels were established and the amount of TLE was determined for each scenario. From the TLE and the probability of occurring, two quantities were calculated: the average error of each method across the various error scenarios and the standard deviation of the TLE for each method. While the average TLE gave a measure of the accuracy for the system, the standard deviation provided a measure of how precise the accuracy values were. Since the scenarios were create to represent the likely parameters the system would face during a mission, the standard deviation thus served as a measure of robustness. It was decided that the maximum standard deviation that would be acceptable for a method was $50ft$ while the objective was $15ft$. A seventh-order polynomial was fit to the following data points:

```
robustnessX = [0, 2, 5, 7, 15, 25, 37, 45, 50, 65];
robustnessY = [1, 1, 1, 1, 0.85, 0.5, 0.22, 0.08, 0, -10];
```

The resulting equation was $-1.6012*10^{-10}x^7+2.9332*10^{-8}x^6-2.1328*10^{-6}x^5+7.7387*10^{-5}x^4-0.0014146x^3+0.010719x^2-0.027232x+1.0068$. A graph of the equation is shown in Figure 4-13.

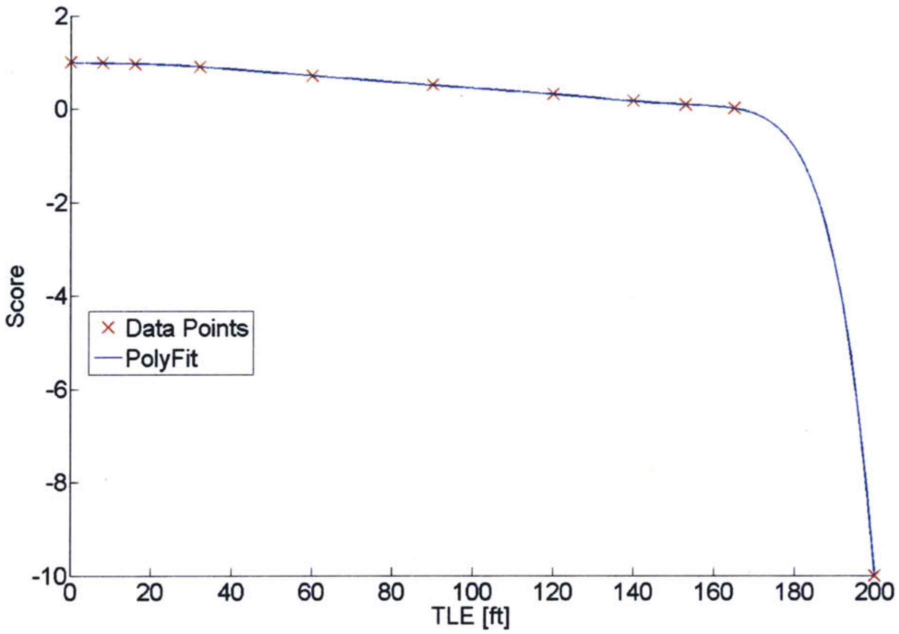


(a) Unit cost bound

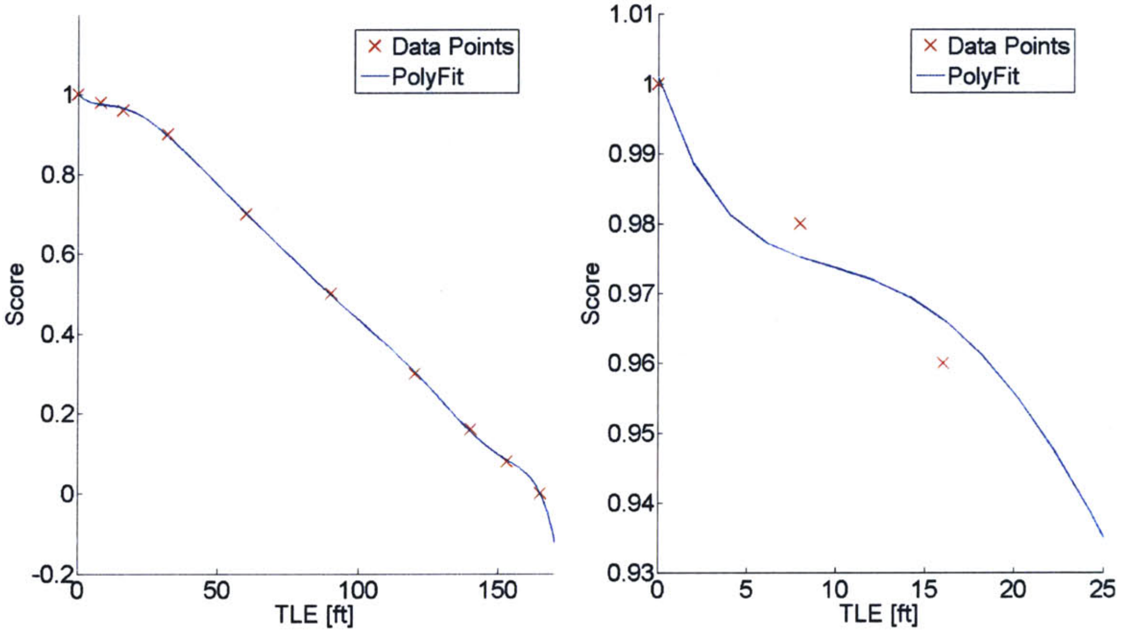


(b) Unit cost bound zoomed in on specific regions

Figure 4-11: The continuous function that defines the translation of the unit cost of the components into a score.

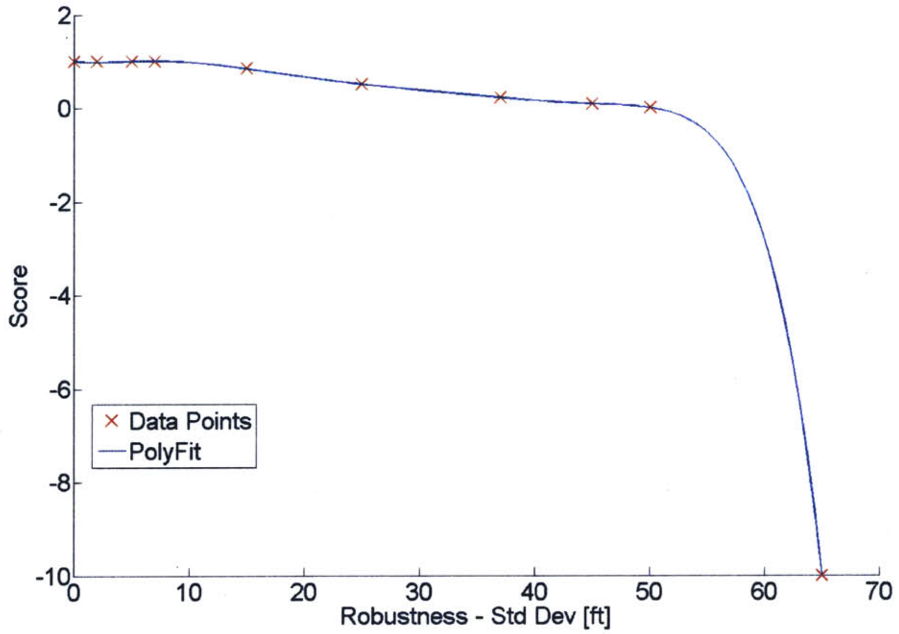


(a) Accuracy bound

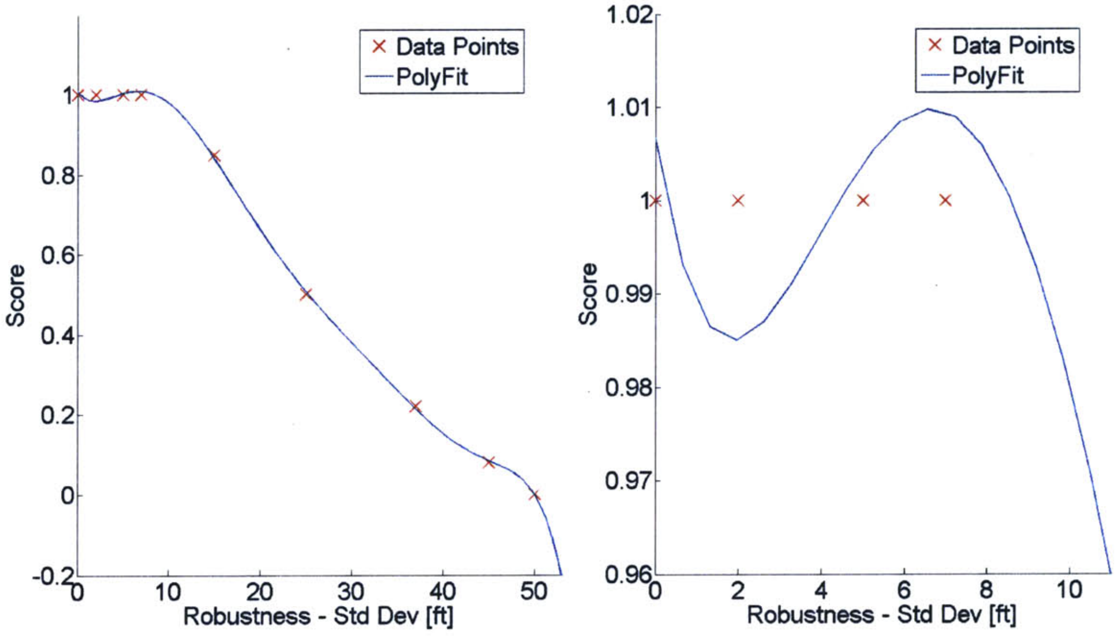


(b) Accuracy bound zoomed in on specific regions

Figure 4-12: The continuous function that defines the translation of the unit cost of the components into a score.



(a) Robustness bound



(b) Robustness bound zoomed in on specific regions

Figure 4-13: The continuous function that defines the translation of the robustness of the solution into a score.

The figure shows the function has a shape similar to those of the first several design criteria. The first part of the figure show the same relatively flat beginning followed by steady decrease and then precipitous drop, as was the case with the others. The view zoomed in around the beginning shows the initial section suffered from some oscillations, with an amplitude of around 0.015 at its maximum. This is not desired, but was necessary to obtain the shape for the rest of the curve.

Latency

The final criteria considered was latency. No requirements were specified in Table 4.1 for latency. The threshold value was then set at 30s, with an objective of receiving a solution in less than 10s. The data points used to fit the polynomial were:

```
latencyX = [0, 2, 5 10, 14, 18, 22, 26, 28, 30, 37];
latencyY = [1, 1, 1, 0.95, 0.78, 0.58, 0.35, 0.15, 0.05, 0, -10];
```

A ninth-order polynomial was fit to the data, yielding an equation of $-2.4994 * 10^{-11}x^9 + 3.1907 * 10^{-9}x^8 - 1.6669 * 10^{-7}x^7 + 4.5824 * 10^{-6}x^6 - 7.1150 * 10^{-5}x^5 + 0.00063227x^4 - 0.0032793x^3 + 0.0090295x^2 - 0.0094421x + 1.0002$. The figure associated with this equation is Figure 4-14. The figure shows behavior that is similar to the other bounds. The initial oscillation had an amplitude of 0.003, making it negligible.

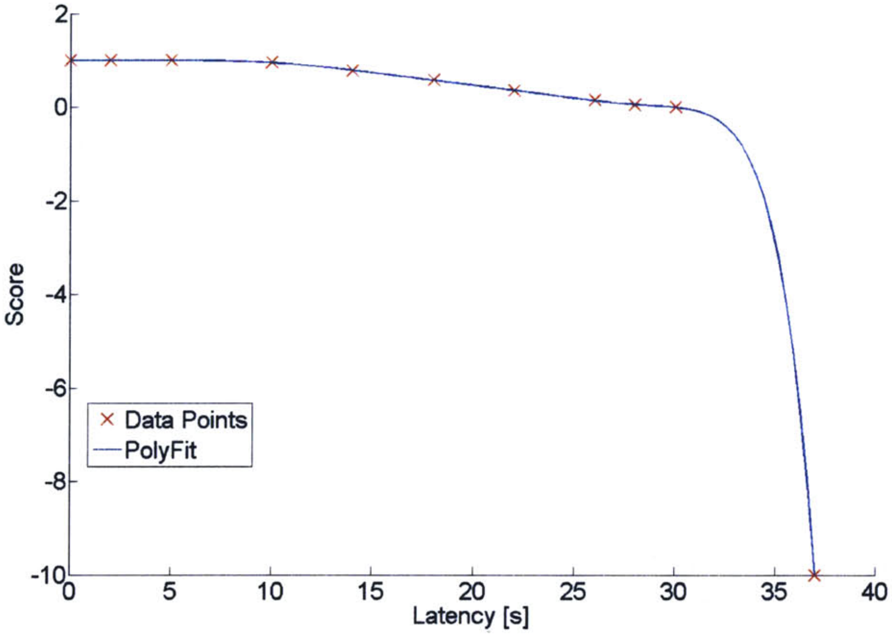
4.4 Define Alternatives

4.4.1 Methodology

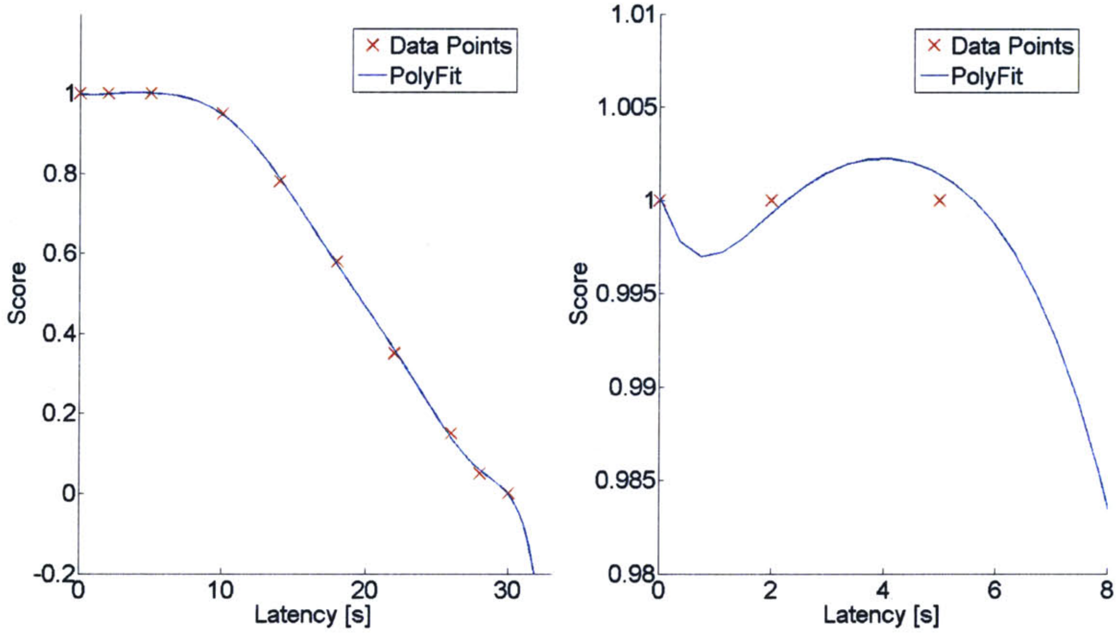
The next step in the design process is to define the alternatives. As was discussed in Section 2.2.4, the two major tasks of this step are to create an initial list of alternatives and then to downselect the list to include only those alternatives that merit future investigation. Of the many ways that exist to create a list of alternatives, three were discussed, namely brainstorming, systematic decomposition, and past experience. Each of these will be discussed in further detail, followed by a discussion of how to downselect the list.

Brainstorming

The first idea generation method is brainstorming. As stated in Section 2.2.4, brainstorming is a creative method whose goal is to come up with as many solutions to a problem as possible. While it can be done individually, brainstorming for a team-based design is often done in a group. Typically, the group is between 5-20 people whose expertise spans the gamut of the disciplines required for the problem. The basic premise of brainstorming sessions is that all the people participating give suggestions as soon as they come to mind. Criticism and analysis are not permitted during the idea generation phase of the session, with the goal being to not spend much time on any given suggestion, but instead to simply generate ideas. By eliminating criticism,



(a) Latency bound



(b) Latency bound zoomed in on specific regions

Figure 4-14: The continuous function that defines the translation of the latency of the solution into a score.

participants are free to share any idea, thereby soliciting responses that may be initially seen as wild or impossible. This usually enhances creativity and expands the solution space beyond the conventional approaches. Also, it allows people to use the outrageous ideas as a basis to piggyback more practical solutions.

The logistics of conducting a brainstorming session will now be discussed. The session is conducted in a single room that is large enough to house everyone comfortably. The room should have a dry erase board, chalk board, flip chart, overhead projector, or some equivalent way of displaying ideas in a prominent fashion so that everyone can see all the ideas at once. The room should also be free from distractions, which may include phones, clocks, or other things of this nature. Before the session begins, the topic, problem, or objective to be considered is displayed, often done in the form of a question. It must be done such that it does not include any implementation details or bias the discussion in any way. A time limit is then established before discussion begins. The time varies depending on the size, complexity, or novelty of the topic, but typically lasts 5-20 minutes. More time can be used if ideas are still being generated.

One person is designated the recorder. His/her job is to display the message on the selected media, which usually involves either writing or typing the idea. Also, a moderator is selected for the session. This person's job is to maintain an environment that is conducive to idea generation. As established above, this entails keeping a criticism-free discussion. Also, the facilitator should keep the ideas focused on the topic and ensure that ideas are not discussed for too long.

Once the time limit for idea generation has passed, the ideas generated are organized by category. Discussion and analysis of the ideas can then be done. By organizing the ideas into categories, new ideas may be generated. This is encouraged.

Systematic Decomposition

The next method considered for defining potential alternatives is systematic decomposition. As described in Section 2.2.4, at least two different types of systematic decomposition exist. Designers could either decompose the problem into different classes of approaches or decompose the desired functionality and then list ways to accomplish the tasks. These two different methods will be discussed in detail below.

Class of Alternatives Decomposition The first alternative for systematic decomposition is to decompose the problem into different classes of alternatives. This approach calls for the designer to create a hierarchy for the problem, defining characteristics that allow for its classification. An example of this was shown in Figure 2-2 related to navigation. First, the designers list initial characteristics that distinguish some alternatives. These do not have to necessarily be at a certain level in the tree, but simply serve as a starting point. Once some characteristics are listed, designers look for ways to expand the tree based upon the existing characteristics. One way is to look for similarities that the characteristics have such that a higher level characteristic could be formed. Also, the designers look for ways to decompose the existing characteristics further. Once a sufficient amount of characteristics are defined, an initial hierarchy is established. As new information is added, characteristics are promoted and demoted as necessary.

Some problems have more than one characteristic that fundamentally divides the solution space without their being an obvious hierarchial relationship between them. In this case, one of the characteristics needs to be arbitrarily selected as the higher level, with the other criteria serving as a way to decompose all the alternatives in the higher level. An example of this is shown in Figure 2-2. In the navigation example, both absolute vs. relative and passive vs. active serve as characteristics by which the solution space could be fundamentally divided. As a result, one of the two characteristics needs to be selected as the higher level. As the figure shows, absolute vs. relative was selected to be the parent. Then, the other characteristic, passive vs. active, was used to divide all the alternatives at the higher level. This is shown as passive vs. active being present under both the absolute and relative blocks in the figure. Ultimately, the leaves of a fully developed systematic decomposition tree are the specific instances of solutions that could be employed to address the problem.

Functional Decomposition The second type of systematic decomposition calls for the breakdown of the end product into its functional components followed by listing ways to achieve these functional goals. In doing so, the designers start with what they need and then list ways to obtain these needs, in essence working backwards. Figure 2-3 contains an example, again applied to navigation. In this example, it was decided that navigation is essentially starting with a position and orientation and then tracking how these states change over time. Thus, the four components of navigation are position, Δposition, orientation, and Δorientation. Once established, each of the components can be treated independently. The designers then list ways to obtain each of the individual functional components. In doing so, the designers are not biased at all by conventional wisdom on how to approach a problem, but instead are focused on providing the functionality required to solve a piece of the overall problem. This may lead to creative combinations that might have otherwise not been developed. A complete alternative can then be achieved by combining the individual pieces of each of the functional components such that all of the necessary components are provided. In the navigation example, an alternative for each of position, Δposition, orientation, and Δorientation would be selected and then combined to form a complete navigation solution. A specific example might be GPS + IMU. The GPS provided the position and Δposition and the IMU provides the orientation and Δorientation.

Past Experience

The final method for idea generation is past experience. The fundamental principle in this method is to list how the problem has been approached in the past. This essentially becomes a search problem involving both human and computer memory. Human experience will be considered first, followed next by media searching.

The first half is to glean information from the experience of all group members or people connected with the group members. Starting first with the group members, each person should list related projects that he/she has worked in the past, is currently working on, will be working in the future, or have heard that others have done/are doing/will do. Whether from a member of the group or indirectly through someone connected with a group member, the information of interest includes what was the attempted solution, what worked, what didn't, what were the major obstacles, what were the lessons learned, what would be done differently if the project were to be repeated, or anything else that would aid the new evaluation.

While it may sound erroneous to include current and future projects in past experience, the reason for the inclusion in the list is to obtain any information that has already been done on these projects. If the project is slated to be done in the future or is currently being worked on, some amount of information is available about the topic and this information could assist in the evaluation of the new topic. Hence, though a matter of semantics, this is not about past projects, but instead about past experience, which includes anything that is known about a topic.

Another valuable source of information is projects that others have worked on outside the immediate contact of members of the group. Finding this information is essentially a literature search. Currently, sources include technical papers, journal articles, conferences, proceedings, websites, press releases, news paper articles, textbooks, email, or any other way that information is shared. Many tools exist to help designers find relevant information. In an increasingly electronic world, Google or some equivalent web-based search engine provides valuable assistance. Google Scholar, Citeseer, and other equivalent websites are good for searching for technical papers. Though not necessarily obvious to the current generation, libraries, microfiche, and other examples of hardcopy media are still useful, especially for older sources that have yet to be digitized. Google Desktop or equivalent executable programs are designed to search the contents of a hard drive or other computer-based media. These are invaluable to search for past projects on a given computer or network as opposed to the internet. As technology continues to grow, new sources of storing or finding data are sure to be developed. The methods of people interested in finding relevant data will have to adapt with the changes in technology.

List Downselection

In almost every circumstance, the list creation process provides more alternatives than can be fully evaluated. In such cases, the list needs to be pruned down to a manageable size. As a result, the alternatives need to undergo a very high level analysis in order to select the possibilities that are most meritorious of future consideration. It is often best to choose a set number of alternatives to evaluate and then to whittle the list down to that number. The number depends on the time and resources that will be dedicated to the evaluation, but is typically around 3-5. The evaluation does not have to be a rigorous, full-blown analysis, as the optimal answer is not looking to be achieved. Instead, this is essentially looking for the top tier of alternatives. As stated in Section 2.2.4, the downselection is typically based on a number of criteria that include the feasibility of the alternatives, the customer's requirements and desired functionality, the technical maturity of the solution, the amount of risk deemed acceptable for the project, the amount of resources (including time and money) available, and the skill set of the workers involved.

The feasibility of the alternative is typically one of the first criterion used to evaluate the possibilities. If a potential solution does not seem feasible at all or within the given resources available for the project, there is no need to consider it further. A similar evaluation criterion is the alternative's technical maturity. A metric for establishing the technical maturity of an alternative is its technology readiness level (TRL), if applicable. The TRL evaluation is a 1-9 rating of the readiness of a technology, ranging from 1 being basic principles observed to a 9 of being fully fielded. If a technology is in its infancy and it needs to be applied to a system with a short time frame, the project may have insufficient development time in order to get the solution to work reliably. The possibility may be feasible eventually, but not for the given instance of the project at hand. This principle can be generalized to include all resources available to the project, including money, machinery, and any other resource. A final limiting factor that would make an alternative feasible in theory but not in practice would be the skill set of the group that will work on the project. Even if others could make a solution work, if the team working on it does not possess the necessary skills, then the solution should be eliminated.

If after applying all the above standards too many alternatives still exist, then the alternatives need to be evaluated based upon a preliminary analysis of how well they meet the customer's requirements and desired functionality. Often, risk is a subset of these requirements, which was another factor listed above and in Section 2.2.4. Again, this does not need to be a flawless evaluation, but can actually be done in a relative manner. While this was seen as a weakness of the matrix-like evaluation, a relative evaluation here is actually allowable since the designers are simply trying to decide which alternatives to evaluate rigorously in the future. At the end of this step, a list of potential solutions for the problem should be created and downselected to an appropriate size for the given resources available in the evaluation.

4.4.2 Application

For the targeting system, the method used for alternative generation was a combination of systematic decomposition and past experience. The combination of these methods generated more possibilities that could be possibly be evaluated in the time frame. As a result, the list of potential solutions was downselected to include three alternatives. The process of list generation and list downselection for the small UAV targeting system is described in the next several sections.

List Creation

The primary method of alternative generation was systematic decomposition, specifically, class of alternatives decomposition. The decomposition was then supplemented with past experience analysis. The final result of the combined analysis is shown in Figure 4-15. This is not necessarily an exhaustive list, but it provided an adequate

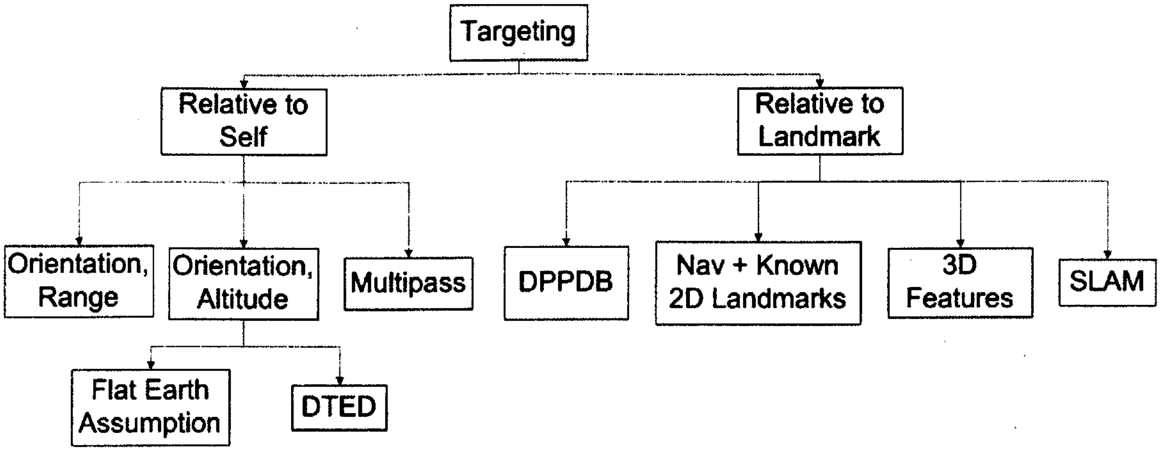


Figure 4-15: Hierarchy of targeting solutions

coverage of the solution space for the problem in terms of the number of alternatives and the quality of alternatives.

It is important to reiterate the statement at the beginning of the chapter that the goal of the targeting system is to provide the absolute coordinates of the target. In Figure 2-2, the navigation problem was divided into absolute versus relative. The absolute solutions depended upon having a sensor onboard that could process the information received and deduce a navigation answer. However, since the UAV does not have access to any sensors onboard the target, the absolute coordinates need to be found relative to a known object. Figure 4-15 shows two ways of doing this, labeled relative to self and relative to landmark. Relative to self refers to methods where the absolute position of the vehicle is known and the position of the target is established relative to the vehicle. Methods classified under the relative to landmark category assume that the position of an object other than the vehicle is known and the position of the target is found relative to that object, referred to as a landmark.

These two classes have distinct implications on the system design and how it is operated. For the relative to self methods, the self position of the vehicle must be known well in order to achieve an accurate targeting solution. As a result, the system must have a high fidelity navigation system. By contrast, the relative to landmark methods of targeting do not care at all about the position of the vehicle, meaning that the navigation system has no bearing on the accuracy of the targeting solution. However, the methods require the absolute position of landmarks to be known a priori, adding an additional layer of infrastructure to the problem. If the mission is being performed in an area that had previously not been scouted, then landmarks might not be known, rendering the methods useless. If the landmark information was obtained after the soldiers were deployed, then they would need a way to download the data while in the field. Thus, relative to landmark methods require a priori supporting information given to the system either prior to deployment or in the field, thereby limiting the spontaneity of the missions it can perform.

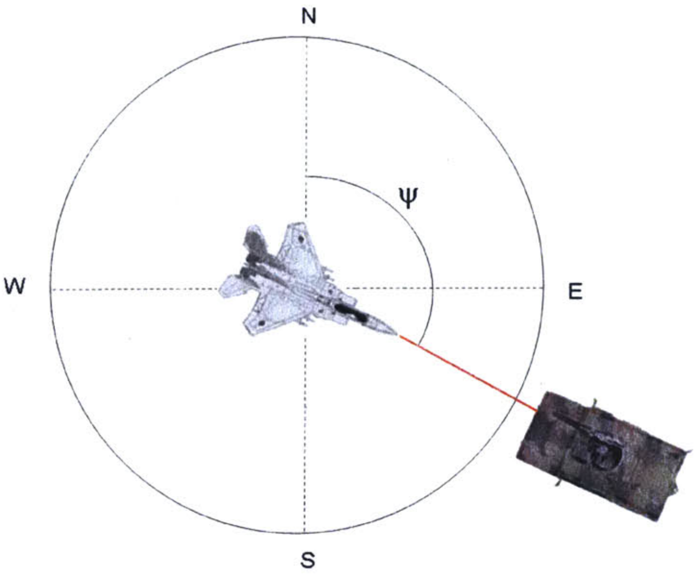


Figure 4-16: Top-view (XY view) of an encounter. ψ is the vehicle's heading.

Relative to Self Alternatives The relative to self alternatives were then decomposed an additional level, resulting in three entries. They were knowing the vehicle's orientation and the range to the target, knowing the vehicle's orientation and altitude, and using a multipass approach. Each of these alternatives will be briefly described in succession.

The first alternative considered was using the vehicle's orientation and altitude to ascertain the target's position. This can best be seen by examining Figures 4-16 and 4-17. The target's position, or any other position, is defined as a point in three dimensional space, thereby requiring an x, y, and z component. On Earth, this could also translate into latitude, longitude, and elevation. The first step in the method is to determine the position of the vehicle. This is used as the reference point for determining the target's location. Once the vehicle's position is determined, then the heading of the vehicle is determined, as shown in Figure 4-16. The important piece of information, however, is not the vehicle's heading, but instead the line of sight (LOS) vector of the camera. If the camera is looking straight forward, as was assumed to be the case in Figure 4-16, then the camera LOS vector and the vehicle heading would line up. If the camera is gimballed, then an additional calculation would be required in order to determine where the camera is pointed given the vehicle's heading. However, as was established in Chapter 3, the configuration of the vehicle was assumed to have a fixed camera pointed in the same direction as the heading vector. The camera LOS vector is important because it defines the line in the xy plane on which the target could lie. Thus, two of the three coordinates of the target's position are constrained once the camera's LOS vector is determined.

In order to pinpoint the target's xy position, along with determining its elevation, the vertical component must also be considered. The geometry of the problem is

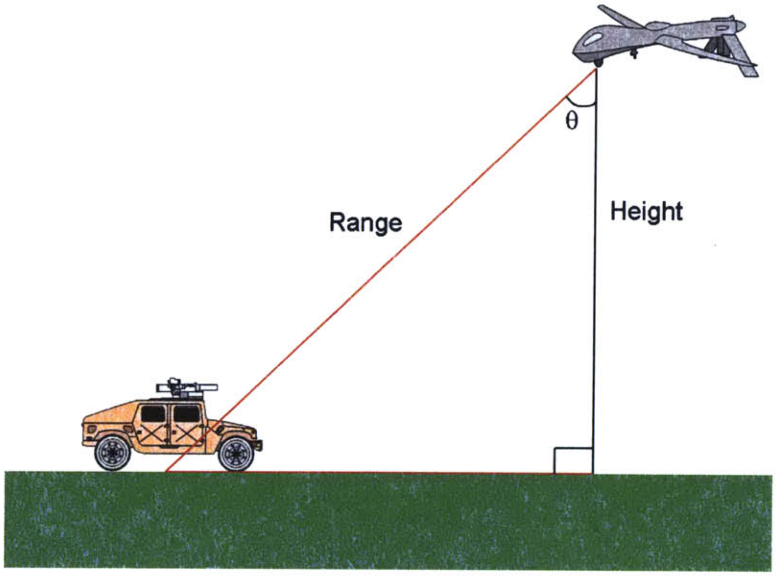


Figure 4-17: Example of Flat Earth method. The range (hypotenuse) and horizontal leg are in the xy direction of the camera's LOS vector, which is the red vector (heading vector) in Figure 4-16.

shown in Figure 4-17. In this example, the hypotenuse of the right triangle corresponds to the camera's LOS vector to the target that was seen in Figure 4-16. The horizontal base of the triangle is also colored red because, when viewed from the top view, this would also be in the direction of the camera's LOS vector. It is thus important to note that though a side view is shown, the red lines (hypotenuse and horizontal leg) are in the direction of the red vector (heading vector) in Figure 4-16. As a result, Figure 4-17 is not in the yz or xz plane.

The geometry shown in Figure 4-17 can be used to determine the target's location. As mentioned above, the current method uses the vehicle's orientation and altitude. Figure 4-16 showed how the vehicle's heading could be used to determine the xy line on which the target was located. The roll and pitch of the vehicle would similarly be used to determine Figure 4-17's θ, which is the angle between vertical and the center of the camera's field of view. The other piece of known information is the vehicle's altitude, which is typically given by an altimeter. This provides the vertical component of the right triangle. With θ and the height known, the remaining two sides of the triangle can be determined. The length of the horizontal leg of the triangle, which corresponds with the length of the camera's LOS vector in Figure 4-16 is used to determine the exact xy position of the target. The z component of the target can be determined a number of different ways, but the simplest is to assume that the Earth is flat. Using this assumption, the elevation at takeoff would be the same elevation of the target. The elevation at takeoff is typically used to calibrate the altimeter and thus serves as the reference for the altimeter for the remainder of the flight.

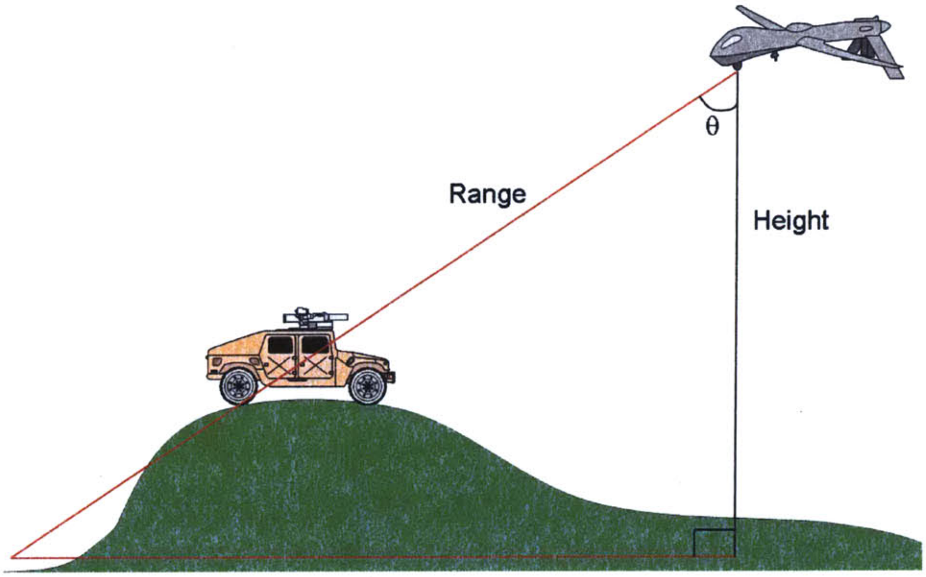


Figure 4-18: Example of changing terrain while using Flat Earth method. Again, red vectors along the heading vector, as in Figure 4-16.

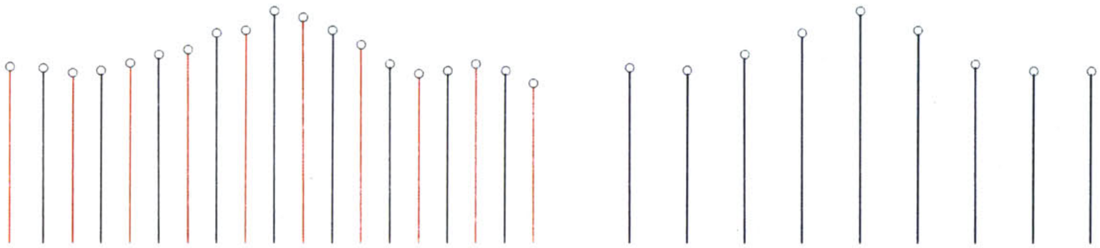
While the assumption of a flat Earth simplifies the calculation process, it also introduces a large source of error. Assuming that the Earth does not change in elevation over the course of 10+ kilometers is not necessarily a realistic assumption in most terrains. Figure 4-18 illustrates the magnitude of the problem. Even if all the other information is perfect, if a change in elevation has occurred between the reference altitude and the target, the coordinates produced by the flat Earth method could be significantly wrong.

The second method shown in Figure 4-15 under know orientation, height addresses this source of error. This method is to incorporate DTED data into the calculation. DTED stands for digital terrain elevation data and it provides a digital record of the elevation of the Earth at regular intervals. An example of the basic concept of DTED is shown in Figure 4-19. The top part of the figure shows a sample terrain with simulated DTED poles. The way that DTED works is by taking elevation data at regularly spaced intervals. These intervals can be thought of as poles that have a height equal to the elevation of the Earth. This is shown in the top figure by the evenly-spaced black and red poles that intersect the surface of the terrain.

The different color poles simulate the different levels of DTED data. DTED has six different levels numbered 0-5, deviating only in the spacing of the poles. The poles for DTED level 0 are spaced approximately 1 km apart (3 arcsec) while the the poles for DTED level 5 are approximately 1 m apart (0.0370 arcsec), with the intermediate levels being 100 m, 30 m, 10 m, and 3 m, respectively.

The different DTED levels are simulated in the second and third part of Figure 4-19. In these two parts, the terrain that was present in the first part of the figure has been taken away, leaving only the poles. The black poles represent the poles present

(a) Sample terrain



(b) All associated poles

(c) Most widely spread poles

Figure 4-19: Example of DTED providing the shape of a given terrain at different levels of fidelity.

at a lower level, such as 1, while the red poles represent the next poles available in the next level of DTED. As shown in the third part of Figure 4-19, the lower level captures the basic trend of the terrain, but obfuscates some of the detail. This is perhaps best seen in the area to the left of the highest point in the terrain. The actual terrain fluctuates a bit between local minima and maxima, but the lower level DTED poles simply show a steady climb to the top.

The second part of Figure 4-19 shows the result of jumping up a level in DTED. All of the poles from the previous level are present (all the black poles are still there), but now additional poles are available in between the existing poles. This is shown by the presence of the red poles. Stepping up a level has the benefit of increasing the resolution of the data, which thereby gives a clearer picture of the terrain. However, increasing DTED levels requires significantly more storage space for the data and as the levels increase, the availability of the data becomes limited, either due to the information being classified or because the information is not available within the required accuracy for every place on Earth.

With DTED data, the problem seen in Figure 4-18 can be readdressed. The DTED data can allow for a construction of a 3D model of the area immediately around the target. Then, instead of just blindly using the height above the reference altitude as the height to the target as with the flat Earth method, the camera LOS vector can be extended from the vehicle until it intersects the terrain. Assuming the target is centered on the camera, the point of intersection is the location of the target. In Figure 4-18, the intersection between the camera's LOS vector, which is the red line labeled range, and the terrain occurs at the top of the hill, which is where the target is located. Since DTED information is known about the terrain, the elevation of the intersection point can be estimated. The vertical leg of the triangle labeled height in the picture is then seen as the difference from between the vehicle's altitude, given by an altimeter, and the elevation of the target, given by DTED, thereby eliminating the error due to change in elevation. Figure 4-18 shows that this is important because small changes in elevation can lead to drastic errors due to the orientation of the camera. The closer θ is to 90°, the more drastic the error due to the fact that the estimate is following the hypotenuse of the triangle.

The next targeting method under the auspice of the relative to self class is using the vehicle's orientation and the range to the target to obtain the target's position. This method is very similar to using the orientation and altitude method and is also best represented through the use of Figure 4-17. The previous method called for the use of θ and height to determine the other sides of the triangle whereas now θ and range are used to define its characteristics. The range is defined as the distance between the vehicle and the target. As before, the known side and angle of the triangle can be used to calculate the length of the horizontal side, which is then used with the heading and camera's LOS vector to determine the target's xy position. An advantage of this method as compared to using the flat Earth method is that it is not susceptible to changes in elevation. Since the range is the distance between the vehicle and the target, the hypotenuse of the triangle is constrained and is therefore unable to be too long or too short due to changes in the terrain, as was the problem with the flat Earth method seen in Figure 4-18. Instead, the vertical component of the triangle is calculated from the hypotenuse and the angle, giving an accurate estimate of the height without the need for the DTED information. However, the range to the target needs to be determined. This is typically done using a laser rangefinder or in software. The discussion of the tradeoffs of this method will be delayed until the appropriate step in the design method.

The final relative to self option listed is in Figure 4-15 is the multipass method. In this method, a time series of estimates of the target's location are taken from a variety of different angles to the target. This is shown in Figure 4-20. In essence, the vehicle flies a circle around the target, estimating the target's position at various different points along the way. The estimates each have errors along the camera's LOS vector and in a sphere surrounding the target's location. However, as more estimates are taken, the errors cancel out and the estimate becomes more precise, thereby converging to an accurate estimate of the target's position. The two major downsides two this method is that it takes longer to achieve a targeting solution and it is susceptible to errors due to a moving target.

Relative to Landmark The previous section covered the class of methods shown in Figure 4-15 as the relative to self methods. These methods were characterized by using the vehicle's position as the foundation and then finding the target's position relative to that location. The other class of methods mentioned in Figure 4-15 is the relative to landmarks method. In this class of methods, as was discussed above, other

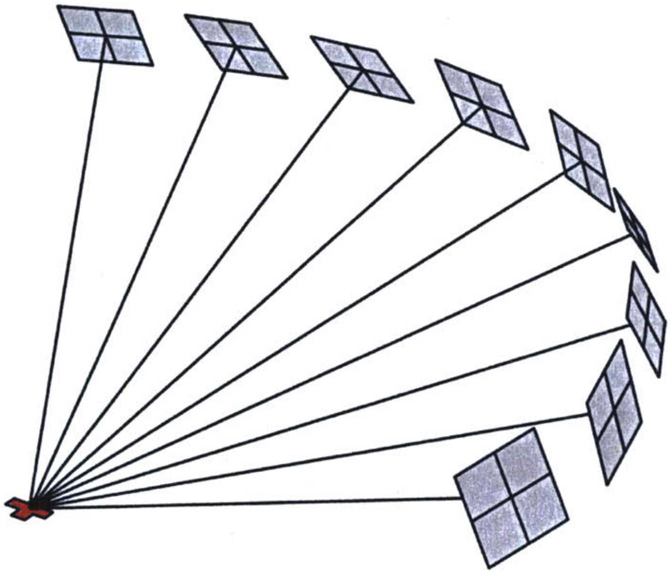
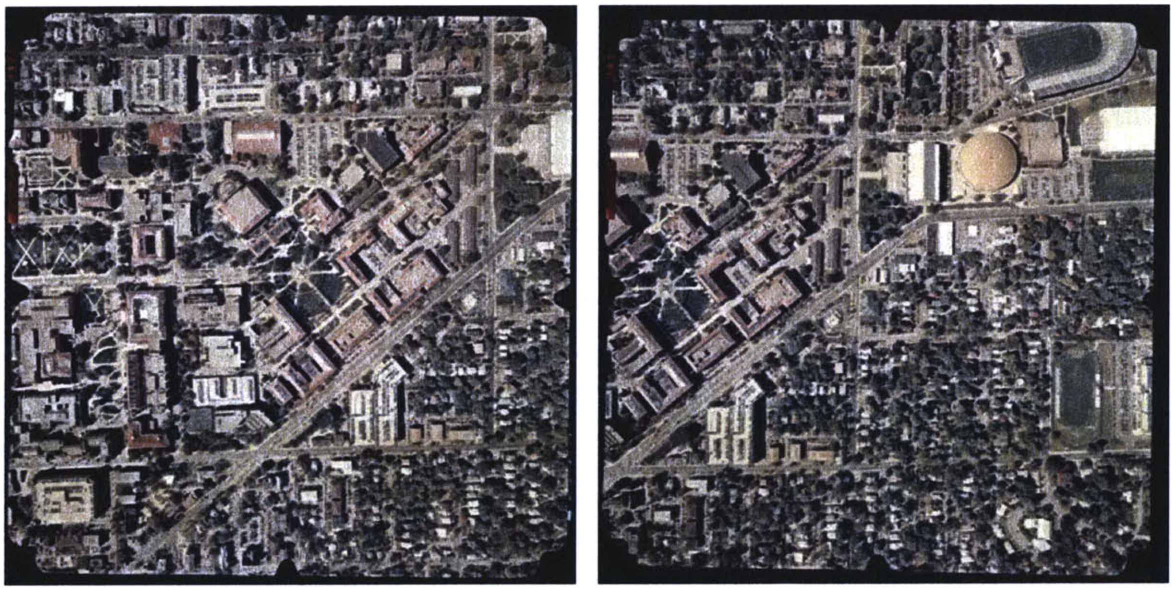


Figure 4-20: Example of how multiple looks at a target can converge to a single location (Taken from [20])

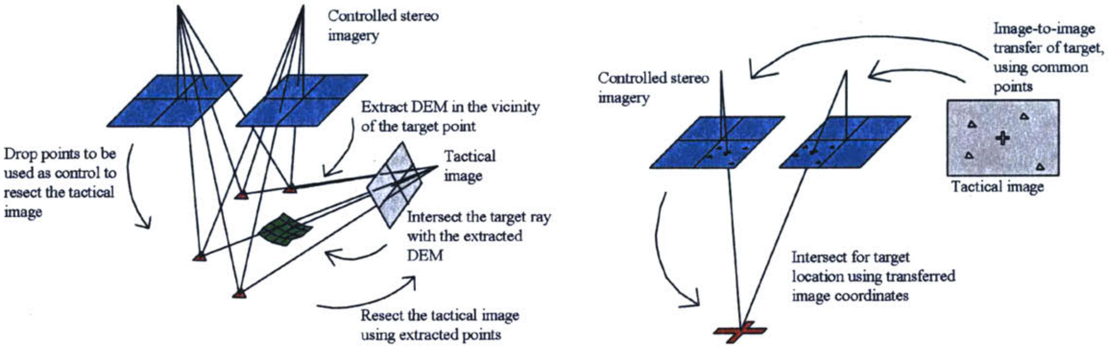
objects are used as the basis for determining the target's position. These objects are referred to as landmarks. Of all the relative to landmark methods possible, four were listed in Figure 4-15 and will be discussed in further detail in the subsequent sections.

The first relative to landmark method is the DPPDB method. DPPDB stands for Digital Point Positioning Data Base. Some of the key components in this method are outlined in Figure 4-21. First, stereoscopic overhead pictures are taken of the area surrounding a target's location, as shown in the first part of the figure. The overhead images are usually taken by reconnaissance satellites and are provided to the system by intelligence agencies. Distinguishable landmarks are then selected from the surrounding area and identified on each of the stereo images. The vehicle then flies into the identified area and its onboard camera supplies what Figure 4-21 refers to as the tactical image. The control points are identified in the tactical image and their location and spacing in the image helps determine the rays from the points to the camera, as shown in the second part of the figure. The target location is then determined through the use of the control points and the intersection of the rays. Figure 4-22 shows an example of a sample high resolution tactical image.

The next two alternatives mentioned in Figure 4-15 are very similar in nature. They are using previously known 2D and 3D landmarks, respectively, to determine the target's location. Both methods rely on having a preexisting database of landmarks surrounding the target. Once the landmarks have been selected and encoded, the system must detect them. Confounding issues related to this task are the scale, rotation, shifting, illumination, and the view of the landmarks. With scale, the landmarks in the database is encoded as a certain number of pixels in size, corresponding to being viewed at a predetermined distance away. However, if in the course of the



(a) Stereo images



(b) Method illustration (c) Further illustration

Figure 4-21: Example of the DPPDB method (Taken from [20])

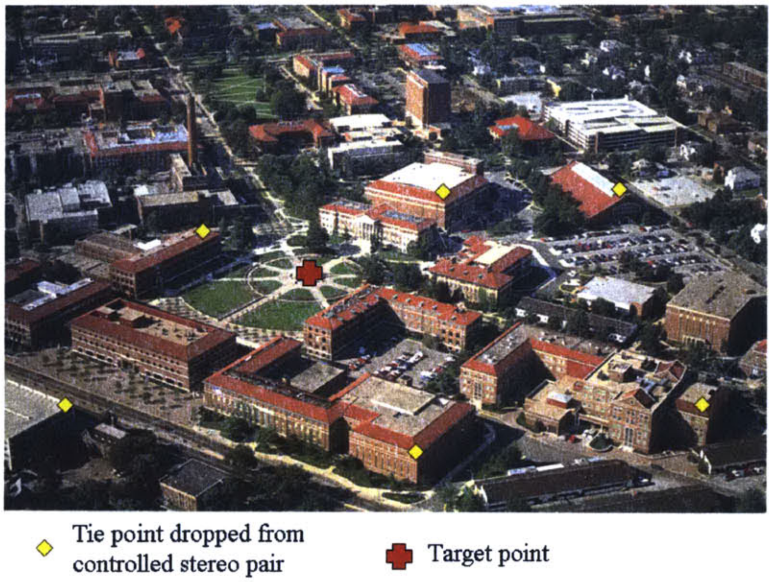


Figure 4-22: Example of a target and some local drop points required for determining the target's location (Taken from [20])

mission, the vehicle is a different distance away, the landmark will be either larger or small than the encoded landmark. Detection still must still occur despite the difference in size. Similarly, a discrepancy may existing in how the landmark is oriented, thereby requiring a rotation or a displacement in order to match the database. The illumination of the landmark might also vary. The database could contain a fully lit representation while the actual object could be partially obscured in darkness. Furthermore, the representation could be during the day while the object is viewed at night. Finally, the representation of the object could be of a top view while the actual view could be non-planar.

In the midst of the mitigating circumstances, an algorithm must still be able to detect a landmark. Once landmarks has been identified, the position of the target relative to the target needs to be determined. This is done by assessing the scale of the image and then estimating the distance and direction to the target from a number of landmarks. The scale of the images can be determined by using the size of the landmarks. If it is known a priori that a landmark has certain dimensions, then the ft/pixel or some equivalent dimension can be determined through a simple division. Then, the number of pixels between the landmark and the target can be counted and the distance between them can be calculated. A variety of different algorithms exist to perform the landmark detection and target location given landmarks. In the 3D case, an area is rendered to produce a 3D model. Then, corners or other easily identifiable landmarks are selected. Fundamentally, however, the concepts and tasks are the same for the different visual detection techniques with known landmarks ahead of time.

The final method listed in Figure 4-15 is SLAM, which stands for Simultaneous

Localization and Mapping. In this method, the system has no a priori knowledge of the landmarks to use or there location. As a result, it determines which landmarks to use and attempts to map them. As more of the world is seen, an increasingly detailed map is created. Error-checking can be performed through revisiting landmarks or filtering the information. SLAM is primarily used in navigation when the system has limited or no a priori knowledge of the environment, making it useful on interplanetary probes such as Martian rovers. In a Martian rover-type case, everything is relative to the landing site. If SLAM were used for small UAV targeting, the system could have some prior landmarks and then also develop new ones throughout the mission. In such cases, known landmarks reduce the error in location of the landmarks.

List Downselection

Even without creating an exhaustive targeting hierarchy in Figure 4-15, the list creation process yielded more targeting solutions than could be evaluated with the given resources. As a result, the list needed to be downselected. As suggested in the methodology section, a number was selected for how many alternatives were desired to be evaluated and the list was whittled down to that number. Along these lines, it was desired to have three alternatives to evaluate.

The downselection process was guided by the criteria specified previously in the methodology section. At an initial analysis, all of the alternatives appeared to be feasible. Money was not a major concern, as all of the alternatives would be simulated first, thereby preventing capital investment in a system prior to knowing how it would function. Of all the criteria listed previously, the two driving considerations were the time available for the analysis and the skillset of the workers.

The time allowed to apply the system design method to small UAV targeting was approximately two months, which includes the time to write up this document, thereby not allowing much time for analysis. Examining the list of alternatives, most of the relative to landmark alternatives have difficult algorithms that need to be written. Because of this, it is hard to accurately predict the accuracy, calculation time, robustness, and other system characteristics prior to implementation. However, the short timeline was prohibitive of even a cursory development of these methods. In addition, the techniques suggested are outside the author's skill set. As a result, the algorithm-based relative to landmark alternatives were eliminated. Similarly, multipass was deemed too time consuming to adequately develop, so it was eliminated. Thus, the three alternatives selected for further consideration were using the vehicle's orientation and the range to target and the two methods associated with using the vehicle's orientation and altitude.

4.5 Define Subcomponents

4.5.1 Methodology

With the list downselected to the alternatives that merit further investigation, the system design proceeds to defining the subcomponents for the remaining alternatives.

As was stated in Section 2.2.5, this step can be seen as an iterative process with the define the alternatives step, thereby allowing for MDO-like decomposition. The iterative decomposition yields a hierarchy of subsystems and components. As discussed in Section 2.2.5, the first iteration produces a list of alternatives for the problem on the system level, with subsequent iterations further decomposing the alternatives into their subsystems and components. Graphically, this process could be represented as a hierarchy where the top level is the alternatives and the leaves are the actual make and model of individual components that could be used for a specific type of method, if the component is a physical object, or the various implementation methods for conceptual objects. Section 2.2.5 finally states that the decomposition should continue until the hierarchy is one level below the level at which the designers want to evaluate.

One of the most common ways to perform the decomposition of the alternatives into subcomponents is through a method similar to the functional decomposition described above in the Define the Alternatives step. Section 4.4.1 described the functional decomposition used for defining the alternatives as the breakdown of the end product into its functional components followed by a listing of ways to achieve these functional goals. This section said that designers essentially work backwards by first figuring out what they need and then listing ways to obtain these needs. Once established, each of the functional needs are treated independently, with complete alternatives coming from selecting a component from each of the functional needs.

The same basic principle applies to the definition of the subcomponents, but it is instead applied to each alternative instead of the overall problem. The alternatives are analyzed to determine what is needed in order for them to work. The designers then list different ways for each item to be provided. For example, if the alternative calculates an answer for a given problem based on properties of the environment or operating scenario, then each of the items needed in the calculation would be analyzed. The list for any given item would then be the ways that the item could be obtained. This would include any sensors that directly measure the property or algorithms that deduce the property from other information. However, the source of the other information would need to be included in the list. For example, if an alternative needs the wind speed, a possible way of obtaining that information is by analyzing the video that is returned for the displacement of the trees and then figuring out the wind speed based upon the deflection. While the software is responsible for the solution, the camera is still a necessary piece of equipment in order to return an answer. As a result, it must be added to the list of subcomponents. Another alternative might be simply to add a anemometer, in which case, the anemometer would be the subsystem and different company's anemometer's would serve as the component choices.

If the designers get stuck during the functional decomposition, another way that they could approach the decomposition is using an And/Or type of approach. In this method, the designers list all of the components that comprise a system. The components are treated as "Ands" while the component options are treated as "Ors". All of the Ands must be selected while only one Or per component must be chosen. If the And/Or process is thought of as a hierarchy, for a full-depth decomposition, all of the leaves of the tree are Ors, but Ors are not necessarily exclusively leaves.

This process is best illustrated through an example, such as the decomposition of a computer. A partial decomposition is shown in Figure 4-23. The decomposition looks at the RAM and hard drive options for the computer. Both are necessary components for a computer, so the first level of the hierarchy is an And connection. On the next level, the two choices for RAM are 184-DDR SDRAM and 240-Pin DDR2 SDRAM, while the two choices for hard drives are IDE and SATA. These nodes form different classes of components; an IDE hard drive is a unique product from a SATA hard drive and a stick of 184-DDR SDRAM is different than a stick of 240-PIN DDR2 SDRAM. As a result, the designers can choose only one of the two categories, making it an Or connection. Finally, the leaves of the tree are all the different RAM and hard drive models that the designers could purchase. These are the actual products that would go into the computer being built. As a result, only one selection can be made, making it an Or connection. A complete system would therefore select one of the six options for RAM and one of the six options for hard drives.

Admittedly, the example shown in Figure 4-23 is a simplistic example, with only one level of And connections. If this was a more complex system, then there would be multiple And levels. This is the case for a system that has complex subsystems. For example, an aircraft would have subsystems that include, amongst others, propulsion, targeting, and controls. The targeting subsystem could then be comprised of a camera, radar, laser rangefinder, and a weapons system. A full decomposition of a aircraft would yield multiple levels of And and Or connections.

Finally, it should be noted that if the designers are adding on to an existing system, they can use the existing equipment and only list new alternatives for the components that need to be added to the system. In doing so, the designers can concretely compare how the existing system is affected and whether the changes are merited. The existing system thus becomes a baseline for the analysis and serves as the basis for comparison for the other alternatives.

4.5.2 Application

The methodology section states that the define the subcomponents section can be seen as an iterative process with the define the alternatives step and that the decomposition should continue to be one level below the level where the analysis takes place. Since the purpose of this design is to evaluate targeting systems at the alternative level (that is, which targeting method works best), the decomposition should proceed to the component level. Thus, if the decomposition is thought of as a hierarchial tree, as was described in the methodology section, the leaves would contain the make and models of the different components to be selected.

The method used in defining the subcomponents was the functional decomposition described above. The methodology section stated that the functional decomposition involved analyzing the alternatives to determine what is needed in order for them to work and then listing different ways for each item to be provided. This analysis will be discussed for each of the three targeting alternatives. Before that is done, however, the commonalities between the methods will be established.

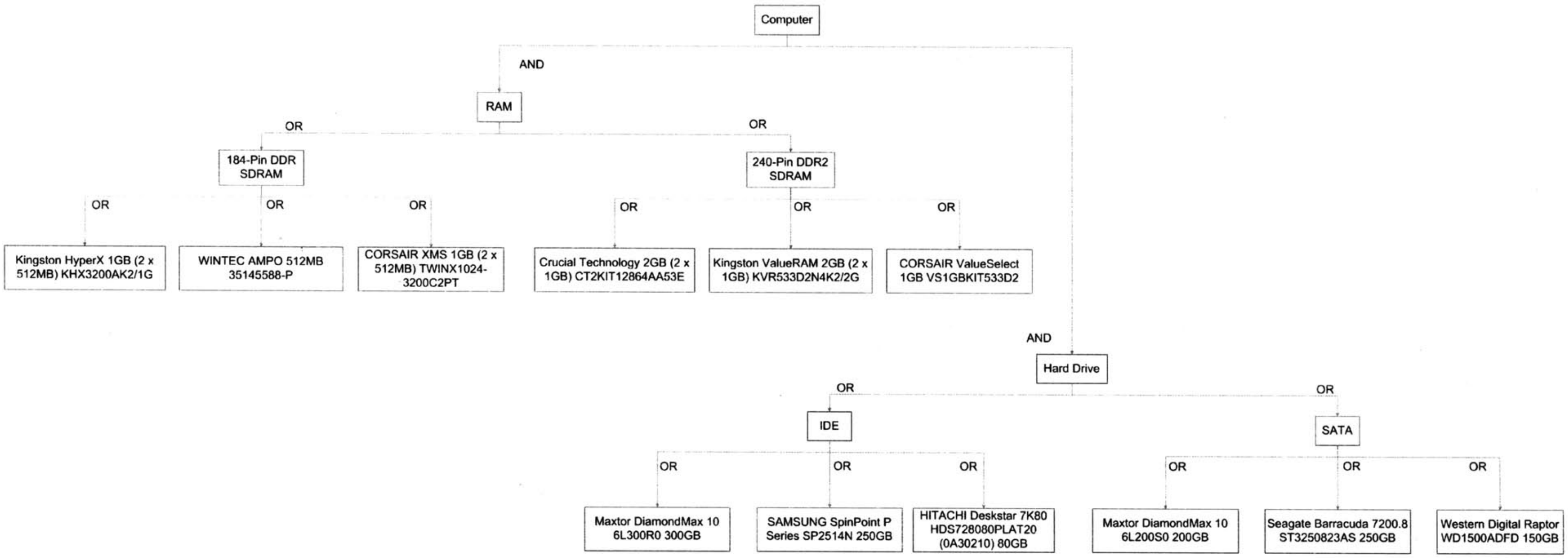


Figure 4-23: Partial And/Or decomposition of a computer.

First Iteration - What is Needed?

As was determined in the previous step of the design process, the three methods selected for further evaluation were:

- using the vehicle's position, orientation, and height, along with the assumption of a flat Earth (referred to from here on as the Flat Earth method)
- using the vehicle's position, orientation, and height while also using DTED data (referred to from here on as the DTED method) and
- using the vehicle's position, orientation, and range (referred to from here on as the Range method).

When put in this format, the commonalities are more easily seen. Since all of the methods selected for future investigation were from the relative to self class of solutions described in Section 4.4.2 and shown in Figure 4-15, all of the solutions require knowing the absolute position of the vehicle. This serves as the anchoring point for determining the target's location. The other commonality to all of the solutions is the need to know the vehicle's orientation. As was stated in Section 4.4.2, the real item of interest is determining where the camera is pointed. However, since it was assumed that the camera is fixed in place and points in the same direction as the vehicle's heading, the vehicle's heading can be thought of as the camera's heading. Then, the camera's pitch can easily be determined through the roll and pitch of the vehicle. Thus, after the calculations are performed, the two common elements between the three methods are the position of the vehicle/camera and the LOS vector for the vehicle/camera.

In addition to the vehicle's position and the LOS vector for the camera, the Flat Earth and DTED method also need to know the vehicle's altitude. As was described in Section 4.4.2, the height serves as the vertical leg of the right triangle shown in Figure 4-17 that is used to determined the target's position. In addition, the DTED method needs to possess the DTED data of the area around the target. By contrast, the Range method uses the hypotenuse of Figure 4-17's triangle, thereby requiring that the range be determined instead of the height.

Second Iteration - Defining Component Alternatives

With the required quantities established for each of the methods, the second iteration of the decomposition delves into how to obtain this data by conducting component level analysis. As before, the discussion will first start with the components that are common to all of the methods and then will proceed to those that are specific to individual methods.

The first component that is common to all the targeting methods of interest is a camera. The targeting process proposed inherently requires the user to select the target, as no automatic detection and acquisition is performed. In other words, a human must be in the loop for any encounter. The way that a human is kept in the loop is by providing the users with a video stream of the vehicle's surroundings. In this way, a camera is so fundamental to the current small UAV systems that every targeting method, not just the three selected, would use one.

As stated above, all three method need to know the vehicle's position and orientation. As was shown in Figure 2-3, these quantities could be determined in a number of different ways. Currently, the most widely accepted method to determine position and orientation for small UAVs is to use a GPS and an IMU, respectively. However, instead of using the two in isolation of each other, it would make more sense to integrate the two so that the GPS information is used to bound the error on the IMU. The two components could be integrated in the form of a COTS autopilot. COTS autopilots provide navigation solutions by typically combining a GPS and IMU in an intelligent way, such as a Kalman filter. Thus, the orientation error is not allowed to grow indiscriminately, but is instead bounded to a manageable error, such as ±10° at greatest. Given the relative simplicity, the high fidelity nature of the solutions, and their low weight, COTS autopilots were selected as the way to determine the position and orientation of the vehicle.

Next, the first iteration established that the Flat Earth and DTED methods required knowing the vehicle's height. While this can be done in a variety of ways, a aircraft's altitude is traditionally found using an altimeter. Different types of altimeters exist. Radar altimeters actively transmit waves towards the ground and calculate height based upon the amount of time to receive a response and the speed of the wave. The same premise is used with laser altimeters.

Another class of altimeters is pressure-based altimeters. Pressure-based altimeters are rooted in the standard atmosphere model created in 1976. The equations used in the standard atmosphere model were adopted on October 15, 1976 by the United States Committee on Extension to the Standard Atmosphere (COESA), representing 29 U.S. scientific and engineering organizations [5]. One of the many capabilities of the standard atmosphere model is the ability to determine the altitude of an object by measuring the pressure. As a result, the static pressure of the air at the vehicle's altitude is measured, typically by a pitot tube, and then the vehicle's altitude is calculated using the equation:

$$h_{alt} = (1 - (\frac{p_{sta}}{1013.25}^{0.190284})) \tag{4.1}$$

where h_{alt} is the pressure altitude, measured in feet, and p_{sta} is the station pressure measured in millibars or hectaPascals ([4]). The station pressure is just the pressure measured by the pressure sensor. Altitude could also be determined algorithmically by using the scaling of pixels or some other calculation method. As with the relative to landmark class of solutions, the fidelity of these algorithms are difficult to ascertain without implementing them. Additionally, the algorithms are outside the scope of the author's knowledge. As a result, only the two types of altimeters were selected for further investigation.

As state above, the Range method also needs to know the range to the target. The traditional way that this is done is with a laser rangefinder. The laser rangefinder emits a laser pulse at a certain frequency and then detects how long it takes to receive a return signal. The distance to the target can then be calculated based upon the known speed of the laser pulse emitted and the time elapsed. This is the same principle as was used in radar and laser altimeters. As before, algorithmically-based methods exist to determine range. For instance, if the size of a landmark is known a priori and then is detected in flight, the number of pixels that cover the landmark can be used to estimate the scale of each pixel. This information can then be used to estimate the range. However, as was the case before, the more tradition option was selected for further investigation.

Third Iteration - Finding Specific Components

The final iteration in the decomposition process focused on finding specific makes and models of the components deemed worth of further investigation in the previous iteration. The analysis provided at least two options for each component. The small number of options was a byproduct of the condensed timeline of the design. If time had allowed, five or more options for each product would have been desired. However, the process would not change when including the additional components. The only difference would be that the design space would be explored with greater resolution. The specific instances of each type of component will be mentioned below. It should also be noted that the specification sheets produced by the company for their product are included in Appendix A.

Two cameras were selected for consideration. They were the Rockwell Scientific UAV-Cam V2M and the Sony FCB-EX780B. The properties of these cameras are found in Table 4.4. The Rockwell Scientific camera is specifically design for UAV applications and as a result, has a combination of low power consumption, light weight, and high resolution. However, the performance comes at the high price of $9,500.00. By comparison, the Sony camera has a lower resolution and a significantly higher power consumption, but costs over an order of magnitude less than the Rockwell Scientific camera.

Three autopilot options were found. They were the Procerus Kestrel, Crossbow NAV420, and the Athena Controls GS-111m. The properties for these products are contained in Table 4.5. The Procerus is significantly less expensive than the other two products, but has a much higher error as well. The Crossbow and Athena Controls have comparable performance characteristics. The main difference between the two products is that the Athena is significantly lighter. As a result, the GS-111m is close to twice the cost of the NAV420.

The next product type that was examined was the altimeters. Two different altimeters were included, those being the Roke Manor Miniature Radar Altimeter (MRA) Mk IVa and the Honeywell HPA. The Roke Manor product is an example of a radar altimeter and the Honeywell product is an example of a pressure-based altimeter. The Mk IVa is bigger and heavier than the HPA, and also requires more power. However, it is more accurate and less than half the cost. It should be noted that the HPA size and weight do not include a pitot tube, since it was not clear whether or not the aircraft would already have one or not. If a pitot tube was included in the estimate, the weights would probably closer in value. It should also be noted that a third product was found in searching, but was excluded based upon a lack of believability of the specs sheet. The product was ZLog MOD3 Recording Altimeter. The MOD3 claimed to have a resolution of "1 ft/m." It was unclear how it could have a resolution of both 1 ft and 1 m, but an educated guess was that the figure was quoted because the product had a digital display that could be set to ft or m. Therefore, the product could claim to have a 1 ft/m resolution. The MOD3 is $40mm$ x $23.4mm$ x $9.4mm$ and has a weight of $8g$. It is primarily used for hobby RC aircraft and has a cost of around \$80.00. Since it is significantly smaller, lighter, and less expensive then the other products yet with the same claimed order of accuracy, this product was deemed either perfect or too good to be true. Given the quality of the company, it's website, the lack of notoriety of the product, and the fact that it is primarily used for hobby RC equipment, the latter conclusion was reached.

The final analysis was conducted on laser rangefinders. Two comparable products were found. They were the Vectronix LRF42 and the Thales Miniature Eyesafe Laser Rangefinder (MELT). The two products are comparable in most pertinent properties other than the accuracy, where the LRF42 is about 10 ft more accurate.

4.6 Establish Relationships Amongst Properties

4.6.1 Methodology

With the alternatives fully defined to the component level and the properties of the components listed in Tables 4.4 - 4.7, the design process moves on to the phase of establishing the relationships amongst the properties. The properties of interest are those that go into the evaluation of the alternative. Since the evaluation is tied to the cost function, which is derived from the design criteria, the properties needed to perform the evaluation are those that directly contribute to the design criteria. Thus, the system-level properties that contribute to the design criteria in some way need to determined.

As stated in Section 2.2.6, the properties are usually determined in one of two ways. If a property is physics-based or involves calculation, then the property is determined by deriving the governing equations. If no governing equations exist for the property, then some sort of measuring metric, simulation, or prototype must be developed. These two methods will be discussed in further detail in the subsequent sections.

Governing Equations

The first class of methods for determining the properties is to define the governing equations. As stated above, governing equations are typically defined when the property is physics-based or involves calculations of some sort. The equations that govern various properties are as varied as the world itself. There is no minimum or maximum complexity. The equation could not involve physics at all but instead require just adding two numbers. It does not have to be rigorous at all. For example, for a given component, the equation for the unit cost might simply be looking up a number

Table 4.4: Camera models selected with their associated properties.

Manufacturer	**Model**	**Weight**	**Size**	**Power**	**Price**	**Resolution**
Sony	FCB-EX780B	230 g	50 mm x 57.5 mm x 88.5 mm	2.7 W	$700.00	680,000 pix
Rockwell Scientific	UAV-Cam V2M	220 g	37mm x 31 mm x 25 mm	0.500 W	$9,500.00	2,110,240 pix

Table 4.5: Autopilot models selected with their associated properties.

Manufacturer	**Model**	**Weight**	**Size**	**Power**	**Price**	**Pitch/Roll Error**	**Yaw Error**	**Horizontal Pos Error**
Procerus	Kestrel	16.7 g	52.7 mm x 34.9 mm x 11.9 mm	1.65 W	$5,000.00	5°	8°	45 ft
Crossbow	NAV420	589.7 g	76.2 mm x 95.3 mm x 76.2 mm	4.2 W	$7,995.00	0.75°	3°	9 ft
Athena Controls	GS-111m	226.8 g	99.1 mm x 66.0 mm x 40.6 mm	4.5 W	$14,500.00	0.75°	1°	9 ft

Table 4.6: Altimeter models selected with their associated properties.

Manufacturer	**Model**	**Weight**	**Size**	**Power**	**Price**	**Altitude Error**
Roke Manor	MRA Mk IVa	400 g	100 mm x 75 mm x 70 mm	8.4 W	$400.00	0.410 ft
Honeywell	HPA	142 g	45.7 mm x 55.9 mm x 24.8 mm	0.165 W	$885.00	1.092 ft

Table 4.7: Laser Rangefinder models selected with their associated properties.

Manufacturer	**Model**	**Weight**	**Size**	**Power**	**Price**	**Range Error**
Vectronix	LRF42	350 g	110 mm x 100 mm x 50 mm	3.575 W	$11,250.00	6.56 ft
Thales	MELT	360 g	100 mm x 75 mm x 50 mm	3.0 W	$10,000.00	16.4 ft

out of a catalog and multiplying by the quantity. In the limit of simplicity, there exist cases such that no calculation would be required because the property would simply be a single number, such as the unit cost or weight of an individual component. On the opposite extreme, the equations could be quite complex, requiring deep understanding of the natural phenomena at work or the interaction of components. An example of this are the Navier-Stokes equations in fluid mechanics.

If the designers cannot define the governing equations themselves, then they have a variety of options at their disposal. The first is web-based search through online search engines such as Google, Yahoo, MSN, or any other of that genre. The designers could also attempt to look in textbooks on the topic, if they exist. Journal articles are another important avenue to try for information. Also, the designers could ask an expert on the topic, or at least someone who has had experience in the area that might be able to help.

Other Methods

Properties that are not governed by equations fall into a broad category, thereby making it hard to have universally applicable generalizations about them. However, some general strategies will be discussed, leaving it up to designers to decide which strategy would work best for the given property.

If the property involves programming, the designer could write a section of the code and then try to extrapolate the necessary properties from the segment, such as run time, development time, or lines of code. They could also examine previous projects that were similar in topic or scope. Also, the internet contains information about average properties of various programming languages, including how many lines of code the average programmer can code in a month, how many lines of code it takes to accomplish a certain task, statistics for how commonly used the language is, and many other properties. This general information may help the designers estimate a property.

Another valuable tool the designers have available to them is simulation. Simulation could very well be included in the governing equations section as well, as it is a broad category. Some simulations involve modeling the "real world," requiring rigorous development of the governing equations. Other simulations involve running a program or algorithm with sample input. Either way, simulation aims at modeling the true behavior of a system. If the simulation is of high enough fidelity, the output from the simulation can be treated as the behavior of the actual system. The designers can then use this behavior to measure or predict what they need to know about the system.

Prototyping is another option. Prototyping usually applies to systems that are physical objects instead of conceptual ones such as software. Many different types of prototyping exist, but they usually involve partial or complete fabrication of the object. Prototyping is usually used on relatively simple products or done in later phases for more complex products. One example of prototyping is when the design team builds an actual working version of the system. This type of prototyping is usually done if the product is going to be mass produced or to prove that a concept will work. When a copy of the actual system is built, the property could potentially be directly measured. In the case of mass production, the designers can see the creation process and decide what needs to be changed. Another type of prototyping is when a reduced-scale model is constructed. This is typically done for systems of considerable size. Creating a reduced-scale model allows the designers to measure properties that can scale up while saving on the materials and labor required to build a full-sized version of the system. The final type of prototyping that will be discussed is rapid prototyping on a 3D printing machine. In this method, the rapid prototyping machine lays down layers of plastic in the shape of the product. This allows for the quick fabrication of parts and allows designers to see the process before it is done with actual materials.

A final method is to implement the system or component and directly measure whatever property needs to be measured. Here implement could refer to building, buying, coding, obtaining, or somehow coming across the component or system needed. If the property is not something that can be directly measured, then the system can be run in a life-like environment so that the designers can get the information that they need.

Tips

As was mentioned in Section 2.2.6, designers must trade the fidelity of how well the properties are known with the amount of time that is required to develop the method of determination. The fidelity required is often affected by the importance of the property. The most important property might merit a rigorous development, as they have a heavy influence on the design, while the ancillary property might just be roughly estimated.

Also, if the equations are such they cannot be calculated easily by hand or if the designers prefer to automate the process, many computer software packages exist to aid their efforts. Matlab, Simulink, and all Matlab-related toolboxes are quite useful in creating simulations or automating calculation. Texas Instruments' LabView is another resource, proving especially useful for data acquisition tasks. For more mathematically intensive problems, Mathematica, Maple, MathCad, and other advanced mathematical software packages exist. Also, any programming language has mathematical capabilities built into the language.

If the designers are going to use a simulation or automate calculations, they should try to make these programs compatible with the optimization capabilities that will be done in the final step. To this end, scripting languages are good at gluing a variety of different activities together in a recipe-like fashion. Languages like Python and shell scripting in DOS or a Unix/Linux-based environment provide the most flexibility, with Matlab offering scripting capability within its domain. More will be discussed about the optimization options during the appropriate step.

4.6.2 Application

As was stated in the methodology section, the system-level properties that contribute to the design criteria in some way need to determined. It is thus natural to analyze the design criteria in succession and determine the component properties and governing equations that need to be determined in order to evaluate the method.

Weight

As established in Section 4.3.2, the mass/weight bound is dependent on the total weight of the targeting system. This is simply a sum of the weights of the individual components, expressed mathematically as:

$$w_{sys} = \sum_{i=1}^{n} w_{comp_i} \tag{4.2}$$

where w_{sys} is the total system weight, w_{comp_i} is the weight of the i^{th} component in the system, and n is the number of components in the system. Thus, the expression is nothing more than a sum of each component's contribution to the total.

Size

The bound section for the size established that the property of interest for this criteria was volume. Expanding this notion further, the total system size was said to be the summation of the volume of each individual component. The volume of each individual component was estimated by determining the volume of the smallest 3-dimensional box that could house the object. As a result, each component's volume, v_{comp}, is

$$v_{comp} = l * w * h \tag{4.3}$$

where l is the component's length, w is the component's width, and h is the component's height. The total system volume, v_{sys} is thus

$$v_{sys} = \sum_{i=1}^{n} v_{comp_i} \tag{4.4}$$

where n is the number of components in the system.

Volume was not the only choice that could have been used for quantifying the size of the system. Other metrics could have included tracking the largest dimension of the object or just simply listing the dimensions. However, volume was chosen because it was felt that volume was the quantity that best represented the desired information.

Power

As with the size, multiple options existed for how best to represent the power criteria for the system. The option that was selected was to calculate the individual power consumption for each component and then summing up the individual contributions to form the total. The equation for the power consumption required for an individual component, p_{comp} is

$$P_{comp} = VI \tag{4.5}$$

where V is the supply voltage required for the component to function and I is the current drawn by the component during operation. The total power consumption for the entire system, P_{sys}, is then

$$P_{sys} = \sum_{i=1}^{n} P_{comp_i} \tag{4.6}$$

where P_{comp_i} is the power consumption of the i^{th} component of the system and n is the number of components in the system.

Another option that existed for power was to track the highest amount of voltage required for any individual component. This quantity will be called V_{max}. The current drawn for operation of each component, I_{comp}, is also recorded. The equation for P_{sys} then becomes

$$P_{sys} = \sum_{i=1}^{n} V_{max} I_{comp_i} \tag{4.7}$$

where n is the number of components in the system.

This method was not used for a couple of reasons. First, it is not necessarily accurate. Some components can operate on a range of voltages and at higher voltages, the current drawn is decreased, making the latter method too high of an estimate. Also, not all of specification sheets give the voltage and current information, but nearly all quote a figure for the component's power consumption. As a result, it is easier to ascertain the power consumption information. Coupled with the fact that using the component power consumption is more accurate than tracking the V_{max} and current draw, the total system power consumption was determined using the summation of the components' power consumption.

Unit Cost

The unit cost continues the trend of simply being the summation of the individual component's contribution to the total. The equation for this is

$$c_{sys} = \sum_{i=1}^{n} c_{comp_i} \tag{4.8}$$

where c_{sys} is the total system unit cost, c_{comp_i} is the unit cost of the i^{th} component, and n is the number of components in the system.

Accuracy

As was stated previously in the thesis, the estimate of the accuracy of the system is the TLE. Since the system's TLE is a physics-based quantity that is method-dependent, the equations governing how to calculate the target's position for each of the three methods need to be derived. After these equations have been derived, the discussion will turn to how the TLE is found.

For the equations in this section, the vehicle's position and height will combine to form an ordered triple such that the planes position in 3D space will be described as (x, y, h). Also, the camera's heading angle will be referred to as ψ and the camera's pitch, measured from vertical as shown in Figure 4-24 will be referred to as θ. The range to the target will be r and d_g will be the camera's LOS vector projected onto the plane of the Earth. The variable d_g was selected because this quantity is the distance along the ground, or the ground distance, from the xy position of the vehicle to the xy position of the target, as shown in Figure 4-24. The desired quantity from each of the methods is the target's position, which will be referred to as (x_t, y_t, z_t). This terminology will be used for all of the methods.

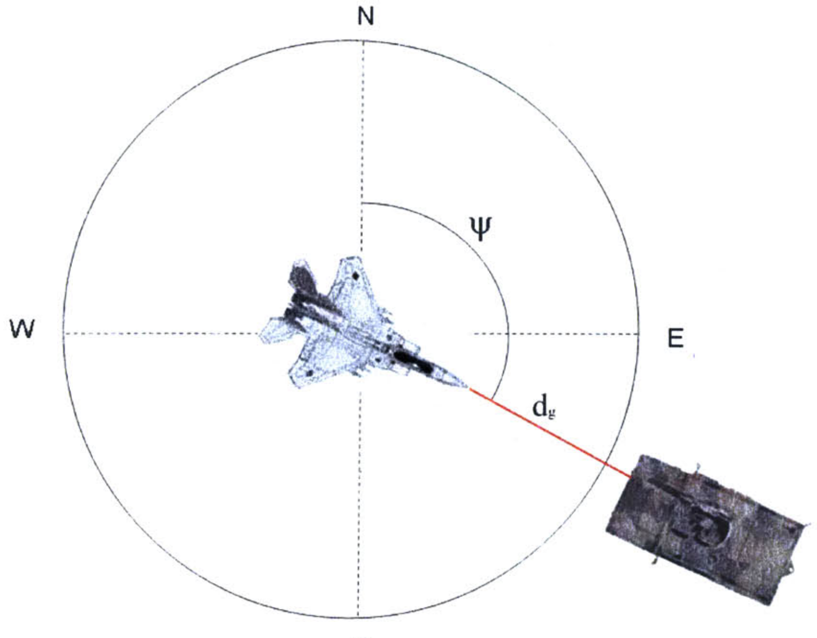
Flat Earth Equations The discussion will first turn to how the target's position is determined for the Flat Earth method. Figure 4-24 reviews the geometry that was previously demonstrated in Figures 4-16 and 4-17, but relabels the sides of the triangle to correspond with the terminology that will be used in deriving the equations. It should be noted that the LOS vector labeled d_g in the XY view and the horizontal leg labeled d_g in the side view are the same length.

As was stated in the Define the Alternatives step, the camera's LOS vector defines the line in the xy plane on which the target's xy coordinates could lie. This is clearly shown in the top part of Figure 4-24. Based on the geometry shown in the XY view in the figure, the equations for x_t and y_t are

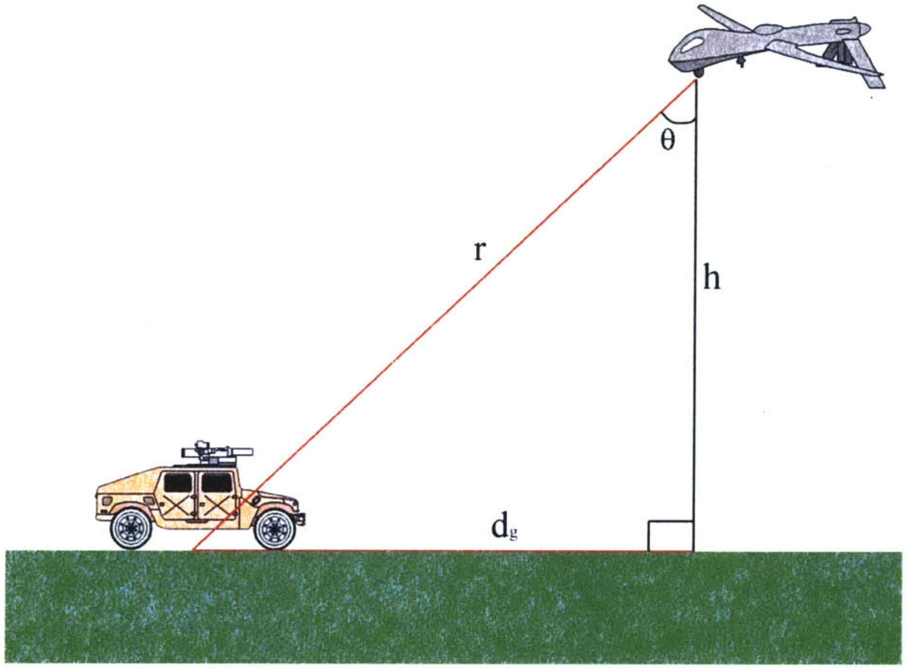
$$\begin{aligned} x_t &= x + d_g \sin\psi \\ y_t &= y + d_g \cos\psi \end{aligned} \tag{4.9}$$

Since x and y are part of the known quantities, the only unknown in Equation 4.9 is d_g. The side view of Figure 4-24 will be used to aid in this calculation. As was stated previously, in addition to the vehicle's xy position and the camera orientation, the Flat Earth method also requires the knowledge of the vehicle's altitude, or h. As a result, the two known quantities for the Flat Earth method in side view of Figure 4-24 are θ and h. Since the triangle is a right triangle and both an angle and side are known, the rest of the quantities can be calculated. The most direct way to calculate d_g from the known information is using the equation

$$d_g = h \tan\theta \tag{4.10}$$



(a) XY View



(b) Side View

Figure 4-24: The geometry of a Flat Earth targeting encounter revisited, with the appropriate dimensions labeled with the corresponding terms to match the equations in the text.

A slightly less direct way would be to calculate the range first by

$$r = \frac{h}{\cos\theta} \tag{4.11}$$

and then finding d_g with

$$d_g = \sqrt{r^2 - h^2} \tag{4.12}$$

The advantage of using Equation 4.12 will be clear later. Substituting Equation 4.12 into Equation 4.9 yields

$$\begin{aligned} x_t &= x + \sqrt{r^2 - h^2}\sin\psi \\ y_t &= y + \sqrt{r^2 - h^2}\cos\psi \end{aligned} \tag{4.13}$$

Equation 4.11 can then be substituted in for r in the above equation and then, after simplification, the equation becomes

$$\begin{aligned} x_t &= x + \sqrt{h^2(\frac{1}{\cos^2\theta} - 1)}\sin\psi \\ y_t &= y + \sqrt{h^2(\frac{1}{\cos^2\theta} - 1)}\cos\psi \end{aligned} \tag{4.14}$$

which is solely in terms of the known quantities. Thus, x_t and y_t have been determined. The only remaining quantity left to determine is z_t. The fundamental assumption of the Flat Earth method is that the Earth is flat and thus no elevation change has occurred from the vehicle's elevation at takeoff. Thus, z_t is equal to the elevation of the takeoff site of the vehicle.

Range Method Equations Instead of considering the DTED method next as has been done in previous sections, the discussion will turn to the Range method. For the Range method, Figure 4-24 still serves as an illustration of the pertinent geometry. In fact, nothing changes at all in the top view. As a result, Equation 4.9 still applies. As before, x, y, and ψ are known in the Range method, so the only unknown is d_g. Also, Equation 4.13 can be ultimately be used again to calculate x_t and y_t, but first an expression must be developed for h. Repeating the same logic from the previous section, the known information contains one side and one angle in a right triangle, which is enough to determine the rest of triangle's information. This time, r and θ are used to find h according to the equation

$$h = r\cos\theta \tag{4.15}$$

With h determined, Equation 4.13 can be used again to find x_t and y_t. Thus, the advantage of using Equation 4.12 was that it yield a generic equation, Equation 4.13, that could be used by both the Flat Earth and Range methods to determine x_t and y_t.

Unlike the Flat Earth method, though, the Range method does not just assume

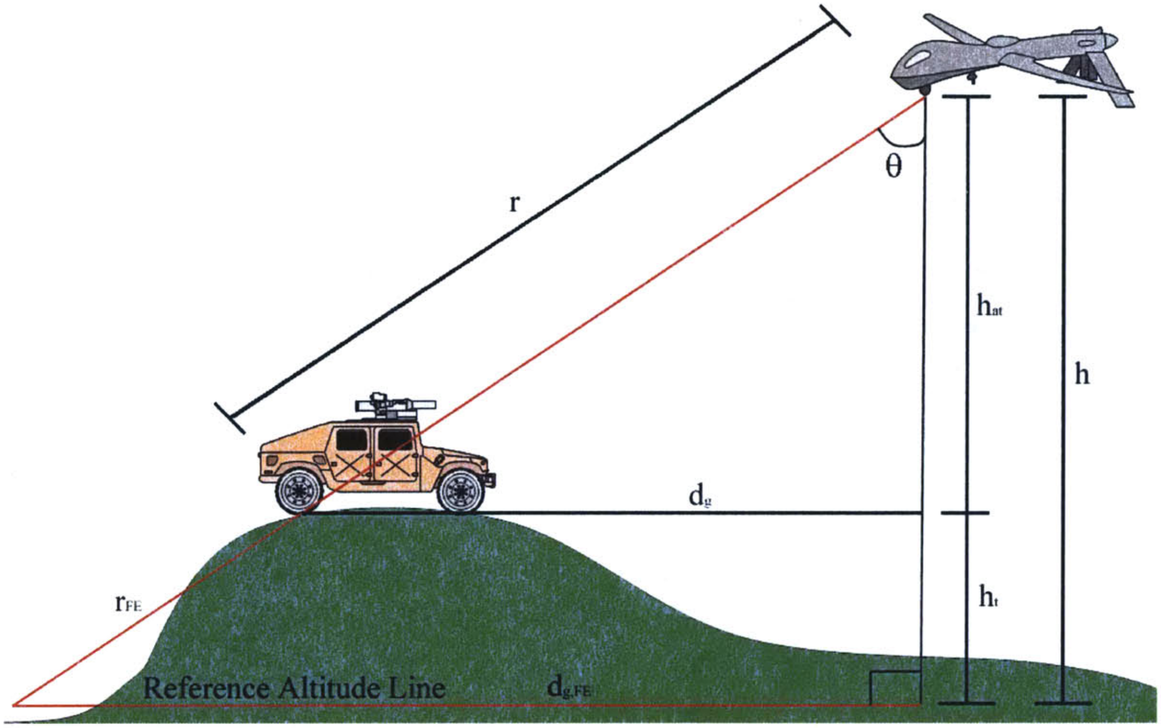


Figure 4-25: Example of an encounter where the target's altitude is above the vehicle's reference altitude.

that the target is at the reference altitude. Instead, it calculates the target's height based upon the vehicle's height and the range to the target. This is perhaps best seen through using Figure 4-25.

In this picture, the target is above the reference altitude by an amount, h_t. As the figure shows, r is not the hypotenuse of a triangle that ends at the reference altitude, but is instead the hypotenuse of a triangle that ends at the target's altitude. As a result, the vertical leg of the right triangle defined by r and θ has a length of h_{at} (height above target), not h. The combination of h_{at} and h_t is the vehicle's height above the reference line, which is h. Thus, the h in Equation 4.15 is not really h, but it is h_at. Thus, the z component of the target can be found by

$$z_t = h_t = h - h_{at} \tag{4.16}$$

where h_{at} is found by

$$h_{at} = r\cos\theta \tag{4.17}$$

yielding a combined equation of

$$z_t = h_t = h - r\cos\theta \tag{4.18}$$

To stay consistent with the notation, the equation for x_t and y_t becomes

$$\begin{aligned} x_t &= x + \sqrt{r^2 - h_{at}^2}\sin\psi \\ y_t &= y + \sqrt{r^2 - h_{at}^2}\cos\psi \end{aligned} \tag{4.19}$$

In terms of only the known quantities, the equation is

$$\begin{aligned} x_t &= x + \sqrt{r^2 - (r\cos\theta)^2}\sin\psi \\ y_t &= y + \sqrt{r^2 - (r\cos\theta)^2}\cos\psi \end{aligned} \tag{4.20}$$

Squaring the terms inside the parenthesis and then factoring out the r^2 yields

$$\begin{aligned} x_t &= x + \sqrt{r^2(1 - \cos^2\theta)}\sin\psi \\ y_t &= y + \sqrt{r^2(1 - \cos^2\theta)}\cos\psi \end{aligned} \tag{4.21}$$

Using the trigonometric identity of

$$\cos^2\theta + \sin^2\theta = 1 \Rightarrow \sin^2\theta = 1 - \cos^2\theta \tag{4.22}$$

with Equation 4.21 yields

$$\begin{aligned} x_t &= x + \sqrt{r^2\sin^2\theta}\sin\psi \\ y_t &= y + \sqrt{r^2\sin^2\theta}\cos\psi \end{aligned} \tag{4.23}$$

Taking the square root give the final equation of

$$\begin{aligned} x_t &= x + r\sin\theta\sin\psi \\ y_t &= y + r\sin\theta\cos\psi \end{aligned} \tag{4.24}$$

Figure 4-25 shows that $r\sin\theta$ is just the horizontal leg of the triangle, d_g. As a result, Equation 4.24 is compatible with Equation 4.9, thereby verifying that the methods are indeed in agreement. This also shows that d_g is in fact the LOS vector projected into the plane of the ground.

DTED Method Equations The top view of Figure 4-24 and side view of Figure 4-25 will again serve to demonstrate the geometry of an encounter. As mentioned in the Define the Alternatives section, for the DTED method, the camera LOS vector can be extended from the vehicle until it intersects the terrain. Assuming the target is centered on the camera, the point of intersection is the location of the target. The equations defined above provide the definition for the camera LOS vector. Since the known information is the same as the Flat Earth method, the equations can be taken directly from the Flat Earth development. In the xy plane, Equation 4.9 can be used. Then, from the side view, the direction of the vector is equivalent to the slope of the line, which is $\frac{rise}{run} = \frac{-h}{d_g} = \frac{-1}{\tan\theta}$. Thus, the equation for any point, (x_p, y_p, z_p) is defined as

$$\begin{aligned} x_p &= x + d_g \sin\psi \\ y_p &= y + d_g \cos\psi \\ z_p &= h - \frac{d_g}{\tan\theta} \end{aligned} \tag{4.25}$$

These are all essentially in the form of generic equation of a line, $y = mx + b$. In this case, d_g is the independent variable. This method assumes that the users have the equation for the ground so that Equation 4.25 can be used to find the intersection of with the ground. This would be done by setting the x, y, and z equations of the surface of the Earth generated from the DTED information equal to Equation 4.25 and then solving for d_g. Once d_g is known, Equation 4.25 can be used to find x_p, y_p, and z_p. Assuming that the camera is centered on the target, this point is then the target's location.

Equation 4.25 can also be used to illustrate the difference between the Flat Earth method and the range method. The only difference is in how d_g is calculated. In the Flat Earth method, one way of calculating d_g was through the use of Equation 4.10, which said $d_g = h\tan\theta$. Substituting that in for d_g yields, after simplification

$$\begin{aligned} x_p &= x + h\tan\theta\sin\psi \\ y_p &= y + h\tan\theta\cos\psi \\ z_p &= 0 \end{aligned} \tag{4.26}$$

In this case, the z coordinate of the target is always equal to the reference height, assumed here to be 0, and the x- and y-coordinates are all in terms of x, y, θ, ψ, and h. In the Range method, $d_g = r\cos\theta$, making the equations, after simplification

$$\begin{aligned} x_p &= x + r\cos\theta\sin\psi \\ y_p &= y + r\cos\theta\cos\psi \\ z_p &= h - r\cos\theta \end{aligned} \tag{4.27}$$

where $r\cos\theta$ was shown in Equation 4.17 to be h_{at}. In this case, r is added to the known quantities. Equations 4.26, 4.25, and 4.27 can serve as the final equations for determining the target position for the Flat Earth, DTED, and Range methods, respectively.

Determining TLE The equations developed above for each method assume that the system is given perfect information. That is, they assume that no error exists in the known quantities. In actuality, however, this is not the case. The information that is required for the above equations come from onboard sensors, all of which has some amount of error. Since the sensors have error, the actual measurements of the desired quantities are

$$\hat{m} = m + e \tag{4.28}$$

where $\hat{m}$ is the estimate of some generic measurement returned by a sensor, m is the actual quantity of the measurement, and e is the amount of error present in the measurement (for example, $\hat{h} = h + e$. Thus, from the perspective of the system, it only knows $\hat{m}$, not m. Because of this, the target position returned by the system is not perfect, but is instead is an estimate given the inexact data that it was given. The system's estimate of the target's position calculated using the sensor data is denoted $(\hat{x}_t, \hat{y}_t, \hat{z}_t)$. The magnitude of the TLE is then the Euclidean distance between the target's estimated position and it's actual position, which is calculated by

$$TLE = \sqrt{(\hat{x}_t - x_t)^2 + (\hat{y}_t - y_t)^2 + (\hat{z}_t - z_t)^2} \tag{4.29}$$

The sources of error that contribute to the hatted quantities will be discussed in the Problem Tradeoffs section during the next step of the design process.

Latency

In examining the latency of the three methods, all are essentially the same. All of the methods require the user to select the target from the video stream, after which the system calculates the target position on the ground. As a result, the only difference in latency is the calculation time, which is negligible compared to the time required to collect the data, package it, send it to the ground, wait for the soldier's response, and calculate the answer. If any of the alternatives had automatically detected the target's location and done the calculation onboard, thus saving the transmission time and the need to wait upon the user to respond, then a significant difference would exist amongst alternatives. However, since no significant difference exists, evaluating the latency does not serve the purpose of distinguishing between the alternatives. As a result, latency was removed from the design consideration.

Robustness

The final criterion to consider is robustness. As mentioned in Section 4.3.2, the standard deviation of the TLE for each method was used to quantify the robustness. The standard deviation was calculated by performing a risk analysis on the different levels of error that could potentially exist in the system. Tables 4.4 - 4.7 were used to establish typical error levels for each type of component. From these typical errors, five levels of error were then created for each type of component, those being zero, small, normal, high, and catastrophic error. These values are shown in Table 4.8 for each of the different components.

The equations developed in the accuracy section were then used to evaluate the magnitude of the TLE under each different level of error. This was done by first finding the various $\hat{m}$ quantities $(\hat{h}, \hat{r}, \hat{\theta}, etc$ by adding the error values in Table 4.8 to the known values. The known values were taken from the mission parameters defined in Chapter 3 (for example, $h = 500ft$, $r = 1200ft$, etc). The hatted values were then substituted into Equations 4.26, 4.25, and 4.27 for the Flat Earth, DTED, and Range methods, respectively, in order to generate the method's estimate of the target's position. The actual position of the target could be found by using the

Table 4.8: Error level taken from the components listed in Tables 4.5 - 4.7

	Level of Error			
Type of Component Error	**Small**	**Normal**	**High**	**Catastrophic**
Altimeter (h)	0.5 ft	1 ft	3 ft	10 ft
Autopilot Horiz Pos (x, y)	5 ft	10 ft	20 ft	30 ft
Autopilot Pitch Error (θ)	1°	2°	5°	10°
Autopilot Yaw Error (ψ)	2°	4°	10°	20°
Laser Rangefinder (r)	5 ft	10 ft	20 ft	30 ft

actual quantities instead of the hatted values. With both the estimated and the actual positions found the TLE was then calculated using Equation 4.29.

With TLE calculated for each scenario, the standard deviation needed to be calculated. However, the scenarios were not all given equal weight since, for example, the probability of the system having a normal error level is definitely not the same as it having a catastrophic error level. The probabilities assigned to each level will be discussed in further detail in the optimize the criteria section. A vector of TLE measurements was then created where the number of entries for any given value was proportional to its probability of occurring. The number of entries was found by multiplying the probability by 100, since all of the probabilities were set at two decimal places. For example, if a certain scenario for one of the methods had a probability of occurring of 0.33, then the TLE for that given scenario was included in 33 times. When all the entries for a given method were included in the vector, the standard deviation of the vector was taken. This served as the quantification of the robustness.

4.7 Problem Tradeoffs

4.7.1 Methodology

The next step in the design process is to perform the problem tradeoffs. Section 2.2.7 states that the trades necessary for the design are dictated by the design criteria. Section 2.2.7 lists three types of analysis that are usually contained within a given design criteria. They are sources of error analysis, option analysis, and future scenario analysis. Each will be described in detail.

Sources of Error Analysis

The general class of sources of error analysis can be divided into two separate tasks, identifying the sources of error and performing analysis on the sources of error. While simplistic, identifying the sources of error should not be overlooked. Many different types of analysis can be performed on sources of error. Since the design methodology is focusing on developing physical systems that are often component based, Section 2.2.7 mentioned sensitivity analysis as a common way to quantify how different sources of error affect the system's performance. The next two sections will first look at suggestions for identifying sources of error and then will discuss sensitivity analysis in further detail.

Identify Sources of Error Sources of error are typically associated with systems that either have governing equations or physical components that take measurements. If governing equations are defined for a part of the system, a simple but effective method for finding sources of error is to examine the terms to see which are constant and which could have error. For a component-based system, functional decomposition could help. As was done before, functional decomposition breaks the system up into what needs to be done or calculated. Then, the designers can list how each is being done and found. The presence of instruments is a good indication that error exists, as no measured quantity can be measured perfectly. Another simple tip is to look at the specifications sheet for each component and see what errors are identified.

Sensitivity Analysis Once the sources of error are identified, analysis of the error can be performed. As was stated above, one of many types of such analysis is sensitivity analysis. As was stated in Section 2.2.7, sensitivity analysis is designed to establish how much of a change in output is derived from a known change to a given parameter. For error analysis, this is accomplished by holding all the sources of error constant except for one, which is varied. The successive variation of the each source of error results in the measurement of the output or property of interest over a range of scenarios. The results are typically graphed, with the varied source of error serving as the independent variable and the performance or property of interest serving as the dependent variable. A sample sensitivity analysis plot is shown in Figure 4-26.

In the case of the Figure 4-26, the source of error is the measurement of the height of the vehicle and the property of interest is the TLE. The error in height is varied from $0ft$ to $\pm 20ft$. In this plot, all of the other sources of error were zero and only one method was examined, since it was solely trying to serves as an illustration of a sensitivity analysis plot.

The figure also shows how sensitivity analysis can also be used as a tool to ascertain derived requirements. From the results of the sensitivity analysis, the designers can see the upper bound required for each source of error in order to still meet a stated requirement. This can be aided graphically by placing the threshold and objective levels of the pertinent requirement on the sensitivity analysis graph, shown in Figure 4-26 as the two horizontal lines on the plot. In the example, the entire error range evaluated meets the threshold requirement, as seen by the fact the resulting error is completely below the threshold line. In order to meet the objective, however, the error in height needs to be bounded to under 12.5 ft.

Also, when the suite of similar such plots from all the sources of error are considered as a whole, the designer can ascertain which source of error dominates the output. With the derived requirement for the source of error and the order of dominance established, the designers can then evaluate the best way to achieve realize the required system. As mentioned in Section 2.2.7, this might necessitate another iteration of the design methodology. It could also mean that the designers need to

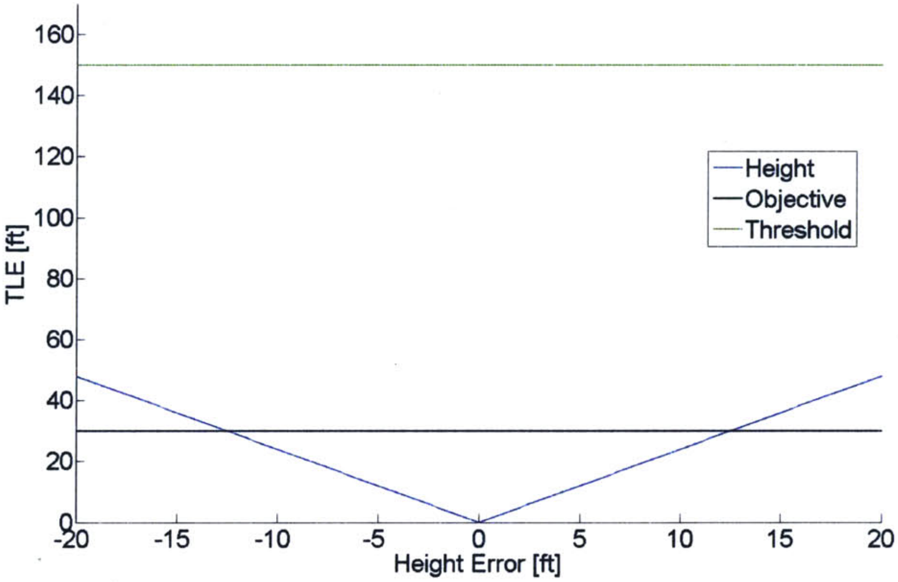


Figure 4-26: Sample sensitivity analysis of how error in height affects TLE.

perform options analysis, which will be discussed in greater detail in the next section.

Options Analysis

As stated in Section 2.2.7, options analysis tries to give the designer information about how the system would be affected if various choices were made. Much like sensitivity analysis, options analysis allows the designers to vary parameters or components in a controlled way such that the difference in performance can be attributed only to the choices being made. In doing so, the designers create an environment that allows valid conclusions to be drawn about the effects of the choices. As Section 2.2.7 says, one of the most common types of option analysis for physical, component-based systems is component analysis. Two types of component analysis were mentioned in Chapter 2, class of component and individual component trades. Due to their inherent similarity, these two types of trades will be discussed together below.

Class of Components and Individual Components Trades Fundamentally, component analysis involves varying components and seeing how the variation affects the criteria of interest. These trades are specific to a given alternative, but are often similar in nature across alternatives. Often times, the variations are visualized in the form of a 2D graph, much like was the case in the sensitivity analysis, with a criterion on each axis. A sample of such a figure is shown in Figure 4-27. Often times, cost is treated as the dependent variable (though it is the independent variable in the figure) and most of the other criteria are treated as the independent variables. Thus, in aviation systems, common graphs are mass vs. cost, power vs. cost, size vs. cost, and performance vs. cost. Performance is also a common dependent variable in the graph. Figure 4-27 is a sample of an example where performance is the dependent variable, with cost as the independent variable.

The GPS + IMU example from Chapter 2 will be continued to demonstrate the trades. Say that the designers were interested in which IMU to use with a given GPS receiver. A class of components trade divides a component into discrete classes based upon some important distinguishing characteristic. This could be any of the design criteria or a property of the specific component. In the IMU example, the classes could be based upon cost, one of the design criteria, or instead on its drift rate, an IMU-specific property. Once the class criterion is selected, then discrete classes are created. These are usually based upon discrete bounds. In Figure 4-27, the transparent rectangles encapsulating four individual components are the classes of components. While the figure was not created for the specific GPS + IMU example since the y-axis label that does not match the IMU-specific criteria, it can still be used to illustrate the basic parts of a typical graph.

Then, one of several different, but very similar, methods could be employed to perform the class of components trade. First, the designers could pick a representative from each class and conduct the evaluation. The evaluation would follow the pattern described in the accuracy section of the previous step where the properties of interest would be calculated and compared, usually in plot form. Another option would be to take a couple of components from each class and average the pertinent properties.

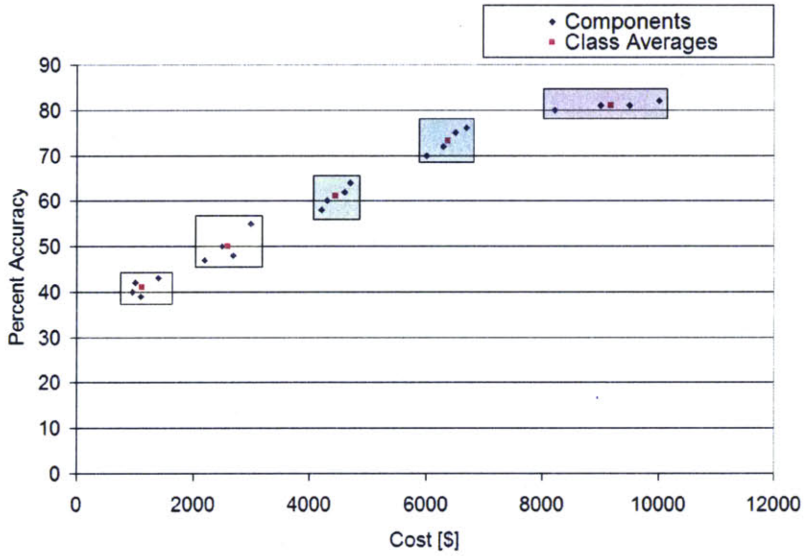


Figure 4-27: Example of Class of Components and Individual Components Trades

This is shown in Figure 4-27 by the pink squares contained within the classes, which is the average of the cost and performance metric for each of the components within the class. Instead of averaging the properties, the evaluation could simply be performed on each of the representatives for the classes. Regardless of the method used, based upon the results, the designers would select which class or classes best merit further investigation.

When specific classes are picked, the analysis essentially becomes one of an individual component trade. In this method, each possible component is evaluated, which would correspond to using all the blue diamonds in the Figure 4-27. In the GPS + IMU example, this would equate to the different options available for IMUs, such as the O-Navi Gyrocube, O-Navi Falcon, Honeywell HG1900, or any other specific IMU model considered by the designers. The individual component trade thus allows for the selection of a specific component whereas the class of components trade allows for the selection of a class of components. Of course, the class of components selected could then be evaluated to pick a specific component that best fits the system for a particular method.

Evaluating classes of components first can save time since designers do not have to evaluate every possibility. It allows for an initial blurred evaluation of the design space, followed by a focused evaluation for only the most relevant components. This is particularly an advantage when a strong trend exists between the class of component and the dependent variable being evaluated. However, if a trend does not exist, the class trade degenerates into an individual component evaluation. Also, the time saved on the evaluation comes at the expense of fidelity. By not evaluating all the possibilities, the designers increase the risk of making an incorrect choice on classes, especially if a strong trend does not exist or a small number of representatives are selected.

Regardless of whether a class of components or individual components trade is performed, the question being answered is the same: For a specific alternative, such as GPS + IMU for navigation, how does a dependent variable or group of dependent variables change when a specific component out of a range of possible components is varied? This information goes towards the question of which component is the best choice for the given alternative, which will be further addressed in the next step.

Future Scenario Analysis

The final type of analysis considered is future scenario analysis. Here, the existing state of the system is altered to a suppositional state in order to ascertain how the system would perform in the new state. The alteration of the state of the system usually comes in the form of a change to parameters of the system or important properties of components contained within the system. In a component-based system, on common form of future scenario analysis is future innovation analysis.

Future Innovation Analysis Future innovation analysis is aimed at answering the "what if" questions of design, usually related to specific properties of components. The analysis is performed by changing a property of interest to be that of some future hypothetical value. In aviation systems, a common theme of future variation analysis is to answer design questions related to weight. One such question is how would the system be affected if the weight of a component was kept the same but the performance improved? A similar such question considers the implications of the performance of the component being constant, but the weight is reduced.

This type of analysis requires more of an overview of the system. Considering one question typically leads to considering many more. For instance, in the case where a component's performance is held constant but the weight is reduced, what is done with the extra weight? Is it best to add an extra component to the system or just fly lighter? If a component is added, what is gained with its addition? What are the downsides of doing so? How much money is required for the reduction in weight to be achieved? What about for the addition of the new component? Can the project budget support such modifications? What does the increased expenditure translate to in terms of performance? The list of corollary questions related to the initial innovation analysis question could go on and on.

The questions are not simply related to the system either. As mentioned in Section 2.2.7, future innovation analysis could also serve as a strategic analysis for a company by deciding the avenues to pursue in the future through the evaluation of "what if" scenarios. A sample question a company could consider in this vain is, "Is it worth investing the resources to accomplish the changes in the component?" Corollary questions could include, "What is the estimated improvement in the system as compared to the estimated investment required?" and "What is the perceived increase in value of the system from the customer's perspective?" Another major question that could be addressed by future innovation analysis is "Which of these potential projects would produce the greatest impact on the system as perceived by the customer?" Information ascertained through answering these and other questions could be used to justify pursuing future proposals or as a marketing strategy for a product that offers substantial improvement over an existing component. In the latter case, the analysis could be used to show the customer the impact the new component would have on the system and the overall value that is added as compared to the additional cost. This analysis makes the case for using the new product much stronger and much harder for the customer to deny.

4.7.2 Application

The methodology section described several different techniques available for exploring the trade space of the design, including sensitivity analysis, class of component trades, individual component trades, and future innovation analysis. Tables 4.4 - Table 4.7 show that too few components were investigated to perform a class of component trade. The individual component trades will be combined with the next step and final step in the design process. Also, time did not permit for conducting future innovation analysis. However, error analysis was conducted on system and will be discussed in the subsequent paragraphs. As was described in the methodology section, error analysis involves the identification of the sources of error, followed by their analysis, which is usually done with sensitivity analysis. As a result, the sources of error will first be identified, after which sensitivity analysis will be conducted.

Source of Error Identification

Figure 4-28 shows the results of identifying the sources of error for a camera-based targeting system with a relative-to-self method. The major sources of error are pointing error, range error, vehicle position error, and camera error. Each of these will be discussed below.

Pointing Error The first major source of error is pointing error. This error pertains to the knowledge of where the camera's LOS vector is pointed. In order to know this, the attitude of the vehicle and the attitude of the camera relative to the vehicle need to be ascertained. The attitude of the vehicle is usually determined by an IMU or, as was described above, by a combined GPS+IMU package, such as an autopilot. The error in the orientation of the camera relative to the vehicle, termed Camera-to-Body Attitude in Figure 4-28, is a larger problem when the camera is gimballed an thus free to move around. If it is fixed on the aircraft, as was assumed to be the case in the system being designed, the error of greater concern is the error in the mounting of the camera. Two types of errors could exist related to the mounting, namely error in the position and error in the orientation. The error in where the camera is mounted on the vehicle (error in position) is a factor when it is considered with the vehicle's orientation. For instance, if the vehicle is pitched up at a positive angle of attack and the camera is mounted further back on the vehicle than is thought, then the camera will have a lower altitude than was thought. Due to the small size of the aircraft, however, and the relatively small errors, this is not typically much of a concern. The

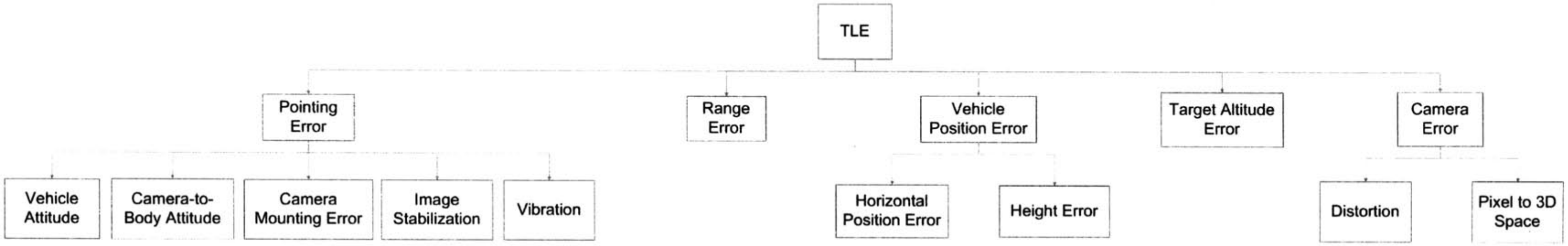


Figure 4-28: Sources of error for a camera-based targeting system with a relative-to-self method

other type of camera mounting error is in how the camera is oriented. An example of this type of error would be if the camera is supposed to mounted so that it is pointed in the direction of the heading vector (out the nose) and with a pitch that is 65° from vertical but is actually mounted 2° off the direction of the nose and at 60° from vertical.

The next source of error comes from a feature that is often added to small UAVs. The feature is image stabilization, which is designed to remove some of the vibration that results from flight. The benefit is that the users are presented with a more stable image and are thus better able to see what is in the camera's field of view. However, the stabilization software often obfuscates the exact orientation of the camera from the system. The vibration against which the image stabilization tries to guard can also be a source of error. If the camera is oscillating at a high rate, the camera's orientation may change non-trivially between measurements.

Range Error, Vehicle Position Error, and Target Altitude Error The next three sources of error are fairly self-explanatory. The first of these is the range to the target. As the name suggests, it is simply the Euclidean distance from the camera to the target. Similarly, the vehicle position error is straightforward. The horizontal position error refers to the x- and y-coordinates of the position while the height error deals with the z-coordinate. Finally, the target altitude error is the error in the presumed value of the target's z coordinate. This comes into play when using the DTED data.

Camera Error The final class of error to consider is the camera error. Two types of errors related to the camera are in distortion due to the lens used and in the mapping of pixels to 3D space. The amount and pattern of distortion in an image depends upon the type of lens used and its characteristics. Explaining the different types of distortions and how it affects the mapping of pixels to 3D space is beyond the scope of this thesis, but is readily available in literature and textbooks.

Errors Considered

Not all sources of error shown in Figure 4-28 are major enough to consider in the sensitivity analysis. Each of the major sources of error discussed above will be revisited below to mention if they will be considered and if so, how they will be measured.

As mentioned in Chapter 3, the system assumes that the camera is fixed, so there is no error in the camera-to-body attitude because the it remains a constant, fixed value. Furthermore, it is assumed that the mounting error and vibration are negligible and that no image stabilization exists. As a result, the pointing error is solely due to the error in the vehicle's attitude. In the conceptualization of the system described in the previous steps, the attitude of the vehicle was measured by the autopilot. Thus, the autopilot's yaw and pitch errors will serve to quantify the system's pointing error.

The range, vehicle position, and target altitude errors were all deemed important enough to consider. In Section 4.5.2, laser rangefinders were selected as the method to measure the range to the target, thereby making the error in these components the quantification of the range error. Similarly, autopilots were decided upon as the method for receiving the estimate of the horizontal position, making the autopilot's horizontal position error serve as the source of the error in x and y. Finally, altimeters were chosen to determine the vehicle's height, making the system's altimeter's error the value for the error in h.

The final source of error, those related to the camera, were decided to be insignificant in comparison to the other sources.

It should be noted that with the above inclusion and exclusion of errors, all the quantities in the equations developed in the accuracy subsection (x, y, h, r, ψ, θ) have errors associated with them and components to which these errors are tied. This will become critically important in the optimization.

Sensitivity Analysis

The shapes of the graphs generated in the sensitivity analysis for the TLE depends on the parameters of the scenario used to simulate the mission and the vehicle. For instance, a scenario where the target location is at the reference altitude does not evaluate the same as when the target is significantly above the reference altitude. This is because the Flat Earth method assumes that the target is at the reference altitude. As a result, the when the target is actually there, the performance is much better than when it is not. Also, the assumed geometry of the Earth around the target also affects the results, especially for DTED. For example, take the scenario where the target is at the edge of a cliff and the vehicle is approaching the target from the side that is still on the cliff. If the pitch is even the slightest bit too high, then the LOS vector of the camera points over the cliff and the target location returned by DTED is those indeterminately wrong. Thus, the scenario parameters must always be considered when analyzing the sensitivity plots.

The sensitivity analysis was performed under two different scenarios, both of which have roots in the vehicle and mission parameters defined in Chapter 3. As a result, both scenarios have $h = 500ft$ and $\theta = 65.38°$. Also, the scenarios arbitrarily set x, y, and ψ to 0, 0, and 45°, respectively. Thus, the scenarios capture a snapshot where the vehicle is heading northeast at the origin of a coordinate system with an altitude $500ft$ above the reference level.

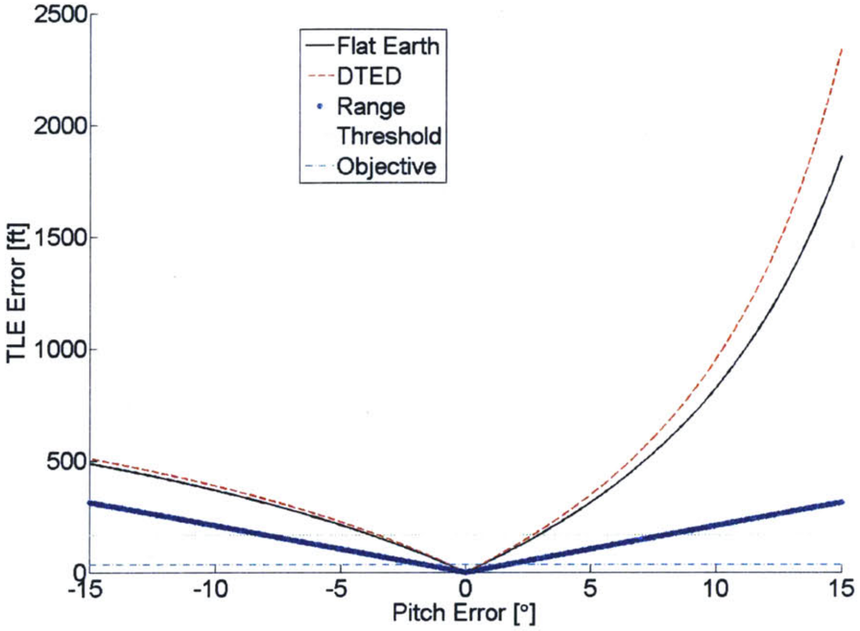
In the first scenario, all of the sources of error are equal to zero, the target is located at the reference altitude (yielding a slant range of $1200ft$), and the ground is sloped downhill at 2°. The point of this scenario is twofold. First, the scenario is designed to demonstrate the shapes of the sensitivity analysis without any other errors confounding the plots. For instance, if the vehicle position was off by 10 ft in the positive x direction, the sensitivity analysis for the xy position would no longer be centered at the origin, but instead would be shifted 10 ft to the left so that the TLE would be zero when the system had a -10 ft error in x. This scenario is supposed to show the effect of just the source of error being varied. Admittedly, in doing so, the Flat Earth method is given an advantage since its assumption about the target's altitude is correct and the DTED method is given a disadvantage because the target is on the backside of a hill. However, a 2° downhill slope is a reasonable geometry for the ground, so the DTED method is not given an unreasonable disadvantage.

Also, the first scenario aims to demonstrate the setting of upper bounds for the errors. The upper bound for each source of error is the value of the error that produces the threshold value of the TLE when all the other sources of error are equal to zero. In this case, all of the error in the target's position is attributed to the source of interest. While it is unrealistic to believe that the other sources of error would ever be zero except on an extremely rare occurrence, the evaluation is used to simply produce the upper bound. The upper bound by definition occurs when the error of interest is the highest it could possibly be while still meeting the TLE requirement and would likely need to be lower due to the presence of the other errors. A similar upper bound could also be generated for the value of the error that produces the objective value in the TLE. The bounds will be discussed below.

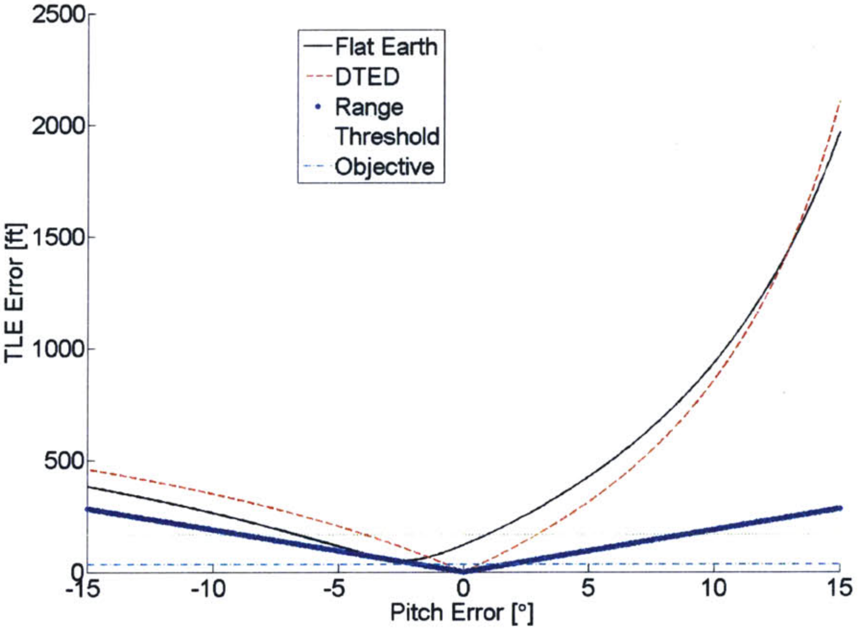
In the second scenario, the only parameter that is changed is the location of the target, which is moved to be $50ft$ above the reference altitude, thereby changing the slant range to $1080ft$. At 10% of the vehicle's altitude, this marks a sizable jump. Doing so will take away the advantage given to the Flat Earth method. Also, it will allow for the demonstration of how the sensitivity plots change when the scenario is altered. However, by just changing one parameter, the change will not be too drastic and will thus allow for a comparison to be drawn.

Pitch Error The first plot considered is Figure 4-29, which deals with the system's sensitivity to errors in pitch. The plots show that the TLE is extremely sensitive to pitch error. Of all the sources of error, pitch error is the one that is the most dominate by a wide margin. For a positive 15° error, the Flat Earth and DTED methods had TLEs of at least $1860ft$. As the plot for Scenario 1 shows, both these methods experienced non-linear growth in the positive regions of the error. By contrast, the range error had a significantly reduced error, with a 15° error producing a TLE of approximately $320ft$, with a much slower growth in the positive region.

The significant difference can be explained by differences in how the methods find their solutions. To aid in the discussion, Figure 4-30 shows what is happening graphically. In the Flat Earth and DTED solutions, the pitch and height are used to define the right triangle shown previously in Figure 4-24. When the pitch measured is greater than the actual pitch ($\theta_e > \theta$, resulting in a positive error), both the estimated range and ground distance are longer than they should be. Since the methods require that the triangle still go all the way down to the target's elevation, the error grows by an increasing amount with increasing θ values. By contrast, in the Range method, the length of the hypotenuse is fixed, since it is the value measured by the laser rangefinder. Because there is no knowledge or assumption about the target's altitude, as the pitch changes, the hypotenuse traces a circle in the plane of the side view. As a result, positive pitch errors result in solution that is too long in the XY dimension and too high. Thus, the TLE is the length of the chord between the estimated location and the actual location. This concept will be described in greater detail in the yaw sensitivity analysis and will be aided by the use of polar plots. Polar plots were not used in the case of the pitch sensitivity due to the lack of



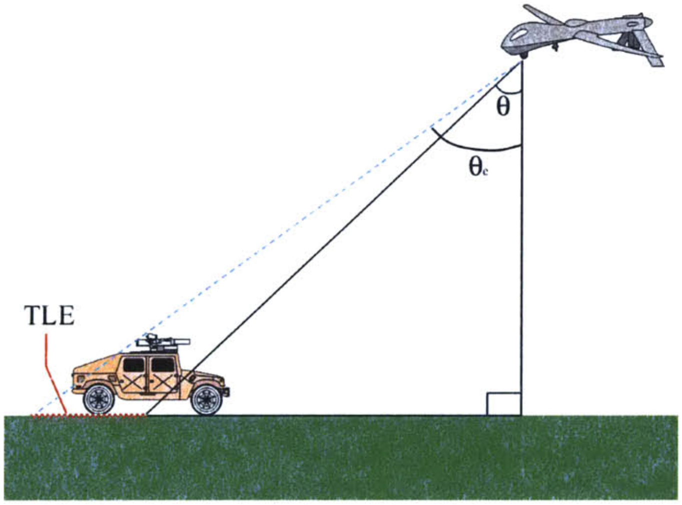
(a) Scenario 1



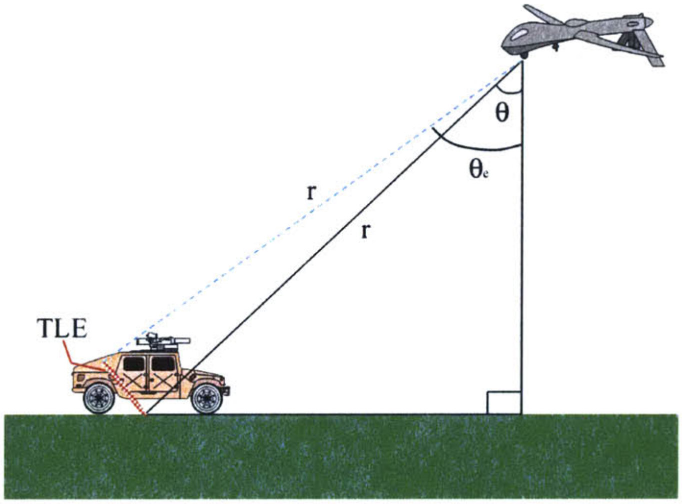
(b) Scenario 2

Figure 4-29: Sensitivity analysis for the pitch error.

visibility of the range method, since its magnitude was dominated by the other two methods.



(a) Flat Earth, DTED



(b) Range

Figure 4-30: Demonstration of why the substantial difference exists between the methods for pitch error.

In the first scenario, the DTED error is worse than that of the Flat Earth method. There are two reasons for this development. First, as was mentioned above, the target is located at the reference altitude, thereby giving the Flat Earth the advantage that its assumption about the target's altitude is actually correct. However, the difference comes in that the DTED method incorporates knowledge of the terrain into its calculates. As stated above, the geometry of the target's location was a 2° downhill slope. Thus, the TLE is even greater because the LOS vector has to intersect a ground that is sloping away from it. If the 2° slope was changed to be 0°, the evaluations would be the same.

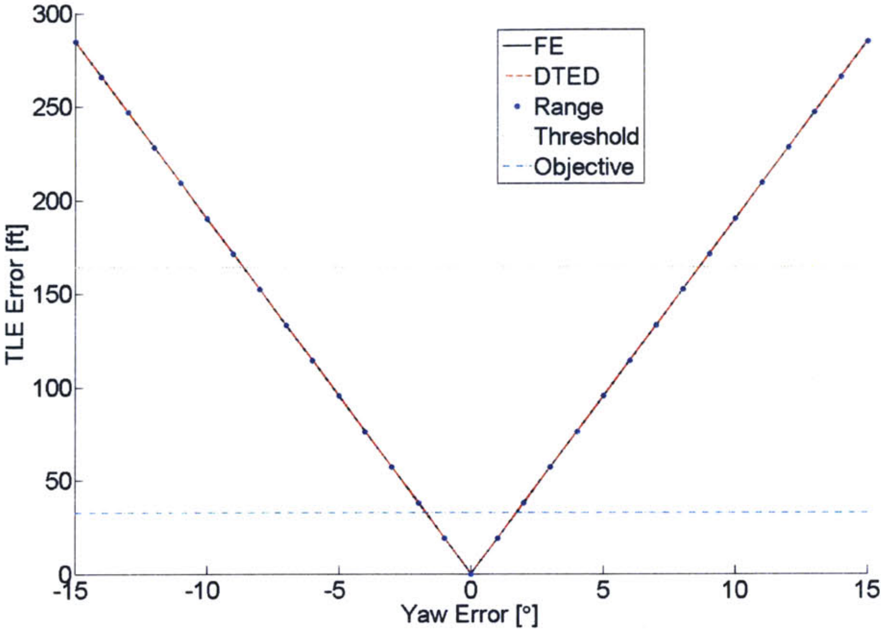
In the second scenario, the target was elevated $50ft$, or 10% of the vehicle's altitude. Doing this also reduces the range from $1200ft$ to $1080ft$, making the target closer to the vehicle. The most noticeable difference is in the Flat Earth method. The basic shape of the graph is shifted to the left. Now, when no pitch error exists, the system still has a TLE of $120ft$. This is because its assumption about the vehicle's altitude is no longer correct and the TLE is therefore contributed to by both the pitch error and the error in the target's altitude. With this in mind, the reason for the shift to the left becomes easier to understand. While positive pitch errors result in the target's location being further downrange than they actually are, negative pitch errors have errors that are just the opposite. The shallowing of the pitch is actually beneficial for the first few degrees in this scenario, however.

This can best be seen by reviewing Figure 4-25. Here, the target is elevated on top of a hill. If the pitch contains no error, then the estimate is located where the figure shows. However, with a negative pitch error, the leftmost point of the triangle would progressively move to the right until it is ultimately directly below the vehicle. At this point, the TLE would be equal to that of the target's altitude. However, as the pitch continues to shallow, the error would begin to increase again. This is indeed what is shown in the sensitivity analysis. The error reaches a minimum of $50ft$, the altitude of the target, at a pitch error of approximately $-2.4°$ and then begin to grow as the negative error continues to grow.

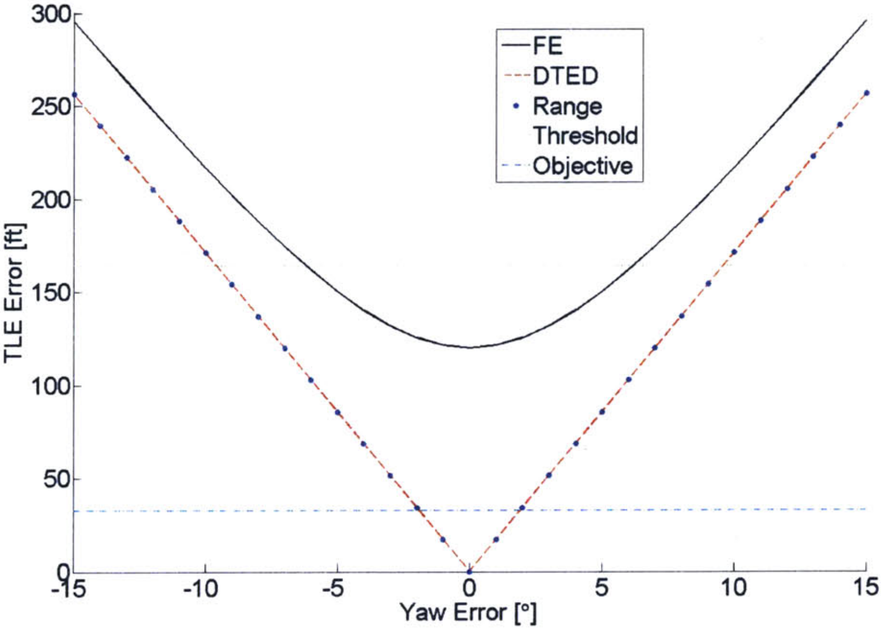
By contrast, the DTED and Range methods have the same basic shape as before, but the magnitude of the error is slightly reduced. This is because of the reduction in the target's range.

Yaw Error The next major source of error to consider is the yaw error. The sensitivity analysis for yaw is shown in Figure 4-31. As the figure shows, for the first scenario, all of the methods behave in the same manner. This is due to the assumed geometry of the world around the target. Vertically, the ground is assumed to be slanted downward at a 2° slope, which, as was seen above, makes a difference in the pitch evaluation. However, horizontally, the world is assumed to be flat. Thus, the world around the target is a downward sloping wedge. The resulting error is thus the same as was seen in the pitch error for the Range method. Since the triangle is perfectly defined due to no errors existing in θ, h, and r, the circle traces out a circle in the xy plane. This would be seen the line in Figure 4-16 tracing out a circle. The error is then the length of the chord from the estimate to the actual location.

The cartesian plots of Figure 4-31 are deceptive, however. They appear to show that the change is linear in nature. However, thinking about what is described in the previous paragraph yields a non-linear change with yaw. If a point on a circle is fixed and then a chord is drawn from that point to every other point on the circle, the result would be that the point 180° away would have the greatest length, equal to the diameter of the circle. As a result, every line drawn between 0° and 180° would grow in length, and then from 180° to 360°, the lines would decrease in length. The symmetry shows that the behavior could not be linear. As a result, the yaw



(a) Scenario 1



(b) Scenario 2

Figure 4-31: Sensitivity analysis for the yaw error.

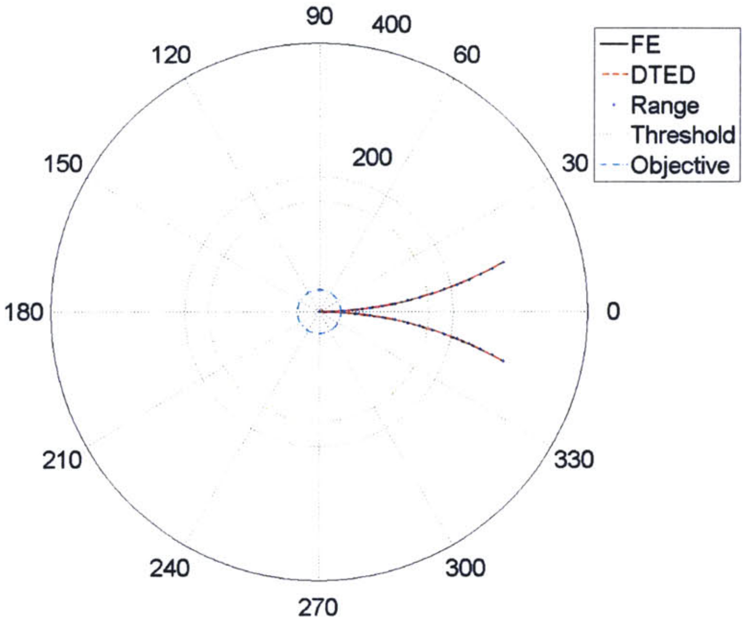
sensitivity analysis was repeated using polar plots in Figure 4-32. The result shows that the error is in fact non-linear. The analysis was then performed on the full 360° of possible values for the yaw error. The result is shown in Figure 4-33. This plot reveals the actual shape of the yaw error is that of a cardioid. The full plot also is the best demonstration of the difference between the two scenarios. The change of height increases the cardioid's size, but the basic shape is still the same.

It is important to note that while the error is indeed rigorously non-linear, in all but the worst cases in small UAV applications, it could be approximated as linear due to the small angle approximation of the curve in Figure 4-33. The autopilot or similar such component should bound the yaw error to the region shown in the plots, which could be approximated as linear with high fidelity. If the error reached the non-linear stage, such as being off by a full 180°, then the vehicle would have such a catastrophic failure in navigation that it would most likely not be used for targeting anyway.

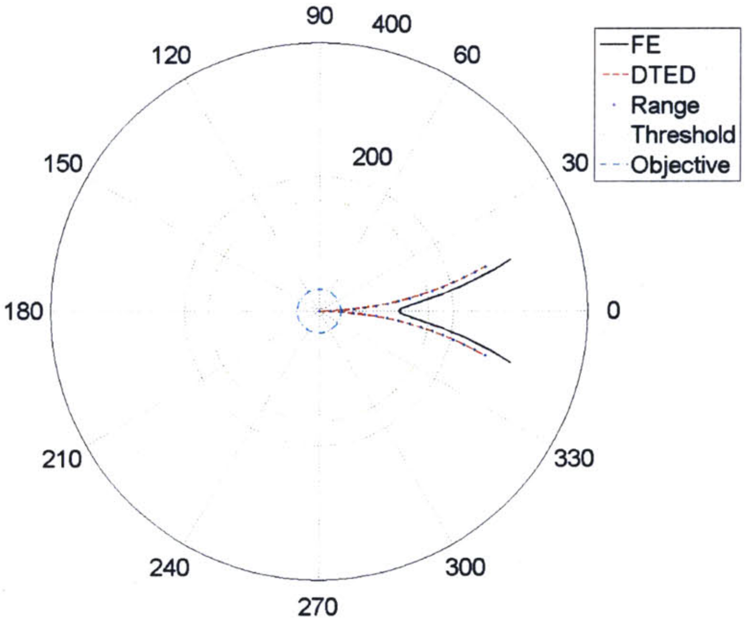
Height Error The next source of error to consider is the height error. The associated plots of the sensitivity analysis for the two scenarios are shown in Figure 4-34. Also, as was done with the pitch error, Figure 4-35 demonstrates the geometry for the methods. Since the error is in the vehicle's altitude only, the resulting change in the geometry is to shift the triangle up or down. The figure shows a positive error in h, thereby moving the triangle up.

As was the case with the pitch error, the fundamental difference between the geometries shown in Figure 4-35 is the fact that the range and ground distance are allowed to change in the Flat Earth and DTED methods but are constant in the Range method. Since the triangle geometry does not change at all in the Range method, the only change in the target's location is the shift upwards. Thus, as shown in Figure 4-35, the estimated location has the correct x- and y-coordinates, but the z-coordinate is too high. In the sensitivity plots, this shows up as a linear change with a slope of magnitude 1, sloping negatively in the negative regions and positively in the positive regions. Thus, the TLE is simply the absolute value of the height error for the Range method.

For the Flat Earth and the DTED methods, the triangle size is affected, but not the shape. Said differently, the two triangles meet the mathematical definition of similar triangles. They have the same angles, but the only difference is in the lengths of their sides (the Range triangles were congruent). The shape of the sensitivity plot is still linear, but the slope of the line depends upon the value of θ. If θ were small, meaning the camera would essentially be looking straight down, changes in height would result in small TLEs in comparison to the changes in height. However, if the θ were large, meaning the camera was looking nearly horizontally, changes in height would result in changes in TLE larger than that of the height. If the camera were at 45°, the changes would be equal. In terms of the sensitivity analysis, this means that the for $\theta < 45°$, the slope of the sensitivity plot would be between 0 and 1, with smaller values of θ being closer to 0. Similarly, for $\theta > 45°$, the slope is greater than 1, with the slope growing with increasing θ. Since the $\theta = 65.38°$ in both the



(a) Scenario 1



(b) Scenario 2

Figure 4-32: Sensitivity analysis for the yaw error graphed with a polar plot.

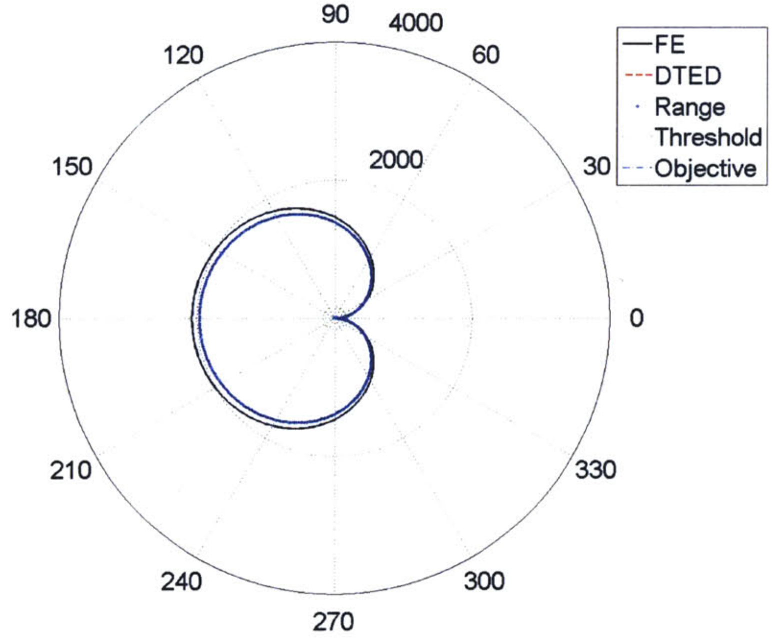
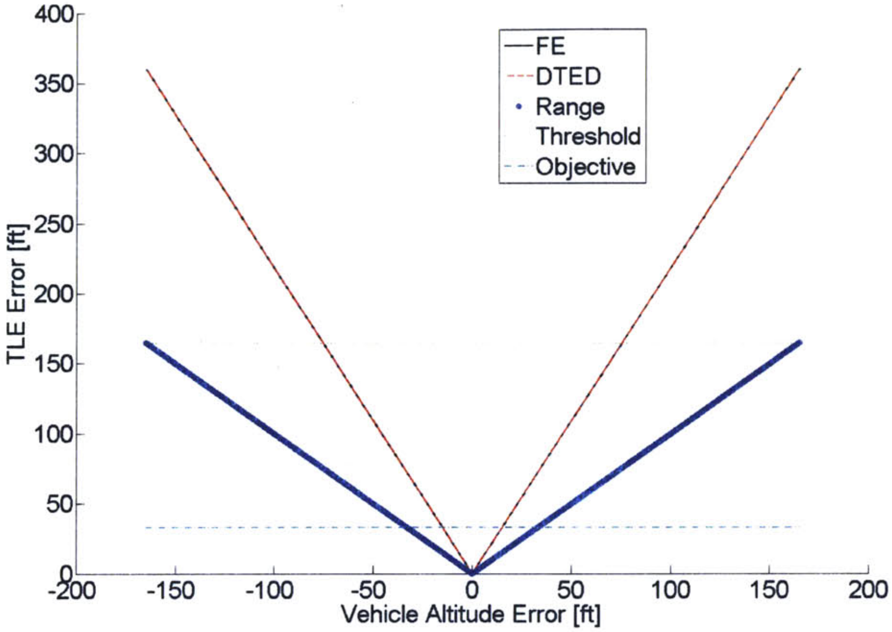
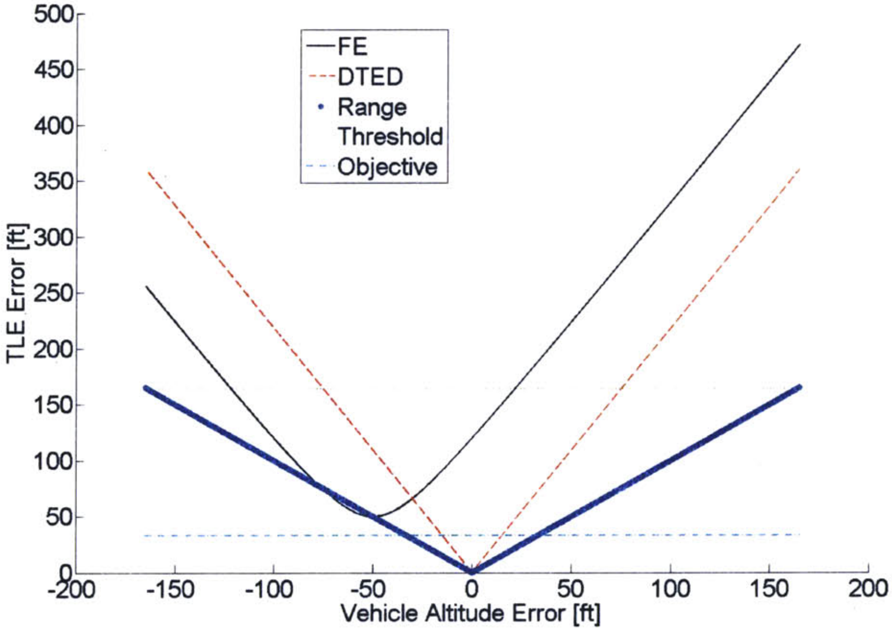


Figure 4-33: Sensitivity analysis for the yaw error over all possible values graphed in a polar plot.



(a) Scenario 1



(b) Scenario 2

Figure 4-34: Sensitivity analysis for the height error.

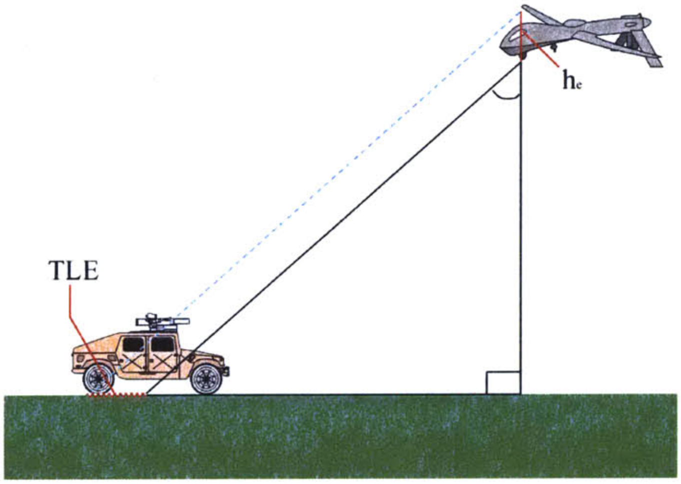
scenarios, the slope is greater than one, thereby explaining why the Flat Earth and DTED methods are more affected than the Range method.

In the second scenario, the only change comes with the Flat Earth method, as the other two are not affected. The explanation of the curve is similar to that of the pitch error. Again, Figure 4-25 can be used to illustrate the process. With no height error, the Flat Earth method has a non-zero TLE due to the inaccuracy in its assumption about the altitude. Negative errors in height mean that the triangle shrinks in size instead of growing. As a result, the leftmost point of the triangle moves to the right until the point it is ultimately directly below the target. This is the point of the least TLE, which is equal to the height of the vehicle. As the negative height continues to shrink the triangle, the TLE begins to grow away from the minimum. This explanation is borne out by the Flat Earth plot for the second scenario. It reaches a minimum TLE of $50ft$ for a negative height error of $50ft$, after which point the TLE begins to grow.

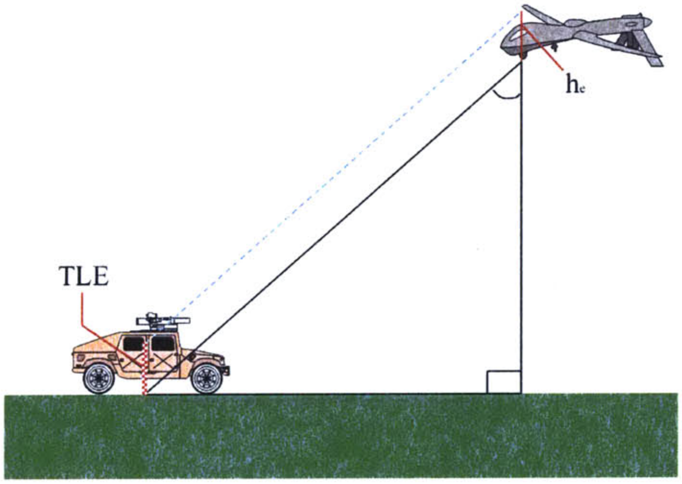
XY Position Error The fourth source of error considered was the XY position error. The resulting sensitivity plots are shown in Figure 4-36. Though the picture is not shown, the geometry of this source of error is similar to that of the height error. The difference is that instead of a vertical shift in the triangle, a horizontal shift occurs. Because there is no movement vertically, there is no difference between the methods when the target is at the reference altitude. Here, the change in geometry is the same as was the case for the Range method in the height error. The two triangles are congruent and are just shifted versions of each other. As a result, all the methods have the same shape in the first scenario. The TLE is just the absolute value of the XY position error.

In the second scenario, the Range and DTED methods exhibit no change. However, a difference is present in the Flat Earth method. The difference is the third appearance of the same basic idea described in the pitch and height error descriptions. The plot has a left shift because doing so puts the triangle directly underneath the target, thereby giving the method a minimum TLE equal to that of the target's height. For a more detailed description, please refer to the other two sections.

Range Error The penultimate source of error considered was the range error. Unlike the previous four sources of error, this source of error does not unilaterally apply to all of the methods. Instead, range error is only applicable to the Range method, as it is the only method of the three that uses range in its calculation of the target's error. Its plots are shown in Figure 4-37. An illustrative way to think about this source of error is that the hypotenuse of the triangle is either too short or too long by the amount equal to the error. Since this is the case, the error is always in the direction of the camera's LOS vector. Because the TLE is measured by calculating the Euclidean distance (which is just the linear distance) between the estimated position and the actual, the TLE is always just equal to the error since the error must be along the LOS. This fact does not change between the scenarios. The only difference in the graph is that the Flat Earth method went from a constant $0ft$ error in the

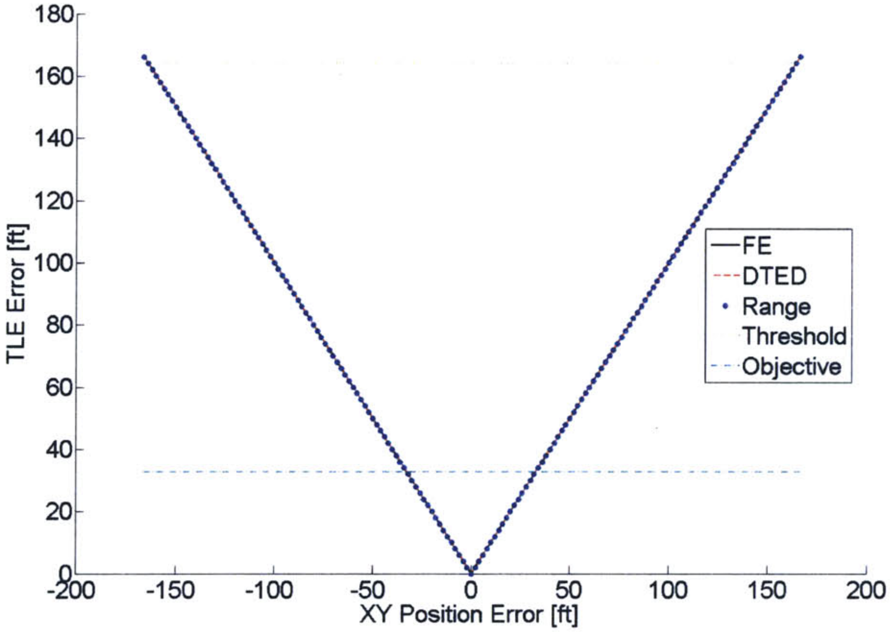


(a) Flat Earth, DTED

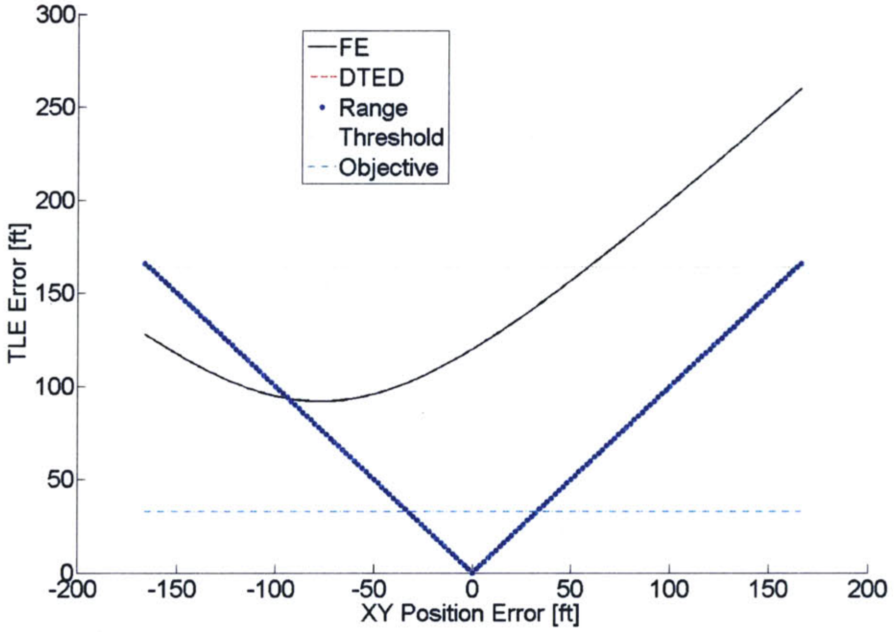


(b) Range

Figure 4-35: Demonstration of the geometry of the height error.



(a) Scenario 1



(b) Scenario 2

Figure 4-36: Sensitivity analysis for the XY position error.

first scenario to a constant $120ft$ error in the second. The difference is because the change in the target's altitude.

Target Altitude Error The last sensitivity analysis performed was for that of the target altitude error. As was the case with the range error, this source of error only applies to one of the three methods, namely the DTED method. This type of error comes from the DTED data and could be contributed to by a couple of factors. Every DTED pole has error associated with the height it. Also, the levels of DTED are distinguished from one another by the spacing of the poles. If the target's location did not occur exactly at a pole location, then the height would need to be found relative to the location of the pole, which has some amount of error associated with it. The origin of the error is not of concern for the evaluation. What is, though, is its affect. A graphical representation of this type of error is shown in Figure 4-39.

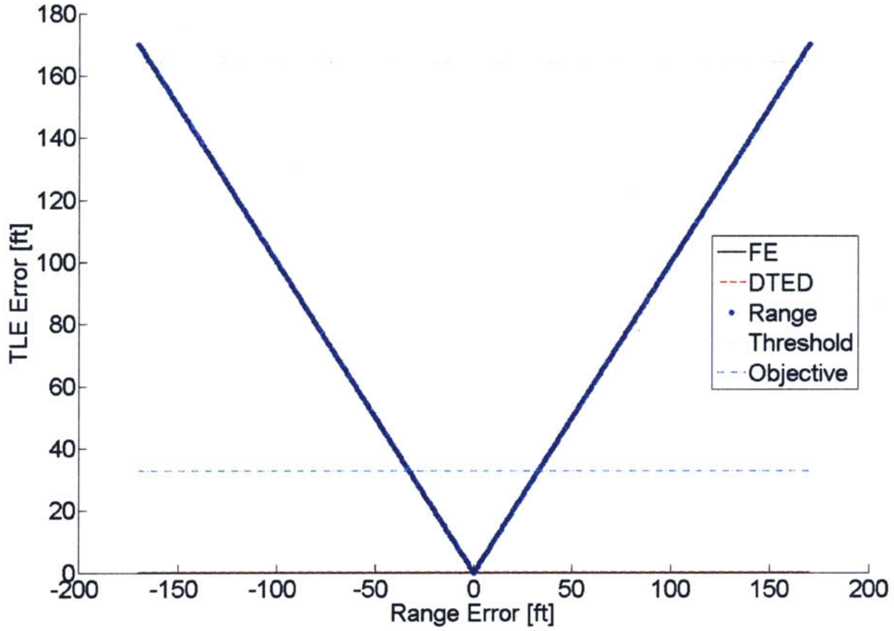
Since all the other sources of error are zero, the shape of the triangle cannot change. What can change, though, is the size of the triangle. Figure 4-39 shows that, due to the target altitude error, the system thinks the triangle is the blue dashed triangle when, in actuality, it is the full triangle. The TLE is then along the direction of the LOS vector, which is the hypotenuse of the triangle. The h_{tae} is the vertical component of a smaller right triangle with TLE serving as the hypotenuse. Since the TLE is the hypotenuse and hypotenuses are always longer than the either leg of the right triangle, the TLE will always be longer than the target altitude error. Also, since the error is along the camera's LOS vector, the error will be linear. Combining this facts, the target altitude error sensitivity will be linear with a slope greater than 1. Also, since the error follows the hypotenuse, having less error moves the red vertical line in Figure 4-39 further to the left edge of the triangle and increasing the error moves the line right. However, the basic picture always remains the same.

Upper Bounds

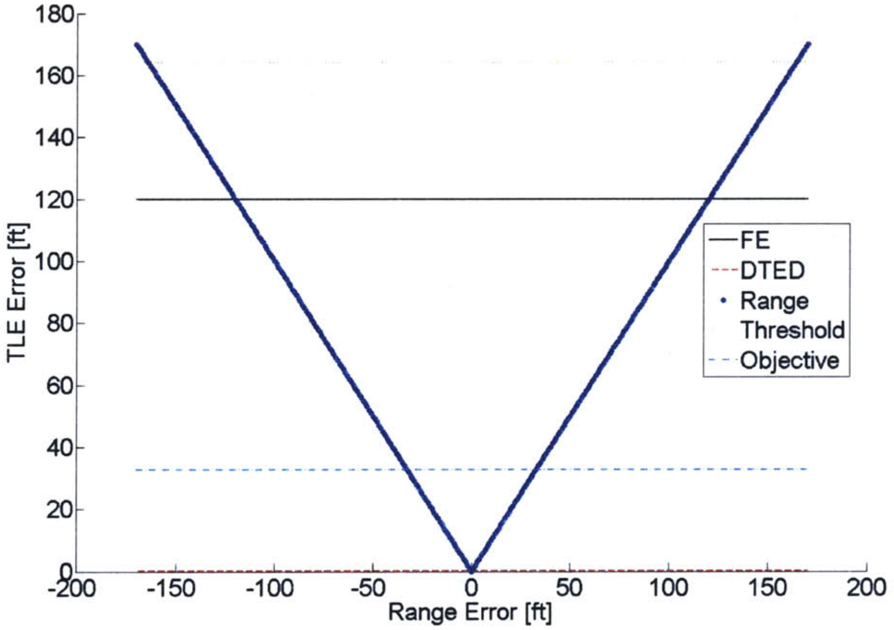
The upper bounds for both the TLE threshold and objective as calculated in the first scenario are shown in Table 4.9 for all the sources of error considered. Similarly, the upper bounds for the sources of error as generated by the second scenario are shown in Table 4.10.

The only changes between the two tables for the DTED and Range methods were in the pitch and yaw bounds. The bounds were relaxed in the second scenario as compared to the first, with the allowable error increasing. This happened because the range to the target decreased from $1200ft$ to $1080ft$ and the altitude change between the vehicle and the target was $450ft$ as opposed to $500ft$. As was seen above, decreasing these quantities lessened the affect of the two sources of error, thereby allowing the bounds to be relaxed.

The major changes were predictably in the Flat Earth method, as it was the method most affected by the change in altitude of the target. The significant changes in the bounds were due to the fact that now two sources of error were present, the source being varied and its assumption about the target's altitude. This combination meant that it could not even meet the objective value for any of the sources of

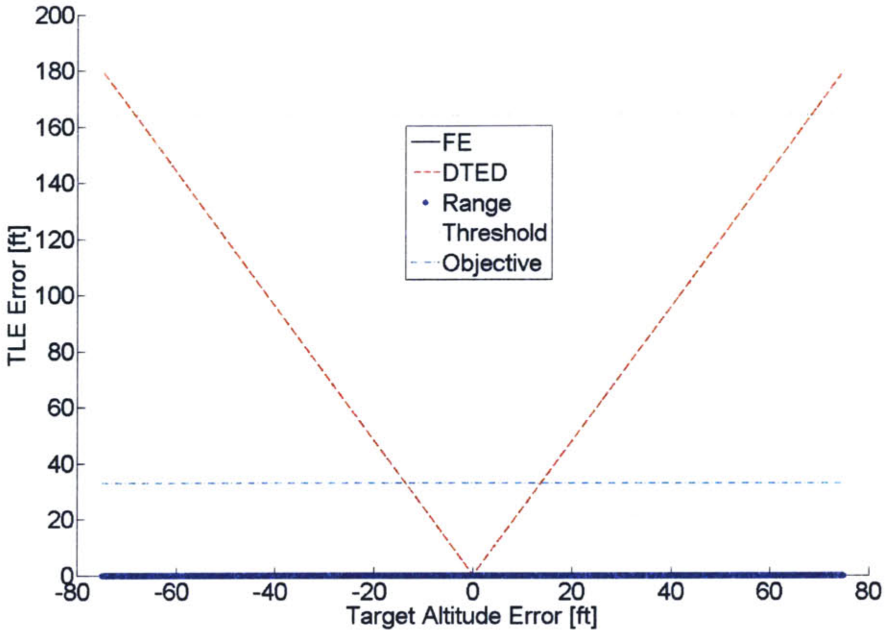


(a) Scenario 1

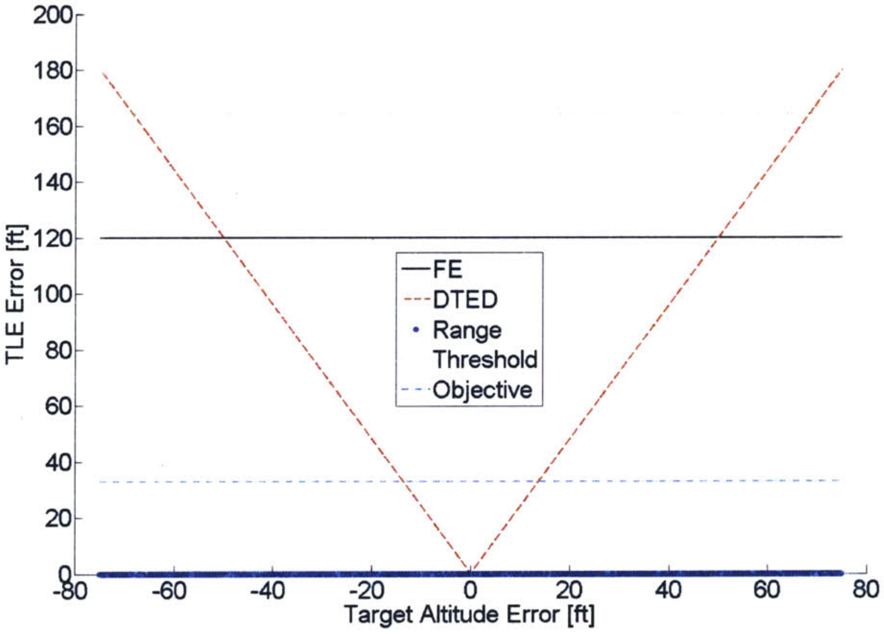


(b) Scenario 2

Figure 4-37: Sensitivity analysis for the range error.



(a) Scenario 1



(b) Scenario 2

Figure 4-38: Sensitivity analysis for the target altitude error.

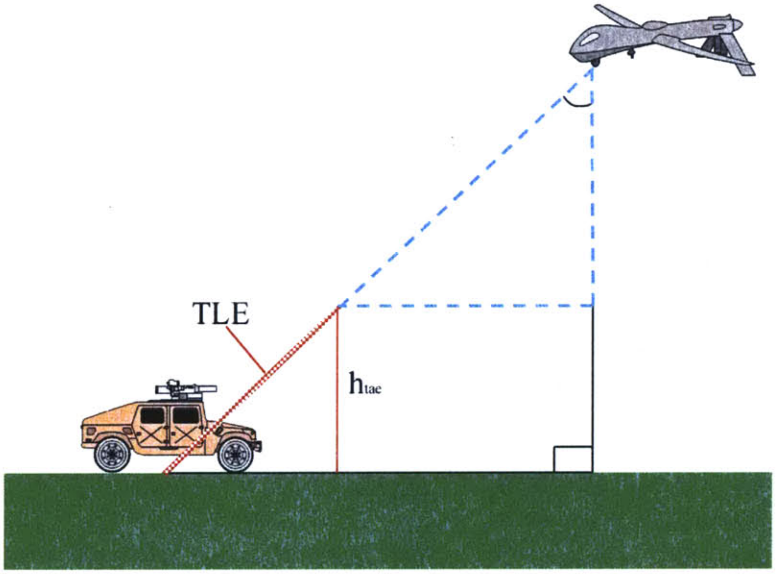


Figure 4-39: Demonstration of the geometry of the target altitude error.

Table 4.9: Upper bounds of the error for both the threshold and objective levels evaluated with the target altitude on the reference line and a 2° downhill slope (UB = Upper Bound).

Type of Error	**Method**	**Threshold UB**	**Objective UB**
Pitch	FE	2.9°	0.6°
	DTED	2.6°	0.5°
	Range	7.8°	1.5°
Yaw	FE	8.0°	1.0°
	DTED	8.0°	1.0°
	Range	8.0°	1.0°
Height	FE	$75.1ft$	$15.0ft$
	DTED	$75.1ft$	$15.0ft$
	Range	$164.0ft$	$32.8ft$
XY Position	FE	$164.0ft$	$32.8ft$
	DTED	$164.0ft$	$32.8ft$
	Range	$164.0ft$	$32.8ft$
Range	FE	N/A	N/A
	DTED	N/A	N/A
	Range	$164.0ft$	$32.8ft$
Target Altitude	FE	N/A	N/A
	DTED	$68.3ft$	$13.6ft$
	Range	N/A	N/A

Table 4.10: Upper bounds of the error for both the threshold and objective levels evaluated with the target altitude $50ft$ above the reference line and a 2° downhill slope (UB = Upper Bound, DNM = does not ever meet the requirement).

Type of Error	Method	Threshold UB	Objective UB
Pitch	FE	0.9°	DNM
	DTED	2.9°	0.6°
	Range	8.7°	1.7°
Yaw	FE	6.1°	DNM
	DTED	9.5°	1.9°
	Range	9.5°	1.9°
Height	FE	$21.6ft$	DNM
	DTED	$75.1ft$	$15.0ft$
	Range	$164.0ft$	$32.8ft$
XY Position	FE	$58.7ft$	DNM
	DTED	$164.0ft$	$32.8ft$
	Range	$164.0ft$	$32.8ft$
Range	FE	N/A	N/A
	DTED	N/A	N/A
	Range	$164.0ft$	$32.8ft$
Target Altitude	FE	N/A	N/A
	DTED	$68.3ft$	$13.6ft$
	Range	N/A	N/A

error that applied. The threshold values were still attainable, but they required considerable tightening of the bounds.

Further conclusions related to the sensitivity analysis will be delayed until Chapter 5.

4.8 Optimize Criteria

4.8.1 Methodology

The final step in the design methodology involves optimizing the criteria. As was mentioned in Section 2.2.8, optimization in this sense is not a strict global optimization. Instead, as was seen in Figure 1-12, the goal of this step could be anything from finding out the feasibility of the system to a pareto solution. With such a wide extreme, generalization becomes difficult. In order to develop a truly optimal system, the requirements would need to be encoded as constraints and the alternatives would need to be modeled in an MDO-like manner. This would require NLP or some sort of constraint satisfaction algorithm. The literature is full of these techniques, but they are beyond the scope of this thesis.

As suggested in Section 2.2.8, one of the simplest forms of optimizing the criteria is the full enumeration and evaluation of all the possibilities that survived the downselection of alternatives. Previously in the design method, the design criteria were established and a method of translation was developed between the properties of the system and the scores. Then, alternatives and components were established, with any governing equations or evaluation schemes needed to find the properties also being created. All that is left for a simple evaluation is to find the aggregate properties of the system, translate them into scores, and the sum the scores according to a user-defined cost function. The score that evaluates the highest would thus be the best alternative. Also, anything with a positive score would be considered within the requirements of the system.

This process is similar to the matrix-like evaluation that was demonstrated in Chapter 1. However, several key differences exist. First, the evaluation is solely property based. Since the discrete or continuous functions that translate the properties into scores were created before the alternatives were even defined, the translation of the scores is free from the bias of the evaluators. Also, the scoring is not relative, but is done on an absolute scale. Plus, the trades performed help the designers understand and justify critical decisions that need to be made.

While this method is simplistic, it answers several critical questions that the designers have about the alternatives. It tells the designers which alternatives meet the bounds, what the total properties of the system are, how the alternatives rank as compared to each other, and, through the trades, provides information about the general trends of the alternatives. It also leaves the opportunity for future, more rigorous simulation of only the alternatives that merit future investigation.

4.8.2 Application

The short timeline involved in the application of the design method was such that a full optimization was not possible. Instead, the suggestion of enumeration and evaluation of all the possibilities was used. This process will be described in the subsequent sections. However, before this is done, a brief summary will be given to review what has already been accomplished in the design so far.

Review of Process

The design criteria established to aid in the evaluation of the problem included weight/mass, power, size, unit cost, accuracy, latency, and robustness. Later, latency was removed, not due to a lack of importance, but instead because all of the alternatives selected were so similar in latency that it did not serve to distinguish between the choices.

The requirement were then used to help establish continuous functions that translated the properties of the system related to these design criteria into scores. These functions were developed in Section 4.3 and were shown in Figures 4-8 - 4-14.

After the translation functions were defined, the alternatives where defined, as was shown in Figure 4-15. From all these alternatives, three were selected, all of which were under the class of relative to self targeting methods. They were the Flat Earth method, the DTED method, and the Range method.

After the alternatives were established, they were decomposed into subcomponents. All the alternatives had three components in common, namely, an optical camera, an altimeter, and an autopilot. In addition, the Range method required the use of a laser rangefinder. Specific makes and models of all the components were found and their pertinent properties were listed in Tables 4.4 - 4.7.

The governing equations were then derived to turn the component properties into the overall system's properties as it pertained to the design criteria. These equations were derived in Section 4.6.2.

Finally, sensitivity analysis was conducted on each source of error in order to establish what the sources of error were, which source of error dominated, and what the upper bounds were for the errors.

Evaluation of Alternatives

With all of this accomplished, the major remaining steps are to define a cost function, enumerate all possible combination of components and methods, apply the equations to translate the components properties into system properties, translate the system's properties into scores, evaluate scores in the cost function, and report the results. Each of these tasks will be described below, with a separate section dedicated to the results.

Cost Function The cost function was designed to capture the feel of a project that addresses an urgent military need while still being a reasonable project. As a result, the cost function used in the evaluation was $TotalCost = weight + power + size + \frac{1}{2} \times UnitCost + 2 \times accuracy + robustness + 0 \times latency$.

The most noticeable weights of the cost function are those of the unit cost, accuracy, and latency. The weight for latency was assigned to 0 in order to reflect the fact that it was an original design criteria, but that was removed in this specific evaluation of the problem. A weight of two was assigned to accuracy, making it the most heavily weighted criteria. This was to reflect the fact that the entire point of the system was to perform targeting and if this could not be accomplished accurately, then the system's usefulness would be in doubt. Finally, the unit cost was given a weight of $\frac{1}{2}$ to reflect the fact the targeting system is an urgent customer need for which they are willing to incur some cost, hence the fact it was weighted less than the others, but not so much cost to make the system unreasonably expensive. In order to give a better feel for how the cost function assigns weights, Table 4.11 shows what percentage each of the criteria contributes to the total cost.

Enumerate and Evaluate Possibilities With the cost function established, the next major task is to enumerate all of the possible solution given the components and methods. To aid in this process, a series of Python scripts were developed. Python is a high level, object-oriented scripting language. A flow chart of the scripts is shown in Figure 4-40. Each part will be briefly described below.

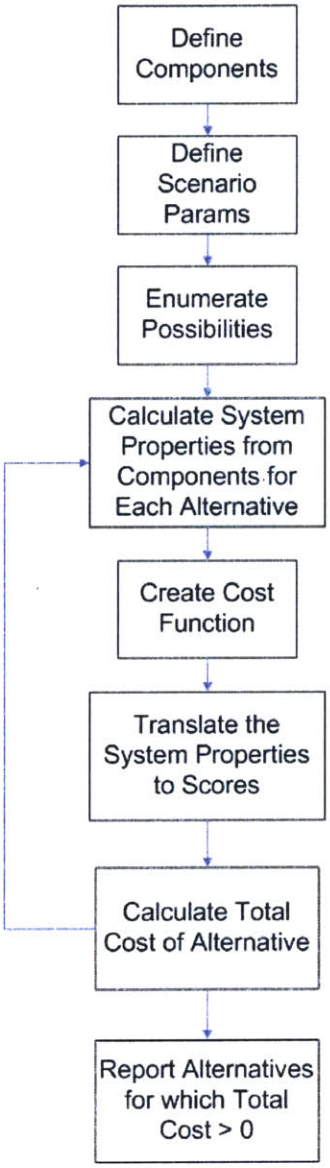


Figure 4-40: Flow chart of the Python scripts used to enumerate and evaluate the possible alternatives.

Table 4.11: Table of the percent contribution to the total cost of each of the design criteria.

Criteria	Weight	Contribution
Weight	1	15.38%
Power	1	15.38%
Size	1	15.38%
Unit Cost	0.5	7.69%
Accuracy	2	30.37%
Latency	0	0%
Robustness	1	15.38%

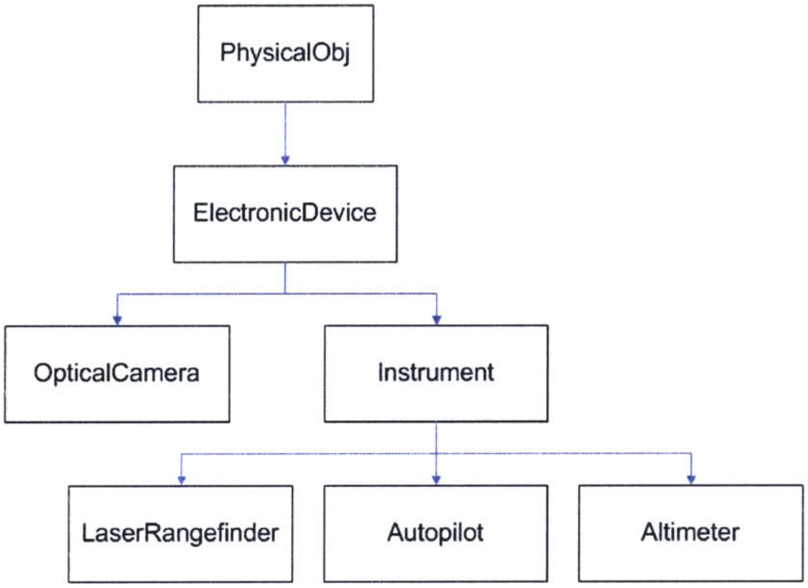


Figure 4-41: Flow chart of the Python scripts used to enumerate and evaluate the possible alternatives.

The first step is to define all of the components in Python. The object-oriented nature of the language was used to construct a class hierarchy in which properties were promoted to the highest level to which they would be common to all child classes. The resulting UML for the classes used to define the components is shown in Figure 4-41.

The top level class is called PhysicalObj. The member variables of the PhysicalObj class are size and mass. This comes from the physics-based definition of an object as anything that has mass and takes up space, as the size criteria was defined as the volume of the object. The size was also a class that contained as member variables the dimensions of the object. Also, the mass was stored as a measure. The measures used in the scripts were perhaps one of the most powerful features of Python. A script was written such that for each type of measure (length, mass, volume, monetary cost, etc), the user could enter the value in any dimension and then work with the value in any other units, just so long as the conversion factor was defined. As a result, the user never had to worry about units.

The next class in the hierarchy was ElectronicDevice, which had properties of power, unitCost, and manufacturer. These were created based on the fact that all electronic devices, regardless of functionality, require power, have some amount of monetary cost, and are created by some company or person. The next level of the hierarchy has two classes, OpticalCamera and Instrument. The OpticalCamera class is the first class of an actual component used in the evaluation process. It had member variables of camera-specific properties, such as resolution and field of view. However, through inheritance from parent classes, it also had all the previously mentioned properties (mass, size, power, unitCost, and manufacturer). Then, the specific camera mentioned in Table 4.4 were simply instances of this class with all of the variables filled in according to the values specified in the table. An additional advantage is that once the specific camera components, or any other type of components, were defined in a file with the properties filled in, they could be imported to any other future project without needing to be recreated. Thus, a database of components could be created simply by creating new instances of the classes and then filling in the variables with the appropriate properties.

The remaining classes in the hierarchy were Instrument, LaserRangefinder, Autopilot, and Altimeter, with the last three classes being children of the first. An instrument was said to be an ElectronicDevice that had the additional variable called error. This was because all instruments are used to take measurements and all measurements inherently have some amount of error in them. The remaining classes then all had variables specific to their functionality.

With the components defined, the next step in the flowchart was to define the scenario parameters. This was done using Python's dictionary functionality. Python contains a data structure called a dictionary that allows the user to define a unique string that serves as the key, or index, into the dictionary used to identify the item and then a definition that is associated with the item. An code snippet of the parameters being defined in the dictionary is shown below.

```
sp = dict();
sp['rng'] = 1200.0;
sp['tgtAlt'] = 50.0;
sp['gndSlopeDeg'] = 2.0;
sp['vehPos'] = Position3D(0.0, 0.0, 500.0);

sp['camEuler'] = EulerAng()
sp['camEuler'].yaw = 45.0 * math.pi/180.0;
sp['camEuler'].pitch = math.acos(sp['vehPos'].z/sp['rng']);
sp['rng'] = (sp['vehPos'].z - sp['tgtAlt'])/
            math.cos(sp['camEuler'].pitch);
sp['tgtAltError'] = 0;
```

There are a couple of interesting features to point out about the code snippet.

First, any variable in the dictionary can be referred to by name. For example, the range can be accessed by

```
sp['rng']
```

Also, two supporting classes were created, Position3D and EulerAng. Position3D had variables of x, y, and z, and EulerAng had variables of yaw, pitch, and roll. This has the advantage of being able to call a value by its name, such as

```
sp['camEuler'].pitch
```

There is no doubt what is being referred to in here whereas if it were done in Matlab without the use of structs, a similar line might be

```
camEuler(2)
```

Next in the flow chart was to enumerate the possibilities. This was done using Python's list data types and nested for loops. As was stated previously, for this design problem, every alternative had a camera, altimeter, and autopilot. The instances of the Range method also had laser rangefinders.

The next four steps in the flow chart contain a loop to indicate that the steps were repeated for all of the alternatives enumerated in the previous step. The first step in this process is to calculate the system's properties from its components. Section 4.6.2 defined the governing equations necessary to determine the system's properties. These equations were then encoded in Python. The properties of the components were then used in the equations. For some of the equations, such as the properties that were simple summations, the calculation is clear. To illustrate this process, the total mass of the system will be calculated for an example. In the example, the components will be the first listed in each table for the camera, altimeter, and autopilot (no laser rangefinder was used). These components are the Sony FCB-EX780B with a mass of 230 g, the Procerus Kestrel autopilot with a mass of 16.65 g, and Roke Manor Mk IVa altimeter with a mass of 400 g. As seen in Equation 4.2, the total system mass is simply the sum of its constituent components, giving the system a mass of $230 + 16.7 + 400 = 646.7g$.

The calculation of the TLE is not so obvious at first look. However, it follows the same pattern. The errors associated with each component were combined with the scenario parameters to form the inputs into the appropriate equations. Again, this will be illustrated through an example, using the same components from before with the Flat Earth method. The Flat Earth method requires knowledge of the x, y, h, ψ, and θ, along with the errors in these values. This was described in the text surrounding Equation 4.28 where it was established that equations developed for the methods were a combination of the actual values and the errors. In the terminology of Equation 4.28, the scenario parameters served as m and the errors served as e, with the addition of the two yielding the desired $\hat{m}$. In the example, the scenario parameter for the vehicle altitude was $500ft$ and the error in the altimeter was $0.410ft$, as seen in Table 4.6. The resulting quantity, $\hat{h}$, was thus $500.410ft$. This was the value that was then substituted into the Flat Earth method's equations for h. While the amount of error in this particular case was small, the alternative had errors of 8° in yaw and $45ft$ in the horizontal position due to the autopilot, so not all error sources were quite so small. With all of the quantities required for the equations established from the components, the TLE was then calculated for the alternative.

With all of the properties, the cost function was encoded in the scripts. This was simply a matter of assigning the weights mentioned above to the design criteria.

The next step requires the translation of the system's properties into scores. The bounds defined in Section 4.3.2 were used to accomplish this translation. Since all of the bounds were continuous functions, the translation of the scores was accomplished by simply using the system property as the independent variable in the appropriate polynomial. The example used previously will be continued to illustrate this process. Section 4.3.2 defined the polynomial function that was used to translate the system's mass into a score. The polynomial that resulted from the *polyfit* in Matlab was a seventh-order equation given by $y = -1.7366 * 10^{-19}x^7 + 5.655 * 10^{-16}x^6 - 7.2504 * 10^{-13}x^5 + 4.5651 * 10^{-10}x^4 - 1.3957 * 10^{-7}x^3 + 1.5261 * 10^{-5}x^2 - 0.00050785x + 1.0004$. The example used above with the Sony camera, Procerus autopilot, and Roke Manor altimeter was shown to have a mass of $646.65g$. Substituting this value in for x results in a score of 0.29. It is important to remember that this is on a scale from 0 to 1. A sanity check of the number reveals that the number makes sense. The requirement stated that the threshold value for the mass was $2lb$ ($907.2g$), with an objective value of $0.5lb$ ($226.8g$). The system's mass is closer to the threshold than the objective, so it make sense that it would be on the lower end of the spectrum. Also, it meets the threshold value, so it make sense that the score is greater than 0.

A similar such process was done to translate all the system's properties into scores except for robustness. As was described in Section 4.6.2, the robustness was calculated by evaluating the methods under both different physical scenarios and different error scenarios. The error was varied according to Table 4.8, which established different error levels. The error levels were none, small, normal, high, and catastrophic, all of which were based upon the values of the errors present in the components, shown in Tables 4.4 - 4.7. Probabilities were assigned to the different error levels. The normal error was assigned the majority of the probability with a value of 0.59. Next were the small and high error, which both had a probability of 0.20. The remaining 0.01 was split between the no error and the catastrophic error levels, giving them 0.005. When they are arranged in the order of increasing error (none, small, normal, high, catastrophic), the probabilities are 0.005, 0.20, 0.59, 0.20, and 0.005. This was meant to very roughly follow the shape of a Gaussian distribution, where the majority of data is contained around the mean, with decreasing probability as the distance from the mean is increased.

As was mentioned in the above paragraph, the physical scenario was also varied. The parameter of the scenario that was varied was the target's altitude. This quantity was given values of $0ft$, $20ft$, $40ft$, $60ft$, $80ft$, and $100ft$. The top range marked a drastic deviation from the Flat Earth's assumption of the vehicle's altitude, as $100ft$ is 20% of the vehicle's altitude. As with the levels of error, the variations in the target's altitude were also assigned probabilities. The probabilities for the altitudes, following the order listed above, were 0.33, 0.30, 0.22, 0.08, 0.05, and 0.02. These

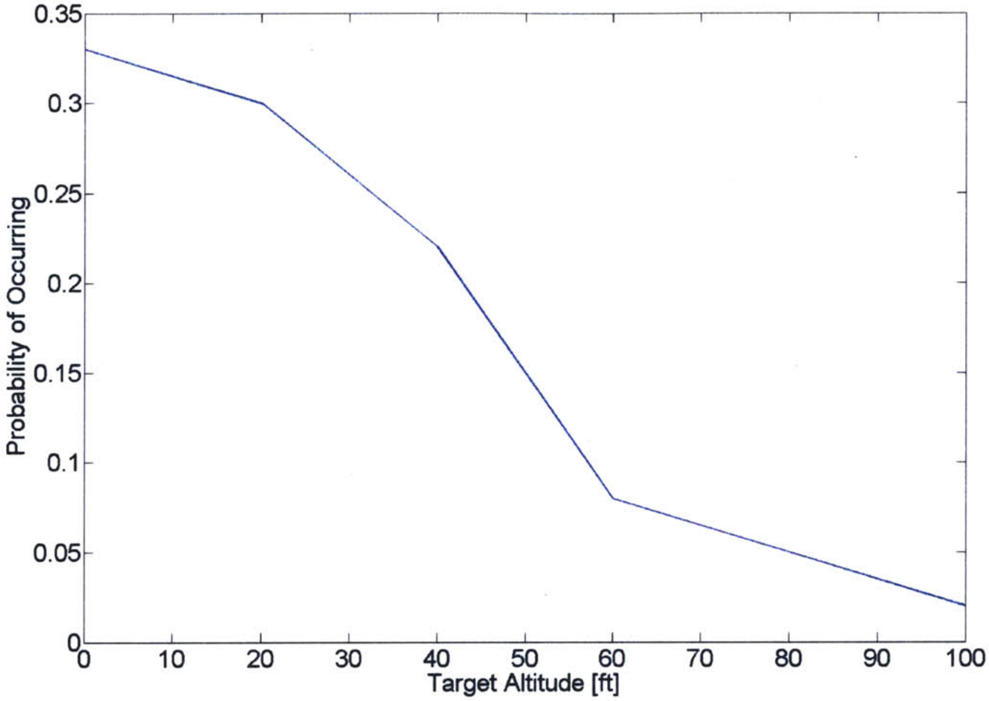


Figure 4-42: Basic shape of the probability of the target altitude as used in the robustness evaluation.

probabilities were picked to follow a basic belief about the probability of how the terrain would change in the given range of the small UAV. The basic shape produced by the probabilities is shown in Figure 4-42. Since the small UAVs have ranges around $10km$, it was assumed that the elevation change would not be too drastically. It was believed that the highest probability for the terrain would be to stay roughly the same. As a result, the highest probability occurs for the target altitude being closest to zero. After a certain change in altitude, it was believed that the probability would began to fall off until it began to asymptotically approach zero. Figure 4-42 tries to roughly capture the shape described as best as could be with six discrete points.

Each error levels was evaluated at each target altitude. That is, the none, small, normal, high, and catastrophic errors were each evaluated with a target altitude of $0ft$, $20ft$, $40ft$, $60ft$, $80ft$, and $100ft$, giving a total of 30 different evaluations. Since the probabilities are independent, the total probability for each of the 30 different scenarios occurring is the probability of the error level multiplied by the probability of the target altitude. As was stated in Section 4.6.2, two different quantities were calculated from this analysis, those being the standard deviation and the average of the TLE for each method. The average was just the summation of the probability of occurring multiplied by the TLE for each of the 30 scenarios. For the standard deviation, however, a vector was created of the TLEs from the scenario. The number of entries into the vector for a given TLE value was proportional to the probability of

Table 4.12: Average error and standard deviation for the three methods.

Method	**Avg Error**	**Std Dev**
Flat Earth	240.79 ft	50.40 ft
DTED	192.02 ft	53.37 ft
Range	107.64 ft	31.68 ft

occurring. The number of entries was calculated by $(p_e \times 100) \times (p_{tgtAlt} \times 100)$. The scenario in which the target's altitude was 0 and the error was normal will be shown as an example. From above, $p_{e=normal} = 0.59$ and $p_{tgtAlt=0} = 0.33$. Plugging these numbers into the above equation yields $59 \times 33 = 1947$. As a result, the TLE for the scenario in which the error was normal and the target altitude was at $0ft$ was put into the vector 1947 times. A total of 10,000 TLE entries were thus included in the vector for each method. The standard deviation for the method was then found by taking the standard deviation of this method. The results of both the average error and standard deviation analysis for the three methods is shown in Table 4.12.

The implications of these results will be discussed in greater detail in Chapter 5. However, the values in the last column of the table are the system properties that were passed into the polynomial function to get the robustness score. It is important to note that this is the one criteria whose score is not tied to the specific components used in an alternative, but instead tied to the method itself. This makes sense, though, as the robustness is a measure of the method, not the components. Hence, the score for robustness should not be tied to the components. Even though the property was calculated in a different way than the others, the same process was used to translate it into a score. The property was simply passed into its respective polynomial and the output was its score.

The final piece of the evaluation was to then substitute the scores for the criteria into the cost function and obtain the total cost for the alternative. Given the simplistic nature of the cost function, this was simply multiplication and addition.

The previous four steps were repeated for all of the alternatives. The final step shown in Figure 4-40 was to report all of the alternatives that met all the requirements, characterized by having a total cost greater than 0. This was done by examining all of the total costs and printing the alternatives that had positive total costs.

Results

As shown in Tables 4.4 - 4.7, there were two cameras, three autopilots, two altimeters, and two laser rangefinders to choose from. The Flat Earth and DTED methods did not need a laser rangefinder, so there were $2cam \times 3autopilot \times 2alt \times 2method = 24$ different choices. Similarly, the Range methodology required the use of all four components, yielding $2cam \times 3autopilot \times 2alt \times 2lrf = 24$ different choices, for a total of 48 different alternatives to evaluate. The results of all 48 evaluations are shown in Appendix B. It should be noted that in the appendix, the Flat Earth method was referred to as height. The entries are sorted by the total cost, with the highest scoring alternative listed first and the lowest scoring alternative listed last. The results include the specific makes and models of the components included in the alternative, the total system properties, and the scores for each of the design criteria, including the total cost obtained when evaluating the cost function.

Discussion of the Top Five Alternatives As Appendix B, none of the possibilities met the specified requirements. In fact, only two possibilities had a total score that was even close to being positive. The two top scores had the same constituent components with the only difference in the alternatives being the methods. The top rated alternative had a score of -8.94 with using the DTED method and the second highest alternative had a score of -10.11 using the Flat Earth method. Both the alternatives used the Sony FCB-EX780B camera, the Honeywell Precision Altimeter HPA for the altimeter, and the Crossbow NAV420 for the autopilot. The two alternatives had three negative scores for the design criteria, occurring in power, mass, and robustness, with power being the most negative. The power requirement set a threshold value of $5W$ and the alternatives required approximately $7W$. This would require a 40% increase in the power allocation to the system. While this is drastic, this might be able to be done, depending on the vehicle and its ability to accept different batteries.

However, the alternatives are also over the mass budget. This is a hard combination to overcome. Usually, greater power could be given to the system by enlarging the batteries. However, if the mass budget is already exceeded, enlarging the batteries would act to exacerbate the problem. The mass is just slightly over the threshold, though, so the alternatives might be feasible on vehicles towards the larger end of the small UAV spectrum. These alternative simply would lot be able to fly on small UAVs that require a mass closer to the objective.

The final criteria that was negative for the scenarios was the robustness. A thorough discussion of the robustness was given in the previous section. As shown in Table 4.12, the standard deviations for the DTED and Flat Earth methods were 53.37 ft and 50.40 ft, respectively. This is just slightly over the 50 ft threshold. Since the robustness is inherent to the method, if a more robust system were required, a different method would need to be used.

A final important note about the two top alternatives is the difference in accuracy. The only difference in the scenarios was the method used to find the solution and DTED had an accuracy of $52.85 ft$ as compared to $160.69 ft$ for Flat Earth, an improvement of over $100 ft$. More discussion will follow on this topic later.

Analyzing the next three highest entries, numbers 3 - 5, show an interesting occurrence, the impact of which will be discussed in greater detail in Chapter 5. The primary reason for the total cost being so low is the score given to the unit cost (the third and fourth alternatives do also have a power problem, but power's contribution is dominated by the unit cost). Like the previous two alternatives, the third and fourth ranked alternatives are the same system, but just have definite methods (again DTED is higher than Flat Earth). As a result, a good comparison can be drawn between the first and third rated scenarios and the second and fourth rated scenarios. In both cases, the lower rate of the two scenarios, three and fourth, are lighter, smaller, more accurate, and require comparable, albeit slightly more, power. However, the cost of the third and fourth scenarios is so much above the threshold, over $6,000, that the huge negative score of the unit cost destroys the benefits of all the other properties. This is the sort of thing that can be shown to the customers to trade the values of the requirements. The designers can show that a superior system could be achieved for a greater cost and then let the customer decide if the added benefit of the system is worth the extra money. As will be discussed in Chapter 5, the extra money added to the budget would profoundly change how the Range alternatives were evaluated.

Method Rankings and Plots of Trends In order to ascertain how the methods ranked in comparison to each other in the evaluation, it is enlighten to track how many are present in the top several alternatives. For instance, in the top ten ranked alternatives, five are DTED, three are Range, and two are Flat Earth. Similarly, in the top 20 ranked alternatives, 10 are DTED, six are Flat Earth, and four are Range. This view of the top 20 is remarkable considering the number of alternatives that were possible for each method. As was stated earlier, there are 12 Flat Earth methods, 12 DTED methods, and 24 Range methods. Incorporating this knowledge into the analysis, 83.33% of the DTED alternatives are in the top 20, 50% of the Flat Earth alternatives are, and only 16.67% of the Range alternatives are present. Thus, it would appear that for the given cost function, the rank order of the methods would be DTED, Flat Earth, and Range.

As a final avenue of analysis, plots were created of several of the design criteria considered against each other. The plots are shown in Figures 4-43 - 4-49. The figures represent the type of trend analysis that was describe in Chapter 2 and earlier in this chapter. As was mentioned in Section 4.7.1, one type of these plots has a performance metric as the dependent variable and another design criteria as the independent. TLE served as the performance metric and in Figures 4-43 - 4-46, the independent variables were mass, power, size, and unit cost, respectively. These figures have the general overview shown in the top part of the figure with a a zoomed in view in bottom half to give a better view of the space where the TLE would be within the threshold value.

Figure 4-46 was also the starting point of another type of common trend analysis plot. Here, the unit cost serves as the independent variable and other criteria serve as the dependent variable. This is much like the CAIV analysis seen in Chapter 1. Figures 4-46 - 4-49 are examples of these types of plots, with TLE, mass, power, and size, respectively, serving as the dependent variables.

In five out of the seven graphs, it is difficult to observe a trend. This could be due to one of two possibilities. The first reason is that a trend might not exist and the second is that a trend does exist but is not easily distinguishable due to the small number of alternatives. Further investigation options on the components would make this answer clearer.

The two plots that seem to have trends present are Figures 4-47 and 4-49. For

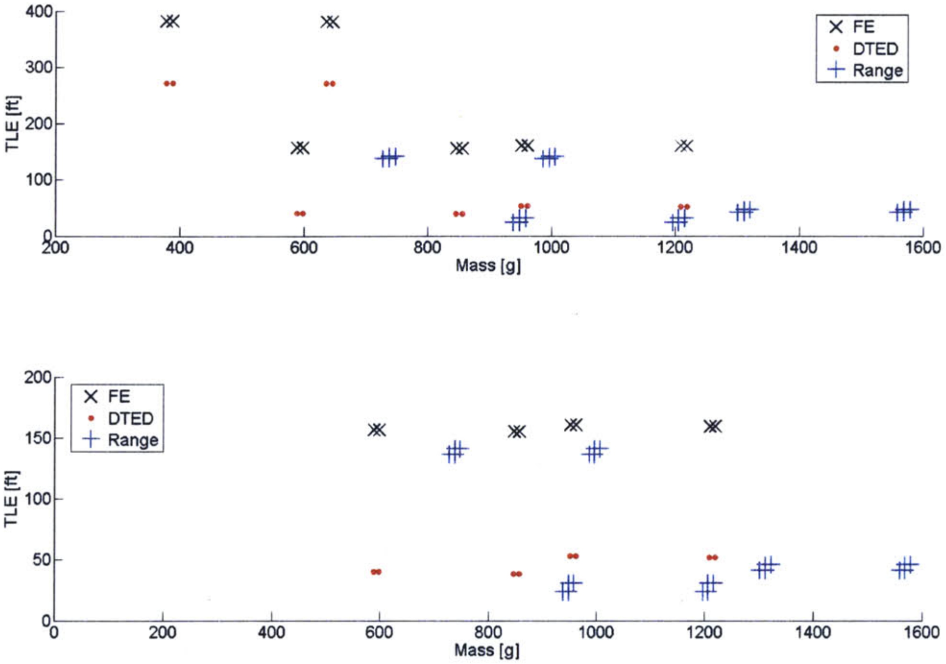


Figure 4-43: Plot of the system's TLE vs. its total mass.

the two figures, the trend is most clearly seen when looking at the Range method alternatives in the plot. In both cases, there appears to be a negatively slope trend indicating that improvement can be achieved in the dependent variable by increasing the unit cost. This means that the system could be made smaller or lighter by spending more money. These plots show one of the fundamental tradeoffs for aviation-based systems. The plots also show that every one of the Range method alternatives is above the threshold value for the unit cost of the system. The implications of this will be discussed in greater detail in Chapter 5.

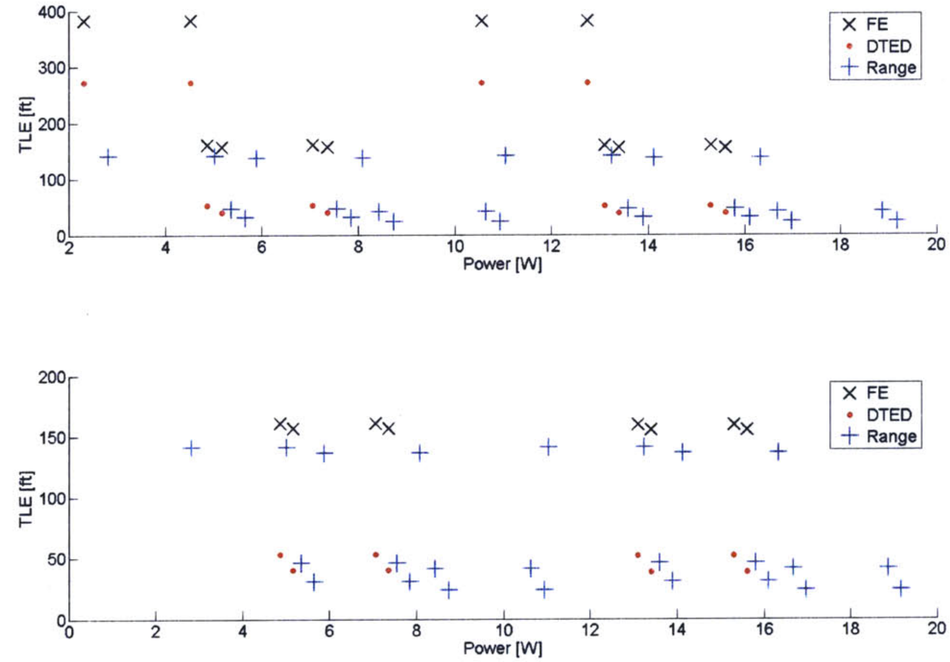


Figure 4-44: Plot of the system's TLE vs. its total power.

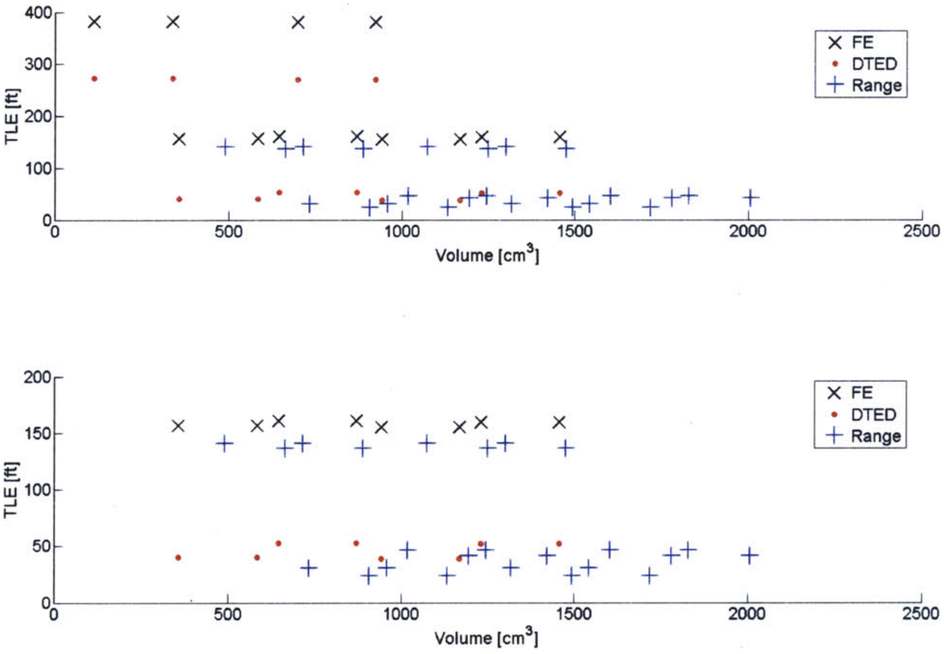


Figure 4-45: Plot of the system's TLE vs. its total size.

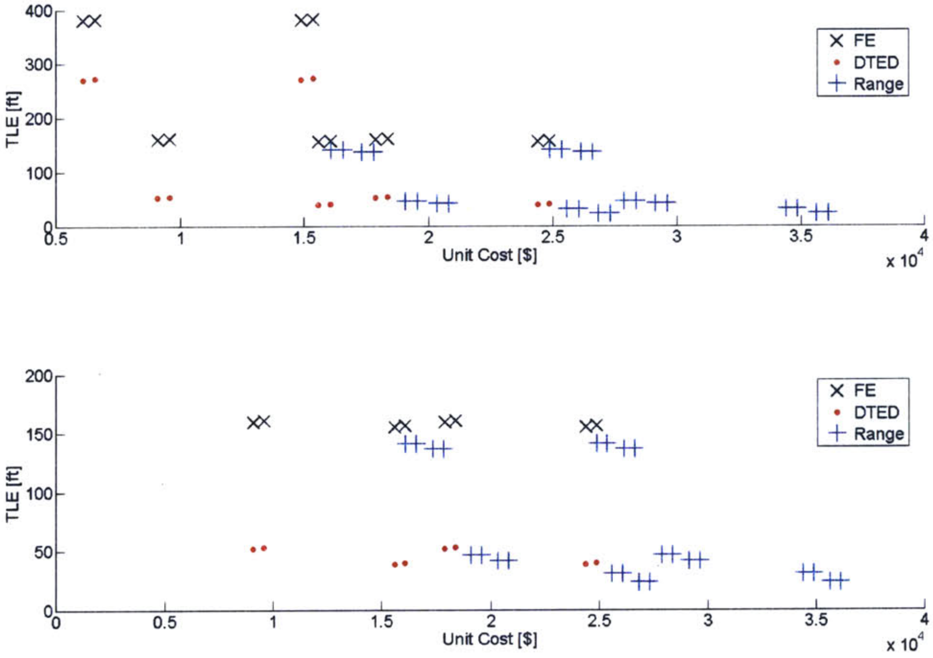


Figure 4-46: Plot of the system's TLE vs. its total unit cost.

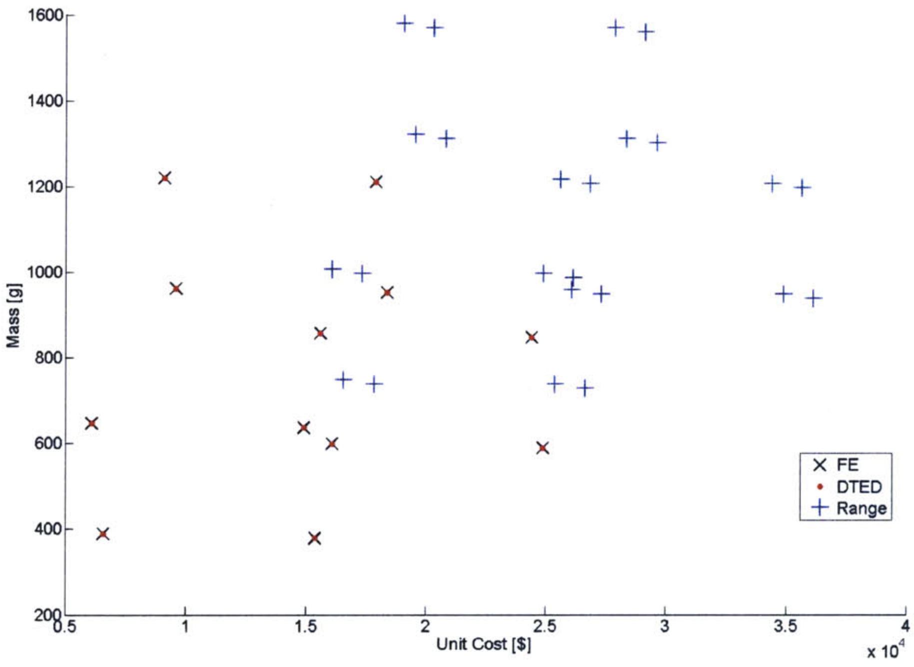


Figure 4-47: Plot of the system's total mass vs. its total unit cost.

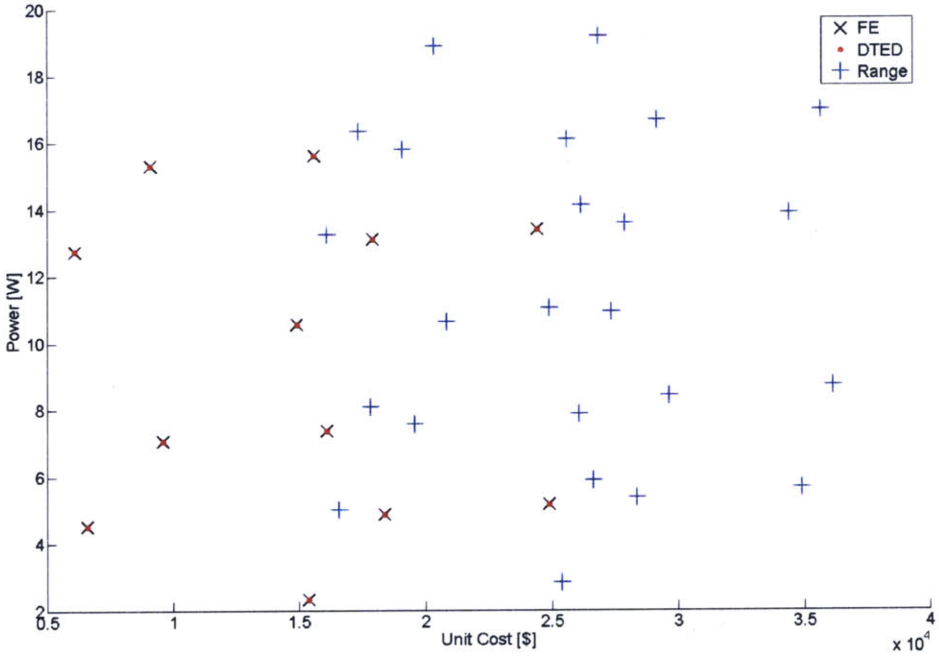


Figure 4-48: Plot of the system's total power vs. its total unit cost.

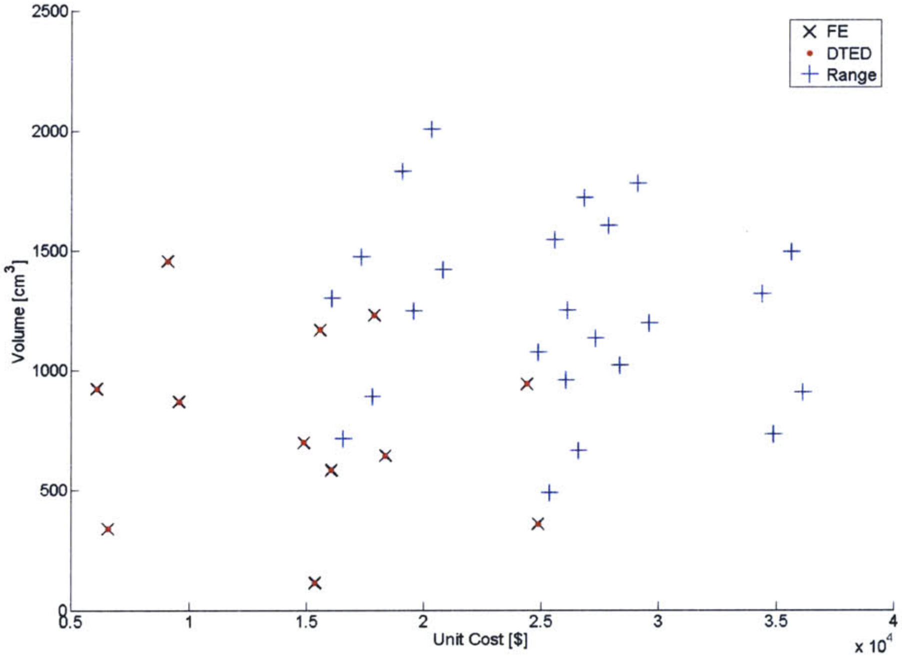


Figure 4-49: Plot of the system's total size vs. its total unit cost.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Conclusions and Future Work

This chapter will discuss the conclusions reached from the application of the design method to the targeting system of a small UAV and future work that could be performed. First, the discussion will focus on the conclusions related to the system design method. Then, the conclusion related to the targeting system will be discussed. After the conclusions are presented, the chapter will finish with discussion on the possible future work.

5.1 System Design Method Conclusions

The design methodology constructed in this thesis demonstrated promise in its first application and seems to fit in its role as an aid for early design. As will be discussed in the next section, the methodology helped demonstrate some key characteristics of the targeting methods and answered some questions related to several critical areas, including their relative effectiveness, their feasibility within the stated requirements, and where to go in the future for targeting design. The system design method also demonstrated that it was much more objective in nature than the matrix-like evaluation and nowhere nears as difficult to construct as an MDO problem. For simple problems or problems without optimization required, the design methodology could be of useful in the state demonstrated here. Complexity of the evaluation methods could be added in incrementally. The beauty of the methodology, however, is that it works well with any level of development, making it unique among the design methodologies examined.

Another important lesson learned in the design process was how effective Python could be as an aid in design. It was able to facilitate the entire problem evaluation, from component property modeling all the way through the enumeration of alternatives and their evaluation. Python was truly the coding-glue that held the entire process together. In the scope of this problem, Python was able to accomplish much of what would typically done in Matlab. The reason that this is significant is because Python is freely distributed while Matlab costs several thousand dollars per license. Of course, Python is no match for Matlab's specialized toolboxes, such as Simulink. However, if designers simply need a scripting language that can perform computations and graphing, Python could be used. Since Python is a high-level language, perhaps even higher than Matlab, it is quite quick for beginners to learn. Given its functionality, free price, and ease of learning, Python might be able to reduce the number of Matlab site licenses needed by a company. This would amount to a huge savings.

5.2 Targeting System Conclusions

The sensitivity analysis showed that the pitch error was the dominate source of error. The upper bounds of Tables 4.9 and 4.10 show that the pitch error is the most restrictive bound, especially in the presence of even just one other error source. While some of the other error bounds could be categorized as loose, the pitch error needed to be under 1° in the certain circumstances. When all the sources that contribute to pitch error are consider, as was seen in Figure 4-28, this becomes a very difficult bound to obtain and dictates, at the very least, the selection of the attitude determination components if not the entire system.

The evaluation conducted in Section 4.8 shows that the targeting system could be achieved with current technology, but not the stated requirements. As a result, either a different targeting method is needed or the requirements must be loosened in some way. The top two rated alternatives seem to suggest an increase in the power budget by $2W$ would allow for a working solution. However, both the alternatives are just on the far side of the mass threshold, showing that they really do not have much design room to play with.

Perhaps the change that would affect the result the most, other than the inclusion of other fundamentally different targeting methods, is in increase in the budget for the system from $10,000 to perhaps $15,000. The increase in budget would most likely have the largest impact on the Range methods. Range alternatives had most of the worst scores, including the final 14 alternatives. Yet, on average, the Range method produced the best accuracies by a wide margin. The reason for the discrepancy between the accuracy of the alternatives and how they were rated was due to the fact that the Range alternatives were far too expensive. While power and mass were of an issue as well (power more so than mass), examining the last 14 alternatives showed that the unit cost dominated the scoring.

In general, it is safe to say that the requirements did not allow for the addition of a component. In order for a laser rangefinder to be included in the system and thereby allow the Range method to become a viable option, the customer would not only need to substantially increase the budgetary restrictions, but they would also most likely need to increase the power budget to compensate for the addition. If the increased financial commitment was deemed worth it and the power was available, the system could be realized. However, the requirements in their current form show what was described in Chapter 1 when it stated that the military has gone away from a solely performance-based system and now also takes into account criteria such as cost. As was shown in the robustness analysis in Table 4.12, the Range method was significantly more accurate and more robust than the other methods. However, as described above, in the actual component evaluation shown in Appendix B, the alternatives rate much worse for the given cost function due to the fact that they are so expensive. In previous times of defense acquisition, the Range methods would most likely have dominated the rankings.

From this analysis, there appears to be a business opportunity for those who could develop a cheap, low power laser rangefinder with comparable performance to those that were considered in the evaluation. A similar business opportunity exists for those that could develop a method that gives the range to the target with the same accuracy as a laser rangefinder without the cost, power, or mass penalties. If such a method were to be developed, it could revolutionize the approach to small UAV targeting, especially if it were software-based and did not have a physical penalty on the system.

Along the lines of a software-based addition to the system with no physical penalties, the analysis in Section 4.8 showed that the DTED method had a lower average TLE than did the Flat Earth method and that it evaluated better than Flat Earth for the given bounds, cost function, and components. Given that the effect of adding the information would be minimal in terms of burden, the evaluation shows that adding DTED information to the system would be beneficial and is thus recommended.

5.3 Future Work

An easy extension to the work that was done in this thesis would be to reevaluate the system with either a different cost function or with modified requirements. As was mentioned above, it would be interesting and informative to the customer what type of results would develop if the monetary budget was increased to \$15,000 and the power budget was increased to $7.5W$. The results from the current evaluation seemed to indicate that the Range method would then dominate, but performing the actual evaluation would confirm this assessment

Another improvement that could be implemented would be to increase the number of alternatives evaluated so as to include the relative-to-landmark class of methods. This avenue was left unexplored by this thesis, but alternatives seemed promising and worthy of further exploration. Another promising avenue that was not explored was the multiple look methods of relative-to-self targeting.

In addition to increasing the breadth of the search, its depth could also be expanded. This could occur in a number of different ways. The first way is to increase the level of fidelity on the models of the methods currently explored. An example would be to further decompose the pointing error to include the factors shown in Figure 4-28. Also, a more thorough evaluation of the robustness of the methods could be created. The new method could use actual DTED data to determine what a typical change of elevation is over the small UAVs operating radius or to construct actual scenarios in which the methods are evaluated that span the gamut of terrain options (mountainous, flat, hilly, edge of a cliff, etc). Another increase in fidelity would be to evaluate the effects of having the different levels of DTED separately instead of just lumping them together and tracking the target altitude error, as was currently done.

Similarly, instead of using a known value and some set amount of error, a probability distribution for the error could be developed. Also, further examination of the COTS market could be done to increase the number of components used in the evaluation, thereby hopefully making the trends of the design criteria more easily visible. A final source of future fidelity that could be added is the use of optimization techniques such as non-linear programming. This would involve encoding the requirements as constraints and trying to find an optimal solution. A corollary bit of future work related to optimization could be to develop a way to extract the properties of the components that would produce an optimal solution, yet still be based in reality.

Another general improvement that could be made is to further develop the Python code. For instance, a GUI could be written for the tool to make it more user-friendly. Also, additional classes could be written to expand the functionality so that it is of more general use instead of being tied to the design of just a targeting system to small UAVs.

Appendix A

Product Specification Sheets

This Appendix contains the company-produced specifications sheet for all of the products included in the system evaluation. It also contains the URLs at which the spec sheets were found. These URLs were current at the time of writing, but are of course subject to change.

A.1 Optical Cameras

- Sony FCB-EX780B - Spec sheet taken from http://www.sony.net/Products/ISP/pdf/catalog/2003/FCB_EXB.pdf.
- Rockwell Scientific High Resolution Visible Imager - Spec sheet taken from https://peoiewswebinfo.monmouth.army.mil/portal_sites/IEWS_Public/RUS/sensorcat/PDF/HighResVisImg-Rockwell.PDF.

SONY®

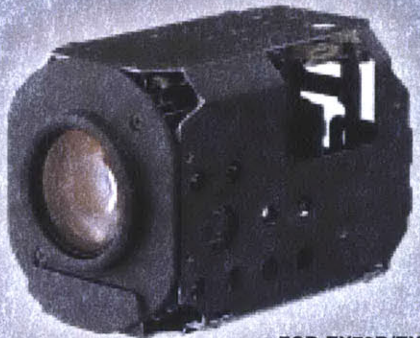
COLOR BLOCK CAMERAS

FCB-EX Series

FCB-EX780B/EX780BP
FCB-EX78B/EX78BP
FCB-EX480B/EX480BP
FCB-EX48B/EX48BP

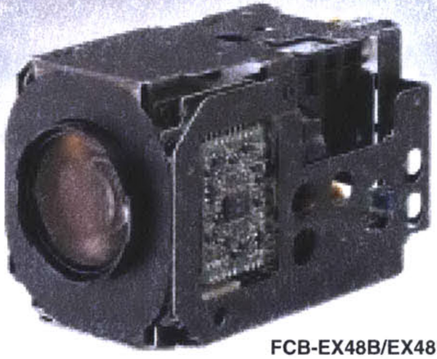
Component/OEM

FCB-EX780B/EX780BP



FCB-EX78B/EX78BP

FCB-EX480B/EX480BP



FCB-EX48B/EX48BP

OUTLINE

Sony's new FCB-EX Series of color block cameras represent an evolution in security dome, police vehicle and traffic monitoring applications. Incorporating familiar and convenient features such as Spot AE, Auto ICR (IR Cut filter Removal)[*1], quick camera control via a high-speed serial interface (max. 38.4 Kb/s), and various customizable settings, these FCB-EX cameras offer superb flexibility and easy operation.

The FCB-EX cameras are equipped with new and unique surveillance features such as an E-Flip function that electrically flips the picture for correct image display and an alarm function that enables changes to be detected within any given area of the picture. In addition, these FCB-EX cameras feature an advanced Privacy Zone Masking function for sophisticated masking controls - a necessity in many surveillance applications. With these new and unique features, the FCB-EX cameras are the ideal choice for indoor and outdoor dome applications.

Incorporating high-performance Digital Signal Processing (DSP), the FCB-EX Series offers greatly enhanced picture quality and operability compared to conventional block cameras. Moreover, all of these FCB-EX cameras use lead-free solder and halogen-free mounting boards, and achieve low power consumption. The dimensions and mounting-hole position of these new FCB-EX cameras are exactly the same as those of earlier FCB-EX Series cameras[*2] allowing them to be easily interchanged. Combining superb picture quality and a variety of unique and convenient features, the new FCB-EX cameras are the perfect match for demanding security dome applications.

*1: FCB-EX780B/EX780BP and FCB-EX480B/EX480BP only
*2: FCB-EX780S/EX780SP, FCB-EX480A/EX480AP and FCB-EX48A/EX48AP

FEATURES

- Advanced Privacy Zone Masking function (max. 24 masking blocks)
- E-Flip function
- Newly developed Alarm function
- Picture freeze during zoom, focus, preset and lens initializing
- Key switch connector (CN701) and DC/video connector (CN903)[*1]
- High-performance Digital Signal Processing (DSP)
- High-speed serial interface (max. 38.4 Kb/s) and TTL signal level control (VISCA™ protocol)
- Auto ICR (IR Cut filter Removal)[*2]
- Image stabilizer[*3]
- Various customizable settings
- Internal/External sync
- Low power consumption (1.6 W with motors inactive)
- EEPROM backup system without battery
- Lead-free solder and halogen-free mounting boards

*1: DC/video connector (CN903): FCB-EX480B/480BP and FCB-EX48B/EX48BP only
*2: FCB-EX780B/EX780BP and FCB-EX480B/EX480BP only
*3: FCB-EX780B/EX780BP and FCB-EX78B/EX78BP only

FEATURES DESCRIPTION

Advanced Privacy Zone Masking

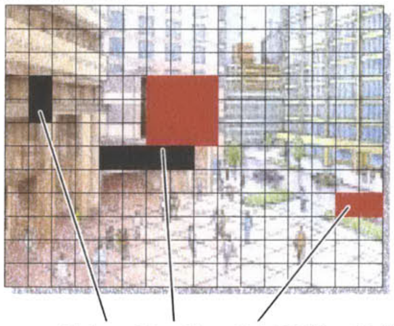
With the advanced Privacy Zone Masking function, unwanted or prohibited areas within an image can be masked appropriately. A maximum of 24 masking areas can be preset to any of 160 horizontal and 120 vertical masking blocks, and up to eight block positions with two colors can be selected together. In addition when zoom is engaged, the size of a masking areas will adjust in proportion to the zoom position. The masking position can be made to interlock with a security dome's pan/tilt to achieve a comprehensive masking operation. For the user convenience, a coordinate grid can be superimposed to easily locate the masking position.

Masking Controls

Maximum number of preset masking blocks	24 blocks
Maximum number of masking blocks to be displayed	8 blocks
Resolution of masking blocks	160 (H) x 120 (V)
Interlock with zoom	Yes
Interlock with pan/tilt [*1]	Yes
Maximum colors to be preset	28 [*2]
Maximum colors to be displayed	2 [*2]
Translucent masking	Yes
Gray scale	Yes [*3]
Individual masking On/Off	Yes
Interlock speed setting [*1]	Yes
Title/day/time superimposition	Yes

[*1] When used in a security dome [*2] Including translucent color [*3] 6 gradations selectable

Masking image

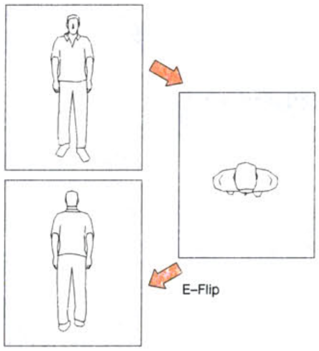


Maximum 24 positions within 160 (H) x 120 (V) can be preset.
(8 positions/2 colors)

Original image

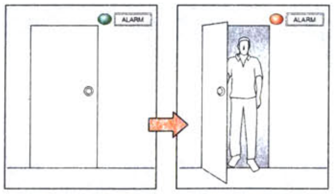
E-Flip (Electrical Flip)

The FCB-EX cameras have an E-Flip function that electrically flips an image upside down. In a dome application for example, if a tracked object moves beneath the camera dome, the image can be inverted to maintain the correct display. This E-Flip function realizes higher mechanical reliability as compared to a mechanical flip function.



Alarm Function

The FCB-EX Series provide a new Alarm function which can detect changes within an area of the picture designated by the user. When a change in AF, AE or both AF and AE is detected, the camera outputs an alarm trigger signal to the external equipment via the VISCA protocol. In combination with the Spot AE, these cameras also detect changes of luminance level, and output an alarm signal. The detecting area can be applied to any of 16 vertical and 16 horizontal blocks.



Day/Night Mode

The FCB-EX cameras feature a new "Day/Night" alarm function. These cameras can output an alarm signal via the VISCA protocol in response to a change in the designated brightness/darkness level. In outdoor dome applications for example, a control center can be instructed to turn lights on and off when a Day/Night alarm is received.

Picture Freeze

The FCB-EX Series is equipped with Picture Freeze function which outputs a freeze-frame picture or a muted picture during zoom, focus, lens initializing, and preset operations. In a dome application for example, if a user does not want to show an image while panning from "A" to "B" points, the camera outputs the "A" point freeze-frame image and then outputs the normal picture once panning is completed.

EXview HAD CCD™

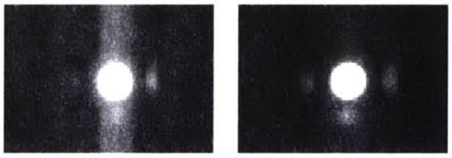
(FCB-EX480B/EX480BP and FCB-EX48B/EX48BP)

The adoption of Sony EXview HAD CCD technology improves basic camera features, providing advantages over earlier FCB models such as superb sensitivity of 0.7 lx (typical), low smear levels and D-range.

IR sensitivity

0.7 lx 3.0 lx

Smear



0.7 lx 3.0 lx

Auto ICR (IR Cut filter Removal) Mode

(FCB-EX780B/EX780BP and FCB-EX480B/EX480BP)

The Auto ICR function automatically switches the settings to attach or remove the IR Cut filter for increased sensitivity. With a set level of darkness, the IR Cut filter is automatically disabled (ICR ON), and the infrared sensitivity is increased. With a set level of brightness, the filter is automatically enabled (ICR OFF). The ICR automatically engages depending on the ambient light, allowing the cameras to be effective in day and night environments.

When auto slow shutter is on

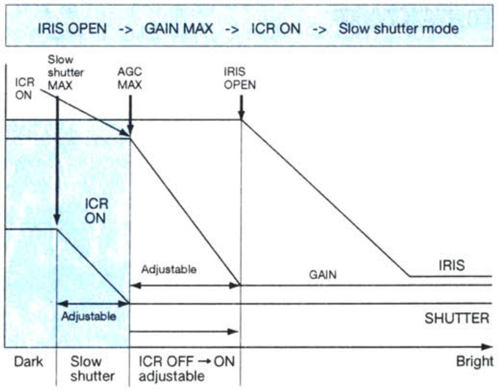


Image Stabilizer

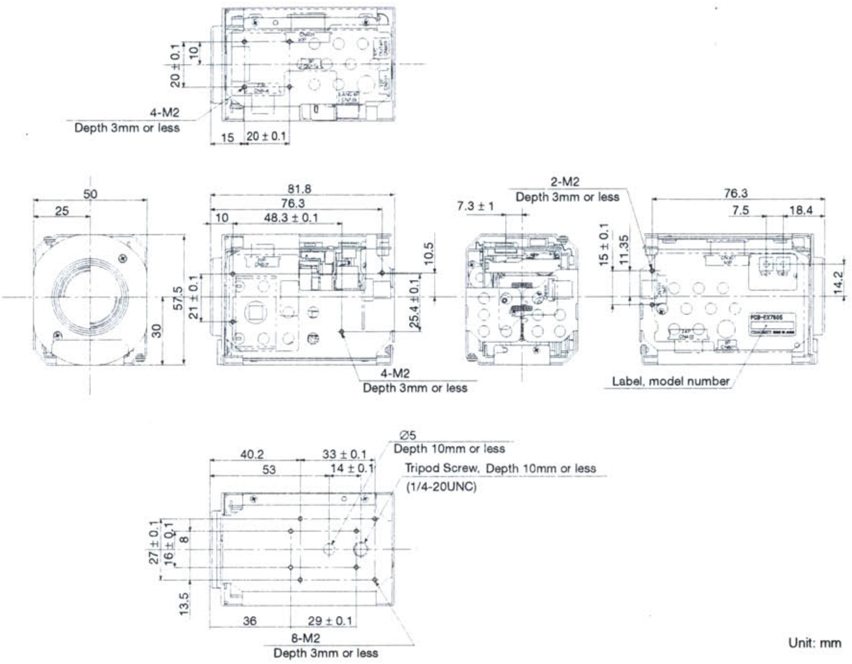
(FCB-EX780B/EX780BP and FCB-EX78B/EX78BP)

The image stabilizer function minimizes the appearance of shaky images caused by low-frequency vibration and maintains a normal horizontal resolution. This function is useful for outdoor surveillance and traffic monitoring applications.

FCB SERIES LINE-UP

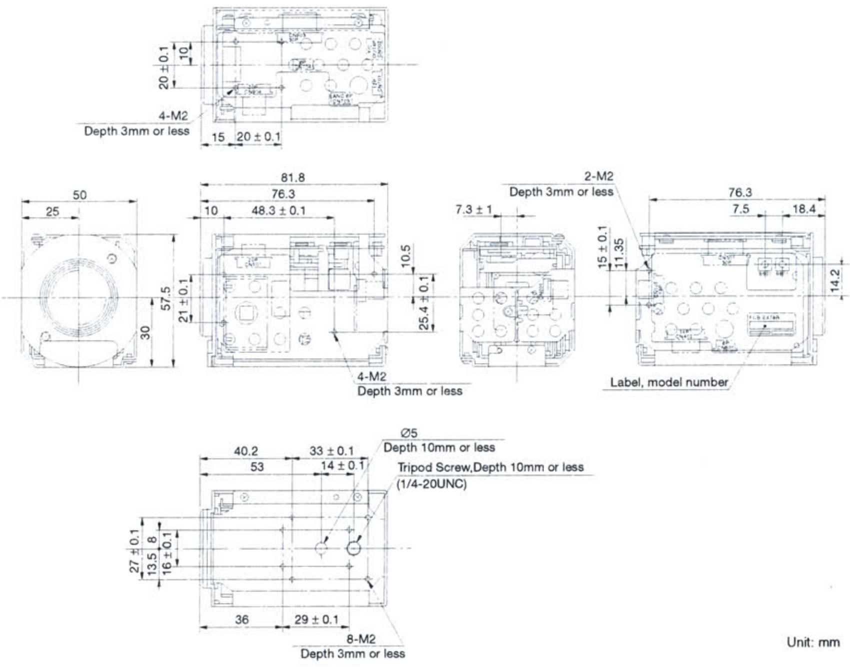
FCB-EX780B/EX780BP

- 1/6 type Super HAD CCD
- 300x zoom ratio (25x optical, 12x digital)
- Auto ICR (IR Cut filter Removal) mode to achieve near-infrared sensitivity
- Image stabilizer
- Advanced Privacy Zone Masking
- E-Flip function
- Alarm function
- Picture Freeze
- Key switch connector (CN701)
- Spot AE
- Electronic shutter/slow shutter
- High-speed serial interface (maximum 38.4 Kb/s) and TTL signal-level control (VISCA protocol)
- Internal/External sync



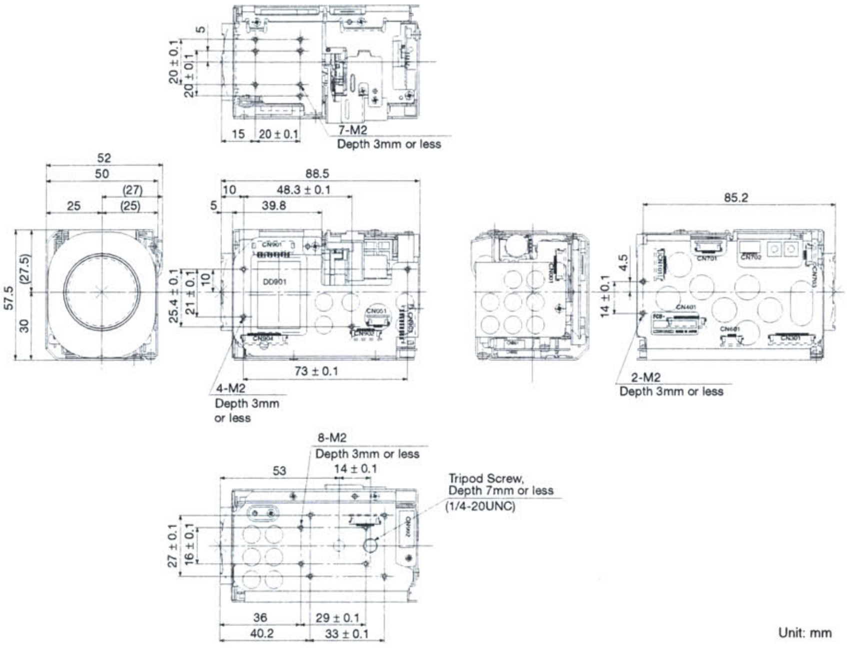
FCB-EX78B/EX78BP

- 1/6 type Super HAD CCD
- 300x zoom ratio (25x optical, 12x digital)
- Image stabilizer
- Advanced Privacy Zone Masking
- E-Flip function
- Alarm function
- Picture Freeze
- Key switch connector (CN701)
- Spot AE
- Electronic shutter/slow shutter
- High-speed serial interface (maximum 38.4 Kb/s) and TTL signal-level control (VISCA protocol)
- Internal/External sync



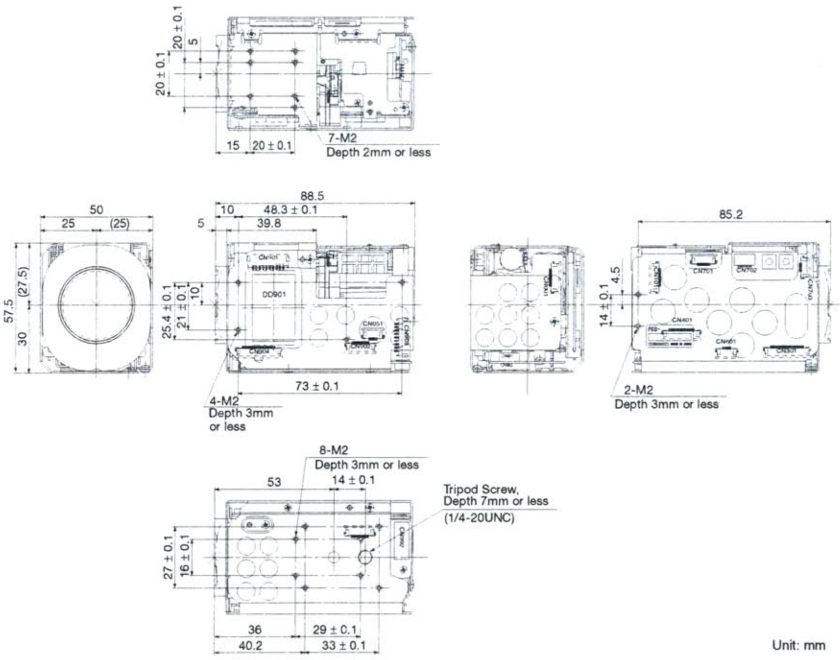
FCB-EX480B/EX480BP

- 1/4 type EXview HAD CCD
- Extremely low minimum illumination of 0.7 lx (typical)
- Auto ICR (IR Cut filter Removal) mode to achieve near-infrared sensitivity
- 216x zoom ratio (18x optical, 12x digital)
- Advanced Privacy Zone Masking
- E-Flip function
- Alarm function
- Picture Freeze
- Key switch connector (CN701) and DC/video connector (CN903)
- Spot AE
- Electronic shutter/slow shutter
- High-speed serial interface (maximum 38.4 Kb/s) and TTL signal-level control (VISCA protocol)
- Internal/External sync



FCB-EX48B/EX48BP

- 1/4 type EXview HAD CCD
- Extremely low minimum illumination of 0.7 lx (typical)
- 216x zoom ratio (18x optical, 12x digital)
- Advanced Privacy Zone Masking
- E-Flip function
- Alarm function
- Picture Freeze
- Key switch connector (CN701) and DC/video connector (CN903)
- Spot AE
- Electronic shutter/slow shutter
- High-speed serial interface (maximum 38.4 Kb/s) and TTL signal-level control (VISCA protocol)
- Internal/External sync



SPECIFICATIONS

	FCB-EX780B	FCB-EX78B	FCB-EX480B	FCB-EX48B
Image sensor	1/6 type Super HAD CCD		1/4 type EXview HAD CCD	
Number of effective pixels	Approx. 680,000 pixels		Approx. 380,000 pixels	
Lens	25x zoom, f=2.4 mm (wide) to 60 mm (tele), F1.6 to F2.7		18x zoom, f=4.1 mm (wide) to 73.8 mm (tele), F1.4 to F3.0	
Digital zoom	12x (300x with optical zoom)		12x (216x with optical zoom)	
Angle of view (H)	45 ° (wide end) to 2.0 ° (tele end)		48 ° (wide end) to 2.8 ° (tele end)	
Minimum working distance	35 mm (wide end) to 800 mm (tele end)			
Sync system	Internal/External (V-Lock)			
Minimum illumination	2.5 lx (typical) (50 IRE)		0.7 lx (typical) (50 IRE)	
S/N ratio	49 dB		More than 50 dB	
Electronic shutter	1/1 to 1/10,000 s, 22 steps			
White Balance	Auto, ATW, Indoor, Outdoor, One-push, Manual			
Gain	Auto/Manual (-3 to 28 dB, 2 dB steps)			
AE control	Auto, Manual, Priority mode, Bright, EV compensation, Back-light compensation			
EV compensation	-10.5 to +10.5 dB (1.5 dB steps)			
Back-light compensation	On/Off			
Privacy Zone Masking	On/Off (24 positions)			
Flicker cancel	Auto			
Focusing system	Auto (Sensitivity: normal, low), One-push AF, Manual, Infinity, Interval AF, Zoom Trigger AF			
Picture effect	E-Flip, Neg. Art, Black & White, Mirror Image			
Camera operation switch	Zoom tele, Zoom wide			
Video output	VBS: 1.0 Vp-p (sync negative), Y/C Output			
Camera control interface	VISCA (TTL signal level), baud rate: 9.6 Kb/s, 19.2 Kb/s, 38.4 Kb/s, Stop bit: 1/2 selectable			
Storage temperature	-20° C to 60° C (-4° F to 140° F)			
Operating temperature	0° C to 50° C (32° F to 122° F)			
Power consumption	6 V to 12 V DC, 1.6 W (motors inactive) 2.7 W (motors active)		6 V to 12 V DC, 1.6 W (motors inactive) 2.5 W (motors active)	
Mass	Approx. 230 g (8.1 oz)			
Dimensions (W x H x D)	50 x 57.5 x 81.8 mm (2 x 2 3/8 x 3 1/4 inches)		52 x 57.5 x 88.5 mm (2 1/8 x 2 3/8 x 3 1/2 inches)	50 x 57.5 x 88.5 mm (2 x 2 3/8 x 3 1/2 inches)

	FCB-EX780BP	FCB-EX78BP	FCB-EX480BP	FCB-EX48BP
Image sensor	1/6 type Super HAD CCD		1/4 type EXview HAD CCD	
Number of effective pixels	Approx. 800,000 pixels		Approx. 440,000 pixels	
Lens	25x zoom, f=2.4 mm (wide) to 60 mm (tele), F1.6 to F2.7		18x zoom, f=4.1 mm (wide) to 73.8 mm (tele), F1.4 to F3.0	
Digital zoom	12x (300x with optical zoom)		12x (216x with optical zoom)	
Angle of view (H)	45 ° (wide end) to 2.0 ° (tele end)		48 ° (wide end) to 2.8 ° (tele end)	
Minimum working distance	35 mm (wide end) to 800 mm (tele end)			
Sync system	Internal/External (V-Lock)			
Minimum illumination	2.5 lx (typical) (50 IRE)		0.7 lx (typical) (50 IRE)	
S/N ratio	49 dB		More than 50 dB	
Electronic shutter	1/1 to 1/10,000 s, 22 steps			
White Balance	Auto, ATW, Indoor, Outdoor, One-push, Manual			
Gain	Auto/Manual (-3 to 28 dB, 2 dB steps)			
AE control	Auto, Manual, Priority mode, Bright, EV compensation, Back-light compensation			
EV compensation	-10.5 to +10.5 dB (1.5 dB steps)			
Back-light compensation	On/Off			
Privacy Zone Masking	On/Off (24 positions)			
Flicker cancel	–			
Focusing system	Auto (Sensitivity: normal, low), One-push AF, Manual, Infinity, Interval AF, Zoom Trigger AF			
Picture effect	E-Flip, Neg. Art, Black & White, Mirror Image			
Camera operation switch	Zoom tele, Zoom wide			
Video output	VBS: 1.0 Vp-p (sync negative), Y/C Output			
Camera control interface	VISCA (TTL signal level), baud rate: 9.6 Kb/s, 19.2 Kb/s, 38.4 Kb/s, Stop bit: 1/2 selectable			
Storage temperature	-20° C to 60° C (-4° F to 140° F)			
Operating temperature	0° C to 50° C (32° F to 122° F)			
Power consumption	6 V to 12 V DC, 1.6 W (motors inactive) 2.7 W (motors active)		6 V to 12 V DC, 1.6 W (motors inactive) 2.5 W (motors active)	
Mass	Approx. 230 g (8.1 oz)			
Dimensions (W x H x D)	50 x 57.5 x 81.8 mm (2 x 2 3/8 x 3 1/4 inches)		52 x 57.5 x 88.5 mm (2 1/8 x 2 3/8 x 3 1/2 inches)	50 x 57.5 x 88.5 mm (2 x 2 3/8 x 3 1/2 inches)

* When used continuously for more than 24 hours, it is recommended to initialize the lens system every 24 hours to extend the life of the lens. The 'Initialize Lens' command takes a little less than 3 seconds to initialize the focus and zoom.

SPECIFICATION COMPARISON CHART

	FCB-EX780B/EX780BP	FCB-EX78B/EX78BP	FCB-EX480B/EX480BP	FCB-EX48B/EX48BP
Lens	25x		18x	
Image stabilizer	Yes		No	
CCD	Super HAD CCD		EXview HAD CCD	
Sensitivity	2.5 lx		0.7 lx	
ICR	Yes	No	Yes	No
Dimensions (W x H x D)	50 x 57.5 x 81.8 mm (2 x 2 3/8 x 3 1/4 inches)		52 x 57.5 x 88.5 mm (2 1/8 x 2 3/8 x 3 1/2 inches)	50 x 57.5 x 88.5 mm (2 x 2 3/8 x 3 1/2 inches)

PIN ASSIGNMENT

■ CN901 --- 9-pin for DC/video out/VD-Lock Pulse/VISCA

Pin No.	Name	Level
1	RxD	CMOS 5.0 V (low: max. 0.8 V, high: min. 2.0 V) Read Data
2	TxD	CMOS 5.0 V (low: max. 0.1 V, high: min. 4.4 V) Send Data
3	GND (for RxD & TxD)	
4	DC IN	9.0 ± 3.0V
5	GND (for DC IN)	
6	VBS OUT	1.0 ± 0.2 V
7	GND (for VBS OUT)	
8	V LOCK PULSE	External VD-Lock Pulse (EX.FV: Negative, 3.0 Vp-p 50% duty)
9	GND (VL PULSE)	

Connector: ELCO 00 6200 509 13000

■ CN902 --- 4-pin for Y/C video out

Pin No.	Name
1	Y_Out
2	GND (for Y signal)
3	C_Out
4	GND (for C signal)

Connector: JST S4B-ZR-SM3A-TF

■ CN903 --- 9-pin for DC/video out

Pin No.	Name	Level
1	DC IN	6.0 V to 12.0 V
2	GND (for DC IN)	
3	NC	
4	VBS OUT	Composite video signal
5	GND (for VBS OUT)	
6	Y_OUT	1.0 ± 0.2 V
7	GND (for Y signal)	
8	C_OUT	
9	GND (for C signal)	

Connector: JST S9B-ZR-SM3A-TF

■ CN701 --- 12-pin for Key Switch control

Pin No.	Name
1	GND
2	GND
3	KEY_AD0
4	KEY_AD1
5	KEY_AD2
6	KEY_AD3
7	KEY_AD4
8	KEY_AD5
9	KEY_AD6
10	KEY_AD7
11	NC
12	Strobe

Connector: Molex 52689-1240 FFC (0.5 mm)

- **Sony Electronics Inc. (USA) HQ** 1 Sony Drive, Park Ridge, NJ 07656 (TEL:+1-800-686-7669)
http://www.sony.com/videocameras
- **Sony of Canada Ltd. (CANADA)** 115 Gordon Baker Rd, Toronto, Ontario M2H 3R6 (TEL:+1-416-499-1414) (FAX:+1-416-497-1774)
- **Sony Broadcast & Professional Europe HQ** Schipholweg 275 1171 PK Badhoevedorp The Netherlands (TEL:+31-20-44-99-351) (FAX:+31-20-44-99-333)
http://www.sony-vision.com

Germany Hugo-Eckener-Strasse 20 D-50829 Koln (TEL:+49-221-537-8923) (FAX:+49-221-537-491)
France 16-26 rue Morel 92110 Clichy (TEL:+33-1-55-90-41-58) (FAX:+33-1-55-90-42-20)
UK The Heights, Brooklands, Weybridge, Surrey KT13 0XW (TEL:+44-990-331122) (FAX:+44-1932-817011)
Nordic Per Albin Hanssons vag 20 S-214 32 Malmo Sweden (TEL:+46-40-190-800) (FAX:+46-40-190-450)
Italy Via Galileo Galilei 40 I-20092 Cinisello Balsamo, Milano (TEL:+39-02-61-83-84-26) (FAX:+39-02-618-38-402)

MK10023V1IW03FEB

Printed in Japan on recycled paper.

High Resolution Visible Imager

ROCKWELL SCIENTIFIC

Date Revised: 17 JAN 03

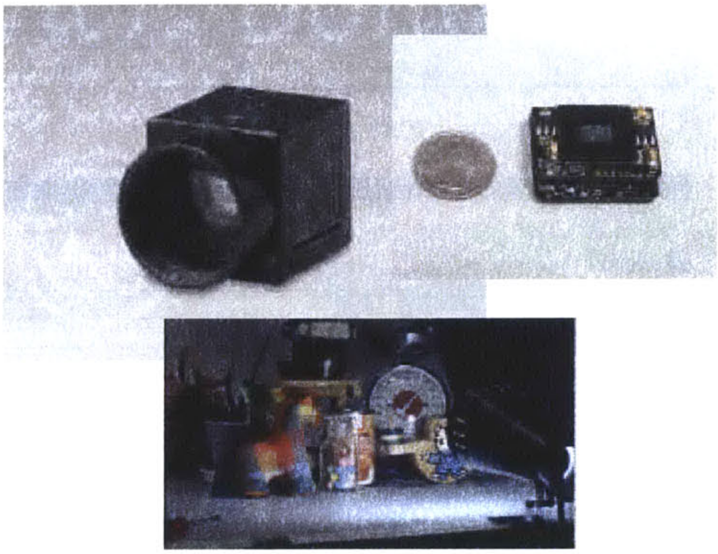
VENDOR DESCRIPTION

RSC has developed a compact, low power, high resolution camera for applications on unmanned platforms. The camera is based on a commercially developed CMOS imager.

The CMOS imaging chips include SOC functional integration and consume 180 mW. Comparable CCD imagers with separate clock and A/D chips consume 2.5 W.

Chip format is HDTV-compatible (2K x 1K) and can be provided as a panchromatic camera or with integrated RGB color filters. Integrated micro-lenses enable high quantum efficiency. Multiple digital or analog interfaces are available.

Selected area readout plus compression enable operation in limited communication bandwidth systems.



Product Manager
Robotic & Unmanned Sensors
Telephone: (732) 427-5827 / DSN 987
Fax: (732) 427-5072 / DSN 987
e-mail: SFAE-IEWS-NV-RUS@iews.monmouth.army.mil

EOIR

Electro-Optical (EO)
Type: CMOS
Resolution: 1936 x 1090, 5 micron pixels
Noise electrons: 25
12 bit integral A/D
Angular Coverage: Can accommodate various commercial lenses
Modes of Operation: 30 Hz Progressive scan
Selective windowing readout
Rolling shutter
Field of View: Can accommodate various commercial lenses
Sensitivity: 0.2-1.0 microns (unfiltered)
Color or B/W: Both

Hardware
Power: 500 mW (Camera), 180 mW (CMOS chip)
Weight: 220 grams (including 3mm lens)
Dimensions: 37mm x 31mm x 25mm

A.2 Altimeters

- Roke Manor Miniature Radar Altimeter (MRA) MkIVa - Spec sheet taken from www.roke.co.uk/download/datasheets/MRA_Mk4a.pdf.
- Honeywell Precision Barometer (HPB) - Spec sheet taken from www.ssec.honeywell.com/pressure/datasheets/hpb.pdf.

Roke
Manor
Research

A Siemens Company

ROKE MANOR
50
YEARS OF INNOVATION
1956 - 2006

Miniature Radar Altimeter Mk IVa

A light-weight, highly accurate, low cost altimeter

Our Miniature Radar Altimeter (MRA) Mk IVa is a revolutionary product designed for the Airborne/ Aerial Target and Unmanned Air Vehicle (UAV) market and is based on our expertise in radar design for the defence sector. Our radar experience ranges from multi-function electronically scanned array radars, to miniature solid state radar sensors for both military and commercial vehicles.

The miniature radar altimeter enjoys huge success with orders received worldwide. We have signed major contracts with Flight Refuelling Limited, a major supplier to the aerospace and defence industries, to provide up to 50 miniature radar altimeters per annum for use in towed aerial targets in air defence training.

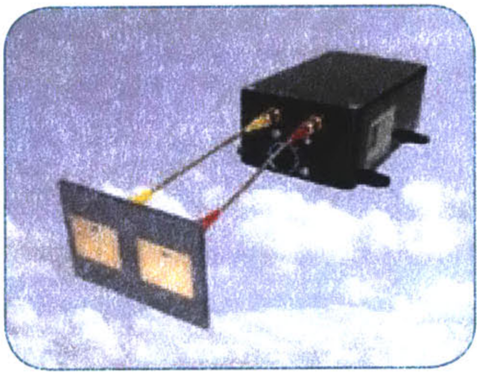
Key Features

- Provides low level height measurement (from 1.5m to 700m)
- Highly accurate
- Ideally suited for on-board small to large UAVs
- Light weight
- Low cost

Applications

Although specifically designed for the unmanned air vehicles market, it has many other applications some of which include:

- Geophysical surveying
- Wave height monitoring
- Airborne filming
- Traffic monitoring



Technical Specification

Parameter	Value	Notes
Maximum operating altitude	700m	Dependent on terrain*
Minimum operating altitude	1.5m	Dependent on airframe, 5m default
Power Output	<100mW cw	
Altitude accuracy	12.5cm	14 bit height value with height in 12.5 cm steps. The system has a higher accuracy mode (figure quoted at left) when at an altitude of less than 100m.
Maximum horizontal velocity of air vehicle	300m/s	
Maximum acceleration	10g	Applied in any axis
Temperature range	-40 to +70°C	
Power supply (Typical)	28V @ 0.3A	Generator or battery
Power supply (Range)	9V - 32V	
Size	150x80x54mm	Excludes antenna fittings*
Weight	400 gramme	
Interface	RS 232	Update rate is 10 times per second
Operating frequency	4.3 GHz	Jamming resistance options
Environmental conditions		Up to 16mm/hour rainfall

Honeywell

Honeywell Precision Barometer HPB

The Honeywell Precision Barometer (HPB) offers outstanding value to instrument builders requiring accurate and stable barometric measurements in real-world conditions. The HPB uses proven silicon sensor technology with microprocessor-based signal compensation, eliminating the need to insulate or temperature-regulate the barometer. The HPB has a pressure range of 500 to 1200 hPa. The HPA, intended for altimeter applications, provides a pressure range of 0 to 17.6 psia.

APPLICATIONS:

- AWOS Weather Systems
- Remote Meteorological Stations
- Ocean Data Buoys
- Environmental Data Logging
- Secondary Air Data
- Alitmeters

±0.4 hPa Accuracy
from -40 to 85°C

Outstanding Value

Small, Rugged Design

CE Qualified

ISO-9001
ISO-14001

FEATURES	BENEFITS
High Accuracy ±0.4 hPa max from -40 to 85°C ±0.03% FS max from -40 to 85°C	**Two-tiered accuracy including temperature errors over -40 to 85°C** – HPB, ±0.04 hPa or ±0.08 hPa; HPA, ±0.03% or ±0.06% FS Max. **Simplifies System Design** – there is no need to insulate, temperature-regulate or provide additional signal compensation.
Multiple Interface Options	**Easy Interface, Plug-and-Play for your system requirements.** **TTL**-for lowest power consumption (33 milliwatts) **RS-232**-receives commands and sends data to a single serial port of a computer. **RS-485**-up to 89 PPTs can be connected to a two-wire multidrop bus.
Proven Honeywell Technology	**Stable and Reliable**-Honeywell has been building the world's highest performance silicon pressure sensors for over thirty years.

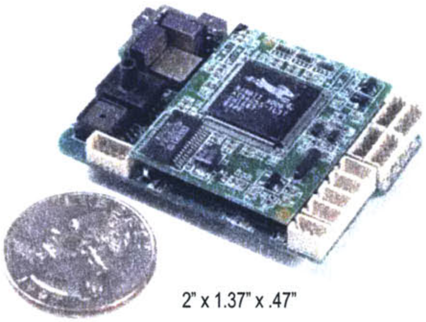
A.3 Autopilots

- Procerus Kestrel - Spec sheet taken from http://www.procerusuav.com/Documents/Kestrel_2.22.pdf.
- Crossbow NAV420 - Spec sheet taken from www.xbow.com/Products/Product_pdf_files/Inertial_pdf/ NAV420CA_Datasheet.pdf. (all one line)
- Athena Controls GS-111m - Spec sheet taken from http://athenati.com/products_services/guidestar/guidestar_gs-111m/ gs-111mproductsheet.pdf (all one line)

KAPv2.22

PROCERUS®
TECHNOLOGIES

Kestrel Autopilot v2.22


2" x 1.37" x .47"

FEATURES

- **16.7 grams**
- Fits within 2.73 in^2 Area & 1.29 in^3 Volume
- 3-Axis Angular Rate Measurement
- 3-Axis Acceleration Measurement
- 2-Axis Magnetometer
- Absolute and Differential Pressure Sensors
- 20 Point Sensor Temperature Compensation
- External Power @ 3.3V & 5V, 500mA
- Efficient Switching Power Regulation
- Battery Voltage and Current Monitor
- 29MHz Processor w/ 512K RAM & FLASH
- 4 Serial Ports (Std.,SPI,I^{2}C) w/ Digital Clock or I/O
- 4 Standard Servo Ports
- 12 Digital I/O (6 bi-directional, 3 input, 3 output)
- 3 Analog Inputs @ 12bit resolution
- Optional Piggy-Back Modem Header
- Wind Estimation
- Multiple Failsafes
- Multi-UAV Support
- Convoy Following Support
- Auto-trim
- GPS-denied take off
- Altitude can be referenced to sea level and can be initialized using onboard GPS, Ground GPS, or entered manually
- Configurable IO support
- Optimal Loiter Radius – Kestrel chooses optimal radius to maintain desired side look field of view
- Loiter offset for wind
- Altitude and Airspeed override through Virtual Cockpit
- New Modes (take off joystick, take off to waypoint, joy stick land. Also, take off timer or count down to motor ramp up.
- Selectable units (Metric, English, Nautical)
- Absolute vs relative waypoints (legal waypoints)
- Improved dead reckoning
- Built-in support for 2-axis gimbal

APPLICATIONS

- Autonomous GPS navigation of UAVs and MAVs
- Inertial Measure Unit
- Slave Processing Unit
- Payload Communication & Control Support
- Data Logger
- Multiple vehicle operations

DESCRIPTION

The Kestrel^{TM} Autopilot v2.2 is designed for autonomous flight control of small UAVs and MAVs. At 16.7 grams, it is the smallest (2" x 1.37" x .47") and lightest full featured autopilot on the market - ideal for all surveillance and reconnaissance applications. The Virtual Cockpit ground control software makes "click N' fly" operation easy while providing powerful mission planning, monitoring, and in-flight adjustment. New "piggy-back" header allows the modem to be plugged directly into the autopilot. The magnetometer can either be on board the autopilot or off board depending on user's setup requirements.

Its IMU is composed of 3-axis rate gyros and accelerometers. Absolute and differential pressure sensors provide barometric pressure and aircraft air speed. 3 temperature sensors combined with a 20 point temperature compensation algorithm reduce sensor drift improving aircraft state measurement and estimation.

Switching power regulation achieves high efficiency, drawing only 0.77 Watts while running cooler and consuming less power (less than half of KAPv1.45). External payloads can be powered at 3.3V and 5V, 500mA each. Battery voltage and current monitoring provides battery life information.

4 serial ports allow for support of payload inter-communication and control. Serial interfaces allow for the use of standard off-the-shelf digital modems and GPS units. The GPS can be placed in a location independent of the autopilot.

Kestrel and Virtual Cockpit are trademarks of Procerus Technologies.

ABSOLUTE MAXIMUM RATINGS

Input Supply Voltage	-0.3V to 18V
Payload Current	500mV @ 3.3V & 5V
Operating Temperature Range	-40ºC to 85ºC
Storage Temperature Range	-40ºC to 125ºC
Maximum Absolute Pressure	400 kPa
Maximum Differential Pressure	75kPa
Humidity	5% to 95%, no condensing
Acceleration	±200 g

Stresses above those listed under the Absolute Maximum Ratings may cause permanent damage to the autopilot. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification are not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING CHARACTERISTICS

Parameter	Conditions	Min	Typ	Max	Units
INPUT VOLTAGE (PWR)					
Operating Input Voltage Range		6.0		15.5	V
Quiescent Supply Current			80		mA
Payload POWER (Each Supply)				≈2.0	W
3.3V Source			3.3		V
5V Source			5.0		V
Supply Current				500	mA
Accuracy			±0.5	±2	%
Noise			15		mV$_{RMS}$
Analog Input Port					
5V Supply (V_S) Current	V_S = 4.95 V		50		mA
5V Supply (V_S) Noise			2.5		mV$_{RMS}$
Input Sample Range		0		5	V
Input Sample Resolution	V_S = 5 V		16 (0.0763)		bits (mV)
Payload Serial & I/O					
Logic High		2.3			V
Logic Low				0.4	V
Current (Sink & Source)				6.8	mA
Rate Gyros	T_A = 25°C, V_S = 5 V, Bandwidth = 9Hz				
Dynamic Range				±300	°/s
Frequency Response (3dB Bandwidth)			9		Hz
Resonant Frequency			14		kHz
Accelerometers	T_A = 25°C, V_S = 5 V				
Dynamic Range				±10	g
Frequency Response (3dB Bandwidth)			22		Hz
Resonant Frequency			10		kHz
Attitude Estimation Error: Roll and Pitch					
Level Flight				5	°
During Turns				10	°
Differential Pressure: KAPv2.20, KAPv2.21	T_A = 25°C, V_S = 5 V				
Range		-0.25		4.7	kPa
Resolution			0.000166		kPa
Differential Pressure: KAPv2.22	T_A = 25°C, V_S = 5 V				
Range		-1.3		15.8	kPa
Resolution			0.000545		kPa
Absolute Pressure: KAPv2.20, KAPv2.21	T_A = 25°C, V_S = 5 V				
Range		101.5		66.5	kPa
Resolution			0.00115		kPa
Absolute Pressure: KAPv2.22	T_A = 25°C, V_S = 5 V				
Range		111.3		41.5	kPa
Resolution			0.00244		kPa
Airspeed: KAPv2.20, KAPv2.21	T_A = 25°C, V_S = 5 V				
Range		0		70 (156)	m/s (mph)
Resolution	13 m/s (29 mph)		0.0076 (0.017)		m/s (mph)
Airspeed: KAPv2.22	T_A = 25°C, V_S = 5 V				
Range		0		130 (290)	m/s (mph)
Resolution	13 m/s (29 mph)		0.025 (0.056)		m/s (mph)
Altitude: KAPv2.20, KAPv2.21	T_A = 25°C, V_S = 5 V				
Range	Standard atmospheric pressure	-13.7 (-45)		3414 (11,200)	m (ft)
Resolution			0.116 (0.379)		m (ft)
Altitude: KAPv2.22	T_A = 25°C, V_S = 5 V				
Range	Standard atmospheric pressure	-792 (-2,600)		6,888 (22,600)	m (ft)
Resolution			0.245 (0.804)		m (ft)
Dimensions			2.073 x 1.375		inches
Accuracy			±0.5		%
Weight			16.65		grams
Accuracy			±4		%

PORT FUNCTIONS

The following tables describe the general pin assignments for each port type.

Power Port

Pin	Description
1	GND
2	PWR (6V – 18V)
3	Current Monitor

Servo Ports

Pin	Description
1	PWR
2	GND
3	Signal

Optional ADC Port

Pin	Description
1	GND
2	PWR (3.3V or 5V)
3	Ch 1
4	Ch 2
5	Ch 3

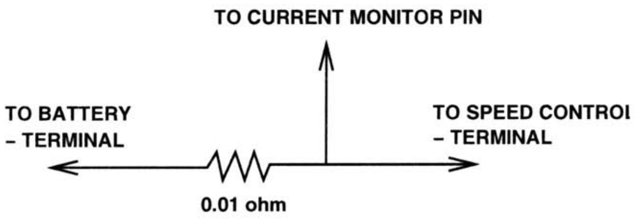
Serial Ports

Pin	Description
1	GND
2	PWR (3.3V or 5V)
3	Autopilot TX
4	Autopilot RX
5	CMD or CLK

Power Port: This port supplies the autopilot power and is typically connected directly to the autopilot or aircraft main battery. The GND and PWR pins connect to the negative and positive battery terminal respectively. The Current Monitor pin is used to detect current draw of the main battery by measuring the voltage drop across a 0.01Ω resistor in series with the battery. This resistor's power rating should be as follows:

$$\textit{RESISTOR POWER} > (\textit{MAX MOTOR CURRENT})^2 \times 0.01 \quad (\textit{WATTS})$$

Typical Current Monitor Circuit:



Analog Input Port: Three analog inputs (pins 3-5) on the Analog Input port allow users to measure 0.0V to 5.0V. Filtered analog 5V supply is available on pin 2. This pin supplies the autopilot analog sensors so take caution not to introduce noise on this pin. For specifications, see Analog Input Port in the Operating Characteristics table.

Serial & I/O Ports: There are 4 serial ports that double as I/O ports. Serial E and Serial A allow users to interface with payload needs. The GPS port is dedicated for the GPS unit. The MODEM port is optional if the modem is not plugged into the modem "piggy-back" header. For each serial port, the autopilot TX and RX lines are found on pins 3 and 4 respectively. All serial ports operate at TTL levels (0V to 3.3V) and can be configured for standard serial, SPI, or I^2C communication. Pin 5 on all serial ports serves as a digital I/O. Pins 2 and 3 can be used as digital I/O if not being used for serial communication. Table 1 shows the pin assignments (connections to Rabbit 3000 processor) of all serial ports.

Pin	SerA	SerE	GPS	Modem
1	GND	GND	GND	GND
2	PWR (3.3V or 5V)	PWR (3.3V or 5V)	PWR (3.3V or 5V)	PWR (3.3V or 5V)
3	TxA (PC6)	TxE (PG6)	TxD (PC0)	TxF (PG2)
4	RxA (PC7)	RxE (PG7)	RxD (PC1)	RxF (PG3)
5	Reset/Smode	ClkE (PG5)	ClkD (PF0)	TClkF (PG0)

Table 1 - Serial Port Pin Descriptions

Servo Ports: These ports are configurable for different aircraft types. Servo connections for standard configurations are as follows:

V-Tail Configuration

Port	Channel
1	Right V-Tail
2	Left V-Tail
3	Throttle

Elevon Configuration

Port	Channel
1	Left Elevon
2	Right Elevon
3	Throttle

KESTREL AUTOPILOT

The autopilot is the heart of the Kestrel system. It is powered by an 8-bit 29 MHz processor. The autopilot board contains a suite of sensors used by the autopilot software to measure and estimate the states of the aircraft. The autopilot interfaces directly to the digital communication link which enables it to send real-time status telemetry to the ground station and receive commands in-flight. The GPS plugs into the autopilot board (optional) and provides inertial navigation information to the autopilot. It also has several additional interface ports to support payloads. The autopilot controls the aircraft through four standard RC hobby servos. If more servos are needed, a servo extender board can be used. Figure 1 shows the Kestrel autopilot with modem attached.

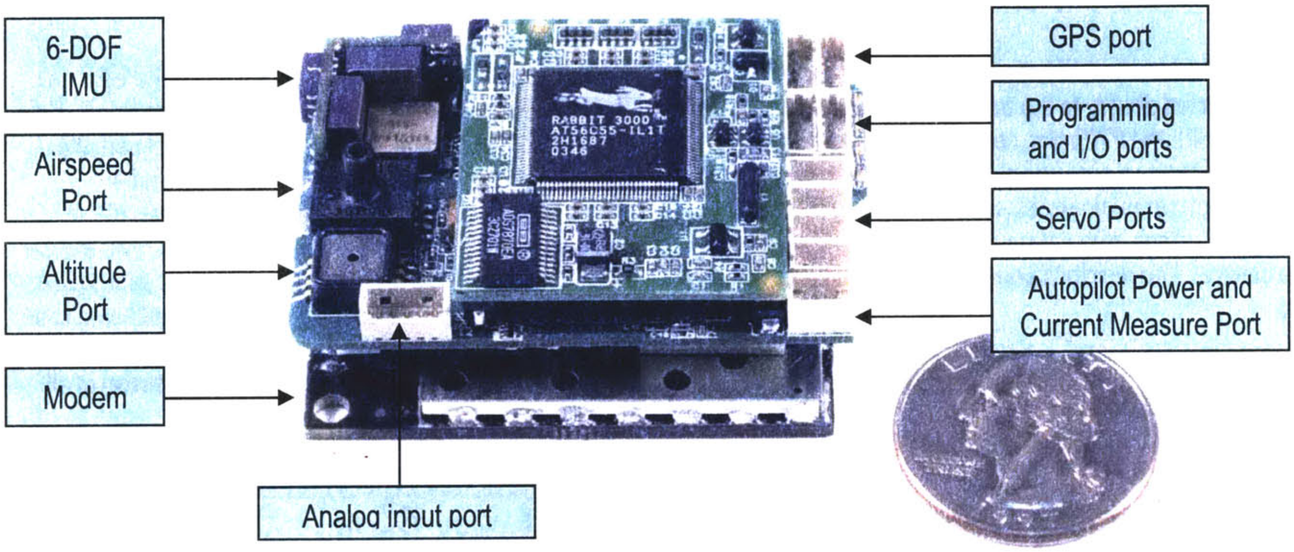


Figure 1 - Kestrel 2.2 autopilot with Aerocomm AC4490 modem attached.

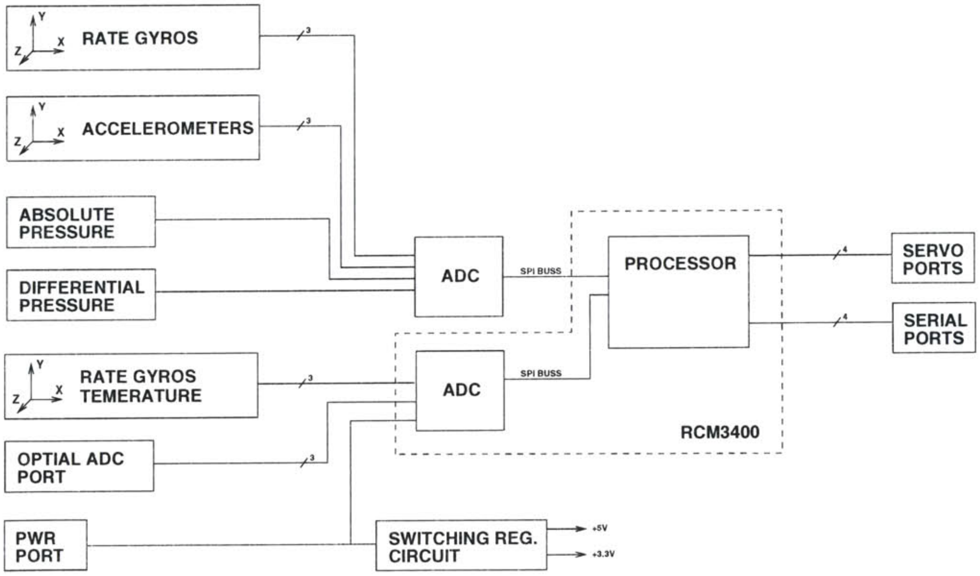


Figure 2 - Kestrel 2.x block diagram.

Port & LED Locations

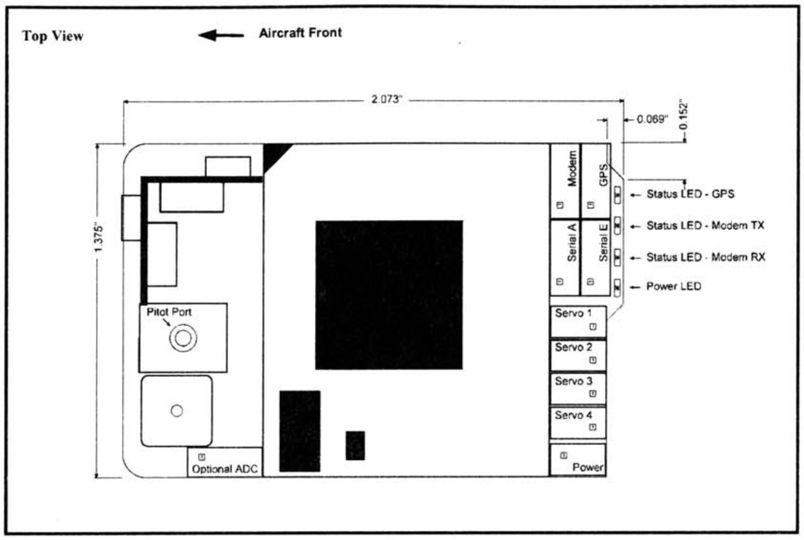


Figure 3 - Port and LED locations on the Kestrel 2.x autopilot. (2" x 1.37" x .47")

Jumper and Header Locations

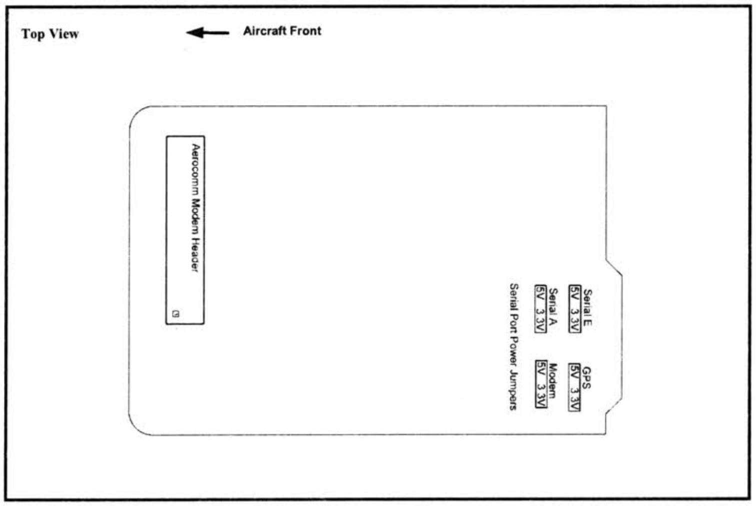


Figure 4 - Jumper and header locations on the Kestrel 2.x autopilot.

Kestrel Autopilot v2.22

Sensors and Attitude Estimation

- Increased resolution on all sensors (8 x increase)
- Acceleration measurements down to 1 mg
- Roll and pitch estimation corrected for coriolis forces (10 - 25% improved roll and pitch estimates)
- 2axis - magnetometer support - compass heading used to calibrate heading gyro on ground and in low ground speed situations.
  - Quicker sensor calibration and improved navigation, path following and altitude hold
  - Assist human pilot in speed and altitude mode (display on video)

3-Sensor Temperature Compensation

- 3 temperature sensors combined with 20 point temperature compensation algorithm significantly reduces sensor drift due to temperature changes. This reduces the need for the user to re-calibrate gyros and pressure sensors, aiding in sea or mobile operations.

Wind estimation

- Real-time wind estimation algorithm relies on airspeed and GPS - continually updating estimate with latest wind data. Good to 5% on wind speed and 2% on wind heading.

Auto-trim

The autopilot can automatically fine tune UAV trim characteristics in the air. Trim values are then saved on the autopilot.

Switching Voltage Regulation

- The nominal regulator temperature remains constant over a broad range of input voltages. The maximum voltage input can be up to 18 volts. (KAPv2.x runs 50% cooler and draws 50% of the power compared to KAPv1.45)

10g Accelerometers

- 10g accelerometers now used vs 2g to better address vibration susceptibility in certain airframe configurations. The Kestrel can also be configured with 2g sensors if desired.

Mode support - single click autopilot configurations

- Manual mode	- rates only, activated by switch on RC controller
- Speed mode	- aircraft holds airspeed using pitch, (roll, airspeed, and throttle commands on ground station)
- Altitude mode	- aircraft holds altitude (roll, airspeed, and altitude commands from on ground station
- Nav mode	- aircraft navigates to standard and loiter waypoints
- Home mode	- aircraft flies home and loiters
- Loiter now mode	- aircraft loiters at current position
- Take off modes (3)	- aircraft uses preset commands to take off - automatically transitions to Nav mode at pre-set altitude
- Land mode	- aircraft flies to landing point on map and lands
- Rally mode	- aircraft flies to Rally point

Multiple user-configurable failsafes

- Loss of communications
- Loss of GPS lock
- Low Battery and Critical Battery
- Manual Mode

New

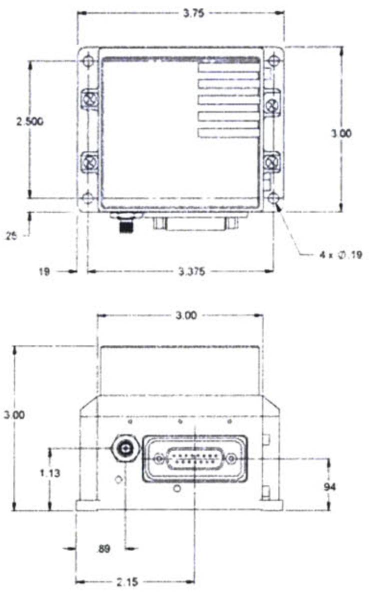
NAV420

GPS-AIDED MEMS INERTIAL SYSTEM

- Real-Time GPS X, Y, Z Position and Velocity Outputs
- AHRS Pitch, Roll, and Heading Output at 100Hz
- Built-In GPS Receiver with RTCM and WAAS Compatibility
- High Stability MEMS Sensors
- Enhanced Performance Kalman Filter Algorithm
- EMI & Vibration Resistant
- Environmentally Sealed

Applications

- Remotely Operated Vehicles
- Land Vehicle Guidance
- Avionics Systems
- Platform Stabilization



Package Dimensions

NAV420CA

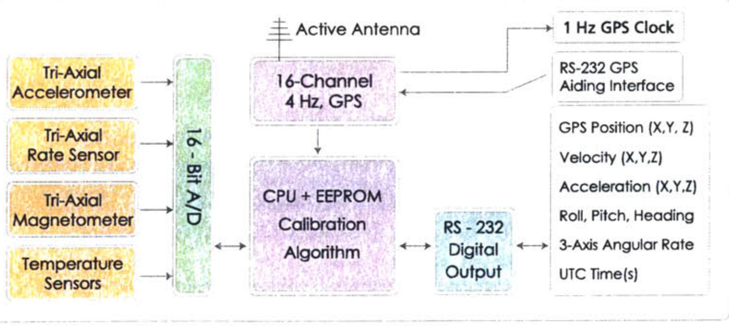
The Crossbow NAV420 is a combined GPS Navigation and GPS-Aided Attitude & Heading Reference system (AHRS) that utilizes both MEMS-based inertial sensors and GPS technology to provide an unmatched value in terms of both price and performance. Developed in response to years of extensive application experience in a wide variety of airborne, marine and land applications, the NAV420 also incorporates many new and enhanced design features including:

- Built-in GPS receiver for position and velocity measurement
- GPS data synchronization clock
- High performance Kalman Filter algorithms
- Water resistant, vibration resistant, light-weight design
- EMI protection for trouble-free operation
- Continuous Built-in-Test

The NAV420 provides consistent performance over a wide temperature range in challenging EMI environments across a broad range of input power conditions. It is designed for use in a number of different applications including remotely piloted vehicles, land vehicle guidance, uncertified avionics and platform stabilization.

This high reliability, strapdown inertial system provides attitude and heading measurement with static and dynamic accuracies that exceed traditional spinning mass vertical and directional gyros. With GPS integration, the NAV420 system also provides GPS velocity data at up to 100 Hz. Velocity data includes aiding from the inertial instruments to improve stability and reduce the latency associated with stand-alone GPS measurements.

Each NAV420 system comes with a GPS antenna and User's Manual. Crossbow's NAV-VIEW software is also included to assist users with system development, evaluation, and data acquisition.



NAV420 Block Diagram

Specifications	NAV420CA-100	Remarks
Performance		
Update Rate[1] (Hz)	2-100	Programmable
Start-up Time Valid Data (sec)	< 1	
Fully Stabilized Data (sec)	< 60	Under static conditions
Position/Velocity		
Position Accuracy[2] (m CEP)	3	Internal GPS, not augmented
X,Y Velocity Accuracy (m/s rms)	< 0.4	GPS available
Z Velocity Accuracy (m/s rms)	< 0.5	GPS available
1PPS Accuracy (ns)	± 50	GPS available
Attitude		
Range: Roll, Pitch (°)	± 180, ± 90	
Accuracy[3] (° rms)	< 0.75	GPS available
(° rms)	< 2.5	GPS unavailable
Resolution (°)	< 0.1	
Heading		
Range (°)	± 180	
Accuracy[3] (° rms)	< 3.0	
Resolution (°)	< 0.1	
Angular Rate		
Range: Roll, Pitch, Yaw (°/sec)	± 200	
Bias: Roll, Pitch, Yaw (°/sec)	< ± 0.05	Kalman filter stabilized
Bias: Roll, Pitch, Yaw (°/sec)	< ± 0.75	Kalman filter off
Scale Factor Accuracy (%)	< 1	
Non-Linearity (% FS)	< 0.5	
Resolution (°/sec)	< 0.06	
Bandwidth (Hz)	25	-3 dB point nominal
Random Walk (°/hr$^{1/2}$)	< 4.5	
Acceleration		
Input Range: X/Y/Z (g)	± 4	
Bias: X/Y/Z (mg)	< ± 15	
Scale Factor Accuracy (%)	< 1	
Non-Linearity (% FS)	< 1	
Resolution (mg)	< 0.6	
Bandwidth (Hz)	25	-3 dB point nominal
Random Walk (m/s/hr$^{1/2}$)	< 1.0	
Environment		
Operating Temperature (°C)	-40 to +71	
Non-Operating Temperature (°C)	-55 to +85	
Non-Operating Vibration (g rms)	6	20 Hz - 2 KHz random
Non-Operating Shock (g)	200	1 ms half sine wave
Enclosure	IP66 compliant	
Electrical		
Input Voltage (VDC)	9 to 42	
Input Current (mA)	< 350	at 12 VDC nominal
Power Consumption (W)	< 5	
Digital Output Format	RS-232	
Physical		
Size (in)	3.0 x 3.75 x 3.0	with mounting flanges
(cm)	7.62 x 9.53 x 7.62	with mounting flanges
Weight (lbs)	< 1.3	
(kg)	< 0.58	
Connector	15 pin "D" male	
GPS Antenna Connector	SMA Jack	

Notes

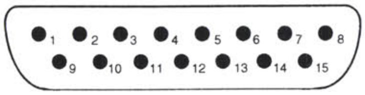
[1] See User's Manual for addtional information

[2] Internal GPS accuracy can be further improved with Radio Technical Commission for Maritime (RTCM) or Satellite Based Augmentation System (SBAS) messages such as the Wide Area Augmentation System (WAAS).

[3] Dynamic conditions, standard Crossbow flight profile

Specifications subject to change without notice

15 Pin "D" Connector Male Pinout



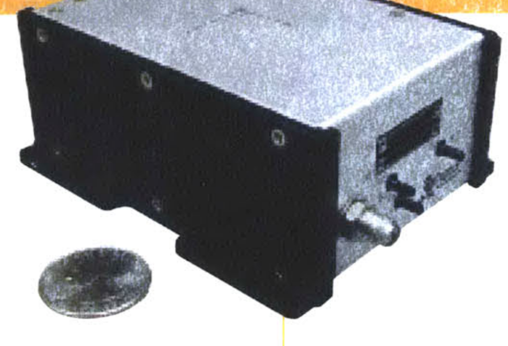
Pin	Signal
1	RS-232 Transmit Data
2	RS-232 Receive Data
3	Positive Power Input (+Vcc)
4	Power Ground
5	Chassis Ground
6	NC – Factory use only
7	RS-232 GPS Tx
8	RS-232 GPS Rx
9	Signal Ground
10	1PPS OUT
11	NC – Factory use only
12	NC – Factory use only
13	NC – Factory use only
14	NC – Factory use only
15	NC – Factory use only

Ordering Information

Model	Description	Gyro (°/sec)	Accel (g)
NAV420CA-100	GPS-Aided MEMS Inertial System	± 200	± 4

CALL FACTORY FOR OTHER CONFIGURATIONS

GUIDESTAR UAV FLIGHT CONTROL SYSTEM (FCS) FUNCTIONAL SPECIFICATIONS



Overview

Size	3.9 x 2.6 x 1.6 in^3
Weight	0.5 lbs (227 g; 8 oz)
Power	4.5 W max. at 9 – 18 V or 18 – 36V

Environmental

Temperature (Operating)	- 40° to + 70° C
Temperature (Non-Operating)	- 54° to + 85° C
Humidity	95% RH, non-condensing
Vibration/Shock	MIL-STD-810

IMU Performance

Update Rate	50 or 100 Hz
Maximum Angular Rates	±200 deg/sec (optional up to ± 600 deg/sec)
Maximum g Range	±10 g
Sampling Resolution	24 bits A/D

Accuracy

Heading	0.3 deg (one sigma)
Pitch	0.3 deg (one sigma)
Roll	0.3 deg (one sigma)
Airspeed	1 knot, typical
Altitude	25 ft at S.L., 100 ft at 40,000 ft
Latitude/Longitude	GPS C/A Code - Differential ready - WAAS enabled

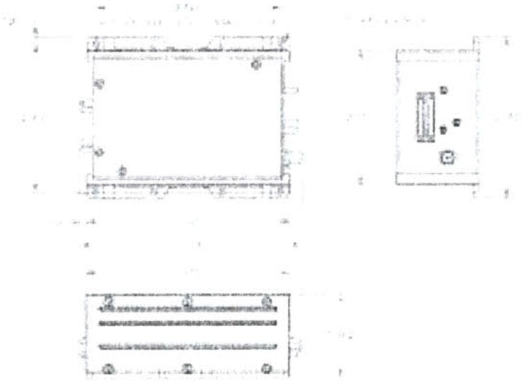
- **GPS Receiver**
- **Solid-State Gyros and Accelerometers**
- **CPU Card (Kalman-Filters, High Speed Sampling, Control Laws)**
- **Air-Pressure Transducers**
- **AoA and Sideslip**
- **Triaxial Magnetometer**

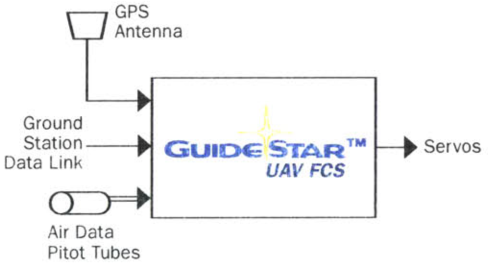
I/O

PWM Input (10) Channels
PWM Output (10) Channels
Discrete Input (4)
Discrete Outputs (5)
Analog Input configurable
Serial Connections (5)
- RS232
- RS422 or RS485

Bus I/O
- CAN bus (2)
- SPI bus

Dimensions and Mounting





Flight Control Functions

- 3-D waypoint navigation
- Joystick or ground station altitude hold, airspeed hold, heading hold, AoA, sideslip, climb rate
- Joystick attitude control (pitch, roll, heading)
- Easy-to-use ground station waypoint navigation and flightplanning
- Automatic takeoff and landing
- Stall protection, speed, attitude, and load factor limiting
- Fault-tolerant design for GPS outage and loss of data-link
- Single design solution for a family of vehicles
- Onboard data-logging

For More Information Contact

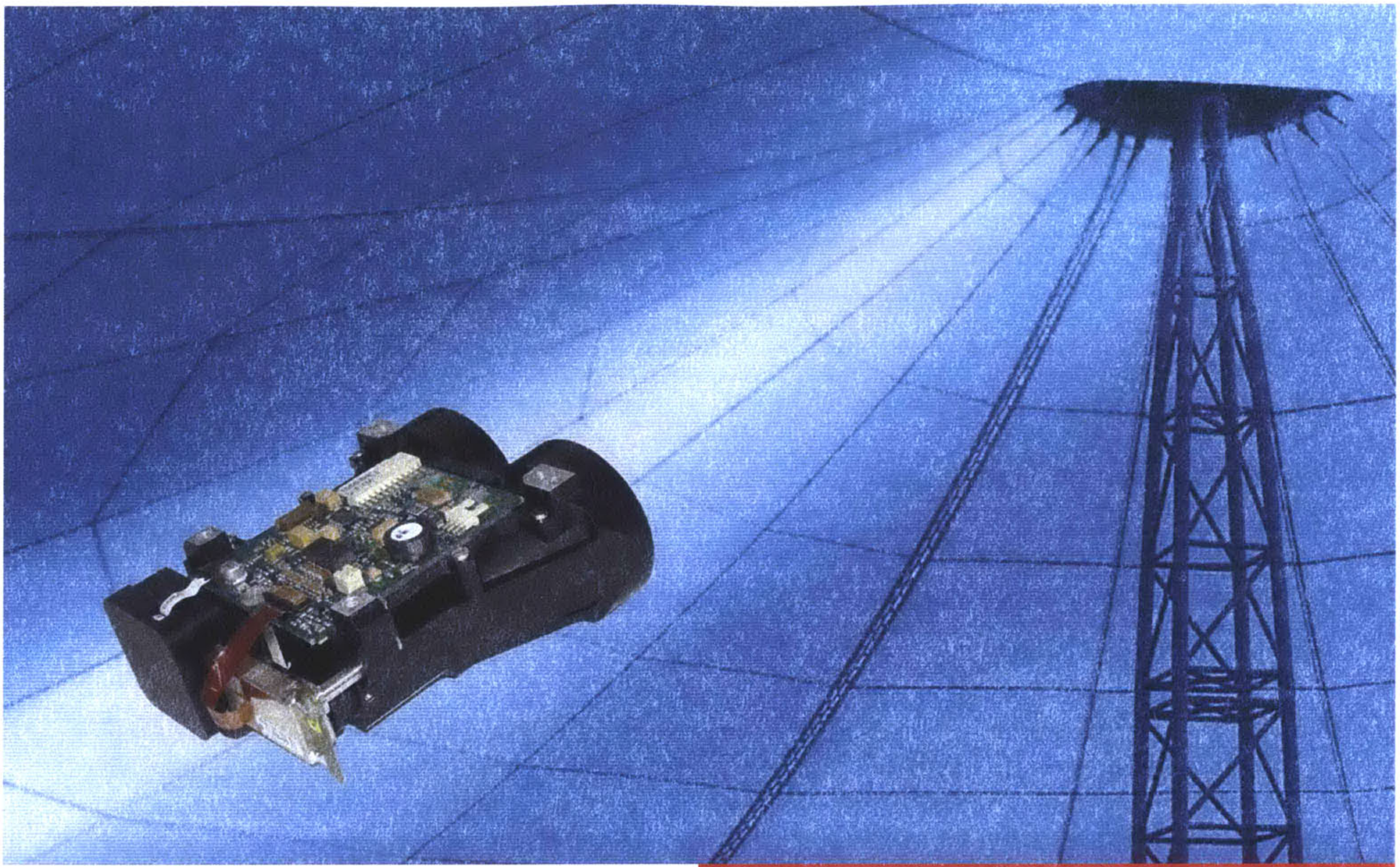
Call James Dotan
VP, Business Development
540.428.3318
info@AthenaTI.com

Athena CONTROLS

www.AthenaTI.com 6876 Watson Court, Vint Hill, Warrenton, VA 20187 USA

A.4 Laser Rangefinders

- Vectronix LRF42 - Spec sheet taken from www.vectronix.ch/files/LRF42.pdf.
- Thales Miniature Eyesafe Laser Rangefinder (MELT) - Spec sheet taken from https://peoiewswebinfo.monmouth.army.mil/portal_sites/IEWS_Public/RUS/sensorcat/PDF/MELT-Thales1.PDF. (all one line)



Rangefinder Modules

LRF42

OEM products for system integrators

Vectronix is a world leader in direct diode eye-safe laser technology. This core competence is utilized in the highly successful family of VECTOR Rangefinder Binoculars and in the Laser Range Finder (LRF) modules widely used by prime contractors in their systems.
A decade of experience working with international partner firms has enabled Vectronix to develop an attractive range of LRF modules. Their electronics are mounted on a chassis together with the appropriate transmitter and receiver optics, perfectly calibrated, boresighted, and equiped with well-defined electronic and mechanical interfaces.

Options:

- Transmitter/receiver aperture 42mm. Modules with 30mm aperture are described in a separate flyer.
- Laser wavelength: 905nm or 1550nm. Both options are eye-safe and invisible to the unaided eye. 905nm can be detected through an image intensifier device, but 1550nm cannot.
- The beamshaper (BS) increases range by reducing the divergence of the laser beam.
- LRF modules can also be provided with the optional DMC-SX Digital Magnetic Compass/tilt sensor assembly. Being fully integrated, it does not increase the modules' overall size. The DMC-SX is described in a separate flyer.

vectronix

Technical Data LRF42

Range performance	**905nm**	**1550nm**	**1550nm with beamshaper**
Beam divergence	0.3 × 1.5mrad	2.0 × 2.0mrad	0.4 × 0.7mrad
Range capability, best conditions	3500m	4500m	7500m
Specified performance* 2.3 × 2.3m target, albedo 0.3, detection rate 90% at visibility 10km	2500m	2500m	4500m

The following applies to all versions	
Accuracy* (1σ)	± 1m to ± 3m
Minimum range	
Functional	5m
Specified	50m
Time per measurement	0.3 to 1.1s
Repetition rate	20 per minute (0.3Hz)
Target discrimination	30m
Eye safety	class 1 per IEC 60825-1 Ed 1.2 (2001-08)

Miscellanous functions	
Multiple target ranging	allows interpretation as first and last return
Range gating capability	on request
Built-in test (BIT)	via serial interface

Electrical	
Power supply voltage	4V to 6V, ripple < 100 mVpp
Power consumption at 5.5V	
Range measurement	Average current while lasing < 650mA, I_{peak} < 1300mA for not more than 500µs
Heading & tilt measurement	0.6W
Standby	30mW (SWT or Com_switch to GND)
Shutdown	55µW (no SWT or Com_switch to GND)
Connector interface	2 PCB mounted FCI Minitek™ connectors
Serial interface	RS232 or RS422, baud rate 9600, 19200 or 38400 bits per second

Environmental conditions*	
Temperature	
Operating	–35°C to +55°C
Storage	–40°C to +85°C
Shock	50g / 11ms half sine, 2000g / 0.5ms half sine
Vibration	Random, 5 to 500Hz, $0.02g^2/Hz$, 120 minutes per axis

Mechanical	
Transmitter/receiver aperture	42mm
Weight with DMC-SX	350g
Weight without DMC-SX	325g
Dimensions (L × W × H)	110 × 100 × 50mm
Interface	4× mounting pad, 2× 3mm positioning hole, 4× M4 threaded hole

* Compliance depends on actual application

Available configurations

Article No.	Description
901 858	LRF42mm-905nm, Laser Range Finder Module incl. Digital Magnetic Compass
901 632	LRF42mm-905nm, Laser Range Finder Module
901 859	LRF42mm-1550nm, Laser Range Finder Module incl. Digital Magnetic Compass
901 860	LRF42mm-1550nm, Laser Range Finder Module
901 861	LRF42mm-1550nm BS, Laser Range Finder Module incl. Digital Magnetic Compass
901 862	LRF42mm-1550nm BS, Laser Range Finder Module

LASER CLASS 1
Eye-safe
in accordance with
IEC 60825-1 Ed 1.2 (2001-08)

vectronix

Vectronix AG
Max-Schmidheiny-Strasse 202
CH-9435 Heerbrugg
Switzerland
Telephone +41 71 727 47 47
Fax +41 71 727 46 79
www.vectronix.ch

Printed in Switzerland –

903 661 – VIII 05 – RDV

A SAFRAN Group company

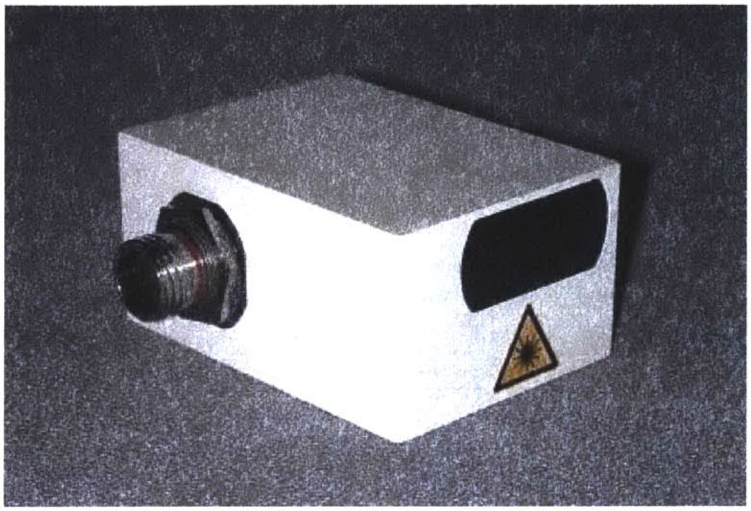
MELT Laser Rangefinder

THALES

Date Revised: 30 JAN 04

VENDOR DESCRIPTION

MELT (Miniature Eyesafe Laser Rangefinder) is a lightweight miniature erbium laser rangefinder. MELT's reduced space claim and minimal weight make it ideal for robotic and unmanned platform applications. The system is currently under development for European customers and demonstration with full rate production is due towards the end of 2003. The MELT unit can be integrated to, or form part of, Surveillance systems, Target Location systems, Laser Designation systems and Fire Control systems where line-of-sight range to target is required. MELT is being offered as either a chassis-based (unboxed) module for inclusion within a larger system or as a discrete LRU (boxed) for bolt-on external applications. MELT is classified as being Class 1 eyesafe to ANSI Z136.1-2000 and TBMED-524.



Product Manager Robotic & Unmanned Sensors
Telephone: (732) 427-5827 / DSN 987
Fax: (732) 427-5072 / DSN 987
e-mail: SFAE-IEWS-NV-RUS@iews.monmouth.army.mil
PM RUS Robotic & Unmanned Sensors

Business Category: Large Business

EOIR

Hardware	
Wavelength: 1.54 micro meters	Operating Temp.: -40°C to +71°C
Eyesafe to Class 1 ANSI 136.1.-2000	Storage Temp.: -55°C to 85°C
Extinction Ratio: 30 dB (Pin diode) or 37 dB (APD option)	Data Interface: RS-485
Range Resolution: 5m	False Alarm Rate: < 1%
Range Accuracy: 5m	Missing Pulses: < 1%
Range Gating Function: Variable between 45m and 9995m	Weight (Boxed): 1 kg or 2.2 lbs
Target Discrimination: Customer-selectable between 5m and 30m	Dimensions (Boxed): 65mm x 104mm x 160mm
Target Selection: First/Last	Weight (Unboxed): 0.4 kg or 0.9 lbs
Beam Divergence: < 0.5 mrads	Dimensions (Unboxed): 100mm x 75mm x 50mm
Beam Stability: < 0.1 mrads	MTBF: > 6000 hrs (calculated)
Processing Range: 50 to 19,995 m	Maintainability: 2-level BIT to LRM level

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Complete Results of Simulation

```
Number 1 Alternative 23 - dted
    Optical Camera - Sony FCB-EX780B
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 961.670086844 g
    Total Unit Cost = 9580.0 usd
    Total Power = 7.065 W
    Accuracy = 52.8503830562 ft
    Total Size = 870.77136599 cm^3
Alternative 23 - dted
      Latency score =       0                Latency weight = 0
   Robustness score = -0.2599             Robustness weight = 1
         Size score = 0.975680991898            Size weight = 1
         Mass score = -0.203402571523           Mass weight = 1
        Power score = -11.0030897709           Power weight = 1
Monetary Cost score = 0.0951040509437  Monetary Cost weight = 0.5
     Accuracy score = 0.752626117651        Accuracy weight = 2

         Total Cost = -8.93790708973
```

```
Number 2 Alternative 11 - height
    Optical Camera - Sony FCB-EX780B
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 961.670086844 g
    Total Unit Cost = 9580.0 usd
    Total Power = 7.065 W
    Accuracy = 160.686306595 ft
    Total Size = 870.77136599 cm^3
Alternative 11 - height
      Latency score =       0                   Latency weight = 0
   Robustness score = -0.0184                Robustness weight = 1
         Size score = 0.975680991898               Size weight = 1
         Mass score = -0.203402571523              Mass weight = 1
        Power score = -11.0030897709              Power weight = 1
Monetary Cost score = 0.0951040509437    Monetary Cost weight = 0.5
     Accuracy score = 0.043427508649           Accuracy weight = 2

         Total Cost = -10.1148043077



Number 3 Alternative 24 - dted
    Optical Camera - Sony FCB-EX780B
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 598.796187248 g
    Total Unit Cost = 16085.0 usd
    Total Power = 7.365 W
    Accuracy = 39.9401484248 ft
    Total Size = 583.571681447 cm^3
Alternative 24 - dted
      Latency score =       0                   Latency weight = 0
   Robustness score = -0.2599                Robustness weight = 1
         Size score = 0.9965703695                 Size weight = 1
         Mass score = 0.351184638762               Mass weight = 1
        Power score = -16.7463249579              Power weight = 1
Monetary Cost score = -1428.15382829     Monetary Cost weight = 0.5
     Accuracy score = 0.843140973131           Accuracy weight = 2

         Total Cost = -728.04910215
```

```
Number 4 Alternative 12 - height
    Optical Camera - Sony FCB-EX780B
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 598.796187248 g
    Total Unit Cost = 16085.0 usd
    Total Power = 7.365 W
    Accuracy = 156.499507186 ft
    Total Size = 583.571681447 cm^3
Alternative 12 - height
      Latency score =      0               Latency weight = 0
   Robustness score = -0.0184           Robustness weight = 1
         Size score = 0.9965703695            Size weight = 1
         Mass score = 0.351184638762          Mass weight = 1
        Power score = -16.7463249579         Power weight = 1
Monetary Cost score = -1428.15382829   Monetary Cost weight = 0.5
     Accuracy score = 0.0677671059133      Accuracy weight = 2

         Total Cost = -729.358349885



Number 5 Alternative 46 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 748.65 g
    Total Unit Cost = 16585.0 usd
    Total Power = 5.015 W
    Accuracy = 140.86174468 ft
    Total Size = 714.661314813 cm^3
Alternative 46 - range
      Latency score =      0               Latency weight = 0
   Robustness score =  0.3392           Robustness weight = 1
         Size score = 0.992445347913          Size weight = 1
         Mass score = 0.17743369235           Mass weight = 1
        Power score = -0.00842923732145      Power weight = 1
Monetary Cost score = -2205.30066194   Monetary Cost weight = 0.5
     Accuracy score = 0.148533253815       Accuracy weight = 2

         Total Cost = -1100.85261466
```

```
Number 6 Alternative 43 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 1006.65 g
    Total Unit Cost = 16100.0 usd
    Total Power = 13.25 W
    Accuracy = 140.86174468 ft
    Total Size = 1299.39086052 cm^3
Alternative 43 - range
      Latency score =      0                Latency weight = 0
   Robustness score =  0.3392            Robustness weight = 1
         Size score = 0.842018498407           Size weight = 1
         Mass score = -0.578754846724          Mass weight = 1
        Power score = -2290.36590838          Power weight = 1
Monetary Cost score = -1447.39362289   Monetary Cost weight = 0.5
     Accuracy score = 0.148533253815       Accuracy weight = 2

         Total Cost = -3013.16318966


Number 7 Alternative 40 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 738.65 g
    Total Unit Cost = 17835.0 usd
    Total Power = 8.09 W
    Accuracy = 136.620784636 ft
    Total Size = 889.661314813 cm^3
Alternative 40 - range
      Latency score =      0                Latency weight = 0
   Robustness score =  0.3392            Robustness weight = 1
         Size score = 0.972525056879           Size weight = 1
         Mass score = 0.187592737828           Mass weight = 1
        Power score = -41.0554093066          Power weight = 1
Monetary Cost score = -5960.81069231   Monetary Cost weight = 0.5
     Accuracy score = 0.178513097092       Accuracy weight = 2

         Total Cost = -3019.60441147
```

```
Number 8 Alternative 22 - dted
    Optical Camera - Sony FCB-EX780B
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 388.65 g
    Total Unit Cost = 6585.0 usd
    Total Power = 4.515 W
    Accuracy = 272.206069715 ft
    Total Size = 339.661314813 cm^3
Alternative 22 - dted
      Latency score =       0              Latency weight = 0
   Robustness score = -0.2599           Robustness weight = 1
         Size score = 0.997364720891          Size weight = 1
         Mass score = 0.617435737801          Mass weight = 1
        Power score = 0.178830144806         Power weight = 1
Monetary Cost score = 0.668805972374   Monetary Cost weight = 0.5
     Accuracy score = -1613.81445278        Accuracy weight = 2

         Total Cost = -3225.76077198


Number 9 Alternative 13 - dted
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 636.65 g
    Total Unit Cost = 14900.0 usd
    Total Power = 10.55 W
    Accuracy = 270.517501984 ft
    Total Size = 698.628360515 cm^3
Alternative 13 - dted
      Latency score =       0              Latency weight = 0
   Robustness score = -0.2599           Robustness weight = 1
         Size score = 0.993340250398          Size weight = 1
         Mass score = 0.305250419409          Mass weight = 1
        Power score = -397.221316666         Power weight = 1
Monetary Cost score = -458.759392896   Monetary Cost weight = 0.5
     Accuracy score = -1478.25904624        Accuracy weight = 2

         Total Cost = -3582.08041493
```

```
Number 10 Alternative 16 - dted
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 378.65 g
    Total Unit Cost = 15385.0 usd
    Total Power = 2.315 W
    Accuracy = 272.206069715 ft
    Total Size = 113.898814813 cm^3
Alternative 16 - dted
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 1.00389240713               Size weight = 1
         Mass score = 0.632691837582              Mass weight = 1
        Power score = 0.806090668995             Power weight = 1
Monetary Cost score = -744.897127137     Monetary Cost weight = 0.5
     Accuracy score = -1613.81445278          Accuracy weight = 2

         Total Cost = -3597.89469422


Number 11 Alternative 17 - dted
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 951.670086844 g
    Total Unit Cost = 18380.0 usd
    Total Power = 4.865 W
    Accuracy = 52.8503830562 ft
    Total Size = 645.00886599 cm^3
Alternative 17 - dted
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 0.995459708227              Size weight = 1
         Mass score = -0.150852468997             Mass weight = 1
        Power score = 0.0655748913387            Power weight = 1
Monetary Cost score = -8879.96161875     Monetary Cost weight = 0.5
     Accuracy score = 0.752626117651          Accuracy weight = 2

         Total Cost = -4437.82527501
```

```
Number 12 Alternative 5 - height
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 951.670086844 g
    Total Unit Cost = 18380.0 usd
    Total Power = 4.865 W
    Accuracy = 160.686306595 ft
    Total Size = 645.00886599 cm^3
Alternative 5 - height
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.995459708227              Size weight = 1
         Mass score = -0.150852468997             Mass weight = 1
        Power score = 0.0655748913387            Power weight = 1
Monetary Cost score = -8879.96161875     Monetary Cost weight = 0.5
     Accuracy score = 0.043427508649          Accuracy weight = 2

         Total Cost = -4439.00217223


Number 13 Alternative 19 - dted
    Optical Camera - Sony FCB-EX780B
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 646.65 g
    Total Unit Cost = 6100.0 usd
    Total Power = 12.75 W
    Accuracy = 270.517501984 ft
    Total Size = 924.390860515 cm^3
Alternative 19 - dted
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 0.966032268547              Size weight = 1
         Mass score = 0.293100984246              Mass weight = 1
        Power score = -1719.31320675             Power weight = 1
Monetary Cost score = 0.730919668802     Monetary Cost weight = 0.5
     Accuracy score = -1478.25904624          Accuracy weight = 2

         Total Cost = -4674.46660614
```

```
Number 14 Alternative 14 - dted
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1209.67008684 g
    Total Unit Cost = 17895.0 usd
    Total Power = 13.1 W
    Accuracy = 51.742714693 ft
    Total Size = 1229.73841169 cm^3
Alternative 14 - dted
      Latency score =       0                   Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 0.871457840251              Size weight = 1
         Mass score = -11.1708569515              Mass weight = 1
        Power score = -2104.66217694             Power weight = 1
Monetary Cost score = -6234.12467581     Monetary Cost weight = 0.5
     Accuracy score = 0.760387286502          Accuracy weight = 2

         Total Cost = -5230.76303939
```

```
Number 15 Alternative 2 - height
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1209.67008684 g
    Total Unit Cost = 17895.0 usd
    Total Power = 13.1 W
    Accuracy = 159.279115377 ft
    Total Size = 1229.73841169 cm^3
Alternative 2 - height
      Latency score =       0                   Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.871457840251              Size weight = 1
         Mass score = -11.1708569515              Mass weight = 1
        Power score = -2104.66217694             Power weight = 1
Monetary Cost score = -6234.12467581     Monetary Cost weight = 0.5
     Accuracy score = 0.0529383711843         Accuracy weight = 2

         Total Cost = -5231.93643722
```

```
Number 16 Alternative 20 - dted
    Optical Camera - Sony FCB-EX780B
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1219.67008684 g
    Total Unit Cost = 9095.0 usd
    Total Power = 15.3 W
    Accuracy = 51.742714693 ft
    Total Size = 1455.50091169 cm^3
Alternative 20 - dted
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 0.770160220101              Size weight = 1
         Mass score = -12.4999356809              Mass weight = 1
        Power score = -6526.71837617             Power weight = 1
Monetary Cost score = 0.174385892314     Monetary Cost weight = 0.5
     Accuracy score = 0.760387286502          Accuracy weight = 2

         Total Cost = -6537.10008411


Number 17 Alternative 8 - height
    Optical Camera - Sony FCB-EX780B
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1219.67008684 g
    Total Unit Cost = 9095.0 usd
    Total Power = 15.3 W
    Accuracy = 159.279115377 ft
    Total Size = 1455.50091169 cm^3
Alternative 8 - height
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.770160220101              Size weight = 1
         Mass score = -12.4999356809              Mass weight = 1
        Power score = -6526.71837617             Power weight = 1
Monetary Cost score = 0.174385892314     Monetary Cost weight = 0.5
     Accuracy score = 0.0529383711843         Accuracy weight = 2

         Total Cost = -6538.27348194
```

```
Number 18 Alternative 21 - dted
    Optical Camera - Sony FCB-EX780B
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 856.796187248 g
    Total Unit Cost = 15600.0 usd
    Total Power = 15.6 W
    Accuracy = 38.4859958032 ft
    Total Size = 1168.30122715 cm^3
Alternative 21 - dted
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 0.895416231386              Size weight = 1
         Mass score = 0.0780824613106             Mass weight = 1
        Power score = -7497.3498313              Power weight = 1
Monetary Cost score = -914.981454517     Monetary Cost weight = 0.5
     Accuracy score = 0.853171821715          Accuracy weight = 2

         Total Cost = -7952.42061622



Number 19 Alternative 9 - height
    Optical Camera - Sony FCB-EX780B
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 856.796187248 g
    Total Unit Cost = 15600.0 usd
    Total Power = 15.6 W
    Accuracy = 155.060150829 ft
    Total Size = 1168.30122715 cm^3
Alternative 9 - height
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.895416231386              Size weight = 1
         Mass score = 0.0780824613106             Mass weight = 1
        Power score = -7497.3498313              Power weight = 1
Monetary Cost score = -914.981454517     Monetary Cost weight = 0.5
     Accuracy score = 0.0743072439435         Accuracy weight = 2

         Total Cost = -7953.73684538
```

```
Number 20 Alternative 47 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1321.67008684 g
    Total Unit Cost = 19580.0 usd
    Total Power = 7.565 W
    Accuracy = 46.1498193529 ft
    Total Size = 1245.77136599 cm^3
Alternative 47 - range
      Latency score =       0                 Latency weight = 0
   Robustness score =  0.3392              Robustness weight = 1
         Size score = 0.864877771514             Size weight = 1
         Mass score = -35.4449931028             Mass weight = 1
        Power score = -21.7761559039            Power weight = 1
Monetary Cost score = -20078.2047403    Monetary Cost weight = 0.5
     Accuracy score = 0.799677142116         Accuracy weight = 2

         Total Cost = -10093.5200871


Number 21 Alternative 37 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 996.65 g
    Total Unit Cost = 17350.0 usd
    Total Power = 16.325 W
    Accuracy = 136.620784636 ft
    Total Size = 1474.39086052 cm^3
Alternative 37 - range
      Latency score =       0                 Latency weight = 0
   Robustness score =  0.3392              Robustness weight = 1
         Size score = 0.761157415813             Size weight = 1
         Mass score = -0.471346141699            Mass weight = 1
        Power score = -10346.762496             Power weight = 1
Monetary Cost score = -4110.68185785    Monetary Cost weight = 0.5
     Accuracy score = 0.178513097092         Accuracy weight = 2

         Total Cost = -12401.1173874
```

```
Number 22 Alternative 44 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1579.67008684 g
    Total Unit Cost = 19095.0 usd
    Total Power = 15.8 W
    Accuracy = 46.1498193529 ft
    Total Size = 1830.50091169 cm^3
Alternative 44 - range
      Latency score =       0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.597658453623              Size weight = 1
         Mass score = -276.660777899              Mass weight = 1
        Power score = -8208.67411525             Power weight = 1
Monetary Cost score = -14576.1507799     Monetary Cost weight = 0.5
     Accuracy score = 0.799677142116          Accuracy weight = 2

         Total Cost = -15770.8740704


Number 23 Alternative 41 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1311.67008684 g
    Total Unit Cost = 20830.0 usd
    Total Power = 10.64 W
    Accuracy = 41.5829645838 ft
    Total Size = 1420.77136599 cm^3
Alternative 41 - range
      Latency score =       0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.786606752055              Size weight = 1
         Mass score = -32.2403677728              Mass weight = 1
        Power score = -425.131389791             Power weight = 1
Monetary Cost score = -43515.9678128     Monetary Cost weight = 0.5
     Accuracy score = 0.831713199297          Accuracy weight = 2

         Total Cost = -22212.5664308
```

```
Number 24 Alternative 38 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1569.67008684 g
    Total Unit Cost = 20345.0 usd
    Total Power = 18.875 W
    Accuracy = 41.5829645838 ft
    Total Size = 2005.50091169 cm^3
Alternative 38 - range
      Latency score =      0                      Latency weight = 0
   Robustness score =  0.3392                  Robustness weight = 1
         Size score = 0.52954323702                  Size weight = 1
         Mass score = -258.333620584                 Mass weight = 1
        Power score = -28424.9062129                Power weight = 1
Monetary Cost score = -32502.5512416     Monetary Cost weight = 0.5
     Accuracy score = 0.831713199297             Accuracy weight = 2

         Total Cost = -44931.9832846


Number 25 Alternative 15 - dted
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 846.796187248 g
    Total Unit Cost = 24400.0 usd
    Total Power = 13.4 W
    Accuracy = 38.4859958032 ft
    Total Size = 942.53872715 cm^3
Alternative 15 - dted
      Latency score =      0                      Latency weight = 0
   Robustness score = -0.2599                  Robustness weight = 1
         Size score = 0.962279212546                 Size weight = 1
         Mass score = 0.0884478780655                Mass weight = 1
        Power score = -2489.43152509                Power weight = 1
Monetary Cost score = -287488.410037     Monetary Cost weight = 0.5
     Accuracy score = 0.853171821715             Accuracy weight = 2

         Total Cost = -146231.139373
```

```
Number 26 Alternative 3 - height
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 846.796187248 g
    Total Unit Cost = 24400.0 usd
    Total Power = 13.4 W
    Accuracy = 155.060150829 ft
    Total Size = 942.53872715 cm^3
Alternative 3 - height
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.962279212546              Size weight = 1
         Mass score = 0.0884478780655             Mass weight = 1
        Power score = -2489.43152509             Power weight = 1
Monetary Cost score = -287488.410037     Monetary Cost weight = 0.5
     Accuracy score = 0.0743072439435         Accuracy weight = 2

         Total Cost = -146232.455602



Number 27 Alternative 18 - dted
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 588.796187248 g
    Total Unit Cost = 24885.0 usd
    Total Power = 5.165 W
    Accuracy = 39.9401484248 ft
    Total Size = 357.809181447 cm^3
Alternative 18 - dted
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.2599               Robustness weight = 1
         Size score = 0.997104865793              Size weight = 1
         Mass score = 0.363230346757              Mass weight = 1
        Power score = -0.108567525228            Power weight = 1
Monetary Cost score = -360902.199456     Monetary Cost weight = 0.5
     Accuracy score = 0.843140973131          Accuracy weight = 2

         Total Cost = -180448.421578
```

```
Number 28 Alternative 6 - height
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 588.796187248 g
    Total Unit Cost = 24885.0 usd
    Total Power = 5.165 W
    Accuracy = 156.499507186 ft
    Total Size = 357.809181447 cm^3
Alternative 6 - height
      Latency score =       0                 Latency weight = 0
   Robustness score = -0.0184              Robustness weight = 1
         Size score = 0.997104865793             Size weight = 1
         Mass score = 0.363230346757             Mass weight = 1
        Power score = -0.108567525228           Power weight = 1
Monetary Cost score = -360902.199456     Monetary Cost weight = 0.5
     Accuracy score = 0.0677671059133        Accuracy weight = 2

         Total Cost = -180449.730826



Number 29 Alternative 31 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 996.65 g
    Total Unit Cost = 24900.0 usd
    Total Power = 11.05 W
    Accuracy = 140.86174468 ft
    Total Size = 1073.62836052 cm^3
Alternative 31 - range
      Latency score =       0                 Latency weight = 0
   Robustness score =  0.3392              Robustness weight = 1
         Size score = 0.927837164726             Size weight = 1
         Mass score = -0.471346141699            Mass weight = 1
        Power score = -573.577128407            Power weight = 1
Monetary Cost score = -363415.527031     Monetary Cost weight = 0.5
     Accuracy score = 0.148533253815         Accuracy weight = 2

         Total Cost = -182280.247886
```

```
Number 30 Alternative 34 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 738.65 g
    Total Unit Cost = 25385.0 usd
    Total Power = 2.815 W
    Accuracy = 140.86174468 ft
    Total Size = 488.898814813 cm^3
Alternative 34 - range
      Latency score =       0                Latency weight = 0
   Robustness score =  0.3392             Robustness weight = 1
         Size score = 0.996739946731            Size weight = 1
         Mass score = 0.187592737828            Mass weight = 1
        Power score = 0.660330619543           Power weight = 1
Monetary Cost score = -453506.010151   Monetary Cost weight = 0.5
     Accuracy score = 0.148533253815        Accuracy weight = 2

         Total Cost = -226750.524146


Number 31 Alternative 1 - height
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 636.65 g
    Total Unit Cost = 14900.0 usd
    Total Power = 10.55 W
    Accuracy = 380.167188867 ft
    Total Size = 698.628360515 cm^3
Alternative 1 - height
      Latency score =       0                Latency weight = 0
   Robustness score = -0.0184             Robustness weight = 1
         Size score = 0.993340250398            Size weight = 1
         Mass score = 0.305250419409            Mass weight = 1
        Power score = -397.221316666           Power weight = 1
Monetary Cost score = -458.759392896   Monetary Cost weight = 0.5
     Accuracy score = -116193.007788        Accuracy weight = 2

         Total Cost = -233011.336398
```

```
Number 32 Alternative 7 - height
    Optical Camera - Sony FCB-EX780B
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 646.65 g
    Total Unit Cost = 6100.0 usd
    Total Power = 12.75 W
    Accuracy = 380.167188867 ft
    Total Size = 924.390860515 cm^3
Alternative 7 - height
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.966032268547              Size weight = 1
         Mass score = 0.293100984246              Mass weight = 1
        Power score = -1719.31320675             Power weight = 1
Monetary Cost score = 0.730919668802     Monetary Cost weight = 0.5
     Accuracy score = -116193.007788          Accuracy weight = 2

         Total Cost = -234103.722589



Number 33 Alternative 10 - height
    Optical Camera - Sony FCB-EX780B
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 388.65 g
    Total Unit Cost = 6585.0 usd
    Total Power = 4.515 W
    Accuracy = 381.866914336 ft
    Total Size = 339.661314813 cm^3
Alternative 10 - height
      Latency score =       0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 0.997364720891              Size weight = 1
         Mass score = 0.617435737801              Mass weight = 1
        Power score = 0.178830144806             Power weight = 1
Monetary Cost score = 0.668805972374     Monetary Cost weight = 0.5
     Accuracy score = -122495.863641          Accuracy weight = 2

         Total Cost = -244989.617647
```

```
Number 34 Alternative 4 - height
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 378.65 g
    Total Unit Cost = 15385.0 usd
    Total Power = 2.315 W
    Accuracy = 381.866914336 ft
    Total Size = 113.898814813 cm^3
Alternative 4 - height
      Latency score =      0                  Latency weight = 0
   Robustness score = -0.0184               Robustness weight = 1
         Size score = 1.00389240713               Size weight = 1
         Mass score = 0.632691837582              Mass weight = 1
        Power score = 0.806090668995             Power weight = 1
Monetary Cost score = -744.897127137     Monetary Cost weight = 0.5
     Accuracy score = -122495.863641          Accuracy weight = 2

         Total Cost = -245361.75157



Number 35 Alternative 45 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 1216.79618725 g
    Total Unit Cost = 25600.0 usd
    Total Power = 16.1 W
    Accuracy = 30.7181185291 ft
    Total Size = 1543.30122715 cm^3
Alternative 45 - range
      Latency score =      0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.728186493752              Size weight = 1
         Mass score = -12.1051336869              Mass weight = 1
        Power score = -9380.02130296             Power weight = 1
Monetary Cost score = -499401.947437     Monetary Cost weight = 0.5
     Accuracy score = 0.903934155774          Accuracy weight = 2

         Total Cost = -259090.2249
```

```
Number 36 Alternative 48 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 958.796187248 g
    Total Unit Cost = 26085.0 usd
    Total Power = 7.865 W
    Accuracy = 30.7181185291 ft
    Total Size = 958.571681447 cm^3
Alternative 48 - range
      Latency score =       0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.958756497967              Size weight = 1
         Mass score = -0.187393066903             Mass weight = 1
        Power score = -31.5693608348             Power weight = 1
Monetary Cost score = -618345.13764      Monetary Cost weight = 0.5
     Accuracy score = 0.903934155774          Accuracy weight = 2

         Total Cost = -309201.219749


Number 37 Alternative 25 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 986.65 g
    Total Unit Cost = 26150.0 usd
    Total Power = 14.125 W
    Accuracy = 136.620784636 ft
    Total Size = 1248.62836052 cm^3
Alternative 25 - range
      Latency score =       0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.86369213718               Size weight = 1
         Mass score = -0.379120119438             Mass weight = 1
        Power score = -3665.09234894             Power weight = 1
Monetary Cost score = -636055.286909     Monetary Cost weight = 0.5
     Accuracy score = 0.178513097092          Accuracy weight = 2

         Total Cost = -321691.555005
```

```
Number 38 Alternative 28 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Procerus Kestrel Autopilot
Totals
    Total Mass = 728.65 g
    Total Unit Cost = 26635.0 usd
    Total Power = 5.89 W
    Accuracy = 136.620784636 ft
    Total Size = 663.898814813 cm^3
Alternative 28 - range
      Latency score =       0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.994854922969              Size weight = 1
         Mass score = 0.198082054278              Mass weight = 1
        Power score = -1.34550382957             Power weight = 1
Monetary Cost score = -783021.460455     Monetary Cost weight = 0.5
     Accuracy score = 0.178513097092          Accuracy weight = 2

         Total Cost = -391510.186568


Number 39 Alternative 39 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 1206.79618725 g
    Total Unit Cost = 26850.0 usd
    Total Power = 19.175 W
    Accuracy = 23.5284630837 ft
    Total Size = 1718.30122715 cm^3
Alternative 39 - range
      Latency score =       0                  Latency weight = 0
   Robustness score =  0.3392               Robustness weight = 1
         Size score = 0.646397036823              Size weight = 1
         Mass score = -10.8115062029              Mass weight = 1
        Power score = -31675.3034738             Power weight = 1
Monetary Cost score = -857271.439676     Monetary Cost weight = 0.5
     Accuracy score = 0.942111850797          Accuracy weight = 2

         Total Cost = -460318.964997
```

```
Number 40 Alternative 42 - range
    Optical Camera - Sony FCB-EX780B
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 948.796187248 g
    Total Unit Cost = 27335.0 usd
    Total Power = 10.94 W
    Accuracy = 23.5284630837 ft
    Total Size = 1133.57168145 cm^3
Alternative 42 - range
      Latency score =       0                 Latency weight = 0
   Robustness score =  0.3392              Robustness weight = 1
         Size score = 0.907985985216             Size weight = 1
         Mass score = -0.137309861628            Mass weight = 1
        Power score = -530.102973109            Power weight = 1
Monetary Cost score = -1048109.25932    Monetary Cost weight = 0.5
     Accuracy score = 0.942111850797         Accuracy weight = 2

         Total Cost = -524581.738531


Number 41 Alternative 32 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1569.67008684 g
    Total Unit Cost = 27895.0 usd
    Total Power = 13.6 W
    Accuracy = 46.1498193529 ft
    Total Size = 1604.73841169 cm^3
Alternative 32 - range
      Latency score =       0                 Latency weight = 0
   Robustness score =  0.3392              Robustness weight = 1
         Size score = 0.698942887002             Size weight = 1
         Mass score = -258.333620584             Mass weight = 1
        Power score = -2776.94306194            Power weight = 1
Monetary Cost score = -1314516.54335    Monetary Cost weight = 0.5
     Accuracy score = 0.799677142116         Accuracy weight = 2

         Total Cost = -660290.910862
```

```
Number 42 Alternative 35 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1311.67008684 g
    Total Unit Cost = 28380.0 usd
    Total Power = 5.365 W
    Accuracy = 46.1498193529 ft
    Total Size = 1020.00886599 cm^3
Alternative 35 - range
      Latency score =       0              Latency weight = 0
   Robustness score =  0.3392           Robustness weight = 1
         Size score = 0.943463740874          Size weight = 1
         Mass score = -32.2403677728          Mass weight = 1
        Power score = -0.300112218066        Power weight = 1
Monetary Cost score = -1592023.48518   Monetary Cost weight = 0.5
     Accuracy score = 0.799677142116      Accuracy weight = 2

         Total Cost = -796041.401054


Number 43 Alternative 26 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1559.67008684 g
    Total Unit Cost = 29145.0 usd
    Total Power = 16.675 W
    Accuracy = 41.5829645838 ft
    Total Size = 1779.73841169 cm^3
Alternative 26 - range
      Latency score =       0              Latency weight = 0
   Robustness score =  0.3392           Robustness weight = 1
         Size score = 0.619249227191          Size weight = 1
         Mass score = -241.044615768          Mass weight = 1
        Power score = -12014.0823288         Power weight = 1
Monetary Cost score = -2135898.94997   Monetary Cost weight = 0.5
     Accuracy score = 0.831713199297      Accuracy weight = 2

         Total Cost = -1080201.98006
```

```
Number 44 Alternative 29 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Crossbow NAV420
Totals
    Total Mass = 1301.67008684 g
    Total Unit Cost = 29630.0 usd
    Total Power = 8.44 W
    Accuracy = 41.5829645838 ft
    Total Size = 1195.00886599 cm^3
Alternative 29 - range
      Latency score =      0             Latency weight = 0
   Robustness score =  0.3392         Robustness weight = 1
         Size score = 0.8852579977          Size weight = 1
         Mass score = -29.2826182804        Mass weight = 1
        Power score = -60.3440720719       Power weight = 1
Monetary Cost score = -2560616.34649   Monetary Cost weight = 0.5
     Accuracy score = 0.831713199297     Accuracy weight = 2

         Total Cost = -1280394.91205


Number 45 Alternative 33 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 1206.79618725 g
    Total Unit Cost = 34400.0 usd
    Total Power = 13.9 W
    Accuracy = 30.7181185291 ft
    Total Size = 1317.53872715 cm^3
Alternative 33 - range
      Latency score =      0             Latency weight = 0
   Robustness score =  0.3392         Robustness weight = 1
         Size score = 0.83401679559         Size weight = 1
         Mass score = -10.8115062029        Mass weight = 1
        Power score = -3259.38625687       Power weight = 1
Monetary Cost score = -12803872.0408   Monetary Cost weight = 0.5
     Accuracy score = 0.903934155774     Accuracy weight = 2

         Total Cost = -6405203.23708
```

```
Number 46 Alternative 36 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Thales Miniature Eyesafe Laser Transceiver (MELT)
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 948.796187248 g
    Total Unit Cost = 34885.0 usd
    Total Power = 5.665 W
    Accuracy = 30.7181185291 ft
    Total Size = 732.809181447 cm^3
Alternative 36 - range
      Latency score =      0                 Latency weight = 0
   Robustness score =  0.3392             Robustness weight = 1
         Size score = 0.991269374758            Size weight = 1
         Mass score = -0.137309861628           Mass weight = 1
        Power score = -0.773824364882          Power weight = 1
Monetary Cost score = -14851994.6486   Monetary Cost weight = 0.5
     Accuracy score = 0.903934155774        Accuracy weight = 2

         Total Cost = -7425995.09709


Number 47 Alternative 27 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Roke Manor Miniature Radar Altimeter Mk IVa
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 1196.79618725 g
    Total Unit Cost = 35650.0 usd
    Total Power = 16.975 W
    Accuracy = 23.5284630837 ft
    Total Size = 1492.53872715 cm^3
Alternative 27 - range
      Latency score =      0                 Latency weight = 0
   Robustness score =  0.3392             Robustness weight = 1
         Size score = 0.752484361341            Size weight = 1
         Mass score = -9.63531434258            Mass weight = 1
        Power score = -13615.231605            Power weight = 1
Monetary Cost score = -18676348.0093   Monetary Cost weight = 0.5
     Accuracy score = 0.942111850797        Accuracy weight = 2

         Total Cost = -9351795.89564
```

```
Number 48 Alternative 30 - range
    Optical Camera - Rockwell Scientific UAV-Cam V2M
    Laser Rangefinder - Vectronix LRF42
    Altimeter - Honeywell Precision Altimeter HPA
    Autopilot - Anthena Controls GS-111m
Totals
    Total Mass = 938.796187248 g
    Total Unit Cost = 36135.0 usd
    Total Power = 8.74 W
    Accuracy = 23.5284630837 ft
    Total Size = 907.809181447 cm^3
Alternative 30 - range
      Latency score =       0              Latency weight = 0
   Robustness score =  0.3392           Robustness weight = 1
         Size score = 0.969244696689          Size weight = 1
         Mass score = -0.0950177482991        Mass weight = 1
        Power score = -82.2696288707         Power weight = 1
Monetary Cost score = -21531468.6705   Monetary Cost weight = 0.5
     Accuracy score = 0.942111850797      Accuracy weight = 2

         Total Cost = -10765813.5072
```

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Anonymous. AeroVironment SUAV Website. http://www.avsuav.com, March 2006.

[2] Anonymous. Johnson's Criteria. Wikipedia, the free encyclopdia:, March 2006. http://en.wikipedia.org/wiki/Johnson's_Criteria.

[3] Anonymous. Pareto front. Wikipedia, the free encyclopdia, March 2006. http://en.wikipedia.org/wiki/Pareto_front.

[4] Anonymous. Pressure Altitude, March 2006. http://www.srh.noaa.gov/elp/wxcalc/formulas/pressureAltitude.html.

[5] Anonymous. Properties of the U.S Standard Atmosphere 1976. Public Domain Aeronautical Software website, March 2006. http://www.pdas.com/atmos.htm.

[6] Anonymous. RQ-11 Raven, March 2006. http://www.globalsecurity.org/intell/systems/raven.html.

[7] Anonymous. Samuel Pierpont Langley. Wikipedia, the free encyclopdia, March 2006. http://en.wikipedia.org/wiki/Samuel_Pierpont_Langley.

[8] Anonymous. Strategic Management: SWOT Analysis. QuickMBA, March 2006. http://www.mindtools.com/pages/article/newTMC_05.htm.

[9] Anonymous. Strategic Management: SWOT Analysis, March 2006. NetMBA: http://www.netmba.com/strategy/swot.

[10] Anonymous. SWOT Analysis: Lesson, March 2006. http://www.marketingteacher.com/Lessons/lesson_swot.htm.

[11] Anonymous. Unmanned Aerial Vehicles and Drones, March 2006. http://www.aviationnow.com/media/pdf/spec05_uav.pdf.

[12] Anonymous. Wright Brothers. Wikipedia, the free encyclopdia, March 2006. http://en.wikipedia.org/wiki/Wright_Brothers.

[13] Anthony C. Danca. SWOT Analysis, March 2006. http://www.stfrancis.edu/ba/ghkickul/stuwebs/btopics/works/swot.html.

[14] Benjamin S. Blanchard and Wolter J. Fabrycky. *Systems Engineering and Analysis*. Industrial and Systems Engineering. Prentice Hall, Upper Saddle River, New Jersey 07458, Fourth edition, 2006.

[15] Brigadier General Mike Cannon. PEO Tactical Missles Sensor to Shooter Timeline Equation. Sensor-to-Shooter and Time Critical Targeting 2005 Conference, February 2006.

[16] Edmund H. Conrow. AWeb Site Devoted to Government Policy and References to Cost as an Independent Variable CAIV, 1998. http://www.caiv.com.

[17] E. Crawley, B. Koo, W. Hostetter, and R. Boas. System Architecture Evaluation Metrics and Tools Development. Technical Presentation, September 2004.

[18] Daniel Schrage, Todd Beltracchi, Laszlo Berke, Alan Dodd, Larry Niedling, and Jarosław Sobieski. AIAA Technical Committee on Multidisciplinary Design Optimization: White Paper on the Current State of the Art. AIAA White Paper, AIAA, January 1991.

[19] EMX Inc. Human Detection at Various Ranges with Different Lenses, March 2006. www.emx-inc.com.

[20] James Bethel, Dominick Andrisani II, Aaron Braun, Ade Mulyana, and Takayuki Hoshizaki. Methods to Improve Airborne Imagery Geopositioning. Precision Target Geolocation from Airborne Platforms - Workshop III, November 2002.

[21] James Manktelow. SWOT Analysis: Discover new opportunites. Manage and eliminate threats., March 2006.
http://www.mindtools.com/pages/article/newTMC_05.htm.

[22] Michael Mok. Precision Guided Munitions, March 2006.
http://www.georgetown.edu/sfs/programs/stia/students/vol.03/mok_pgm.htm.

[23] Department of Defense Systems Management College. *Systems Engineering Fundamentals*. United States of America Department of Defense, January 2001.

[24] University of Wisconsin Center for Cooperatives. Cooperatives: A Tool for Community Economic Development. Chapter 5: Conducting a Feasibility Study, March 2006. http://www.wisc.edu/uwcc/manual/chap_5.html.

[25] Olivier de Weck. Multidisciplinary Design Optimization. Technical presentation given at Charles Stark Draper Laboratory, September 2005.

[26] Olivier de Weck and Karen Willcox. Multidisciplinary System Design Optimization: Introduction. Open Courseware at Massachusetts Institute of Technology, March 2006. http://ocw.mit.edu/OcwWeb/Aeronautics-and-Astronautics/16-888Spring-2004/CourseHome/index.htm.

[27] Associated Press. Lockheed Awarded Contract: Defense Department Taps Lockheed-Martin for $200b Joint Strike Fighter. CNN Money Website, October 2001. http://money.cnn.com/2001/10/26/companies/strikefighter/.

[28] Randall S. Hansen and Katharine Hansen. Using a SWOT in Your Career Planning, March 2006.
http://www.quintcareers.com/strategy/SWOT_Analysis.html.

[29] Richard P. Hallion. Precision Guided Munitions and the New Era of Warfare, March 2006. http://www.fas.org/man/dod-101/sys/smart/docs/paper53.htm.

[30] Lombardo Technical Services. Target Acquisition, 2003.

[31] Defense Acquisition University. *Defense Acquisition Handbook.* United States of America Department of Defense, November 2004.