TOOLS FOR EVOLUTIONARY ACQUISITION: A STUDY OF MULTI-ATTRIBUTE TRADESPACE EXPLORATION (MATE) APPLIED TO THE SPACE BASED RADAR (SBR)

by

2nd Lieutenant Timothy J. Spaulding, USAF

B.S. Physics
United States Air Force Academy, 2001

M.P.P. International Security and Political Economy
Harvard University, John F. Kennedy School of Government, 2003

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 2003

© 2003 Massachusetts Institute of Technology
All rights reserved

Signature of Author………………………………………………………………………………
Department of Aeronautics and Astronautics
9 May 2003

Certified by…………………………………………………………………………………………
Eric Rebentisch
Research Associate, Center for Technology, Policy, and Industrial Development
Thesis Supervisor

Certified by…………………………………………………………………………………………
Debbie Nightingale
Professor of the Practice, Department of Aeronautics and Astronautics
Aeronautics and Astronautics Reader

Accepted by…………………………………...……………………………………………………
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Opinions, conclusions, and recommendations expressed or implied within are solely those of the author and do not necessarily represent the views of the Air Education and Training Command, the United States Air Force, the Department of Defense, or any other US government agency.

TOOLS FOR EVOLUTIONARY ACQUISITION: A STUDY OF MULTI-ATTRIBUTE TRADESPACE EXPLORATION (MATE) APPLIED TO THE SPACE BASED RADAR (SBR)

by

2nd Lieutenant Timothy J. Spaulding, USAF

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

ABSTRACT

The Multi-Attribute Tradespace Exploration (MATE) process was applied to the Space Based Radar (SBR), a space system under study by the United States Air Force. A system-level model of possible SBR architectures was created using data and analysis from previous high-level studies. Competing designs were evaluated through MATE's universal utility metric.

The MATE model was qualitatively compared against a high-level design study and MATE's advantages were noted, specifically its ability to trace modeling assumptions and present a holistic view of the space of competing designs. A quantitative comparison revealed significant differences between MATE's recommended system design and that of the comparison high-level study.

The potential for a simplification of the MATE method was explored through the use of several approximations to revealed user preferences. Comparisons were made through both a proportional utility loss metric and a general Spearman's Rho rank order correlation. Using these measures it was shown that while a linear or subjective approximation to utility curves resulted in excessive errors, and approximation to weighting relationships did not.

Finally, MATE's potential applicability to the Air Force acquisition process was studied. In general MATE was shown to be useful to any acquisition effort that derives its benefit from a networked approach and is of sufficient technical complexity as to make tradeoff decisions opaque to casual analysis. Specifically, MATE was shown to be useful in the analysis of alternatives process as well as an aid to early milestone sourcing decisions.

Thesis supervisor: Dr. Eric Rebentisch
Title: Research Associate, Center for Technology, Policy, and Industrial Development

BIOGRAHICAL NOTE

Tim Spaulding was born on June 14, 1979 in Stillwater, Minnesota. Moving around while young, Tim grew up mainly in Centennial, Colorado, a suburb south of Denver. It was there that he was first exposed to the Air Force Academy, where he enrolled in June 1997 after graduating from Aledo High School outside of Fort Worth, Texas. He earned his commission as a 2nd Lieutenant upon graduating from the Academy in May, 2001.

As a cadet, Tim majored in physics and minored in math, specializing in problems involving the intersection between wireless communication and space physics. Named the 2001 outstanding physics major, Tim was also selected in 2000 for the Truman Scholarship, a nationwide graduate scholarship given to students who are pursuing careers in public service.

While not thinking about physics, Tim was a cadet instructor at the Academy's 94th Flying Training Squadron, teaching underclass cadets how to fly the Schweitzer 2-33 sailplane. In the fall of 2000, he served as the Wing Commander of the Air Force Academy's 4000 member cadet wing—the top cadet leadership position at the Academy.

After graduating, Tim earned his private pilot's license before beginning study for a Master's in Public Policy degree at Harvard University's Kennedy School of Government where he focused on national and international security studies. While a student at the Kennedy School Tim also took classes at MIT, following the example of recent Academy graduate Nathan Diller. After splitting his summer between an internship at MIT's Lincoln Lab and a position with the Air Force Scientific Advisory board, Tim was offered a research assistantship with MIT's Lean Aerospace Initiative. Working in this capacity, he was accepted as a Master of Science candidate in the Aeronautics and Astronautics department at MIT the next semester.

The Air Force is the common thread between Tim's research at Harvard and MIT—specifically, the Air Force's efforts to embrace and implement a strategy of "evolutionary development." At the Kennedy School Tim investigated the problems of budgeting for such a development strategy, while at MIT he investigated how tools of systems engineering could be harnessed for the same efforts.

Before graduating from Harvard and MIT, Tim will do the one thing that can top both—get married to Ms. Dorothy Ann Mackay, whom he was introduced to while a cadet. After their wedding on May 31st 2003, Tim and Dori are moving to Sheppard Air Force Base in Wichita Falls, Texas where Dori will pursue her teacher's certification while Tim attends Euro-NATO Joint Jet Pilot Training.

ACKNOWLEDGEMENTS

Others' contributions to my research effort have been both varied and extensive. What follows is only a partial list of the support I have received—I offer my deepest apologies to anyone not mentioned here.

My first thanks go to the Lean Aerospace Initiative, whose support in office space, technical resources, and financial assistance were essential to the completion of this thesis. LAI was not only generous in this support, but was also willing to extend it before I was officially accepted as an MIT student.

Dr. Eric Rebentisch of LAI has been unwavering in his support for me and my research, despite that fact that his plate was already full before he suddenly found himself mentoring yet another student. His efforts on behalf of all of his students at LAI have been exemplary.

The ACE group at LAI has been an invaluable source of feedback and discussion. Jason Derleth, Bobak Ferdowsi, and Chris Roberts made contributions to understanding and development that are evident throughout the research. Nirav Shah lent his never-to-be-underestimated technical abilities to my efforts. It is likely that I would have been in dire straits without his considerable and timely assistance. Finally, I am indebted to Adam Ross and Nathan Diller, whose visions of MATE have provided fertile ground for mine (and many others') research efforts. Furthermore, Nathan's trailblazing efforts meant that my path through Harvard and MIT was far smoother than it would have been otherwise.

At Lincoln Lab I am indebted to Dr. Joe Chapa, who was willing to take me on for a summer internship. His broad experience in engineering and Air Force program management gave me a greater perspective on my efforts on both lines of research. Mr. Bob Coury's help was also invaluable. His willingness to both share his RPAT software tool and provide support for its use was remarkable and critical to my efforts. I am also grateful to Mr. Larry Tonneson for his willingness to take utility interviews and develop a set of system attributes for the Space Based Radar. Dr. Tony Phillip and Mr. Robert Harvey also provided key pieces of information for the construction of the MATE model. My Lincoln Lab internship experience was filled with these and many others whose professionalism and dedication to the pursuit of knowledge led them to assist me in a number of ways.

My thanks also go to Dr. Ray Sedwick, whose expertise in space systems helped complete the final parts of the radar coverage model. Also, I am grateful for the teaching team of Dr. Joyce Warmkessel, Dr. Hugh McManus, and Dr. Daniel Hastings, whose leadership of the 16.89 Space Systems Design course provided me the detailed experience to embark on a MATE analysis of the Space Based Radar.

None of these pages would have been written were it not for Dr. Joyce Warmkessel, whose vision permeates my and many others' work at MIT. My debt to her guidance is unique since it was her personal efforts that brought me to MIT, found me a home at LAI, and formed the very basis, direction, and goals of my research. Even while falling ill, Joyce was personally dedicated to seeing that I received the support and direction that I needed. Joyce, you are sorely missed.

The support I received from the Air Force was crucial as well. Captain Melissa Flattery initially enabled me to branch my efforts out to MIT, and 1st Lieutenant Angela Bjorge continued this support through her professional, courteous, and rapidly responsive administrative help. I am also in the debt of Captain Kent Broome of the Air Force Scientific Advisory Board, who not only enabled me to gain a world of experience while working with him at the board, but also helped me get into MIT.

My family's support during my time in Cambridge has been unconditional, even if it meant that phone calls and e-mails home were sometimes few and far between. At a deeper level, I am eternally grateful for having grown up in a home with parents who loved both each other and their children, and who dedicated themselves to seeing them succeed. I attribute any success I have had in life to having been raised in such an environment.

Finally, my sincere thanks go to the person who has the most direct and personal experiences with the downsides of a busy research agenda—my soon-to-be wife Dori. Dori not only happily coped with a fiancé who had two demanding mistresses named "Harvard" and "MIT," but performed admirable duty as both travel agent and taxi service. As if this weren't enough, she gladly submitted to being a test subject for a MATE analysis and happily proofread an untold number of arcane policy memos and term papers, pointing out and removing superfluous commas without a word of complaint. Dori kept me on track and in high spirits even when I was desperately trying to take myself too seriously. Thank you, Dori, for your support and encouragement—after this, pilot training should look like a vacation!

For Joyce

TABLE OF FIGURES

TABLE OF CONTENTS

1. Introduction and Motivation

The purpose of this thesis, broadly stated, is to further the state of knowledge regarding the evolutionary acquisition of weapons systems for the United States Air Force. With that umbrella objective in mind, it attacks a much narrower field of study related to the tools and processes that must be developed in order to enable the evolutionary acquisition strategy. Specifically, one such tool is examined: the Multi-Attribute Tradespace Exploration (MATE) method developed at the Massachusetts Institute of Technology by Adam Ross, Nathan Diller, Dr. Dan Hastings, Dr. Joyce Warmkessel, Dr. Hugh McManus, and others.

The MATE tool is applied to a system currently under study by the Air Force—the Space Based Radar (SBR). SBR was one of the "pathfinder" programs identified by the Air Force Acquisition Center of Excellence (ACE) office to be a candidate for evolutionary acquisition. As of this writing, the Space Based Radar is in the "analysis of alternatives" stage of development, which means competing system architectures are being evaluated, and the decision to go forward with full funding for detailed design (milestone 0 in the Air Force Acquisitions parlance) has yet to be made.

1.1 Evolutionary Acquisition

Before endeavoring to discuss an evolutionary acquisition strategy, it is important to clarify the terms used in the acquisition community, ensuring clear distinctions are drawn between competing visions of ideal design processes:

The standard model of acquisition is often called the “waterfall” or “top-down” method, and involves designing to a set of requirements defined as detailed design-work begins. These requirements are rigid so that the designers know what capabilities the product will posses before it is finished, at least along the dimensions specified by the requirements document.

An alternate means of acquisition is the “pre-planned product improvement” (P^3I) method. This allows for scheduled product upgrades, which amount to improvements on the original finished product. It is critical that these are usually small changes which can be accomplished quickly and do not affect the overall design of the product in any significant way. An example is upgrading a car’s engine to a higher performance version after the car has already been designed and built. Another key characteristic of this method is that these changes are “pre-planned,” which means designers foresaw the change when they designed the original.

Evolutionary acquisitions (EA), by contrast to P^3I, involves several full cycles of the traditional engineering process, each building on the last and providing some incremental capability. Products that function as networks are ready examples: evolutionary acquisition of a network of sensors might, in its first iteration of the design cycle consist of one sensor, placed in a strategic location. Further iterations would augment this capability, adding more or better sensors to the network. These iterations are commonly called “spirals” and are meant to be a full iteration of the engineering processes of design, construction, and testing. Each cycle is intended to be performed faster than the overall project would be, since each provides only an incremental capability.

This concept, that engineering work might be better accomplished in short, repeating cycles, is not a new one—proponents of this method have been thinking and writing for years. The initial and most successful applications of the method have been in software development (Highsmith, 2000).

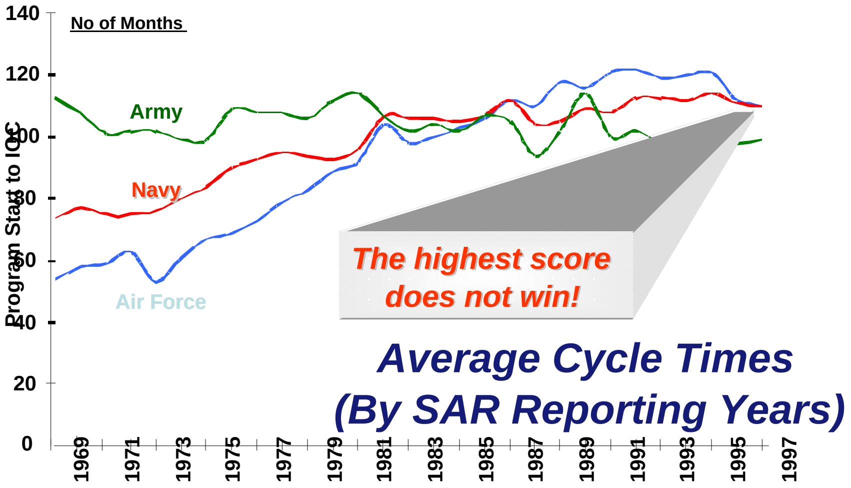
It is important here to note distinctions in the Air Force lexicon regarding this process. In various fields of study, evolutionary acquisition is referred to as spiral development, spiral acquisition, and evolutionary development. For the purposes of the Air Force acquisitions community and this thesis, “evolutionary acquisition” refers to the concept of developing a product in cycles rather than in one linear process. “Spiral development” refers to the process of executing one of these cycles. Using these definitions, there could be several episodes of spiral development in one evolutionarily acquired product (Alderidge, 2002).

There is one further helpful distinction regarding evolutionary acquisitions that involves designer knowledge about the end-state of the design. “Type I” evolutionary acquisition sets an end goal for the system, executing spiral developments to incrementally approach that goal. “Type II” does not set an end goal, instead executing each spiral development according to changing user needs. These might be called “goal driven” and “blind” strategies respectively. For further explanation of this distinction and its implication, see the work done by Chris Roberts (Roberts, 2003) on the subject.

Recent Acquisition Woes

The Air Force, as well as the Department of Defense as a whole, has been moving toward using an EA paradigm for some time. This move has been solidified by the recent memorandum released by Deputy Secretary of Defense Paul Wolfowitz. This memorandum replaced the Department's acquisitions regulations (called the 5000 series) with Secretary Wolfowitz's interim guidance. In it he writes "Evolutionary acquisition strategies shall be the preferred approach to satisfying operational needs. Spiral development shall be the preferred process" (Wolfowitz, 2002).

This move toward EA is a response to the perceived problems in traditional acquisitions, specifically the time it takes to field a complex product.



Source: DSB Briefing, Dan Czelusniak, 12 June 1998

Figure 1: Why Change is Needed (Little, 2000)

The chief of Air Force Acquisitions, Dr. Marv Sambur, recently noted: "On average, Air Force programs' cycle times run about 10 years, and that's only the average; some programs take up to 25 years to get to the field" (Paone, 2002). There is a further need to have more flexible acquisitions processes because the threat base is more fluid today that it was during the Soviet era.

Although there have been applications of this type of acquisitions process in the past, both in and out of the Air Force (Birker, 2000), the Air Force is the leader in institutionalizing such strategies. The Acquisition Center of Excellence, seeking to do just this, selected several programs to be pathfinders, pioneering the EA paradigm. Space Based Radar was one of these programs (Little, 2002).

Applicability of MATE

Having seen the need for acquisition change and the recent tilts toward EA as a way to institutionalize this change, what role does MATE have to play? Why should MATE be utilized by those trying to actualize EA? The answer comes in part from further comments by Dr. Sambur, who noted that the level of communication between major players often determined which programs succeeded and which were plagued with problems (Paone, 2002).

The MATE process was originally designed to enable just this kind of communication. Diller and Ross note that MATE seeks to remedy “limited incorporation of interdisciplinary expert opinion and diverse stakeholder interest,” as well as “disconnects between perceived and actual decision maker preferences” (Ross, 2002). As a universal metric, utility can serve as a “boundary object” between users and designers, opening a communication channel that ensures the right final product is created.

MATE has more advantages that make it ideal as a tool for EA. Many observers of EA efforts note stories where premature and rigid requirements definition to perverse and suboptimal development outcomes (Boehm, 2000).   MATE’s strength as an analysis tool is the ability to evaluate potential options with respect to flexible attributes instead of rigid requirements. Boehm cautions that in executing EA-like acquisitions, a risk exists that an architecture will be chosen that is "compatible with the users' expectations, but not with the customer's budget expectations" (Boehm, 2000). As will be seen in Chapter 4, MATE analysis yields precisely these relationships between a user's desires for capability and his or her willingness to pay. In this respect then, MATE is an ideal tool for enabling EA.

Further Reading

This thesis is limited in scope however, and so can only address a limited subset of the questions surrounding MATE's applicability for EA. These questions, developed in Chapter 2, are as follows:

1*) How does the MATE study of the Space Based Radar problem compare with traditional analysis of alternatives studies like the effort made at Lincoln Lab? Are the predictions equivalent? What are the similarities and differences?*

2) Should MATE be simplified to ease the stress placed on the decision-makers?

3) How might MATE be used in the requirements community? How might it account for the preferences of those who make system acquisition decisions?

Other research on the subject includes Chris Robert's study of the implications of using MATE over several design iterations, Bobak Ferdowsi's study about what types of systems are ideal candidates for EA, Jason Derleth's study on MATE applied to a non-space system, and Nirav Shah's work on using MATE to develop portfolios of design options. Each of these forthcoming MIT Master's theses are strongly recommended for the reader interested in further applications of the MATE method to the problem of EA.

1.2 Space Based Radar (SBR)

The concept of a ground-looking, space based radar is a direct descendent of the "Joint Surveillance and Target Attack Radar System" (commonly known as JSTARS). JSTARS is an airborne radar platform, based on the Boeing 707 airframe. Used for the first time during the 1991 Desert Storm operation, JSTARS provided Ground Movement Target Indication (GMTI) for commanders. This information was used to arrange air-strikes on enemy forces. The drawback of an airborne system is that aircraft have to get within radar range of the area of interest. This sometimes presents problems due to both airspace restrictions and the difficulty of arranging and sustaining long loiter times.

Accordingly, planners began to investigate the possibility of taking a similar radar antenna and placing it in orbit. This would resolve the airspace problems (allowing for surveillance deep inside hostile territory), and, given a sufficient constellation of satellites, ensure adequately persistent coverage in time.

This idea for a space based radar system has been floating in defense circles for many years, but has never gained enough support to become an acquisition program. Technical difficulties in launching and running a large, complicated radar antenna have served to slow the full scale funding of such a system. The recent efforts toward developing SBR may or may not find enough promise in the system to push it further into the acquisition process.

Recent Studies

There are two recent studies on Space Based Radar relevant to this thesis. The first is a 120-day utility study performed by the Joint C4ISR Decision Support Center (Keithly, 2001).[1] This study sought to understand how a space based radar would be useful to existing forces. After this study, the Chief of Staff and Secretary of the Air Force, along with the Undersecretary of the Air Force for Space, wanted further details and options, and therefore commissioned a summer study from Lincoln Lab. The Lincoln Lab study also focused on how a system could be useful to existing forces, but examined scenarios for use in far more technical detail. A summary of the study can be found in Chapter 4. At this time of this writing, the Space Based Radar program remains in the analysis of alternatives phase, awaiting further definition and funding.

---

[1] It is important here to note the difference between the way the Decision Support Center uses the term "utility," and what is meant by the more rigorous definition used in utility theory. Apart from referring to the 120-day study, the more rigorous definition is intended.

2. Literature Review

2.1 MATE

MATE History

The single best source of information for a reader interested in MATE is "The MATE Book" (Ross, 2002). Another helpful reference is Nathan Diller's MIT Master's thesis (Diller, 2002). The treatment here will differ from theirs in two important ways. First, it will not approach the level of detail found on those documents. Secondly, it will focus only on the part of MATE dealing with high level architecture studies. Though much of the recent work with MATE involves integration with concurrent engineering (MATE-CON), only the architecture level aspects of the MATE process are germane to this thesis.

MATE's development began with system analysis work done in the MIT Space Systems Lab, which was eventually embodied in a process called Generalized Information Network Analysis (GINA). GINA's goal was to model satellites as information networks, focusing especially on distributed satellite systems (Shaw, 1999). Accordingly, GINA used metrics appropriate for information systems to construct a tradespace of possible designs. Systems engineers could then explore these spaces, finding optimal trades between various metrics and cost.

This work was taken up by students in an MIT graduate course, "Space Systems Design" (16.89), jointly taught by Dr. Dan Hastings, Dr. Joyce Warmkessel, and Dr. Hugh McManus. The students in the course sought to apply the GINA process to an ionospheric sensing mission for Air Force Research Labs' Space Vehicles Directorate. In the course of application, however, the GINA metrics seemed ill-suited for the sensing mission. A need to evaluate a different set of metrics became apparent. Though the students completed their analysis on the mission which they named “A-TOS,” the results were not satisfying.

The next year’s course picked up on a similar mission, again attempting to use GINA. During this iteration, called B-TOS, a pair of students, Adam Ross and Nathan Diller, began to develop a method to include various metrics that the user felt were important. The method was an application of the multi-attribute utility theory (MAUT), a common tool in the fields of economics, operations research, and decision analysis. Utility theory itself has arisen only fairly recently (it was introduced in the work of von Nuemann and Morgenstern, 1947). The multi-attribute version is a late 20th century phenomenon, with its most lucid expositors in Ralph Keeney and Howard Raiffa. Their book, Decisions with Multiple Objectives, originally printed in 1973, still forms the bedrock of the multi attribute utility cannon. From this background—GINA and MAUT, Diller and Ross laid out the groundwork for MATE.

The first full MATE application was the X-TOS project, which was the third iteration of analysis on the same type of ionospheric mapping mission. Here students in the 16.89 course applied the MATE method, creating a tradespace of potential system architectures that traded off utility (as measured by Keeney and Raiffa’s multi attribute utility theory) against life-cycle cost. Having learned lessons about how to refine the procedures, Diller and Ross set about formalizing the process, eventually naming it MATE-CON. This formalization included both a high level architectural component and a further link to concurrent engineering processes. This most recent iteration of the 16.89

course, armed with the benefit of past experience and implementation tools, produced a detailed MATE study (Long, 2001). As a student on the "X-TOS" project, the author was exposed to MATE and the associated methodology.

MATE Process Overview

As mentioned above, this MATE review will only briefly discuss the finer points of MAUT and the MATE process steps. It will paint a general picture of how the process works, at a level of detail appropriate for general understanding. Graphically, applying MATE follows the steps in the figure below:

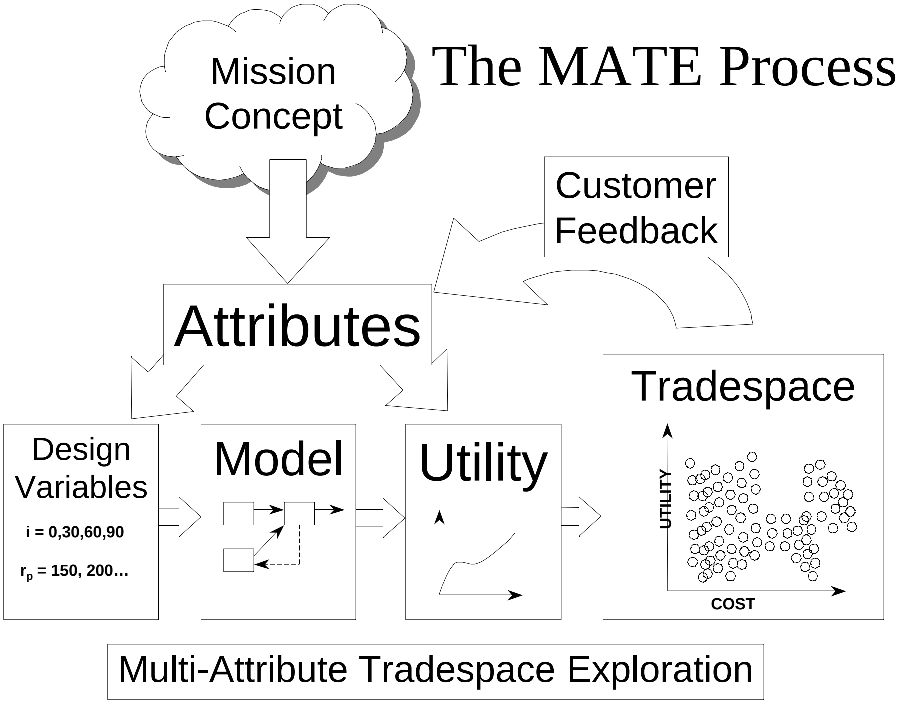


Figure 2: The MATE process

Before further elucidation of this graph, it is helpful to list a glossary of terms. Some of these terms appear in the general systems engineering lexicon, but others are unique to MATE or have narrow and important definitions. The following list is adapted from Ross, 2002.

Objective: a decision maker-desired end state or outcome.

Attribute: a decision maker-perceived metric that measures how well a decision-maker defined objective is met.

Utility: a dimensionless parameter, ranging from zero to one, that reflects the "perceived value under uncertainty" of an attribute.

Multi-attribute Utility: a dimensionless parameter, ranging from zero to one, that reflects the value of an aggregation of single utility values.

Design Variable: a designer-controlled quantitative parameter. Typically these represent physical aspects of a design (i.e. mass)

Design Vector: a collection of design variables that represent the characteristics of a given system.

Architecture: a potential system, which is defined by its design vector—a unique combination of design variables

Tradespace: the set of all architectures under consideration.

With these definitions in mind, we can now examine figure 2, noting that the process breaks into several large activities: defining a set of attributes, choosing a set of design variables, creating a model that links the variables to the attributes, creating a set of utility curves, and actually evaluating each architecture with the model. On the roughest of levels, these steps are serial:

- choose attributes
- choose design variables
- link variables to attributes (create model)
- define utility curves
- evaluate architectures

Choosing attributes involves engaging the user and determining what aspects of system performance are truly important. This is often done through a top-down hierarchy process, wherein the user lists high level goals and works downward in detail from them. For instance, a high level goal of the SBR system is to provide high-resolution radar images of stationary ground targets. An attribute coming from that goal might be the level of maximum resolution of those images. The goal of this step is to find a short list of attributes (three to seven) that can be used as metrics in evaluating a proposed system.

Choosing design variables is a task performed by the system architects, who select a set of design “knobs” that they would like to examine. The potential system architectures come from combinations of the possible values of these design variables. For space systems, these design variables often include the number of satellites in a constellation and the orbital parameters of each. In the SBR context, there are also design variables that involve the radar itself—the aperture size for instance.

With these two lists in hand, system architects create models (usually parametric models) that link the combinations of design variables to the attributes. These links provide a model for predicting how well a given system will satisfy the user’s selected attributes. Using the above examples, one could imagine a model that gave the best resolution possible from a system with a given orbital altitude and radar aperture size. These models can often be adapted from existing technical analysis.

Defining the utility curves is the step in the process that is hereafter referred to as “preference elicitation.” Here the analysts explore the user’s preferences on his or her list of attributes. First maximally useful and minimally acceptable values for each attribute are decided upon. Then the user's preferences between these boundaries are evaluated using a utility theory tool called "lottery equivalent probability." This tool described in more detail in section 2.3.

Evaluating the architectures means assigning a multi-attribute utility score to each architecture based on how well it performs on each of the attributes. It is during this step that the mathematics of utility theory are exploited. In order for this operation to be axiomatically rigorous, each attribute's utility must be independent of the others (mutual utility independence) in order to allow for an aggregation of the single attribute utility measures (Keeney, 1976). In the MATE framework, this aggregation is multiplicative, and follows the following formula:

$$KU(\underline{X})+1=\prod_{i=1}^{N}\left(Kk_iU(X_i)+1\right)$$

Where

$U(\underline{X})$ = *multi attribute utility*
$U(X_i)$ = *utility of attribute i*
k_i = *weighting factor for attribute i*
K= *overall weighting factor*

Figure 3: Multiplicative Multi-attribute Utility Function (Keeney, 1976)

Once this utility value is calculated for each architecture, the results are plotted against life-cycle cost. This allows the decision maker to choose among a set of pareto-optimal architectures that maximize the tradeoff between overall utility and cost.

In practice, there is iteration between these steps, particularly in building the model, where systems analysts must ensure that the set of design variables they choose shows clear relationships to the set of user-selected attributes. For an analysis of the optimal ordering of these steps, see Ross, 2003.

Research Question 1:

This discussion of the MATE method leads to the first set of questions this study will answer: *How does the MATE study of the Space Based Radar problem compare with the effort made at Lincoln Lab? Are the predictions equivalent? What are the similarities and differences?*

2.2 Other Methods Similar to MATE

MATE's efforts at formalizing the multi-attribute evaluation techniques into a repeatable process are just one in a field of contenders that has been growing by fits and starts since Keeney and Raiffa's seminal work. Recent efforts at developing MATE-like processes include Tim Bedford and Roger Cooke's generic model (Bedford, 1999) which was first applied to a decision problem at the European Space Agency. Like several other attempts at applying the Keeney and Raiffa framework, Bedford and Cooke make simplifications that soften some of utility theory's mathematical rigor. In their development, Bedford and Cooke note that "…no set of attributes in the real world are really utility independent." Because of this view, they propose the concept of "conditional preferential independence," which is a less stringent condition on the decision attributes.

Another MATE-like formalization is Anandalingam's multi-stage decision model (Anandaligam, 1989). Like Bedford and Cooke, Anandaligam makes an effort to circumvent some of the strictures of the Keeney and Raiffa framework. The emphasis in these changes is to ease the demands placed on the decision maker during preference elicitation. This is a theme strongly running through the decision theory literature. Taking as an example a project decision regarding water provision in Virginia, Anandaligam creates two filters to eliminate clearly unacceptable solutions, allowing for a decision to be made over only a few remaining alternatives. This filtering allows for the elicitation of value functions which are far simpler to obtain than utility functions.

A final MATE-like method is the Simple Multi-Attribute Rating Technique (SMART), presented in (Edwards, 1994). Originally developed in 1977, the authors have updated it, noting that "SMART should be dead; SMARTS replaced it some time ago." They say this because there was a key flaw in the axiomatic development of the original SMART formulation, which the authors contend they have corrected with both the SMARTS and SMARTER protocols.

The SMARTS process is very similar to MATE. It begins with the top-down hierarchy process described by Keeney and Raiffa, and moves into the development of a value hierarchy. Then the "objects of evaluation" are defined (for MATE, this is equivalent to defining what system combinations included in the tradespace). Next an "objects by attributes matrix" is created, essentially scoring each option's predicted attribute level. This process basically replaces the modeling relationships created in MATE (SMARTS can do this since, in general, it is concerned only with policy choices where the link between a choice and its attribute level is clear and simple). Clearly dominated solutions (those whose inadequacy is clear even without detailed analysis) are then eliminated, as well as any attributes that seem to have little effect on any of the options. The authors then assume utility relationships from this "objects by attributes" matrix, with the understanding that they must all be linear. This is the first of the simplifying assumptions that enable SMART to be conducted quickly. The second assumption is that the utility functions can then be additively aggregated. Though they offer no axiomatic defense for this belief, they note that they have developed “rules of thumb” establishing when errors caused by this assumption are acceptably small. The final assumption regards rank weighting, which is the method they use to find weights on the attributes for the additive aggregation. The authors justify these shortcuts with their belief that the error in generating utility curves in the more rigorous techniques is so large as to leave them practically useless.

The authors go on to describe in some depth their method of “swing weighting” (essentially a way to convert a rank-order to a “range adjusted” rank order), citing contemporary research that shows these approximate weighting methods to be reasonably close to more formal ones. The main thrust of the process is “heroic approximation,” by which they mean they would rather construct a method that is easy to use and quick than one that is mathematically justified.

All three of these MATE-like methods highlight just a portion of the literature on decision theory and offer alternative ways to handle problems in the formalization that Keeney and Raiffa originally presented. SMARTS is especially interesting, as it presents a well known and simpls way to achieve the same type of results as one can get from MATE. The two methods are ideal for comparison.

2.3 Utility Theory Analysis Tools

Before comparing MATE to other methods of assessing multi-attribute utility, one must first have a method and framework for comparison. Fortunately, along with the widening field of utility assessment techniques there is a wealth of evaluative tools.

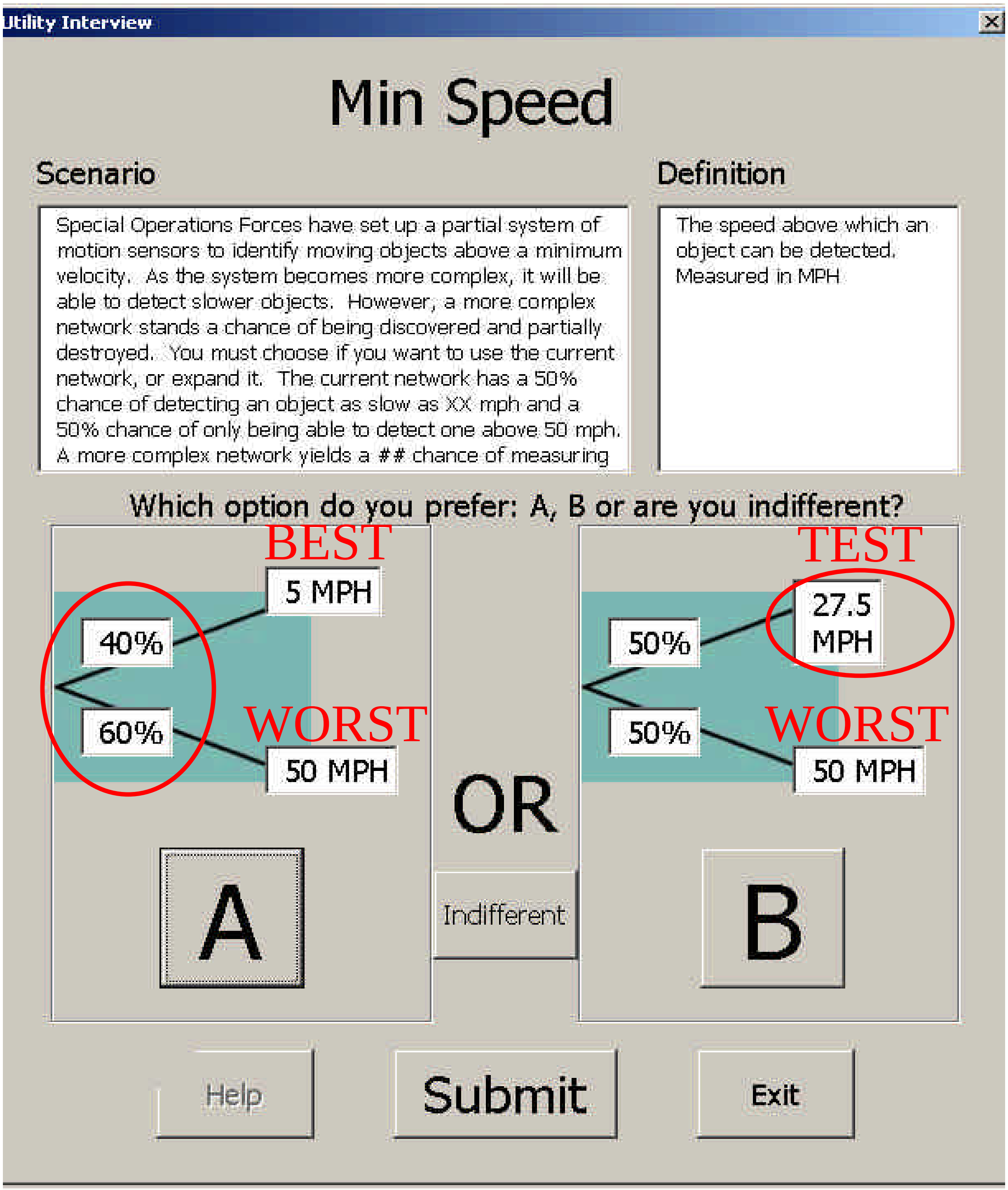
One interesting technique is David Olson's work with baseball statistics (Olson, 2001). This paper is unique in that it applies multi-attribute decision theory to a decision problem that has already happened, and has a wealth of data associated with it. This post facto analysis offers a unique chance to actually evaluate the predictive results of different assessment techniques. Olson evaluates three methods: SMART, PROMTHEE (an alternate multi-attribute utility method), and a centroid weighting method that he develops.

Using professional baseball statistics from 1901-1991 for his data, Olson examines the attributes of hitting, power, speed, fielding, and pitching. He molds these into a multi-attribute utility function by analyzing statistical data in the first five years of each decade. This function is then used to predict the outcomes for the second five years of the decade, given each team's attribute values in a given season. All three methods show high predictive validity over the data sets, with little loss associated with more drastic assumptions. It would be helpful to use such a technique to test the validity of the MATE method. However, in the field of space systems, there is no data set comparable to the set of baseball statistics with which Olson worked.

Vu Ha, in another effort to simplify the elicitation process, develops a measure of "closeness" between two preference structures (Ha and Haddaway). A measure of closeness can be a useful tool, especially when trying to measure the stability in responses to an elicitation method. More work on stability of responses was done in Laskey 1987, Fischer 1977, and Hershey 1982. These works, and others, document the errors that may be encountered when eliciting preferences from users in various elicitation schemes.

Moving beyond the recognition of these errors to their classification and impact, T.A. Farmer (Farmer, 1987) tests the robustness of multi-attribute utility theory to these errors. This was taken a step further by Fry, Rinks, and Ringuest, who compared robustness to errors across several types of preference elicitation (Fry, 1996). Fry notes that it is important that "the preference assessment strategy employed be structurally capable of encoding valid models of choice in the presence of elicitation errors" (Fry, 1996). Fry goes on to point out that there are, in general, two types of error in preference structures. The first is random error on the utility curves themselves—small deviations from the decision-maker's true preferences. The second is in incorrect attribute specifications—attributes that fail to be included in the analysis for either lack of capability (the MATE method is practically limited to seven attributes), or mistakes in the value hierarchy creation.

These efforts are important to understand because one of the criticisms of the MATE process is that it places considerable stress on users who provide utility data. This stress is both cognitive stress (the interview process is mentally taxing and somewhat frustrating), and time stress (busy users do not often have time to spend hours constructing utility curves). Currently, the MATE process uses a method called “lottery equivalent probability” to construct utility curves. This method asks the user to choose between two lotteries, where one of the lotteries is a 50/50 lottery between the worst and best acceptable values of a given attribute, and the other lottery is one with changing percentages between a test value and the worst value. An example is shown below.



Utility Interview

Min Speed

Scenario

Special Operations Forces have set up a partial system of motion sensors to identify moving objects above a minimum velocity. As the system becomes more complex, it will be able to detect slower objects. However, a more complex network stands a chance of being discovered and partially destroyed. You must choose if you want to use the current network, or expand it. The current network has a 50% chance of detecting an object as slow as XX mph and a 50% chance of only being able to detect one above 50 mph. A more complex network yields a ## chance of measuring

Definition

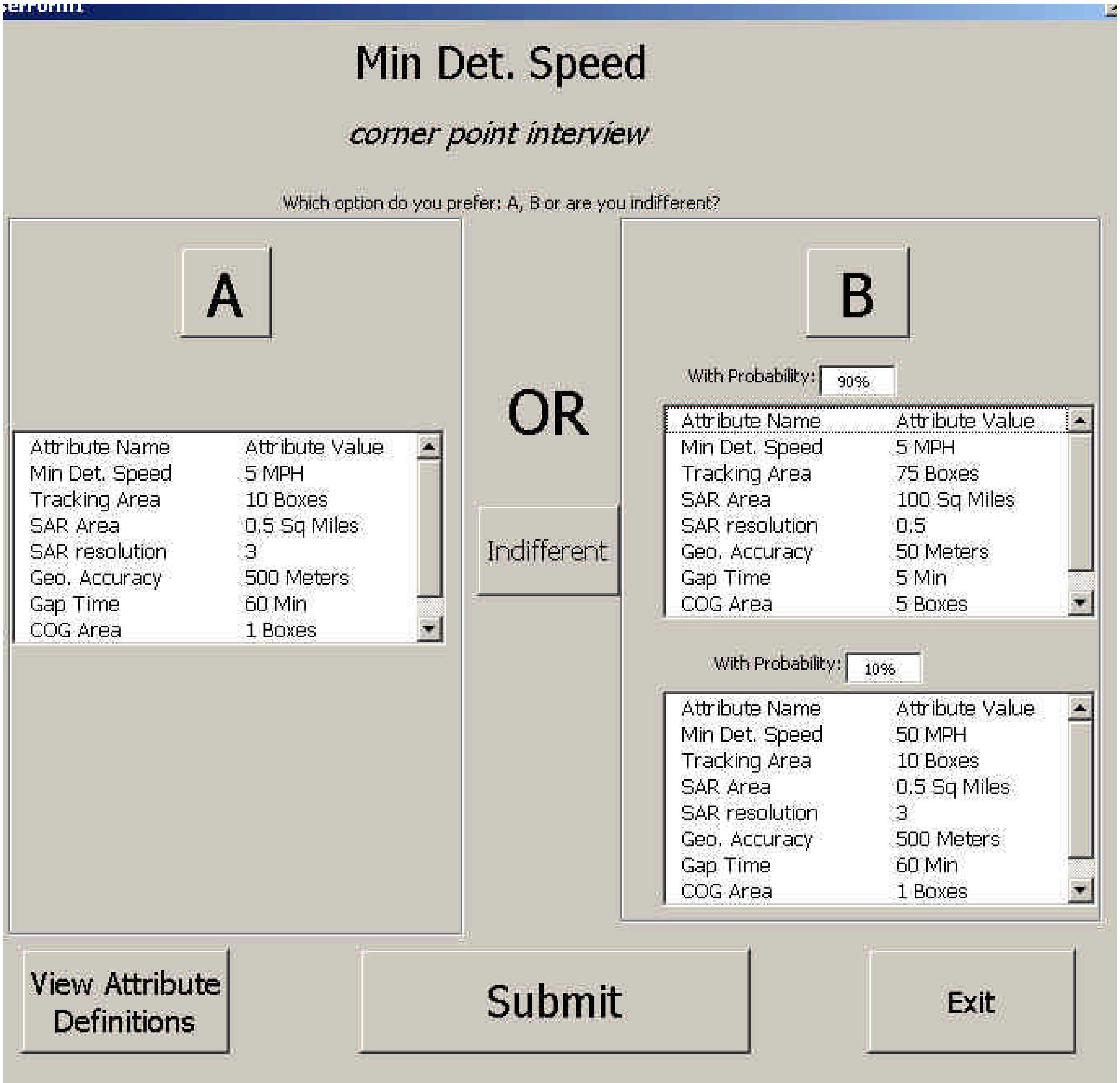
The speed above which an object can be detected. Measured in MPH

Which option do you prefer: A, B or are you indifferent?

Figure 4: Lottery Equivalent Probability

Figure 4 is a screen-shot from the Multi-Attribute Interview Software Tool (MIST) developed at MIT by master’s student Satwik Seshashi. Depending on how the interview proceeds, the user has to choose between many tens of these lotteries, each of which demands his or her attention and mental energy.

After this process is completed for each attribute, the user must then attempt to make tradeoffs among all of the attributes simultaneously. As mentioned above, it is this process that limits the number attributes to seven. An example screen-shot is shown in Figure 5.



Min Det. Speed

corner point interview

Which option do you prefer: A, B or are you indifferent?

Attribute Name	Attribute Value
Min Det. Speed	5 MPH
Tracking Area	10 Boxes
SAR Area	0.5 Sq Miles
SAR resolution	3
Geo. Accuracy	500 Meters
Gap Time	60 Min
COG Area	1 Boxes

With Probability: 90%

Attribute Name	Attribute Value
Min Det. Speed	5 MPH
Tracking Area	75 Boxes
SAR Area	100 Sq Miles
SAR resolution	0.5
Geo. Accuracy	50 Meters
Gap Time	5 Min
COG Area	5 Boxes

With Probability: 10%

Attribute Name	Attribute Value
Min Det. Speed	50 MPH
Tracking Area	10 Boxes
SAR Area	0.5 Sq Miles
SAR resolution	3
Geo. Accuracy	500 Meters
Gap Time	60 Min
COG Area	1 Boxes

Figure 5: Corner Point Interview

It is likely clear to the reader from a cursory glance at the decision between option A and option B above that making these judgments is taxing both in terms of cognition and time.

Research Question 2:

This elicitation process, though axiomatically rigorous, is difficult, time consuming, and still prone to errors in the user's inputs. These problems lead to the second question this thesis will endeavor to answer: *Should MATE be simplified to ease the stress placed on the decision-makers?*

2.4 Space Based Radar

As mentioned above, there are two key studies regarding SBR—the Decision Support Center (Keithley, 2001) and the Lincoln Lab study, which is detailed in Chapter 4. Both of these studies highlight the important parts of the SBR design problem, to include the distinction between the two types of products expected from the system: Ground Moving Target Indication (GMTI) and Synthetic Aperture Radar (SAR) imaging. These two products can be delivered by the same system, though not at the same time by the same radar antenna. GMTI involves measuring returns from moving objects in real-time (similar to an air traffic control radar but tracking instead terrestrial objects) while SAR imaging involves keeping the radar focused on one area for a long time in order to take high resolution images of objects.

There is a wealth of data on the design of such systems which will not be reviewed in this study. Design rules for electronically steered arrays and details of the performance of radar through different environmental and ground clutter conditions are far below the level of this thesis. A good, general review of the considerations for a SBR system can be found in Hacker, 2000. Further general principles of radar systems design can be found in a variety of sources, including the Space-Based Radar Handbook by Leopold Cantafio and Introduction to Radar Systems, by Merrill Skolnik.

Although applying MATE to space systems has been performed reliably (Long, 2001), there is some uncertainty in applying it to a space system like SBR, which is much different from the atmospheric sensing satellites of B, C, and X-TOS. The SBR's usefulness, though provided primarily by its sensing capabilities, is also a function of its role as a network. There is some precedent for analyzing military networks in a MATE-like framework (Davis, 2000).

Davis engages the problem of analyzing potential command and control networks, taking a very MATE-like approach. He writes "this study illustrates the use of value focused thinking (VFT) to capture the potential conflicting objectives of network expansion. A network planner, or a more senior decision maker, delineates objectives in a hierarchical structure down to measurable attributes which fully define each objective" (Davis, 2000). This model of "value focused thinking" has been applied in other settings, particularly for the Australian Defense Force.

Although the Davis study's method (subjective utility curves combined into a multi attribute utility theory) is somewhat shaky, its decomposition of the network problem into the meta-attributes of service, survivability, and flexibility is a helpful one to begin thinking about how to break down the SBR problem.

2.5 Utility Theory with Multiple Decision Makers

MATE, in its exposition by both Diller and Ross (Diller, 2002 and Ross, 2002), involves accounting for the preferences of three decision-makers—a user, a customer, and a firm. Although this multiple stakeholder approach is embedded in the method, no application of MATE to this date has actually pursued it. Instead all MATE results have been with respect to only the user's preferences. It is not clear how these multiple perspectives would be taken into account, though there are many possibilities.

Any method of analyzing preferences from multiple stakeholders must be careful to avoid the pitfalls predicted by Arrow's Impossibility Theorem (Arrow, 1970). In this seminal work, Arrow laid out a set of reasonable conditions which, if agreed upon, eliminate the possibility of constructing a consistent group preference over a set of alternatives. This only applies to groups of greater than two.

Arrow's theorem, though important, should not discourage all efforts at realizing a MATE analysis that includes three or more preference perspectives. As pointed out by Michael Scott and Erik Antonsson, the engineering decision-making paradigm is significantly different from that of social choice, and therefore one or more of Arrow's restrictions may be inappropriate (Scott, 2000). Keeney and Raiffa shed further light on the possibility of including multiple perspectives in a decision analysis (Keeney, 1976).

The idea of using multiple perspectives, particularly the customer's (who would be the acquisition community in the Air Force setting) is an intriguing one, and means that MATE could serve as more than just a decision aid. Its interface with the requirements process might allow it to serve as a “boundary object” between the communities of users, engineers, and customers involved on any large-scale system development.

The requirements process in the Air Force is complicated enough to have its own vast literature. A cautionary note is in order here: changes in the requirements process are common, and especially with the rise of a new 5000 series acquisition regulation, ways of managing requirements may drastically change.

The official version of the requirements generation process is published by the Joint Chiefs of Staff (CJCS, 2001), and details the labyrinth through which requirements documents develop. A more detailed and realistic look can be found in Rob Wirthlin's master’s thesis (Wirthlin, 2000), which presents both an idealized and actual Air Force requirements process. Nathan Diller also attacked the issue of requirements generation, specifically analyzing how MATE might be used to aid the process (Diller, 2002).

Research Question 3:

The difficulty in adding in multiple stakeholder preference to MATE analysis, and the effect this might have on the way the Air Force might acquisition process leads to the final question this thesis endeavors to answer: *How might MATE be used in the requirements community? How might it account for the preferences of those who make system acquisition decisions?*

2.6 Research Questions

As an effort to better understand that way MATE might be used as a tool to aid spiral development, this thesis poses and answers the following questions, developed above:

1*) How does the MATE study of the Space Based Radar problem compare with the effort made at Lincoln Lab? Are the predictions equivalent? What are the similarities and differences?*

2) Should MATE be simplified to ease the stress placed on the decision-makers?

3) How might MATE be used in the requirements community? How might it account for the preferences of those who make system acquisition decisions?

This Page Intentionally Blank

3. Research Methods

The centerpiece of the research efforts involved shadowing the Lincoln Lab' summer study of the space-based radar and integrating their analysis into a holistic, system-level MATE model. This model, detailed in Appendix B, used a combination of data directly from Lincoln Lab, cost modeling relationships from the Air Force Cost Model (7th ed.), and results from the RPAT radar performance software developed by Robert Coury. The broad outlines of the model are presented in section 3.1. The Lincoln Lab's data was used to build this model to allow for direct comparison between their recommended system configurations (which are as of yet still classified) and the results of the MATE process. The three interrelated research questions presented above center around this model. Each is answered through varying methods, which are detailed below.

3.1 Model Description

Below is a short description of the MATE model, focused mainly on the overall structure and assumptions. A more detailed description can be found in Appendix B, along with the source Matlab code.

Attributes

Following the MATE process described above, the model is based around a set of user-defined attributes. Defining these attributes and setting their minimum and maximum acceptable range is one of the most sensitive parts of any MAUT process (Keeney, 1976). Standing in as a proxy user was Mr. Larry Tonneson, of Zeltec Inc., who was a contractor working on the Lincoln Lab summer study. It was Mr. Tonneson's task in the Lincoln study to develop possible scenarios for the SBR's use. Since this involved imagining what a military user might want the system to do, he was an ideal person to represent an end user's needs and develop a set of attributes and associated utility curves. Mr. Tonneson was ideal for this task in a another respect due to his prior Air Force experience as an officer and crewmember aboard the AWACS aircraft, which functions in an operational context as an integrator of several sources of battlefield intelligence.

In defining attributes there is always a fundamental tension regarding where along the user-analyst spectrum the attributes will fall. For a user to express meaningful preferences, the attributes needs to be close to the user. That is, they should be something that makes no reference to the system that produces them. They should use language and concepts familiar to the user. In the military context, this often means that attributes will be things that related directly to combat effectiveness, and are therefore tied through a number of models to the actual system performance. For instance, if the user has a preference over the number of tanks a force can destroy, one can imagine a model that links this attribute to the performance of a given SBR architecture. The informational distance one must travel to make this link is significant however.

For the analyst, attributes are easier to deal with if they are closer to the system under study. These technical performance parameters are usually easy to model parametrically, and can often be defined more readily than can user attributes. However, they often place a burden and limitation on the user, who has to try and think in non-familiar technical terms. A user might be able to express a preference on the system's orbital inclination, but the validity of this preference is doubtful.

The attributes Mr. Tonneson chose represent a mix of loacations along this user-analyst spectrum. They were derived from interaction between the author and Mr. Tonneson over the course of ten days and approximately five iterations. Several different value hierarchies were explored to arrive at this list, each building on information from the current as well as previous studies:

ATTRIBUTE	WORST	BEST
TRACKING AREA: The number of 10 square mile boxes inside which objects can be reliably tracked.	10 boxes	75 boxes
MINIMUM DETECTABLE SPEED: The speed above which an object can be detected.	50 mph	5 mph
SAR RESOLUTION: The best possible resolution of the SAR images.	3 meters	.5 meters
SAR AREA: The square mile size of SAR images possible. (can be split into any number of smaller images)	_ x _ miles	10x10 miles
GEOLOCATION ACCURACY: The average error ellipse on a GMTI return	500 meters	50 meters
GAP TIME: The average time during which the enemy can "hide" (when there is no coverage)	60 min	5 min
CENTER OF GRAVITY AREA: The number of 100 square mile boxes inside which center of gravity can be reliably calculated.	1 box	5 boxes

Figure 6: SBR Attributes and Ranges

Utility Curves

After settling on a list of attributes and their associated ranges, Mr. Tonneson's preferences on these attributes were explored. This was done using the MIST software described in Chapter 2. Two separated MIST interviews were conducted, separated by approximately one week. Before the first interview, Mr. Tonneson was provided with training on the MIST tool and its associated method of preference elicitation. After this training the first utility interview was completed in approximately one hour. After this interview (and before seeing the utility curves that his answers had generated) Mr. Tonneson was asked to sketch what he thought is utility curves on each attribute would look like. This was done on graph paper that was scaled equivalently to the MIST graphs so a direct comparison could be made. The first MIST interview's data is hereafter referred to as MIST1. The first hand drawn subjective utility curves are referred to as HAND1. One attribute was accidentally omitted from the HAND1 data set.

The second interview session was performed in the same manner, with Mr. Tonneson completing the interview in approximately 45 minutes. This data is referred to as MIST2 and HAND2. For the purposes of analysis, the MIST2 data was taken to represent Mr. Tonneson's "true" preferences, with any differences between MIST1 and MIST2 being taken the errors one might typically expect from the MIST elicitation process. This data is summarized in Appendix A.

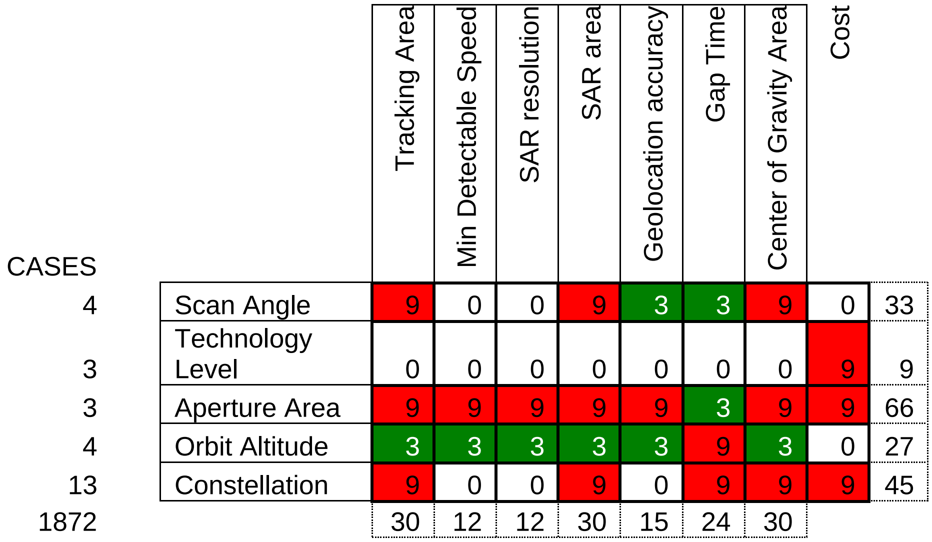
Design Variables

Forming the model around these attributes, system design variables were then listed and considered in terms of their effects on achieving various levels of these attributes. Since narrowing down the list of possible design variables is inescapably arbitrary, the scoping was accomplished in such a way that interesting technical design trades would be included.

There are two notes of interest here: first, contrary to previous applications of the MATE method, a sharply limited number of designs were considered. This was because the basis of the technical modeling was results from detailed analysis rather than parametric estimating tools. For example, instead of varying the aperture size from 40 square meters to 100 square meters in 5 square meter increments (which would have been the standard MATE procedure), only three increments were considered (40, 70, and 100 square meters). This was done in order to use the analysis already performed at Lincoln Lab. In other MATE applications, parametric relationships are used so analysis can be easily repeated for a number of design possibilities. In this case however, the analysis was somewhat more mature, so only the options under study by Lincoln Lab were considered.

The second note of interest involves a design trade that was not included: mechanical versus electrical beam steering. Although this represents an interesting system design trade, information on how to model the cost and performance penalties associated with mechanical steering became difficult to obtain. A further study of the SBR would do well to include this trade-off.

Below is a list of the design variables included in the model, presented in a QFD chart. The strength of relationship was measured on a 0-3-9 scale, in order to get a sense of how important each design consideration was to each attribute. Cost is also included next to the attributes, though it should be noted that it is not treated in the analysis as such. Rather, cost is an independent variable, against which utility is considered in the final tradespace output.



CASES		Tracking Area	Min Detectable Speed	SAR resolution	SAR area	Geolocation accuracy	Gap Time	Center of Gravity Area	Cost	
4	Scan Angle	9	0	0	9	3	3	9	0	33
3	Technology Level	0	0	0	0	0	0	0	9	9
3	Aperture Area	9	9	9	9	9	3	9	9	66
4	Orbit Altitude	3	3	3	3	3	9	3	0	27
13	Constellation	9	0	0	9	0	9	9	9	45
1872		30	12	12	30	15	24	30		

Figure 7: QFD for SBR Model

Model Assumptions

Using this relational matrix as a guide, a series of Matlab modules were developed. These modules translated the combinations of design variables into potential system architectures and rated each architecture's performance on the various attributes. These levels of performance were translated into utility through the methods described in Chapter 2. The modules and their interconnections are described in detail in Appendix B. It is helpful here though to examine quickly the fundamental assumptions that go into the model.

The first and most important assumption is one that limits the total designs under consideration to 1872. This “identical spacecraft” assumption requires that any potential architecture contains one and only one type of spacecraft. Therefore, the situation where one might have three satellites with 100 square meter radars combined with one satellite with a 70 square meter radar is ignored. This assumption is necessary for two reasons: first, it provides a limit for the architectures under study—there are an enormous number of architectures that could be considered if one were to relax this constraint and allow for “mix and match” combinations. Secondly, it provides for modeling ease—the radar performance for one satellite can be evaluated and used to extrapolate across the entire constellation. It would be difficult to model two different types of satellites performing together. This assumption, however, is as limiting as it is useful. Especially in the case of an evolutionary system design, one would like to be able to model the interaction between several different types of satellites simultaneously. Future iterations of a SBR MATE model would be strengthened by the ability to model these situations.

The second major assumption comes in the cost model, which is an adaptation of the Air Force’s 7th edition cost model, using the minimum percent error calculation. A ten-year life cycle was assumed, with both recurring and non-recurring costs are included. In order to satisfy the originators of the study, only a medium-lift delta class launch vehicle was considered. This was a restriction that Lincoln Lab used in its study, so it was mirrored here. Additionally, there is a rubber spacecraft assumption (i.e. the sizing for the launch vehicle is done entirely by mass—it is assumed that it will be able to physically fit on the launch vehicle if it is light enough). The model places as many satellites as possible on the same launch vehicle, provided they are going to the same orbital plane.

In order to simplify the radar calculations, performance characteristics were evaluated for each satellite assuming that it was performing either the SAR or the GMTI function exclusively. This of course is impossible in practice, since each satellite would be performing each task with some duty-cycle. Therefore, performance numbers should be viewed as total potential performance (instead of total actual performance). Further model details are described in Appendix B.

3.2 Question 1

How does the MATE study of the Space Based Radar problem compare with the effort made at Lincoln Lab? Are the predictions equivalent? What are the similarities and differences?

To answer this question, two approaches were taken. The first is a brief study of the Lincoln Lab study methodology. Data for this analysis were taken from the author's personal involvement during an internship at Lincoln Lab. This research was conducted from June-September 2002. The summary includes both the idealized and actual progression of events. Alongside this summary the details of the MATE process are described, and the likeliness and differences between the two study methodologies are explored.

Having made this comparison of methods, the results from each are compared. Results from the two studies are broadly comparable since the author's MATE study utilized the same data as the Lincoln Lab analysis. Furthermore, the MATE analysis only considered systems architectures also under study by Lincoln Lab. This parallelism essentially meant that the MATE study did not add any further technical refinement to the Lincoln Lab effort, but rather started from the same basis and used the unifying metric of overall utility to make a decision among alternatives.

These comparisons between the results of the two studies were to not only made on a broad, qualitative level (i.e. which types of systems did each study recommend, how precise were these recommendations, etc.), but also on a more quantitative basis. This quantitative comparison was made by taking the Lincoln Lab recommended architecture (or architectures) and assessing their utility according to the user provided utility curves. This procedure was to determine if the study leaders at Lincoln Lab, using techniques typical of analysis of alternatives studies, arrived at a system recommendation that conformed to the set of optimal architectures predicted by the MATE method.

3.3 Question 2

Should MATE be simplified in order to ease the stress placed on the decision makers?

The first question sought to examine MATE in relation to other methods for front-end, analysis-of-alternative studies, and to gain some understanding about how it compares both in the process it uses and the results it produces. The second question focuses on the MATE process itself, exploring its robustness to the errors inherent in the attribute generation and preference elicitation processes. As discussed in the literature review, both of these sources of error have been observed in prior studies and are real concerns for MATE if it is to be applied to “real-world” engineering problems.

To judge MATE’s robustness to these errors, and to ask whether or not the process should be simplified, the methodology from previous studies (Fry, 1996) was adopted. In order to make comparisons across different tradespaces, Fry used both a measure of proportional utility loss and a general rank-order correlation. The metrics were used to make comparisons across three methods of preference elicitation—the

MIST interview technique, a hand drawn subjective utility measurement made in parallel with the MIST interview, and a linear, risk averse preference relation as would be used by the SMARTS method (Ward, 1994). The second MIST interview data (MIST2) was taken to be the “true” user preference for this analysis, with the first MIST interview data (MIST1) being an example of the kind of elicitation errors one might expect to find. The subjective (HAND2) and linear data were taken as alternative simplifications of the process. Comparing their predictive validity helps answer the questions around simplifying MATE’s MIST elicitation process

To calculate proportional utility loss, a set of architectures along an approximately iso-cost line is considered, and the utility loss that an incorrect specification of preference incurred is calculated using the formula from Fry, 1996. The formula is as follows:

$$PUL = P_i^* - \frac{RP_i^*}{P_i^* - P_i^o}$$

Where

P_i^* *= the utility of the preferred alternative as measured by the MIST2 data*
RP_i^* *= the utility of the preferred alternative as measured by the model under study*
P_i^0 *= the utility of the least preferred alternative (for a given cost) as measured by the MIST2 data.*
(Definitions and formula adapted from Fry 1996).

Figure 8: Proportional Utility Loss Formula

If the two tradespaces under comparison predicted the same architecture to be the optimal for a given cost level, then the proportional utility loss is zero. Otherwise, proportional utility loss gives a measure of how close one tradespace’s prediction of the best alternative was to the other’s. This calculation is performed over the entire tradespace, with the results plotted along the axis of lifecycle cost.

The proportional utility loss metric, though useful, only tells part of the story since it is calculated solely on the basis of how the pareto-front changes under different sets of utility data. This information, while useful, is incomplete. One might also want to know how the rest of the two tradespaces compare to each other. In order to make this kind of comparison, a more general metric is needed that takes into account not only the differences or similarities in what each method predicts as the best alternative for a given cost, but how it ranks the whole range of alternatives at a given cost level.

To make these comparisons, a general rank-order correlation was performed using the Spearman's Rho statistic. This formula is as follows:

$$D = \sum_{1}^{n} (d_i)(d_i)$$

$$r_s = 1 - \frac{6D}{n(n-1)(n+1)}$$

Where

d_i = *Difference between ranks in lists*
r_s = *rank order coefficient*
n = *number of items in list*

Figure 9: Spearman's Rho Formula

A rank order correlation coefficient of one means that the two rank orderings are identical. A correlation of zero means that they deviate as much as possible from one another—that is, they are exactly reversed. The more out of order the one list is with respect to the other, the lower the Spearman's Rho correlation.

In order to get a better understanding of the performance in the hand and linear approximations, these calculations were performed twice—once where the attribute weights were assumed to be simply rank ordered, and again where the attribute weights were assumed to have a weighted rank order. (This distinction between simple ranking and weighted ranking is the same one that Edwards makes between SMARTS and SMARTER). For this analysis, it was assumed that weighted rank ordering could reveal the "true" weighting of the attributes. Therefore, the MIST2 weights were used for this second calculation. The specific weights used for these calculations are given in Chapter 5.

Also, for comparison, the proportional utility loss and rank order correlation statistics were calculated for the MIST2 tradespace under a "missing attribute" case. This case removes the least and most important (by k-value) attribute from the utility calculation and compares the resulting tradespaces to the original MIST2 tradespace. This gives some measure of how the tradespace would be affected if the attribute generation process failed to capture one of the attributes that was important to the decision maker. These results are presented here simply for comparison, to show how the errors from elicitation compare to the errors from attribute generation.

It is of important methodological note that in using the MIST1/MIST2 differences as representative of typical MIST interview errors it is assumed that errors are entirely due to the inherent variance in a decision maker's preferences over the attributes. This means they are not due to the user's preferences changing over time. Since the two sets of preference elicitation sessions occurred over a fairly compressed time-space (less than two weeks) this is a reasonable assumption. At a minimum, the errors are directly representative of the variance in the SBR decision maker's preferences, and are therefore useful to understanding how accurate the MIST elicitation technique is to the decision problem at hand.

3.4 Question 3

How might MATE be used in the acquisition community? How might it account for the preferences of those who make system acquisition decisions?

Answering the final question involved a strategy of exploration. Although MATE’s implications for the Air Force acquisition process had been theoretically examined (Diller, 2002), the process had yet to be actually applied or demonstrated to the communities in the Air Force that make acquisition and development decisions. To make this application, the author worked with the Joint Program Office (JPO) for Space Based Radar, located at Los Angeles Air Force Base. During this interaction two efforts were conducted. The first was to demonstrate the MATE method to the leadership of the JPO, highlighting the similarities and differences between it and typical analysis of alternatives processes like the Lincoln Lab study. After this demonstration, the author conducted an interview, gaining insights into the ways in which the process and products might help the JPO better perform its tasks.

The second task was to explore ways to include the JPO's perspectives into the tradespace. There is precedent for including this “second opinion” in the tradespace of possible architectures, in the form of the “customer” in the original formulation of MATE-CON (Ross, 2002). Although this idea of taking utility curves from different decision makers had long been part of MATE, it had never actually been applied to the process and/or results. Therefore, there was much uncertainty regarding how the preferences of a second decision maker might be interpreted and resolved with the first decision maker’s.

This Page Intentionally Blank

4. Data

4.1 Lincoln Labaoratory Process Summary

Below is a brief process summary of the Lincoln Laboratory summer study. As an example of a typical Analysis of Alternatives (AoA) study, this process description of the Lincoln Lab study forms the basis from which the AoA and MATE efforts can be compared.

Lincoln Lab is one of the few remaining Federally Funded Research and Development Corporations (FFRDCs). FFRDCs have long played a crucial role in the federal government's research and development efforts although their numbers have dwindled significantly since the heady research days of the cold war. Originally created in 1951 to study the problems and opportunities in the field of air defense systems, Lincoln Lab has since become a laboratory strong in communications, radar and other sensors, and missile defense technologies. Owned by MIT, Lincoln Lab gets the vast majority of its funding from the Air Force and Navy, who contract out to the lab for various technology related projects. Lincoln Lab prides itself on building one-of-a-kind systems, "following a project from the concept stage, through simulation and analysis, to the development of hardware and the ultimate demonstration of an integrated system." (http://www.ll.mit.edu/about/about.html).

Study Guidance

The summer study's efforts in analyzing the space based radar were independent of the larger analysis of alternatives effort conducted by the Air Force Space Command.

The study's initial mandate came from the Secretary of the Air Force. During a meeting on 27 February 2002, the Secretary of the Air Force, the Chief of Staff of the Air Force, and the Under Secretary of the Air Force for Space were identified as the "Configuration Control Board" (CCB) for the command and control constellation of which SBR is to be a part. (Space Surveillance Summer Study Statement of Work, Lincoln Lab internal). During this meeting, the CCB requested that Lincoln Lab perform a five-month study, independent of other acquisition activities, to "define the warfighting characteristics of potential future space-based-radar system options," taking as a guide the fact that the SBR would be the "primary space-based complement" of the larger constellation. The CCB asked that the analysis be performed from a Theater Commander-in-Chief's perspective.[2] This emphasis is important since until recently space assets have been designed for the national intelligence community (Defense Intelligence Agency, Central Intelligence Agency, etc.) and not for the more tactical uses of the combatant commanders. Finally, the CCB emphasized that "persistence" be an important factor in evaluating the effectiveness of any proposed constellation.

The following final products were listed as requested outcomes of the five month process:

1) A description of the surface targeting kill chain to include space-based radar, the performance of each element of the kill chain and the impacts on technology requirements and readiness
2) "Incubation" strategies for those technologies requiring further development
3) "Capability hill" charts to enable knee-in-the-curve analyses of investment versus warfighter capability
4) Draft operational, system, and technical architectures that could reduce acquisition risk

---

[2] "Commander-in-Chief" here refers to the commander most directly responsible for the conduct of battle in one of the United States' geographic command areas. They are usually referred to as "CINCS" (pronounced "sinks") or "combatant commanders."

5) Specific concepts for balancing operational needs against technology trajectories and system integration risks.
(Statement of Work, Lincoln Lab internal)

To complete this list of tasks, the CCB provided funding for travel, program management, engineering support, and contractor fees. This totaled $2 million, and included 90 staff-months worth of Lincoln Lab time, along with the efforts of subcontracted individuals.

Study Process

With the above goals in mind, the leader of the study set about devising a process that would produce appropriate and timely results. As with most idealizations, the reality of the process was somewhat different than what is described below. I will endeavor to discuss these disparities, and note what, if any, impact they had on the success of the process.

First the study participants were broken down into two teams, one called the “buyer” team and another called the “seller” team. Each was to pursue a different angle of analysis, while the synthesis of their efforts was to form the final deliverable product.

The seller team represented the technical experts. Much like a product development team would in a typical product development process, their job was to explore the technical possibilities of the Space Based Radar system, presenting in the end a system (or class of systems) that represented the best possible designs. They broke themselves up into two functional areas: Sensors Issues and Orbit and Constellation Issues. Forming as they do the two major design considerations, these two areas were broken down further and specialized engineering analysis was performed at these lower levels. The organization chart showing this construction appears below:

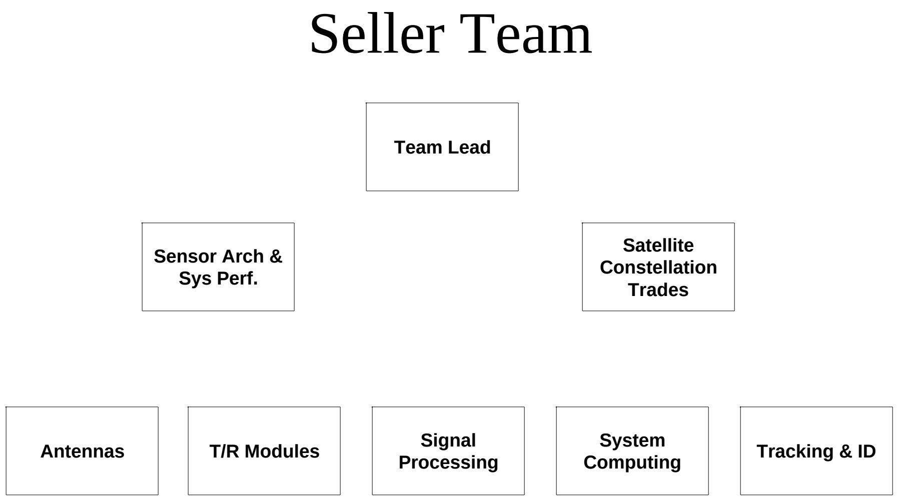


Figure 10: Summer Study Seller Team (SBR Staff, Lincoln Lab Internal)

The seller team's mandate included most of the technical analysis of the system itself, including how much it would weigh, how much it would cost, and how much time it would take to develop.

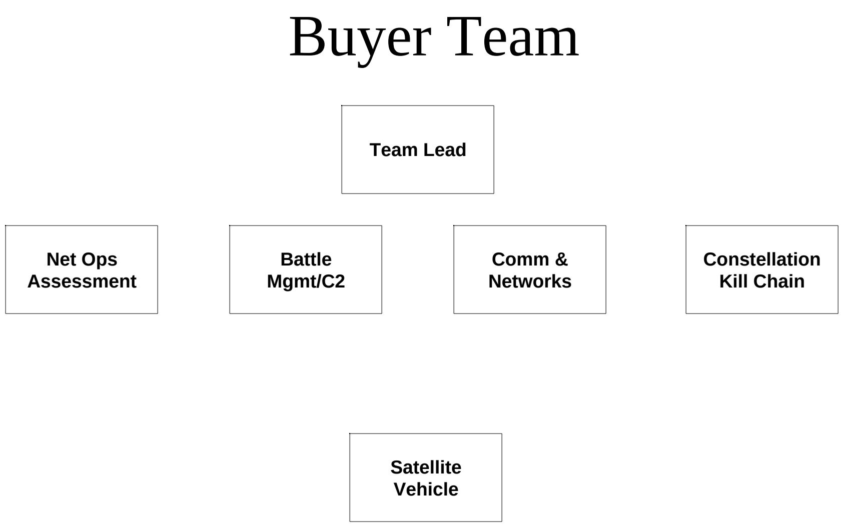


Figure 11: Summer Study Buyer Team (SBR Staff, Lincoln Lab Internal)

The buyer team, on the other hand was set up to act as the representative of the user, seeking to understand what the user would want out of any potential SBR architecture. This included everything from general capabilities to specific products. The buyer team was also responsible for defining how the SBR could be lashed into existing and proposed future systems, making some assessment about how various systems might be used in concert. Finally, the buyer team was responsible for quantifying and assessing the performance of the various systems that the seller team proposed.

In these respects, the buyer team essentially acted like a proxy for the Combatant Commander who could, among others, be envisioned as the ultimate end-user of the system. Pursuant to the summer study's unique mandate to consider systems built "for the warfighter," the buyer team focused especially on trying to understand and represent the needs of the warfighting user. The subgroup dedicated to this area was the one tasked with describing the "kill chain," which is the idealized path that leads from the use of the Space Based Radar to the destruction of a military target. By thinking in great detail about how such a path might proceed, the buyer team could better understand how the product would actually be used in the field.

With these two teams in place, a system for interchange was set up, whereby the two teams exchanged information and findings. These interchanges formed the heart of the process' workings, and consisted of a lengthy group meeting where each team presented their work up to that point. At the end of the presentations, the two teams had time to interact and compare their briefings, taking note of where capabilities of the seller and the desires of the buyers were matched or mismatched.

For the first interchange, the seller team developed what was informally called "The Poor Man's SBR." This was an inexpensive system that would surely not satisfy all the needs of the buyer team. However, by presenting it in the first team exchange, it could form a baseline from which the sellers could progress to a better, more buyer oriented system. Starting with a lower cost system was also effort to ensure the final recommendation was driven toward the more low-cost solutions.

After this first interchange, with some sense of the systems' possibilities and the buyer's requests, each team worked toward the second interchange. It was hoped that through this process, the two sides would converge on a solution, which would then represent the best design choice (or set of choices). The scheduled interchanges are shown below:

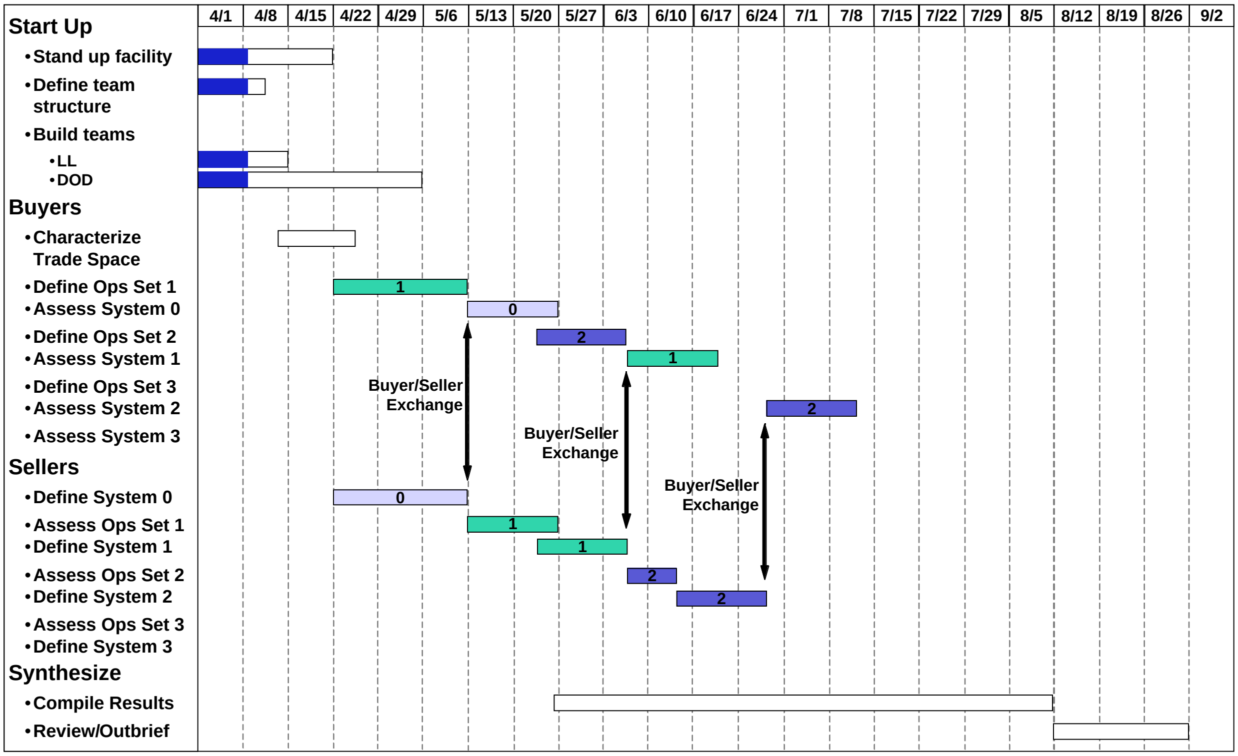


Figure 12: Summer Study Schedule (Top Level Schedule, Lincoln Lab Internal)

This structure was more or less adhered to in practice. The first exchange meeting took place on 14 May, and the second on 18 June. The third exchange, however, never occurred since the study participants wished to deepen the work already done on the first two interchanges and complete the other tasks necessary for the CCB final briefing.

The level of detail in each exchange varied, and was highly dependent on the various dimensions of the analysis. As a handy example, the system's possible orbital configuration was studied, but not in the kind of detail that would incorporate long-term orbital perturbations and effects. On the buyer side, possible uses were analyzed in great detail, even included a notional set of targets that would be of interest in a given theater. In general, the analysis was more detailed in the second exchange that the first, especially in the areas that proved interesting after the first round of work for each team. After the second exchange, the level of detail pursued was even greater, since the analysis was being complicated in order to give a cogent and detailed final report to the CCB.

Knowing that this exchange structure did not in itself guarantee that the buyer and seller teams would examine enough of the interesting potential design space to generate an optimal outcome, the buyer team was further instructed to ensure that their three statements of user need spanned the space of possible user needs.

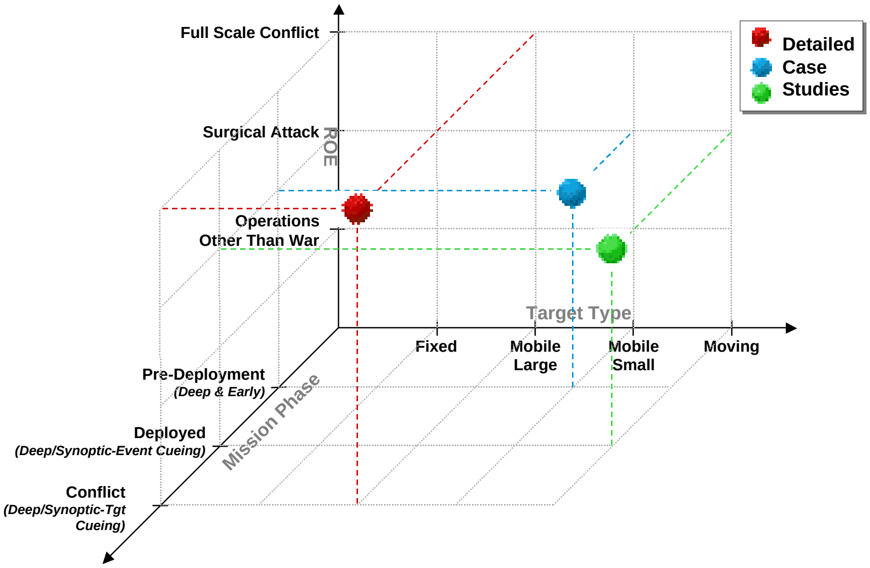


Figure 13: Summer Study User Needs Space (Lincoln Lab Internal)

This space was originally defined by three axes: Mission Phase, Rules of Engagement, and Target Type. By consciously attempting to span these three axes, it was believed that the statements would be widely scattered in this space of potential need, increasing the likelihood of producing a truly optimal outcome.

In the actual process this framework was abandoned though there were, in fact, four scenarios developed. As scenarios were finally developed in the buyer team they involved a military conflict between China and Taiwan, the breakout of war in the Caspian Sea, battle in Southern Europe, and a homeland defense scenario. The China/Taiwan and Caspian Sea were the two used in the interchange meetings, while the Southern Europe and homeland defense scenarios were developed only for informational purposes.

In order to span the range operations, each of these scenarios was then allowed to progress in several stages. That is, the buyers imagined their needs in a pre-conflict information gathering mode, during an overt buildup toward hostilities, and finally during full-scale combat in each scenario. The scenarios also included targets of different types so that the "target type" axis would not be ignored.

Final Report

After the two interchange meetings, the remainder of the summer study was dedicated to completing the technology readiness reviews, developing acquisition strategies, and finishing the other requirements for the final briefing. The final briefing took the following form:

- Introduction
- Buyer Team View
  - History
  - CINC centered view
  - Utility analysis
- Seller Team View
  - Radar Trades
  - Constellation Trades
  - Capabilities vs. Cost
- Buyer/Seller Adjudication
- Acquisition Strategies

Unfortunately, due to technical details of the system, the final report was classified and is therefore unavailable for this report. General results are, however, available. The Lincoln Lab final recommendation was a system with a 40 square meter aperture, a 1200 kilometer circular orbit, 45x15 degree electronic scanning, and a 2005 technology level (Chapa, 2003).

No number of satellites were explicitly recommended though the Air Force reacted by planning enough funding to create a nine satellite constellation. For the purposes of comparison the architectures with these satellite characteristics and nine (or nearly nine) satellites were taken as the Lincoln Lab recommendation.

4.2 Tradespaces

Simple Tradespaces

The following figure are the tradespaces for the Space Based Radar using the utility data from various elicitation methods. Each point in the following plots represents a unique system architecture, or combination of the design variables. Of the more than 1800 possible architectures, only approximately one-third appear on these plots. The remaining two thirds are either two heavy to fit onto the required launch vehicle or fail to achieve the minimum required performance on one of the attributes. Either of these conditions means that the architecture represents an unacceptable alternative to the user.

Note that each set of utility curves yields a different tradespace, both in terms of its shape and its placement on the utility scale. The plotted line in each connects all of the system architectures that populate the pareto-front. This front represents those systems that offer the optimal tradeoffs between utility and cost. The details of these pareto-optimal systems follow the tradespace plots.

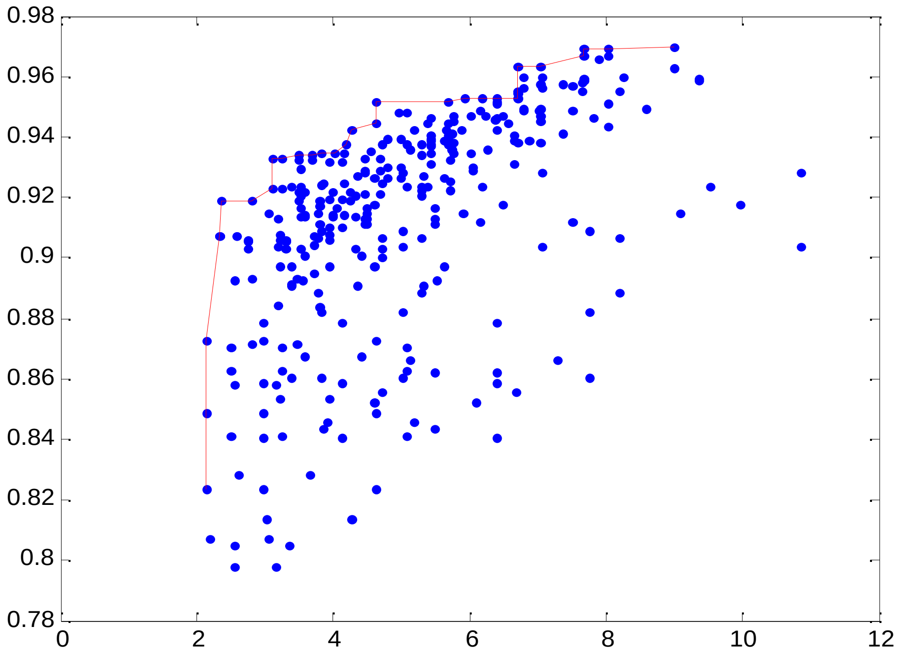


Figure 14: Tradespace Using MIST1 Data

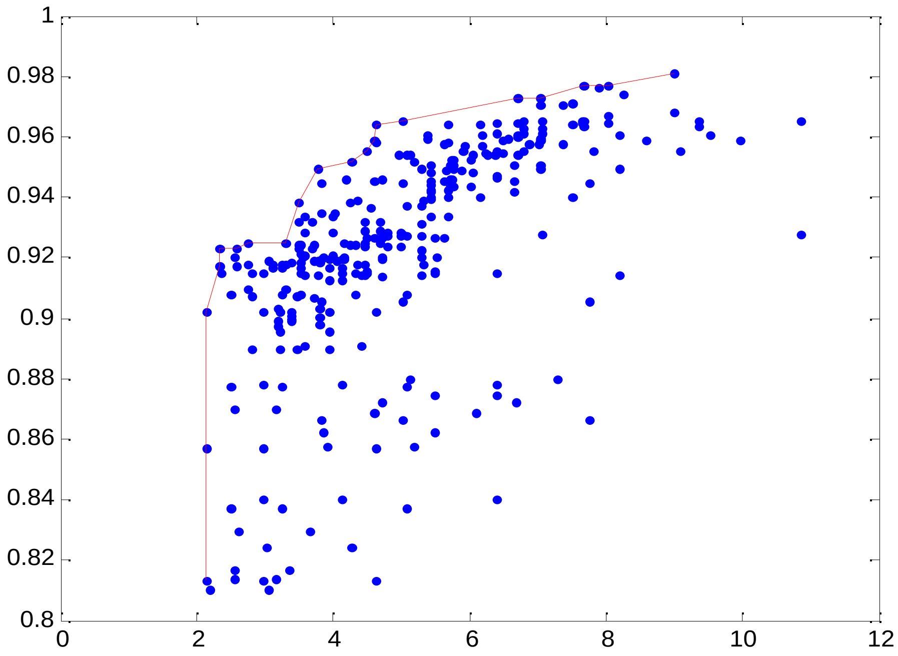


Figure 15: Tradespace Using MIST2 Data

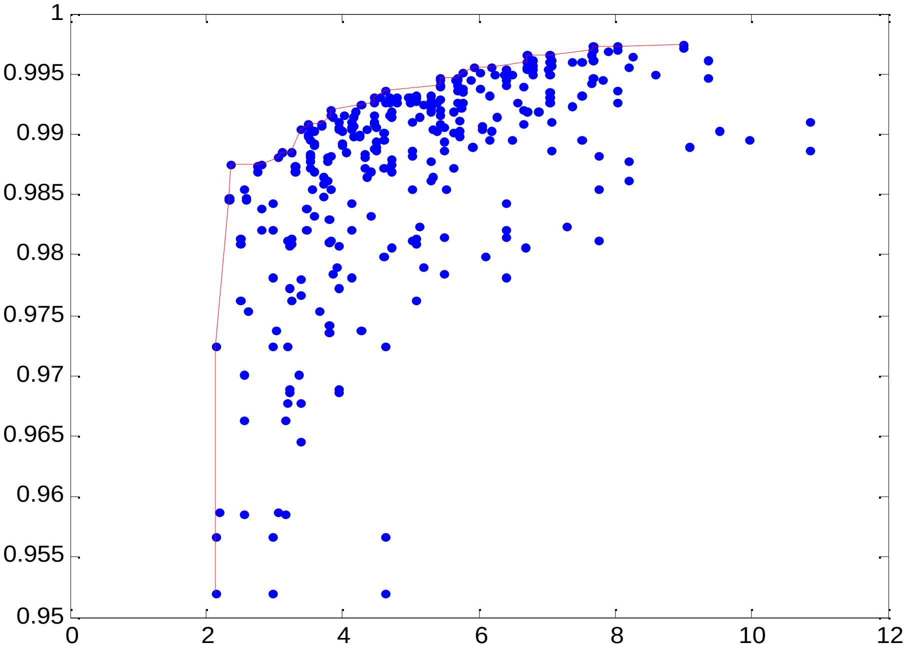


Figure 16: Tradespace Using HAND2 Data

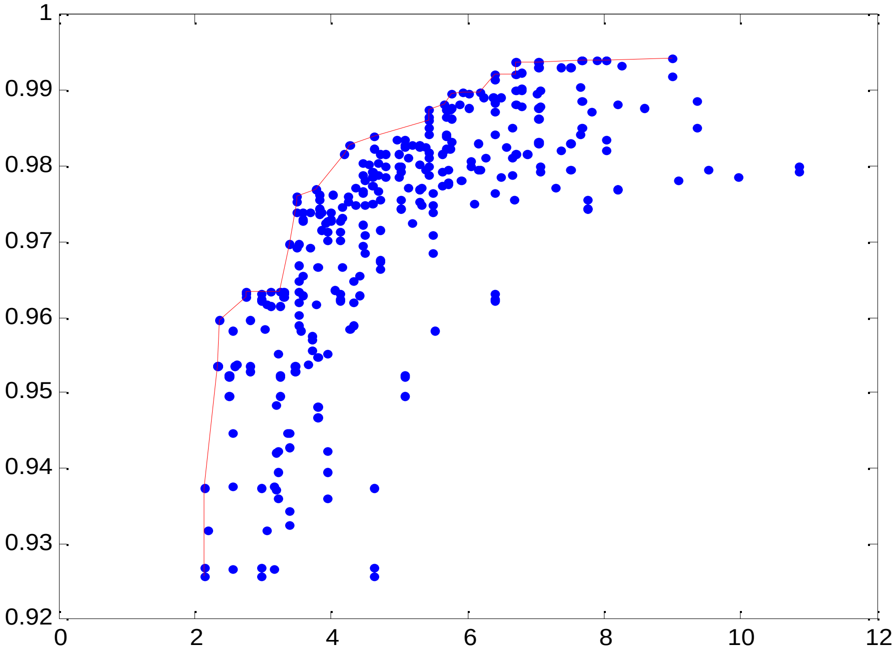


Figure 17: Tradespace Using LINEAR Data

Pareto Optimal Fronts

The details of the pareto-optimal fronts are below. The scanning column refers to the electronic scanning capability of the system, where 1 represents 5 degree scanning azimuth and elevation, 2 is 15 degrees in azimuth and 30 degrees in elevation, 3 is 45 degrees and azimuth and 30 degrees in elevation, and 4 is 30 degrees in azimuth and 45 in elevation.

As described in Chapter 3, the costs listed in these tables are predicated on the assumption of a 10 year life-cycle, and include both recurring and non-recurring costs. They are based on inputs to the Air Force Satellite Cost Model (7th ed). Technology level is factored in to cost by the assumption that future versions of the radar would be able to generate the same performance and lighter weight, and therefore less cost.

The absolute accuracy of the life-cycle costs is somewhat dubious, as are all parametric cost estimations, and several observers of SBR efforts have made system cost predictions greater and less than the figures shown below. Since the costing model's internal consistency is its strongest attribute, costs should be primarily understood as comparisons between competing architectures instead of highly accurate predictions of total program expenditure.

Architecture	Cost($b)	Utility	# of sats	# of rings	Altitude(m)	Aperture (m^2)	Scanning	Tech Level
1275	2.145583	0.82383	8	4	1200	40	3	2010
1262	2.145583	0.84877	8	4	1000	40	3	2010
1249	2.145583	0.87287	8	4	800	40	3	2010
28	2.328021	0.90704	9	3	1200	40	1	2002
15	2.328021	0.90719	9	3	1000	40	1	2002
3	2.362148	0.91881	10	5	800	40	1	2002
159	2.81602	0.91881	10	5	800	40	1	2005
223	3.097809	0.92327	9	3	1000	70	1	2005
236	3.097809	0.93288	9	3	1200	70	1	2005
80	3.23995	0.93288	9	3	1200	70	1	2002
288	3.507674	0.93384	9	3	1200	100	1	2005
132	3.695986	0.93384	9	3	1200	100	1	2002
263	3.828233	0.93453	10	5	800	100	1	2005
107	4.02896	0.93453	10	5	800	100	1	2002
8	4.195163	0.93769	18	6	800	40	1	2002
10	4.263416	0.94213	20	5	800	40	1	2002
25	4.64367	0.94461	22	11	1000	40	1	2002
12	4.64367	0.95152	22	11	800	40	1	2002
168	5.674276	0.95152	22	11	800	40	1	2005
217	5.93816	0.95296	19	19	800	70	1	2005
61	6.178687	0.95296	19	19	800	70	1	2002
268	6.3927	0.95308	18	6	800	100	1	2005
112	6.692748	0.95308	18	6	800	100	1	2002
685	6.697106	0.95435	19	19	800	70	2	2005
1309	6.710198	0.95497	19	19	800	70	3	2010
269	6.713259	0.96349	19	19	800	100	1	2005
113	7.025722	0.96349	19	19	800	100	1	2002
272	7.674934	0.96667	22	11	800	100	1	2005
285	7.674934	0.969	22	11	1000	100	1	2005
129	8.024643	0.969	22	11	1000	100	1	2002
1364	9.006744	0.9695	22	11	800	100	3	2010

Figure 18: Pareto Optimal Architectures with MIST1 Data

Architecture	Cost($b)	Utility	# of sats	# of rings	Altitude(m)	Aperture (m^2)	Scanning	Tech Level
1275	2.145583	0.81298	8	4	1200	40	3	2010
1262	2.145583	0.85704	8	4	1000	40	3	2010
1249	2.145583	0.90196	8	4	800	40	3	2010
15	2.328021	0.91753	9	3	1000	40	1	2002
28	2.328021	0.92299	9	3	1200	40	1	2002
184	2.577832	0.92299	9	3	1200	40	1	2005
30	2.742402	0.92492	12	6	1200	40	1	2002
186	3.292396	0.92492	12	6	1200	40	1	2005
19	3.502909	0.93848	16	8	1000	40	1	2002
1254	3.781979	0.94918	16	8	800	40	3	2010
10	4.263416	0.95176	20	5	800	40	1	2002
1256	4.503078	0.95512	18	6	800	40	3	2010
1258	4.600177	0.95855	20	5	800	40	3	2010
25	4.64367	0.96382	22	11	1000	40	1	2002
1260	5.009277	0.96536	22	11	800	40	3	2010
269	6.713259	0.97259	19	19	800	100	1	2005
113	7.025722	0.97259	19	19	800	100	1	2002
272	7.674934	0.97672	22	11	800	100	1	2005
285	7.674934	0.97718	22	11	1000	100	1	2005
129	8.024643	0.97718	22	11	1000	100	1	2002
1364	9.006744	0.98093	22	11	800	100	3	2010

Figure 19: Pareto Optimal Architectures with MIST2 Data

Architecture	Cost($b)	Utility	# of sats	# of rings	Altitude(m)	Aperture (m^2)	Scanning	Tech Level
1275	2.145583	0.95198	8	4	1200	40	3	2010
1262	2.145583	0.95662	8	4	1000	40	3	2010
1249	2.145583	0.97243	8	4	800	40	3	2010
15	2.328021	0.98456	9	3	1000	40	1	2002
28	2.328021	0.98473	9	3	1200	40	1	2002
3	2.362148	0.98745	10	5	800	40	1	2002
159	2.81602	0.98745	10	5	800	40	1	2005
627	3.062476	0.98807	10	5	800	40	2	2005
236	3.097809	0.98854	9	3	1200	70	1	2005
223	3.097809	0.98855	9	3	1000	70	1	2005
67	3.23995	0.98855	9	3	1000	70	1	2002
211	3.381844	0.99049	10	5	800	70	1	2005
275	3.507674	0.99066	9	3	1000	100	1	2005
288	3.507674	0.99087	9	3	1200	100	1	2005
132	3.695986	0.99087	9	3	1200	100	1	2002
263	3.828233	0.9916	10	5	800	100	1	2005
679	3.833767	0.99197	10	5	800	70	2	2005
10	4.263416	0.99241	20	5	800	40	1	2002
277	4.46935	0.99257	12	6	1000	100	1	2005
290	4.46935	0.99311	12	6	1200	100	1	2005
12	4.64367	0.99363	22	11	800	40	1	2002
681	5.424511	0.9941	15	5	800	70	2	2005
265	5.431025	0.99412	15	5	800	100	1	2005
278	5.431025	0.99447	15	5	1000	100	1	2005
291	5.431025	0.99461	15	5	1200	100	1	2005
135	5.693828	0.99461	15	5	1200	100	1	2002
279	5.751583	0.99507	16	8	1000	100	1	2005
217	5.93816	0.99559	19	19	800	70	1	2005
61	6.178687	0.99559	19	19	800	70	1	2002
685	6.697106	0.99593	19	19	800	70	2	2005
269	6.713259	0.99651	19	19	800	100	1	2005
113	7.025722	0.99651	19	19	800	100	1	2002
272	7.674934	0.99701	22	11	800	100	1	2005
285	7.674934	0.99734	22	11	1000	100	1	2005
129	8.024643	0.99734	22	11	1000	100	1	2002
1364	9.006744	0.99745	22	11	800	100	3	2010

Figure 20: Pareto Optimal Architectures with HAND2 Data

Architecture	Cost($b)	Utility	# of sats	# of rings	Altitude(m)	Aperture (m^2)	Scanning	Tech Level
1262	2.145583	0.92581	8	4	1000	40	3	2010
1275	2.145583	0.927	8	4	1200	40	3	2010
1249	2.145583	0.93753	8	4	800	40	3	2010
28	2.328021	0.95346	9	3	1200	40	1	2002
15	2.328021	0.95351	9	3	1000	40	1	2002
3	2.362148	0.95969	10	5	800	40	1	2002
17	2.742402	0.96278	12	6	1000	40	1	2002
4	2.742402	0.96321	12	6	800	40	1	2002
30	2.742402	0.96333	12	6	1200	40	1	2002
223	3.097809	0.96347	9	3	1000	70	1	2005
67	3.23995	0.96347	9	3	1000	70	1	2002
211	3.381844	0.9698	10	5	800	70	1	2005
6	3.502909	0.97538	16	8	800	40	1	2002
19	3.502909	0.97597	16	8	1000	40	1	2002
1254	3.781979	0.9769	16	8	800	40	3	2010
1267	3.781979	0.977	16	8	1000	40	3	2010
8	4.195163	0.98158	18	6	800	40	1	2002
10	4.263416	0.9828	20	5	800	40	1	2002
12	4.64367	0.98407	22	11	800	40	1	2002
681	5.424511	0.98598	15	5	800	70	2	2005
265	5.431025	0.98663	15	5	800	100	1	2005
278	5.431025	0.98749	15	5	1000	100	1	2005
216	5.654125	0.98808	18	6	800	70	1	2005
279	5.751583	0.9896	16	8	1000	100	1	2005
217	5.93816	0.98976	19	19	800	70	1	2005
61	6.178687	0.98976	19	19	800	70	1	2002
268	6.3927	0.99223	18	6	800	100	1	2005
112	6.692748	0.99223	18	6	800	100	1	2002
269	6.713259	0.9937	19	19	800	100	1	2005
113	7.025722	0.9937	19	19	800	100	1	2002
285	7.674934	0.99401	22	11	1000	100	1	2005
272	7.674934	0.99405	22	11	800	100	1	2005
116	8.024643	0.99405	22	11	800	100	1	2002
1364	9.006744	0.99432	22	11	800	100	3	2010

Figure 21: Pareto Optimal Architectures with LINEAR Data

MIST2 Tradespace Broken Down by Design Vector

Below are several graphs depicting detail in the tradespace using the MIST2 utility data. For the purposes of the analysis, the MIST2 data was taken to be the user's "true preferences." Each graph below shows the same tradespace, organized with respect to one of the design variables. This method of visualization allows for an analyst to quickly understand how the design variables impact the cost and utility space. In some cases this leads to quick insights about which designs are preferable to others.

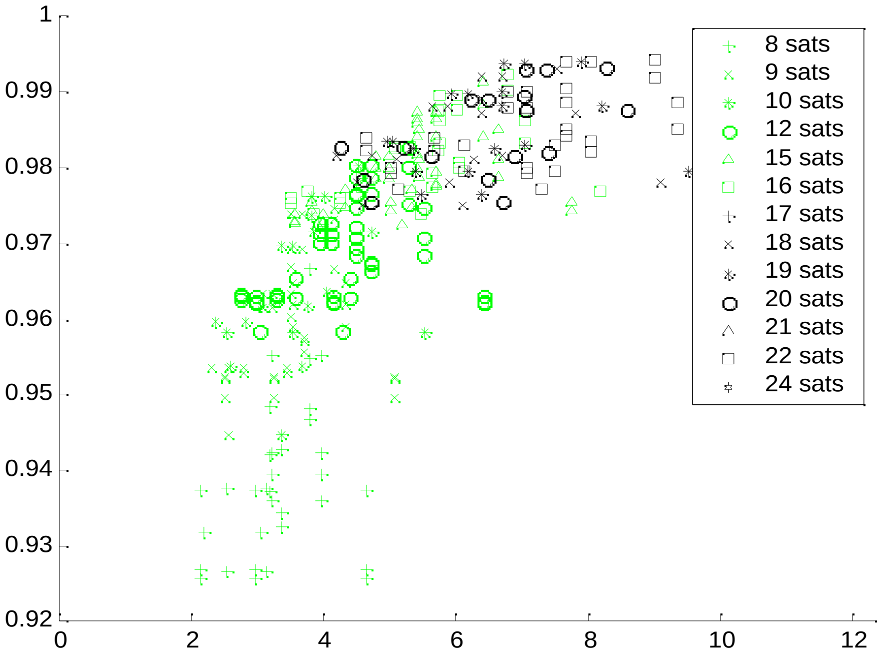


Figure 22: MIST2 Tradespace by Number of Satellites

One can immediately see the first order effect that the number of satellites has on cost. Generally, architectures with the same number of satellites are clumped together and move from left to right.

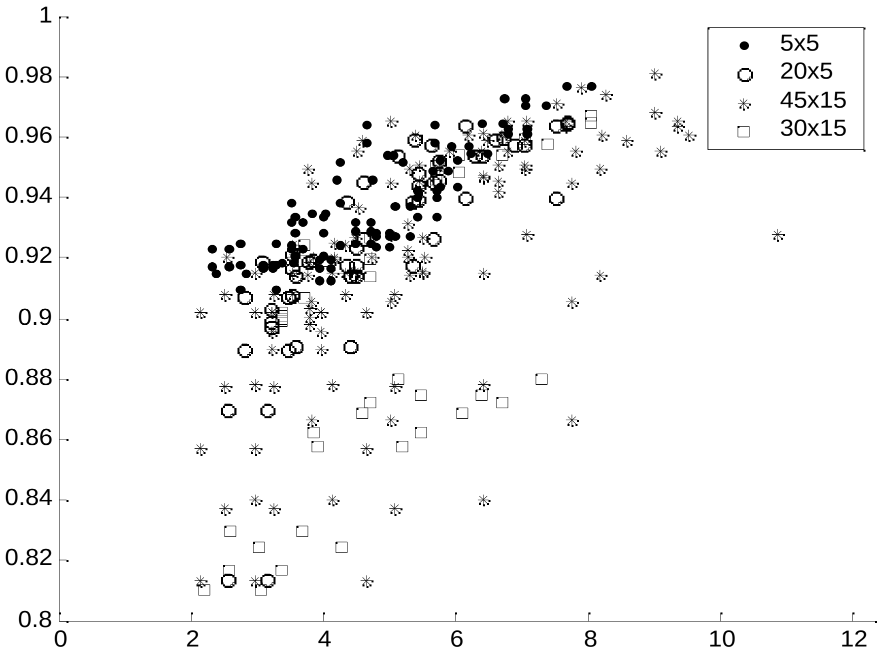


Figure 23: MIST2 Tradespace by Scan Angle (degrees Azimuth and Elevation)

In this depiction scan angle appears to be not directly linked with either cost or utility. Instead different scan angles create architectures that are spread throughout the space. If anything, the 45x15 and 5x5 scanning cases produce the majority of architectures that populate the pareto-front. This is an interesting result—essentially it means that the mid-range choices of 20x5 and 30x15 architectures do not represent and optimal tradeoff between cost and performance.

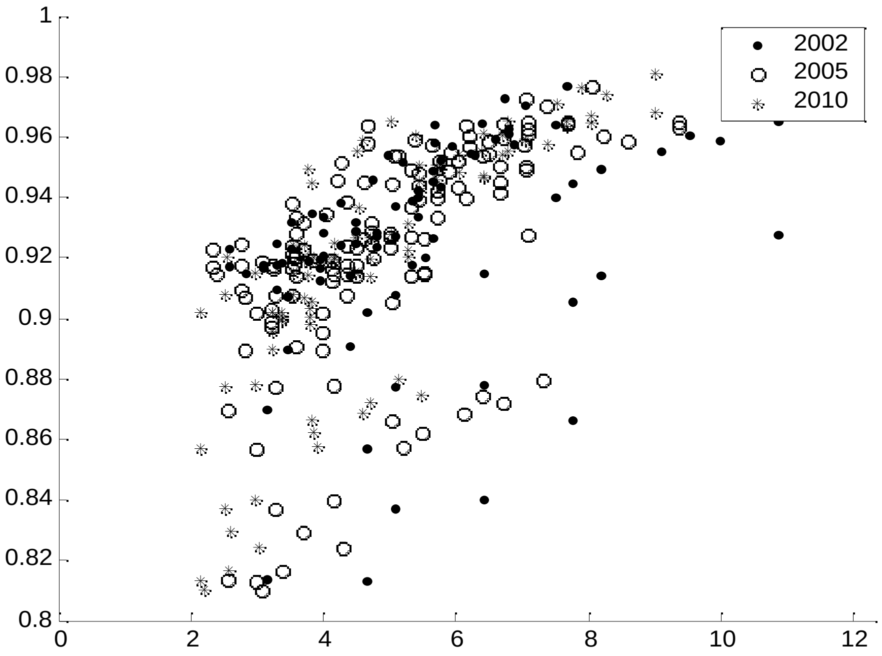


Figure 24: MIST2 Tradespace by Technology Level

Here technology level is also not directly related to cost or utility. Architectures of all three technology levels can be found on the pareto-front, as well as spread throughout the remainder of the tradespace.

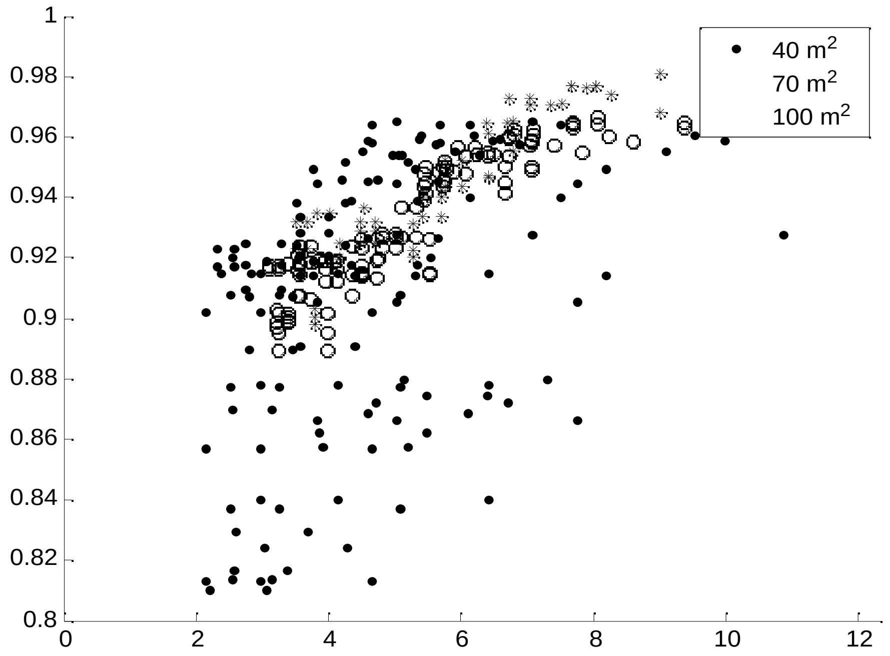


Figure 25: MIST2 Tradespace by Aperture Size

In this figure, it is clear that the 40 square meter aperture size represents the ideal tradeoff between cost and performance. This result is discussed further in Chapter 5.

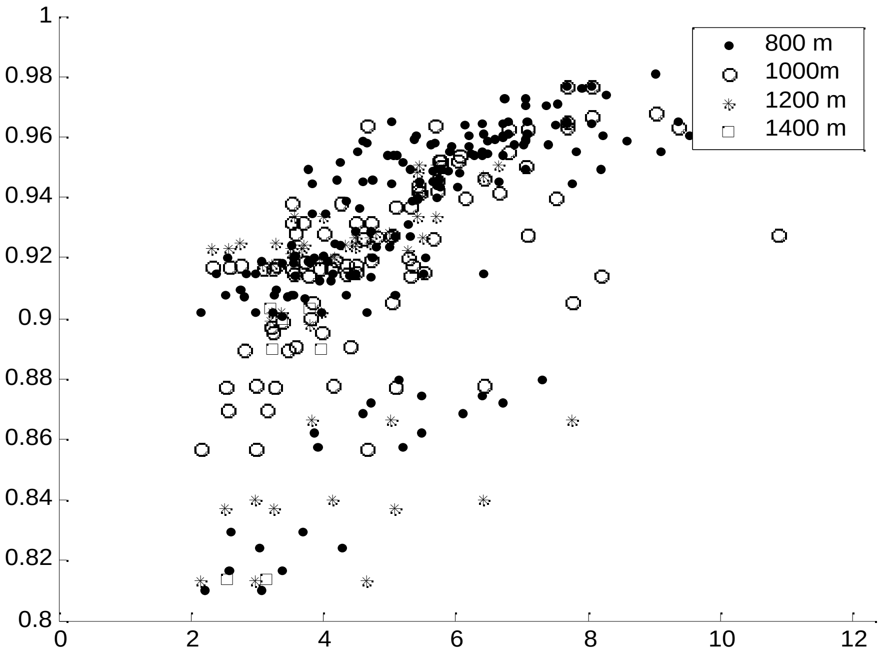


Figure 26: MIST2 Tradespace by Orbit Altitude

In this depiction, orbit altitude has an indeterminate effect on the tradespace. Like technology level, architectures of all altitudes can be found on the pareto-front.

This Page Intentionally Blank

5. Analysis/Discussion

5.1 Question 1

In light of the data and results in Chapter 4, how are the three questions posed in Chapter 2 answered? The first set of questions (*How does the MATE study of the Space Based Radar problem compare with the effort made at Lincoln Lab? Are the predictions equivalent? What are the similarities and differences?)* can be answered by comparing the Lincoln Lab process summary to the MATE process and the Lincoln Lab recommendations to the tradespace results in Chapter 4.

In general, the MATE and Lincoln Lab are broadly parallel—they both seek to analyze potential system configurations in a high-level way, relying on technical analysis to point designers toward the best parts of the tradespace. To do this, they both incorporate feedback from the user (or a proxy for the user), which allows the technical system trades to be made in view of what will best benefit the users of the system. In this respect, one can view both methods as creating a “boundary object” between the technical designers of a system and the people who will make the program decisions about it. The final product of each is meant to allow designers and decision-makers to communicate and come to a conclusion about what type of system is best to pursue.

Viewed from this perspective, the methods have two broad differences. The first regards the manner in which user preferences are incorporated into the technical investigation. The Lincoln Lab study took an evolutionary approach to incorporating this feedback through its system of planned interactions of the buyer and seller teams. Such a strategy imagined that through a number of such interactions, the proxy user’s preferences would be fully explored and incorporated in the study. In order to ensure that the full range of preferences was explored, the study architects made sure that the scenarios under consideration spanned the space of possible missions.

Such a strategy meant that there were planned times during which information could flow back and forth between the buyer and seller groups. During these interactions, there was clearly intended to be a flow from the buyers to the sellers so that future technical investigation was geared toward accomplishing the types of goals envisioned by the buyers. There was also a less explicit flow of information from the sellers to the buyers—expectations on system performance were adjusted in response to what were seen as the technical possibilities of the potential system designs. This flow of information meant that user preferences would change as the study went on since they were being constantly scaled against the possible technical solutions.

This back-and-forth information flow is useful in that it ensures that buyers and sellers are talking on equal terms, and that the users know what to ask for in their system. There is a danger as well however, since technical solutions, especially those proposed in the “poor man’s SBR” (the first round of interaction) might overly influence the future preferences of the users. Instead of focusing on what might really be militarily useful, users might instead think in terms of the system they have been presented, artificially adjusting their preferences to reflect the nature of this system. This is dangerous because innovative systems designs are unlikely to emerge from such a process—preference and technical considerations get wrapped up together and can no longer be understood in isolation.

The MATE process, on the other hand, attempts at its core to separate the two spheres of concerns as much as possible. In MATE the user preferences are elicited by thinking about performance objectives. By constructing a hierarchy of objectives and lower-level system attributes, user preferences are explored without any reference to any specific potential systems. In fact, the utility questions are worded such that no reference to the system design is made at all. By starting from such a general hierarchy, MATE seeks to ensure that the full space of the user's preferences is explored. Provided that the desired attributes are of a small enough number MATE can span the full space of user preference.

In the MATE process feedback is given to the user only after the technical investigation has been made and the user (or proxy) can see the results of their preferences on the systems under consideration. After this feedback is given, the user may choose to change their preferences encoded in the utility data. In the context of the MATE analysis, this feedback is given at the end of the study, not at intervals during the study as in the Lincoln Lab strategy.

On this point the Lincoln Lab and MATE methods carry different strengths and weaknesses. The Lincoln method allows for an evolution of user preferences since feedback flows back and forth between buyers and sellers. This strength introduces a weakness as well—the end preferences shown by the user might be unduly influenced by the early technical options that were presented. The MATE study avoids this weakness by clearly separating the user's preferences from the technical possibilities. In this process, an evolution of user preferences becomes unlikely.

The second difference involves the final output of the studies. It is in this respect that MATE presents the greatest advantage. In the Lincoln Lab study, the final output relied on the sum of the learning from buyer and seller interaction, but could only actually present the final decision. Although there was much interaction between the two teams, and much was learned about user preference, there was no formal method by which the learning could be unified, understood and presented. The Lincoln Study essentially traced a path through the tradespace of possible designs, but wouldn't necessarily be able to document that evolution.   The problem with such an approach is that "re-work" becomes very difficult. If the recipient of the study findings were to change slightly one of assumptions made early in the exchange process, the work done after that point would be invalidated in some sense. At a bare minimum it would be difficult to go back into the study team and find out what changes on the final decision would result from this new assumption.

The MATE study's process produces a product that is ideal for such further investigation. Though rework is still necessary should the study recipient want to change an assumption, the cost of rework in the case of MATE is far lower. Furthermore all of the learning—all of the interactions between user preference and technical possibility—are illustrated and available in the tradespace graphs that show the relationships between utility and cost. By examining these tradespaces with respect to various attributes, the tradeoffs between system performance and cost are clearly shown. This product alone clearly accomplishes the third objective that the CCB laid out ("capability hill charts"). Throughout the MATE study, the learning from the analysis is clearly contained in the outputs. As an example, the results in the MIST2 tradespace that is organized by aperture size reveals that 40 square meter aperture designs are almost the universally dominant solution, and that only for the higher cost systems are larger aperture systems efficient. This type of broader lesson would have been difficult to present in the Lincoln analysis, despite the fact that the study would have likely come to the conclusion. The MATE framework and its universal metric of utility allows analysts, users, and decision makers to all work from the same page and visualize the relationships between cost and performance. Another example is found in the MIST2 plot that is arranged by orbit altitude. It is clear from this plot that no architectures with a 1400 km orbit altitude are pareto-optimal. This insight would lead to dropping those high orbits from further detailed analysis. The rationale for such a decision would be clearly laid out for all involved to see.

It is this unification and capture of information that is the strength of the MATE method. In essence, MATE creates and utilizes a universal metric that can be used to integrate all of the learning that goes on in the kind of studies that are common to pre-acquisition activity. This benefit comes at little cost as well—the MATE analysis in this paper was executed by the efforts of only a handful of graduate students. This small extra effort, coupled with the technical efforts already performed, represents a different and more accessible end-product for decision makers.

Though the details of the final Lincoln Lab recommendations are classified, a quantitative comparison can still be made between the two methods’ recommended architectures. In the MATE model, any architecture lying on the pareto-front can be thought of as a recommended architecture. In the Lincoln Lab study, any constellation of satellites that has the characteristics described in Chapter 4 is a recommended architecture. How do these two recommendations compare?

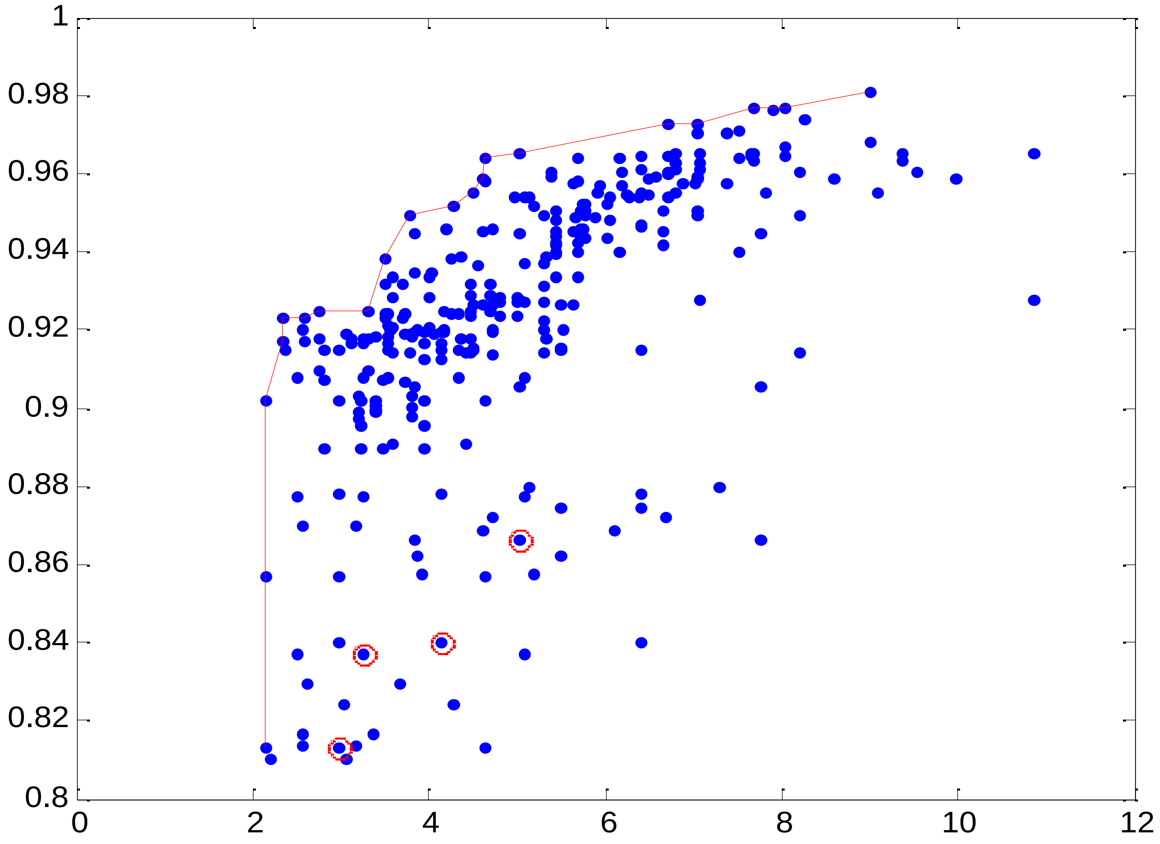


Figure 27: Lincoln Lab Recommendations in MIST2

Figure 27 shows the comparison, with the circled architectures representing the Lincoln Lab recommendations. Immediately one notices that these architectures, even in the most generous of views, cannot be considered to be close to the MIST2 pareto-front. Since this is the case, one can only conclude that the MATE and Lincoln studies arrived at fundamentally different answers to the same technical questions.

There are host of reasons why this might be so. First of all, there was much modeling done in the MATE analysis that went beyond the Lincoln Labs' data. In order to calculate attribute values, models were constructed that might very well have led to different technical conclusions. Futhermore, this modeling was done using radar performance at the unclassified level. Since Lincoln Lab analysis was performed with secret-level performance numbers, conclusions about which system is optimal may very well differ.

There is another class of possible differences involving the proxy user, Mr. Tonneson. One can confidently conclude that the data he provided in the MIST interviews represented the same user preferences that he expressed during team interactions. Even if this holds true, Mr. Tonneson was only one among many on the buyer team, and his opinions might not have been dominant throughout. It this was the case then the Lincoln Lab predictions might be skewed because of this additional input.

Whatever the reason for the divergent results, interesting perspectives regarding the strength of the MATE analysis arises. The four circled architectures represent the same type of satellite in a different orbital configuration. The far left architecture is a eight satellite constellation, and the three to its right are nine, ten, and twelve satellite constellations, respectively. The Lincoln Lab recommendation did not specify which of these was preferable. As a decision-maker, it would then be difficult to gain an understanding of which constellation to choose. Although a decision-maker might understand that more satellites incur more cost, they would still lack any formal way to make the corresponding tradeoff. With the MATE model one can explicitly see the utility/cost tradeoff of each option. This reveals that adding more satellites (especially going from ten to eleven) adds very little to user utility. This ability to continue to make tradeoffs even after an architecture recommendation is one of the strengths of the MATE method.

From a technical perspective, the biggest difference between the MIST2 pareto-front architectures and the Lincoln Lab recommendation is the orbital altitude. The MIST2 tradespace indicates that satellites of the type recommended by Lincoln Lab should be flown at low altitudes (800 or 1000 km). The Lincoln Lab recommended altitude is 1200 km. This disparity essentially means that the Lincoln Lab technical analysis showed performance to be enhanced at higher orbit altitudes, and that 1200 km represented the optimal tradeoff between good area coverage and adequate radar performance. The MATE study's utility curves indicated that this optimal tradeoff location was in the lower altitudes. As mentioned above, this difference could have come from two sources: Lincoln Lab's technical modeling (performed at the classified level) might have revealed better performance at high altitudes, or Mr. Tonneson's revealed preferences might have been different from the group preferences that drove the Lincoln recommendation to the higher orbital altitude.

As a design matter though, the recommendations are very similar—the MATE method revealed both that 40 square meter apertures were preferred, and that scanning of either 5x5 or 45x15 was necessary for optimality. These findings are represented in the Lincoln Lab recommendation. In this respect then, the two methods arrived at similar technical answers. The difference in orbit altitude, though important, is not a critical design parameter as the next phase of system development is entered.

5.2 Question 2

To answer the second question (*Should MATE be simplified?)* it is necessary to analyze and understand how the tradespace is affected by various utility inputs and other changes. To answer this question the fundamental assumption described in Chapter 3 must be invoked; the MIST2 data set must be assumed to represent the true preferences of the user. Under this assumption, the alternate data sets can be seen as approximations to that ideal. The linear and hand-drawn curves represent approximations that are easier to elicit than the MIST2 curves. The MIST1 curves represent the kind of error in elicitation that we might expect to find when using the MIST elicitation techniques in the MATE process.

With this framework in place, we can make comparisons among the data sets, comparing how much each approximation skews the results. If it can be shown that making a linear approximation or using hand drawn curves introduces no more error than can be expected from a typical MIST elicitation, then it would seem advantageous to skip the formal elicitation process and move directly to a linear or hand-drawn approximation for utility curves.

To make these judgments, the metrics discussed in Chapter three were employed—Spearman's Rho and proportionality utility loss (PUL). These metrics are calculated in similar ways. Each point in the MIST2 tradespace is selected in turn, architectures with similar cost are identified, and comparisons are made between the results in the MIST2 tradespace and the tradespace under study. The details of this methodology can be found in Chapter 3. The results that follow were calculated by using an iso-cost band that extends plus and minus $ 0.5 billion. This size cost band was chosen for the analysis because it is large enough that there are always a sufficiently large number of architectures inside it.

The calculations of Spearman’s Rho and PUL were performed twice, using the two different weighting methods discuss in SMARTS and SMARTER (Edwards, 1994). When figures are labeled “no weighting,” the attributes were simply rank-ordered from greatest to least important. The assumption here is that a user, even without going through a rigorous preference elicitation process, could still correctly rank the list of attributes.

When the figures are labeled as “rank order weighted,” the weights from the more rigorous MIST elicitation process were used in the hand and linear approximations. These weights contain more information than is contained in a simple ranking scheme. The assumption here is that by following the methods in Edwards, 1994, a decision maker can provide a weighted rank-ordering with little additional investment of time or effort. The choice to use the same weights as were derived from the MIST interview means that no errors are introduced through the weighting technique. Any errors that remain are necessarily due to the utility curves themselves. The two sets of weights are shown below:

Rank order weighted
k_min_speed = 0.35;
k_tracking = 0.20;
k_sar_area = 0.30;
k_sar_resolution = 0.10;

k_gap = 0.25;
k_cog = 0.05;

No weighting
k_min_speed = 0.7;
k_tracking = 0.4;
k_sar_area = 0.6;
k_sar_resolution = 0.3;
k_geo = 0.2;
k_gap = 0.5;
k_cog = 0.1;

Figure 28: Weighting Values

By using these two sets of weights, comparisons could be made between the errors introduced by ranking and those introduced by the utility curves.

The first set of comparisons used the proportional utility loss metric—a measure of how similar the pareto-optimal fronts were between different data sets.

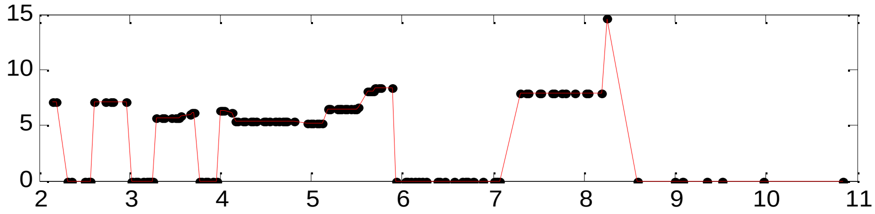


Figure 29: Proportional Utility Loss (PUL) for Alternate Utility Sets -- No Weighting

The figure above compares the proportional utility loss for the various methods. Each dot represents the proportional utility loss results from a cost band centered at the cost at its x-coordinate. If the proportional utility loss is zero the pareto-optimal architecture at that cost was the same one that was predicted by the MIST2 method. For instance, the HAND data set produced no proportional utility loss in the range above $6.5 billion. This means that the pareto-fronts generation by the MIST2 data and HAND data are equivalent above $6.5 billion. A PUL of zero across the whole range of cost is the goal of any approximation technique.

The first striking result is that the MIST1 data, taken as a representation of the typical kinds of errors one might expect from rigorous preference elicitation, does indeed introduce changes to the pareto-optimal front. These changes show a tendency to be clustered toward the low cost end of the tradespace. This accords with intuition—all utility curves are nearly identical in areas of high and low utility (since they all begin a one and end at zero). Since the high cost architectures are likely to be the ones with almost maximal utility (this can be confirmed by an examination of the tradespaces in Chapter 4), then we should expect there to be little disagreement between the alternate sets of utility curves. Indeed, this is the case for all three sets of data.

Where there is a difference between the MIST1 and MIST 2 pareto-fronts, the PULs seem to cluster around a value of around 6 or 7%. Upon examination of the data, this percentage typically represents a slip from a position of first in the list of architectures to second. Since the utility values separating two architectures near the pareto-optimal front are broadly similar (and small), this results in proportional utility loss of approximately 6 or 7%. Another way to think about this PUL is that the MIST1 data causes the decision-maker to choose an alternative that is 6 or 7% off of the pareto-front in the MIST2 tradespace. This gives a graphical feel for how great the errors are from preference elicitation.

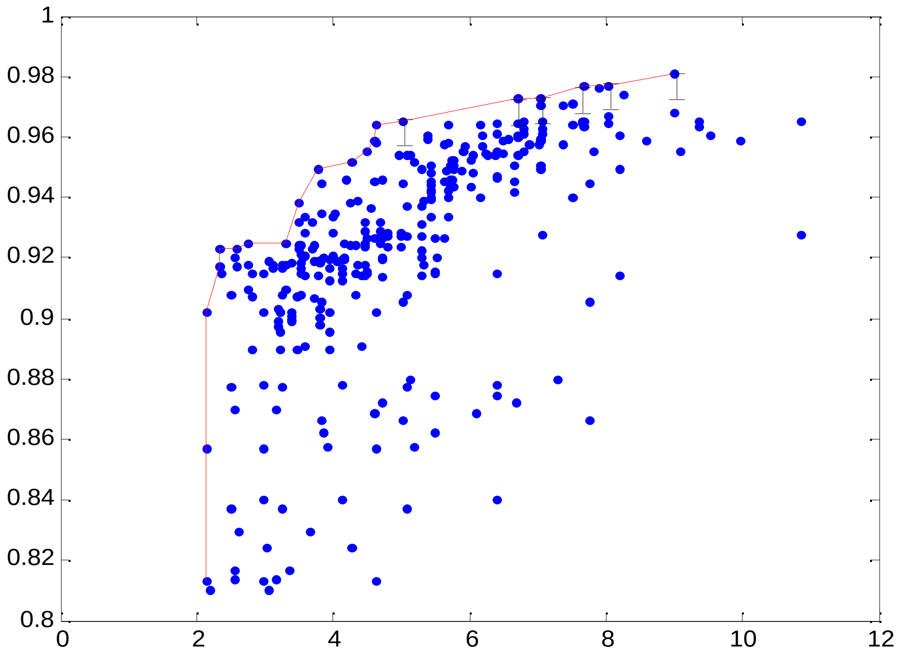


Figure 30: Graphical Representation of Errors from Elicitation

Above this uncertainty is shown on the MIST2 tradespace, with error bars that represent 6% proportional utility loss. This representation is helpful in that is shows the kind of "fuzziness" that one can expect on a pareto-optimal front.

The other striking thing from the proportional utility loss results is that while the MIST1 data are exactly accurate more often than the hand or linear data sets, the magnitude of errors they produce are comparable. For instance, although the hand data hardly ever picks the correct architecture, it produces errors very similar to the MIST1 case. The same is true of the linear data set, which produces slightly better results than the hand case.

This is remarkable since the linear approximations are quite severe—not only is a linear utility curve assumed, but the various attributes are simply ranked in order, with no additional weighting information provided. Even after these rather severe assumptions, the errors produced are not appreciably larger than for the MIST1 case.

This is not to say, however, that the mistakes produced by the simplifications are strictly on a level with those produced by errors in the elicitation process. Rather, the simplifications produce far more errors—in fact the fact that the PUL for these simplifications is almost always non-zero means they almost always pick the wrong architecture for a given cost level. This is especially true in the lower cost range.   These errors, however, are not catastrophic.

These conclusions are of course all predicated on two assumptions. First is that the MIST2 tradespace does indeed represent the user's "true preference." This assumption is fundamental to the structure of the analysis, and cannot be changed without finding a new method of analysis—it is necessary to have a baseline for comparison. The second assumption is that the deviation between the MIST1 and MIST2 data represents the size and types of errors one can expect from the MIST elicitation technique. This assumption could be changed were there further data collected on typical MIST elicitation errors. If the size of error expected was shown to be smaller, the approximation techniques would be worse by comparison. More research would be helpful in order to draw solid conclusions about the comparison of errors introduced by elicitation and those introduced by approximation.

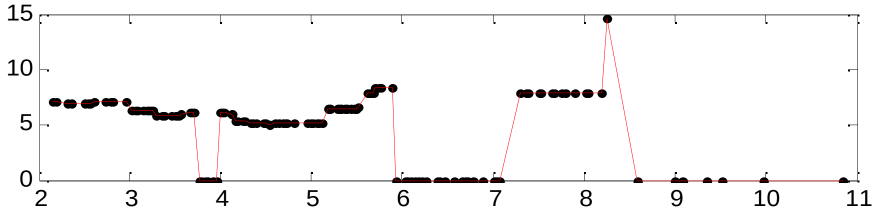


Figure 31: PUL for Alternate Utility Sets -- With Weighted Rank Ordering

The figure above shows the same analysis, this time with the weighted ranks used for the hand and linear approximations. This represents an increase in fidelity for the hand and linear approximation methods since they now encode the "true" weights on the various attributes. Comparing these results to those shown above, we see that adding in the weighted rank structure adds little benefit. In fact, in the case of the linear approximation, the results are worse that when a simple rank structure is assumed. This occurs for two reasons. First, the preference weights produced in the MIST interview process are more-or-less evenly spaced, meaning there is little difference between the ranked list and the weighted rank list. Secondly, it is likely that the errors that are shown when the weighting structure is used are actually damped out by the simple ranking scheme. That is, since the most important attribute receives less weight, the roughness of the linear approximation doesn't affect the results as strongly as it could.

Analyzing these proportional utility loss graphs is fruitful but incomplete without a point of reference. In order to provide such a point, two comparison cases were generated. These cases compare tradespaces that both use the MIST2 data. In each the errors are generated by dropping one of the attributes out of the tradespace.

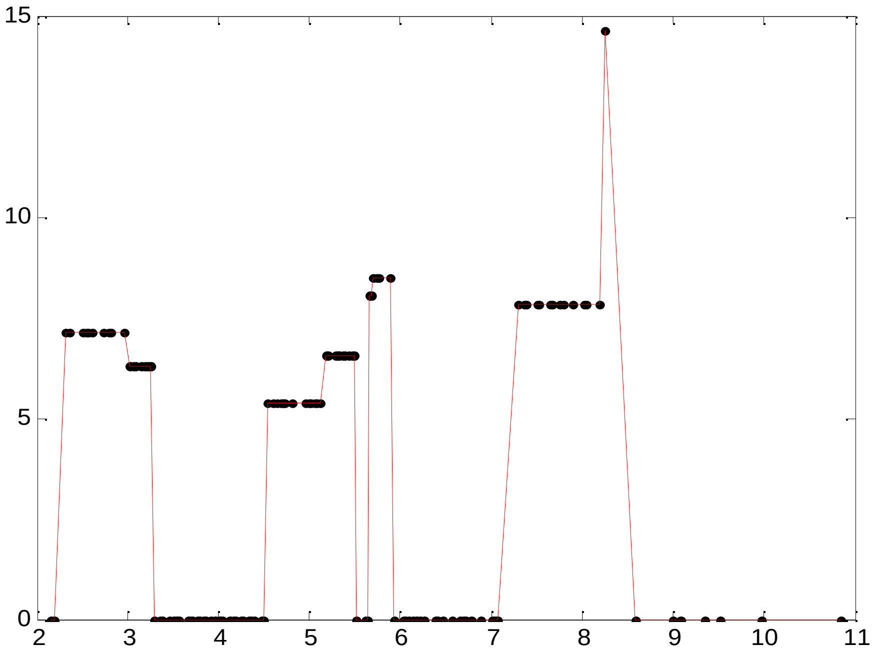


Figure 32: PUL from dropping least weighted

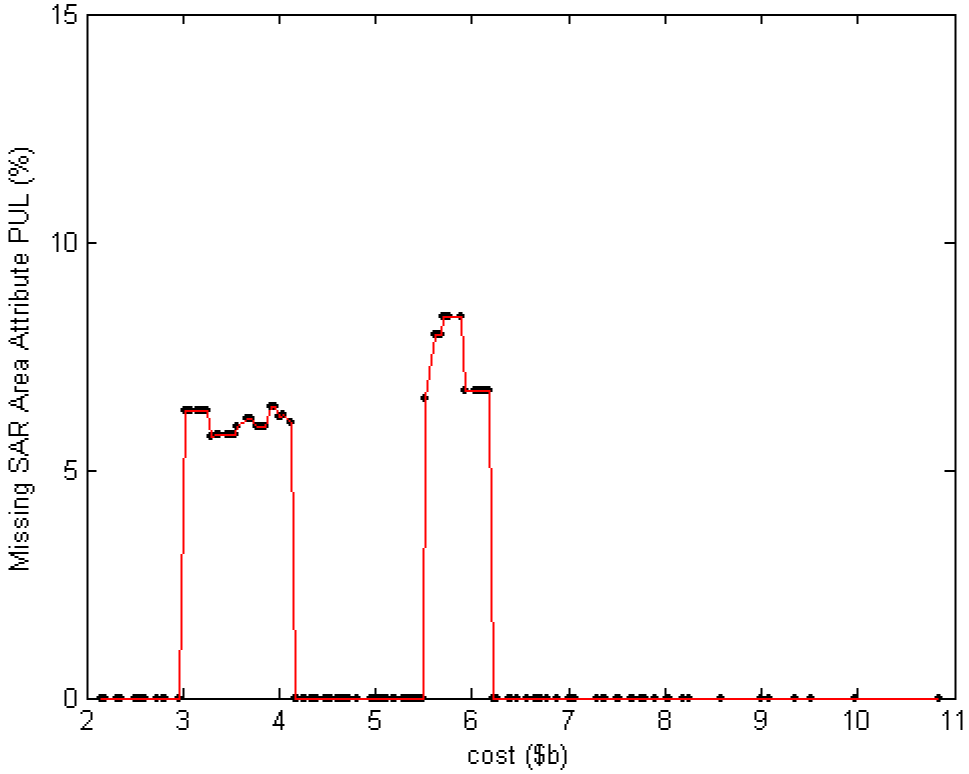


Figure 33: PUL from dropping most weighted

In both cases the levels of error produced by dropping an attribute are broadly similar to the errors introduced by the approximation methods. If anything, the errors are slightly greater. Just like the approximations, these differences are greatest in the lower cost portions of the tradespace. From this comparison, we see that the errors produced through the approximations are roughly the same as from dropping one of the attributes from the tradespace. It is interesting to note here that these results indicated that dropping the most heavily weighted attribute creates very few errors. This counter-intuitive result is explained by the fact that an attribute's effect on the tradespace is a combination of its impact on the user's preferences and its relationship to the design variables. In this case it appears that SAR area, though important to the user, does not affect the tradespace as much as the Center of Gravity attribute. This is because the designs considered don't vary widely in their performance on the SAR area attribute.

The changes in the pareto-optimal front are only part of the story. In order to get a better sense of the errors that are introduced by the approximations, it is necessary to examine the entire ranking structure at each iso-cost level. To do this a general rank correlation is necessary.

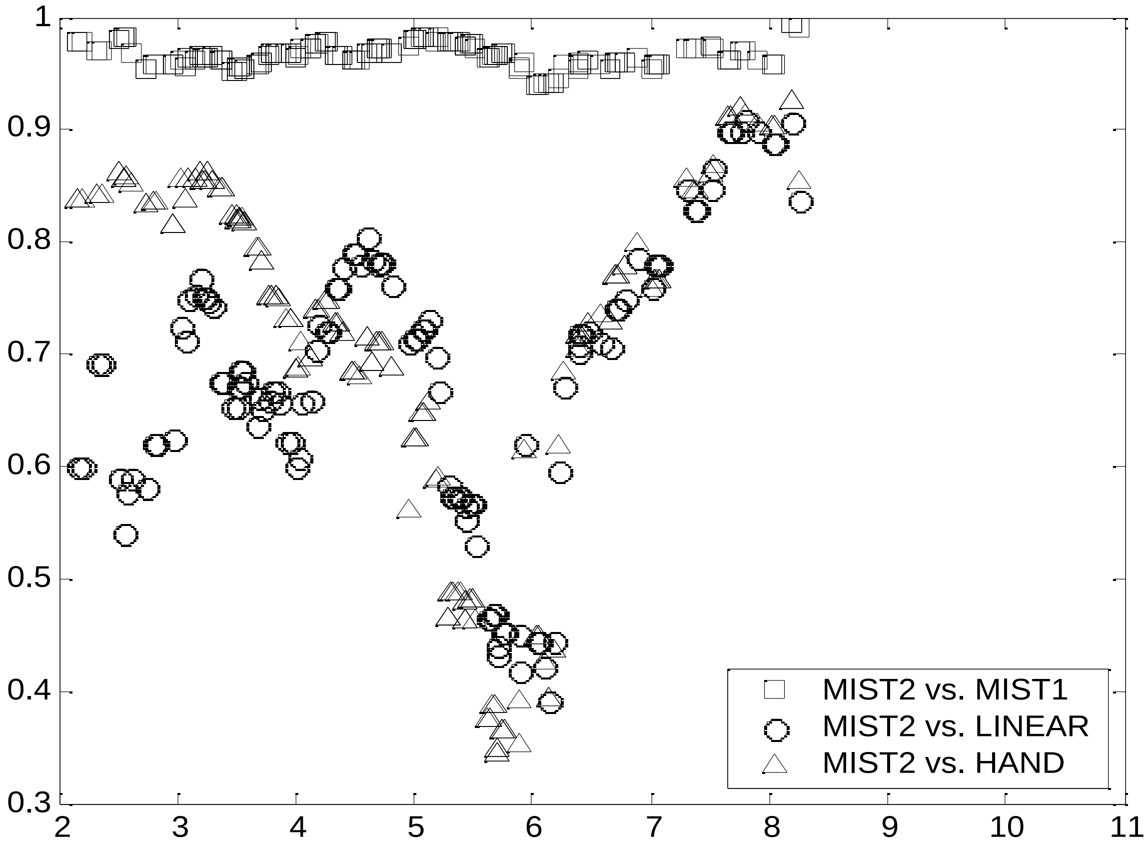


Figure 34: Spearman's Rho – No Weighting

Above are the results of such a measure—the Spearman’s Rho statistic that was discussed in Chapter 3. Spearman’s Rho is a measure of the similarity of two ranked lists. A value of one means the lists are identical.

The most obvious feature of the results is that the MIST1 and MIST2 tradespaces are almost identical, while there are significant differences between MIST2 and the hand and linear tradespaces. As was seen in the proportional utility case, these differences are smaller in the high cost/high utility parts of the tradespace where there is little difference in the preference structures. The hand drawn curves seem to fare better in this analysis, though this advantage is only in the lowest cost part of the tradespace. In general, the linear and hand approximations are equally inaccurate.

This method of analysis reveals that the errors caused by the approximation strategies are not confined only the architectures on the pareto-front. It also reveals that the errors in the ordering of the remainder of the architecture are actually far greater than in the MIST1 case. This is an insight that was not obvious from the analysis of proportional utility loss.

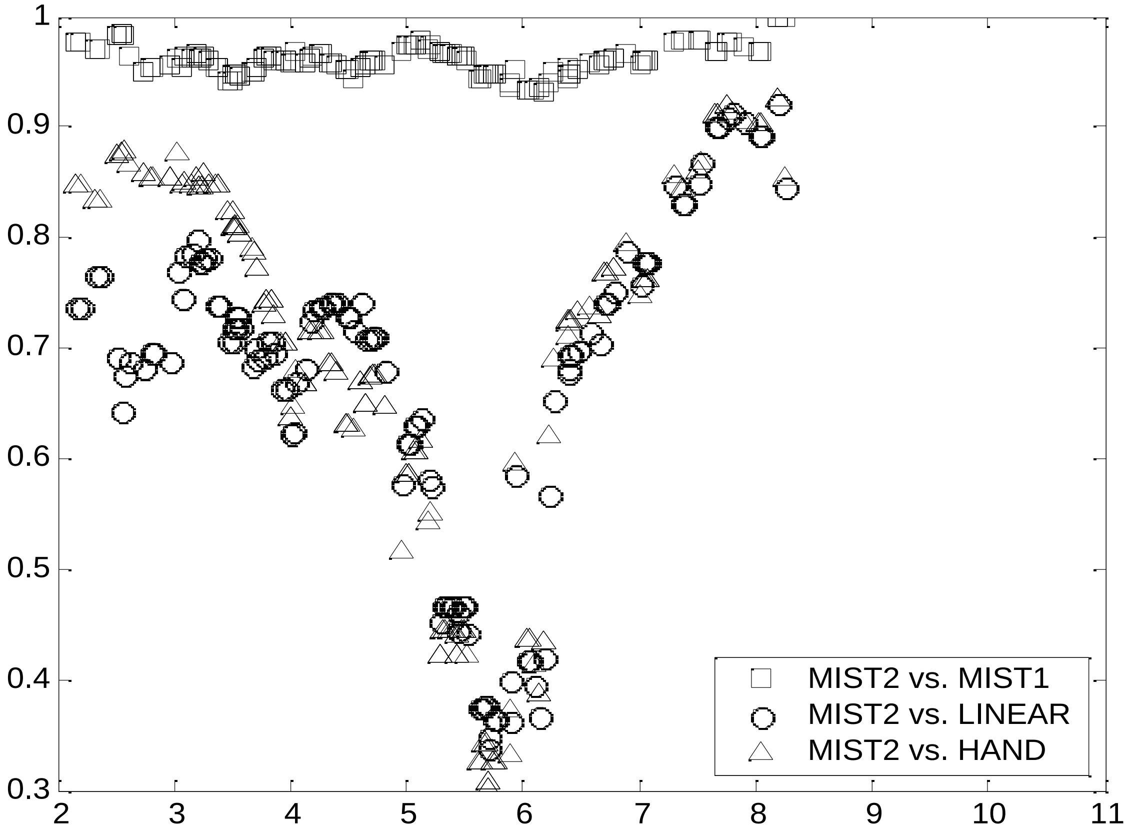


Figure 35: Spearman's Rho – With Rank Weighting

Above the same correlation is conducted, with the rank weighting included. As in the case of proportional utility loss, adding in the additional information of rank weightings adds almost nothing to the predictive validity of the approximation methods. Though both correlations are improved by adding the weighting, the differences are inconsequential.

5.3 Question 3

The third research question (*How might MATE be used in the requirements community? How might it account for the preferences of those who make system acquisition decisions?)* was answered in two parts. The first involved demonstrating the MATE technique to the JPO and receiving feedback about its usefulness and potential. The second involves experimenting with ways in which the preferences of multiple decision makers might be incorporated into the same tradespace. Both activities are exploratory—accordingly, the results are only preliminary and speculative.

Working with a member of the JPO team, the author presented the MATE method and the Space Based Radar analysis results. Over the course of a phone interview, three questions were explored: Would MATE be useful to a JPO? How do MATE's outputs compare to the outputs of the AoA process? Were the JPO involved in defining attributes, what would its system attributes be? The answers to these questions are revealing in several ways, though it must be noted that they represent the opinion of only one member of the JPO team.

The first question addressed is the comparison between the AoA analysis and the MATE analysis. The interviewee noted first and foremost that the primary difference between the AoA output and the MATE output was the former's concentration on a few point designs. He noted the AoA concentrated on in-depth analysis of a few well defined system architectures that were built around a couple of key performance parameters. This analysis was completed not to compare among alternatives, but to demonstrate the system's utility to the user. A secondary objective was to start solidifying a set of requirements that would be used for the more rigorous requirements definition processes.

In the case of SBR, the AoA processes centered around designs that were launchable on a medium sized launch vehicle, and had the area rate performance of the existing JSTARS airborne radar. It is interesting to note that the Lincoln Lab study essentially centered itself on the same two parameters. The MATE analysis, while keeping the medium range launch vehicle requirement, did not center itself exclusively around the JSTARS' radar performance.

The interviewee also noted the difference in the level at which the analysis was conducted. He noted that in the AoA, very specific combat scenarios were envisioned, and very specific performance was measured. It was his feeling that while the MATE analysis considered a multitude of systems, the AoA analysis considered a multitude of employment scenarios.

He also expressed some difficulty with the concept of utility in general, and the abstractness of the attributes in particular. Like many who are exposed to the non-cardinal measure of utility for the first time, he wondered about the absolute relationships between various architectures. This point was clarified, but caused initial confusion. It was his opinion that the attributes were at a level that would be difficult for users to understand. Whereas typical AoA studies used very concrete measures (targets destroyed) the MATE attributes for the SBR analysis were fairly "high level" (see Chapter 3 for a discussion of how the attributes were chosen).

When asked about when a MATE-like analysis would be useful to the JPO, the interviewee answered "as soon as possible." He recommended MATE early in order to provide a supplement to the AoA activity and to broaden its scope. He also indicated that MATE might be a good analysis method for making sourcing decisions.. He noted that the JPO was quickly going to have to make a sourcing decision to move from the five contractors who had submitted proposals to one, two or three in the next round of development. Although the AoA had helped the office to set preliminary requirements, they had little help in choosing which of the contractors would continue development. Lacking a holistic and quantitative way to make these comparisons, the JPO relied on the experience of its members. While this base of knowledge is considerable, judgments made in such a way are unaxiomatic.

On the question of the appropriateness of the attributes, the interviewee generally thought the attributes selected were the right ones. He did note that he might have included attributes for the quality of Digital Terrain and Elevation Data that the system produced, the minimum target cross section that could be detected, and a metric (R-NEARS) that it used by national intelligence agencies to score the quality of an image. He also noted that he would have rated the tracking area capability as by far the most important attribute. Taking his desire to more heavily weight the tracking area attribute, a new set of k-values was used with the MIST2 data in order to experiment with ways to represent the two sets of preferences.

Rank order weighted

```
   k_min_speed = 0.35;
   k_tracking = 0.20;
   k_sar_area = 0.30;
   k_sar_resolution = 0.10;
   k_geo = 0.10;
   k_gap = 0.25;
   k_cog = 0.05;
```

Rank order weighted with JPO Feeback

```
   k_min_speed = 0.35;
   k_tracking = 0.70;
   k_sar_area = 0.30
   k_sar_resolution = 0.10;
   k_geo = 0.10;
   k_gap = 0.25;
   k_cog = 0.05;
```

Figure 36: Weighting Values, JPO Feedback

The challenge is to display both the proxy user and the JPO's preferences simultaneously. There are many ways one could think to display both utility spaces together. The most obvious is to create a three dimensional graph with the first utility on one axis, the second utility space on another, and cost on the third. This results in a pareto-optimal *surface* rather than a *frontier,* where the architectures on the surface represent an optimal trade between the first utility, the second utility, and cost. Though conceptually simple, such a process is practically difficult—trying to visualize this much information is difficult.

Instead, two separate tradespaces were created, one using the first utility data and another using the second. Once each tradespace the pareto-optimal architectures were identified, and a set of architectures that were pareto-optimal under *both* utility measures isolated. This list of architectures and their positions on the original tradespace are shown below.

Arch Number	Cost($b)	Utility	Satellites	Rings	Altitude	Aperture (m)	Scanning	Tech Level
1275	2.145583	0.81298	8	4	1200	40	3	2010
1262	2.145583	0.85704	8	4	1000	40	3	2010
1249	2.145583	0.90196	8	4	800	40	3	2010
15	2.328021	0.91753	9	3	1000	40	1	2002
28	2.328021	0.92299	9	3	1200	40	1	2002
184	2.577832	0.92299	9	3	1200	40	1	2005
186	3.292396	0.92492	12	6	1200	40	1	2005
25	4.64367	0.96382	22	11	1000	40	1	2002
285	7.674934	0.97718	22	11	1000	100	1	2005
129	8.024643	0.97718	22	11	1000	100	1	2002

Figure 37: Common Pareto-Front

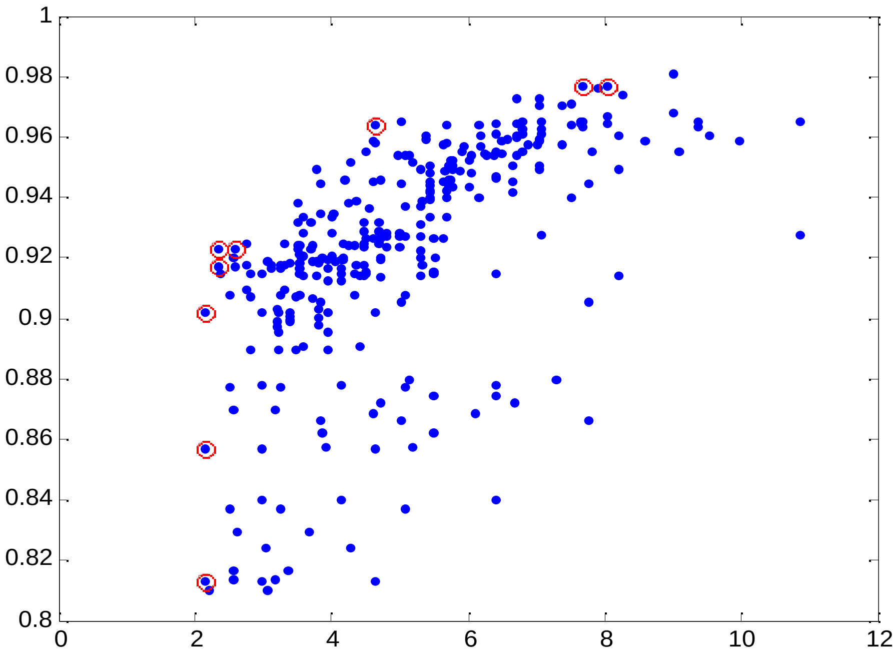


Figure 38: Common Pareto Front Graphically

The figure above is the tradespace of possible architectures as measured by the MIST2 data from Mr. Tonneson. Circled architectures represent architectures that are both on the MIST2 pareto-front and the pareto-front of the tradespace created by using the JPO attribute weights. One can graphically see the intersected set of pareto-front architectures eliminates some of the former possibilities. This is an easy way to visually see which architectures both the JPO and the proxy user agree on. The limitation of such a display is that it only shows the comparison along the pateto-front—the differences in the rest of the tradespace are ignored.

5.4 Conclusions

Taking stock of the analysis and data presented above, what conclusions can be drawn regarding the questions posed by this thesis? First of all, it is clear that the MATE analysis represents a new approach to the kind of analysis that was done at Lincoln Lab. The MATE process separates out the user's preferences in order to analyze possible designs from an unbiased standpoint. This is different from the Lincoln Lab approach, which prized evolution of user preferences.

This is not to say that MATE does not also provide communication between users and designers. This in fact is the goal of the MATE process, and is accomplished through the tradespace analysis shown in Chapter 4. By providing a metric by which all the technical and cost tradeoffs can be understood, MATE functions as a boundary object between the two groups, clearly laying out the tradeoffs for all to see. If this kind of "high quality information channel" is important for enabling spiral acquisition, then MATE certainly offers an ability currently lacking in studies like Lincoln Lab's. More development of how the MATE process can be directly applied to the problems and promises of evolutionary acquisition, see Roberts, 2000.

Though not directly mentioned in this analysis, it was the author's experience that the scheme of adding a MATE analysis onto the existing study architecture is not the preferred way to perform the analysis. Though it might be attractive to simply add the MATE analysis on as a "back-end" to a study already completed, this severely limits the potential the MATE process has for revealing the important aspects of the tradespace. Was a study like Lincoln Lab' begun with the MATE process in mind, the results would certainly have been much more revealing. MATE's potential lies not only in its ability to propagate a universal metric, but also in its ability to change the focus and direction of a study.

A convergence between the Lincoln Lab recommendations and the MATE analysis pareto-front was not shown—in fact quite the opposite. As discussed in Chapter 5, there are a number of reasons why this might be so. It has been the goal of several observers of MATE to validate the revealed preference metrics by making a direct comparison to a project under study. This effort highlights the difficultly in doing so—in the process of calculating attributes, modeling differences are introduced that make results from MATE and a traditional process difficult to compare. A MATE validation through comparison might still be possible, but it is more likely that MATE's benefits over traditional processes will remain un-provable.

This thesis also asked whether MATE should be simplified by using faster and less rigorous preference elicitation techniques. In this case, the answer is clearly no. If one takes the difference between the MIST1 and MIST2 utility data as representative of the kinds of error inherent in the MIST (or any other rigorous) preference elicitation technique, then the approximation techniques add considerable additional error into the tradespace. This is not to say that one might not choose to use the approximations in order to do a "quick and dirty analysis." However, if the physical modeling is going to be rigorous, it is preferable to do a rigorous preference elicitation since other approximations add additional error.

An opposite conclusion can be drawn about the preference weightings—here there was little difference between using a simple ranking scheme and a more rigorous ranking and weighting scheme. Since it is this process that limits the MIST preference elicitation technique to only seven attributes (since decision makers cannot easily balance more than seven attributes at one time in order to reveal their weightings), it is advisable to investigate using less rigorous techniques to find these weights

In the process of seeking to know whether simplification and approximation were appropriate, two useful metrics were developed that can aid future MATE studies. The proportional utility loss and spearman's rho calculations are extremely useful since they offer two ways in which tradespaces can be compared to one another. Along with the other methods of analyzing MATE outputs (like classifying tradespaces by design variables), these tools should be a part of the standard MATE post-study analysis.

The final question asked if MATE would be useful to the Air Force acquisition community. Clearly here the answer is yes, at least for certain types of acquisition efforts. The SBR MATE model shows there are opportunities for MATE to build on the studies typically produced by an AoA for this type of space system. By evaluating a multitude of architectures that are not bound by a strict requirement, MATE can provide valuable analysis that aids the JPO's decision-making and knowledge. It is hoped that MATE can be applied to other types of acquisition efforts, though this is speculative until there are successful applications of MATE beyond the realm of space systems. In general, any acquisition effort that derives its benefit from a networked approach, and is of sufficient technical complexity as to make tradeoff decisions opaque to casual analysis, should be an ideal candidate for MATE.

Furthermore, MATE could be useful for making early sourcing decisions, where there is currently little support provided by the AoA process. One could imagine that the

MATE method could indeed be used as a way to deal with alternative proposals from contractors. The members of the JPO could develop a list of attributes, produce utility curves, and then require the contractors to provide information about how well their proposal would meet the various attributes. This is roughly analogous to providing the contractors a set of requirements to which they can design their proposals, but would provide for more flexibility and a metric through which the various proposals could be evaluated. The benefit of this strategy is that the contractors submitting proposals would bear the burden of creating the technical models that linked their designs to the various attribute levels. The JPO would only have to create the list of attributes and the utility curves, using these to evaluate the proposal. A single member of the JPO, instructed in the MATE method, could perform this task. The drawback back here is that if the contractor builds the performance models, there is incentive to make them overly optimistic.

Another option is for the JPO to create the models itself, taking instead design variables from the contractor proposals. This would mean the technical modeling was owned by the JPO, and could therefore be more closely controlled. Under such a system however, more expertise would be needed, and it is likely that many people would be involved. In such a case, the modeling would probably need to be done by an outside organization, most likely an FFRDC. In either scenario the MATE method would be useful in evaluating proposals from competing contractors.

Representing multiple stakeholders with the MATE method remains problematic—creating an intersection of pareto-optimal fronts was useful in the SBR case but is not guaranteed to be useful generally. One could easily imagine a case where two sets of preferences are different enough as to result in no overlap between pareto-fronts. It is more likely that MATE will continue to be best applied by using a single set of utility curves that are a product of consensus among various interests. Creating this consensus is not easy itself however. As learned from prior experience with the MATE method, much iteration and communication is necessary to arrive at a set of utility curves that accurately represent the group preferences of even a small number of decision-makers. This consensus would be best achieved by a series of iterations where a number of decision-makers give their preferences through the MIST software, and then compare and decide upon a common set of preferences.

5.5 Further Research

Many areas for further research have been revealed in the course of this study. First and foremost, it would be useful to further develop the body of data from repeat interview with the same user (or proxy user). Is the MIST1 data used in this analysis really a good representation of the kind of errors likely to be found in the preference elicitation process? If it can be shown that elicitation errors from MIST interviews are larger on average than those used in this study, approximations to utility curves may become a more appealing option.

Further studies should attempt to derive attribute weightings by methods that do not rely on the laborious MIST interview process. There is much decision theory literature that would be helpful in this regard. Several method could be tried, with the most user friendly method eventually becoming part of the MATE method. This would allow entirely new kinds of interviews and analyses, specifically ones that included many more than seven attributes. Such a capability would allow for easier application of the MATE process to larger scale problems like those involving complicated networked systems.

The interaction with the JPO indicates that there is further work that could be done in attempting to use MATE in an actual Air Force acquisition context. It would be interesting to see if it were possible to use MATE to aid in a sourcing decision. Preliminary research in this regard might start by analyzing a prior sourcing decision, using that experience to develop a method by which a JPO could use MATE in coordination with its bidding contractors. The future of MATE, at least in the Air Force context, is likely to involve working with program managers and analysts at JPOs and SPOs.

This Page Intentionally Blank

Appendix A: Utility Data

Four sets of utility curves were elicited from the proxy user. A member of the Lincoln Lab buyer team was chosen to play this role. This selection was ideal since members of that team were already tasked with trying to think through the kind of capabilities that would be desired by the combatant commanders. The proxy user was Mr. Larry Tonneson, of Zel-Tec, Inc. Mr. Tonneson was a government contractor working on the study specifically tasked with thinking of the scenarios in which Space Based Radar would be useful. Mr. Tonneson was ideal for this task in a another respect due to his prior Air Force experience as an officer and crewmember aboard the AWACS aircraft, which functions in an operational context as an integrator of several sources of battlefield intelligence.

Mr. Tonneson developed a set of attributes over the span of several weeks, and identified the ranges of acceptable values on these attributes. He then provided utility data on these attributes during two separate interviews. Two of these were hand-drawn curves (subjective utility) and two were from the MIST interviews. Also, a linear approximation was made for comparison. The data are shown below, first in tabular format, then again graphically. All four sets of Mr. Tonneson’s responses are shown together on these graphs.

A.1 Summary

	MIST 1		HAND 1		MIST 2		HAND 2		Linear	
	# of Boxes	Utility	# of Boxes	Utility	# of Boxes	Utility	# of Boxes	Utility	# of Boxes	Utility
Tracking Area	10.00	0	10.00	0	10.00	0	10.00	0	10.00	0
	20.83	0.6	20.83	0.03	21.00	0.85	12.00	0.1	20.83	0.1667
	31.67	0.8	31.67	0.15	32.00	0.85	14.00	0.3	31.67	0.3333
	42.50	0.9	42.50	0.43	42.00	0.8	20.00	0.56	42.50	0.5
	53.33	0.9	53.33	0.82	53.00	0.85	35.00	0.76	53.33	0.6667
	64.17	0.9	64.17	0.96	64.00	0.85	55.00	0.92	64.17	0.8333
	75.00	1	75.00	1	75.00	1	75.00	1	75.00	1
	MPH	Utility	MPH	Utility	MPH	Utility	MPH	Utility	MPH	Utility
Min Speed	5.00	1	5.00	1	5.00	1	5.00	1	5.00	1
	12.50	0.9	12.50	0.95	12.50	0.8	15.00	0.95	12.50	0.8333
	20.00	0.7	20.00	0.7	20.00	0.75	23.00	0.54	20.00	0.6667
	27.50	0.4	27.50	0.31	27.50	0.75	30.00	0.27	27.50	0.5
	35.00	0.2	35.00	0.08	35.00	0.55	36.00	0.15	35.00	0.3333
	42.50	0.2	42.50	0.01	42.50	0.05	42.00	0.06	42.50	0.1677
	50.00	0	50.00	0	50.00	0	50.00	0	50.00	0
	Square miles	Utility	Square miles	Utility	Square miles	Utility	Square miles	Utility	Square miles	Utility
SAR Area	0.50	0	0.50	0	0.50	0	0.50	0	0.50	0
	17.08	0.7	17.08	0.27	17.00	0.85	18.00	0.24	17.08	0.1667
	33.67	0.7	33.67	0.52	34.00	0.85	35.00	0.7	33.67	0.3333
	50.25	0.9	50.25	0.88	50.00	0.85	45.00	0.85	50.25	0.5
	66.83	0.9	66.83	0.97	67.00	0.85	60.00	0.94	66.83	0.6667
	83.42	0.9	83.42	1	83.00	0.85	80.00	0.96	83.42	0.8333
	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
	Meters	Utility	Meters	Utility	Meters	Utility	Meters	Utility	Meters	Utility
SAR resolution	0.50	1	0.50	NDA	0.50	1	0.50	1	0.50	1
	0.92	0.9	0.92	NDA	0.92	0.85	0.75	0.97	0.92	0.8333
	1.33	0.9	1.33	NDA	1.33	0.85	1.00	0.93	1.33	0.6667
	1.75	0.9	1.75	NDA	1.75	0.8	1.25	0.76	1.75	0.5
	2.17	0.5	2.17	NDA	2.17	0.8	1.50	0.45	2.17	0.3333
	2.58	0.1	2.58	NDA	2.58	0.4	2.00	0.2	2.58	0.1677
	3.00	0	3.00	NDA	3.00	0	3.00	0	3.00	0
	Meters	Utility	Meters	Utility	Meters	Utility	Meters	Utility	Meters	Utility
Accuracy	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1
	125.00	0.7	125.00	0.82	125.00	0.85	100.00	0.85	125.00	0.8333
	200.00	0.6	200.00	0.67	200.00	0.85	160.00	0.65	200.00	0.6667
	275.00	0.7	275.00	0.51	275.00	0.8	250.00	0.45	275.00	0.5
	350.00	0.6	350.00	0.25	350.00	0.8	330.00	0.25	350.00	0.3333
	425.00	0.1	425.00	0.06	425.00	0.2	410.00	0.1	425.00	0.1677
	500.00	0	500.00	0	500.00	0	500.00	0	500.00	0
	Minutes	Utility	Minutes	Utility	Minutes	Utility	Minutes	Utility	Minutes	Utility
Gap Time	5.00	1	5.00	1	5.00	1	5.00	1	5.00	1
	14.17	0.7	14.17	0.92	14.00	0.85	15.00	0.8	14.17	0.8333
	23.33	0.9	23.33	0.63	23.00	0.85	25.00	0.425	23.33	0.6667
	32.50	0.7	32.50	0.21	32.00	0.8	35.00	0.1	32.50	0.5
	41.67	0.4	41.67	0.06	42.00	0.7	45.00	0.05	41.67	0.3333
	50.83	0.1	50.83	0	51.00	0.2	55.00	0.05	50.83	0.1677
	60.00	0	60.00	0	60.00	0	60.00	0	60.00	0
	# of Boxes	Utility	# of Boxes	Utility	# of Boxes	Utility	# of Boxes	Utility	# of Boxes	Utility
COG Area	1.00	0	1.00	0	1.00	0	1.00	0	1.00	0
	2.33	0.3	2.33	0.32	2.00	0.4	2.00	0.3	2.00	0.25
	3.67	0.7	3.67	1	3.00	0.85	3.00	0.75	3.00	0.5
	5.00	1	5.00	1	4.00	0.85	4.00	0.95	4.00	0.75
					5.00	1	5.00	1	5.00	1

Figure 39: Utility Data

A.2 Graphs

Below are the graphs of Mr. Tonneson's utility preferences. In general it is interesting to note that MIST generated utility curves show more risk aversion than do the hand-drawn subjective utility curves. This is to be expected, since the subjective utility represented by the hand-drawn curves does not formally include any accounting for risk.

In the MIST1 data there are several instances of "preference reversal," which is not consistent with formal utility theory. Preference reversal occurs when the utility curve is not a monotonically increasing or decreasing function. These were interpreted to be mistakes on the part of Mr. Tonneson, who was initially unfamiliar with the MIST interview process. In the analysis, the MIST1 data is interpreted as the kind of errors one might expect for this type of preference elicitation.

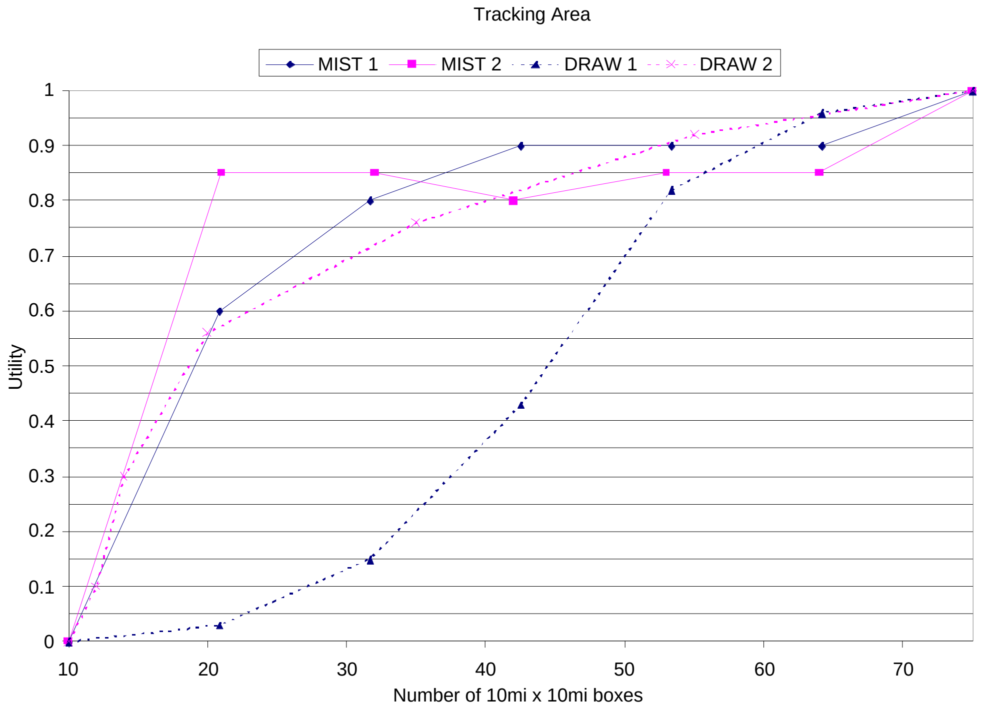


Figure 40: Tracking Area Utility Curves

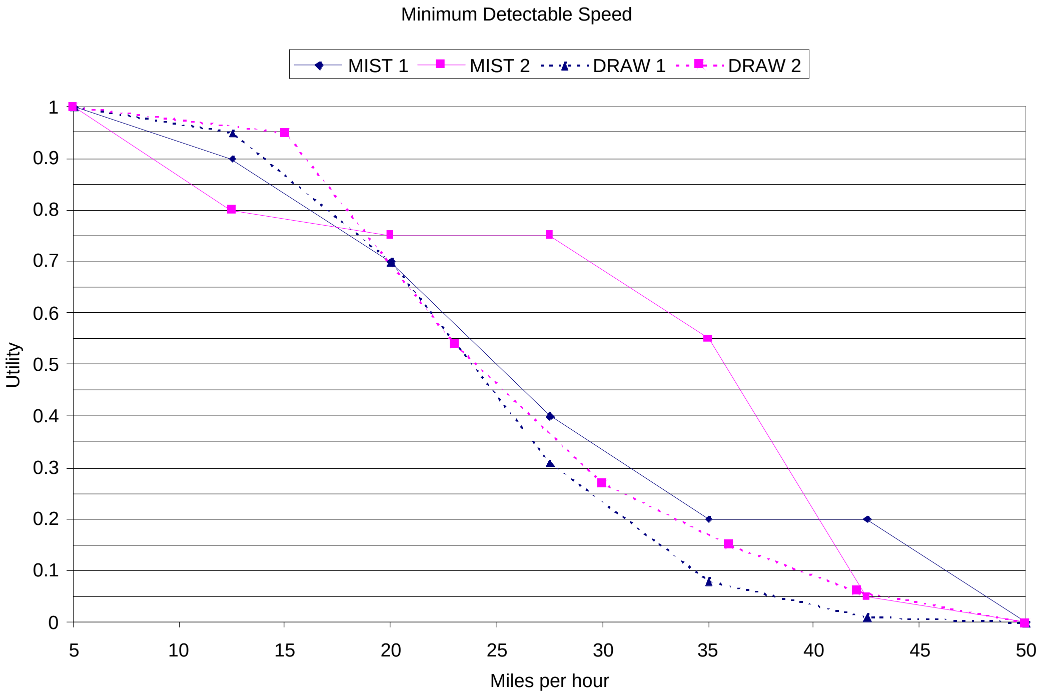


Figure 41: Minimum Speed Utility Curves

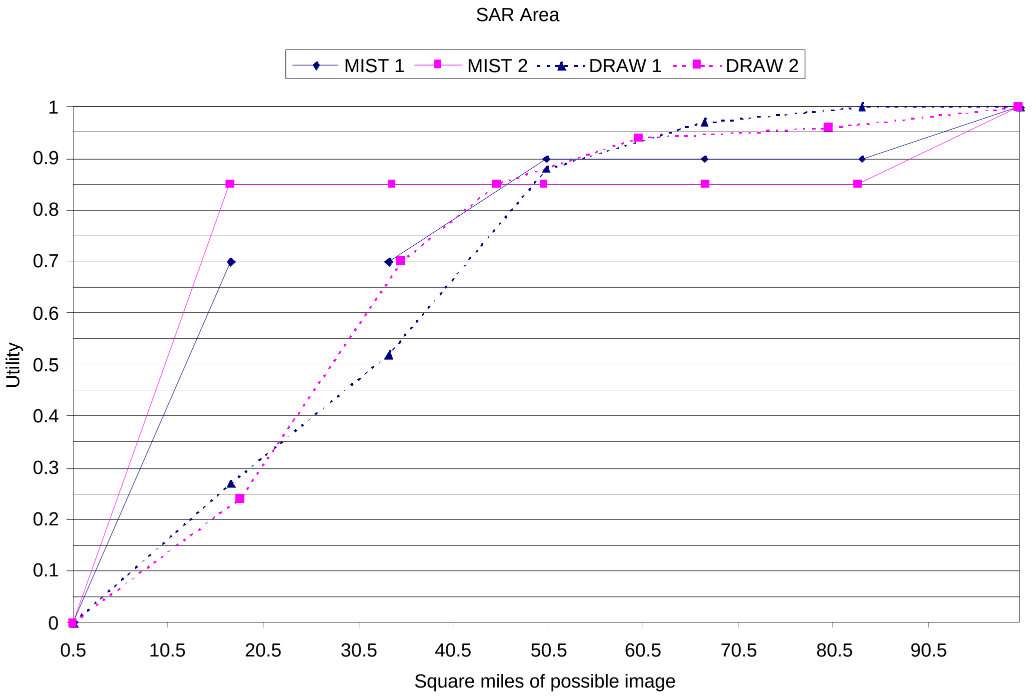


Figure 42: SAR Area Utility Curves

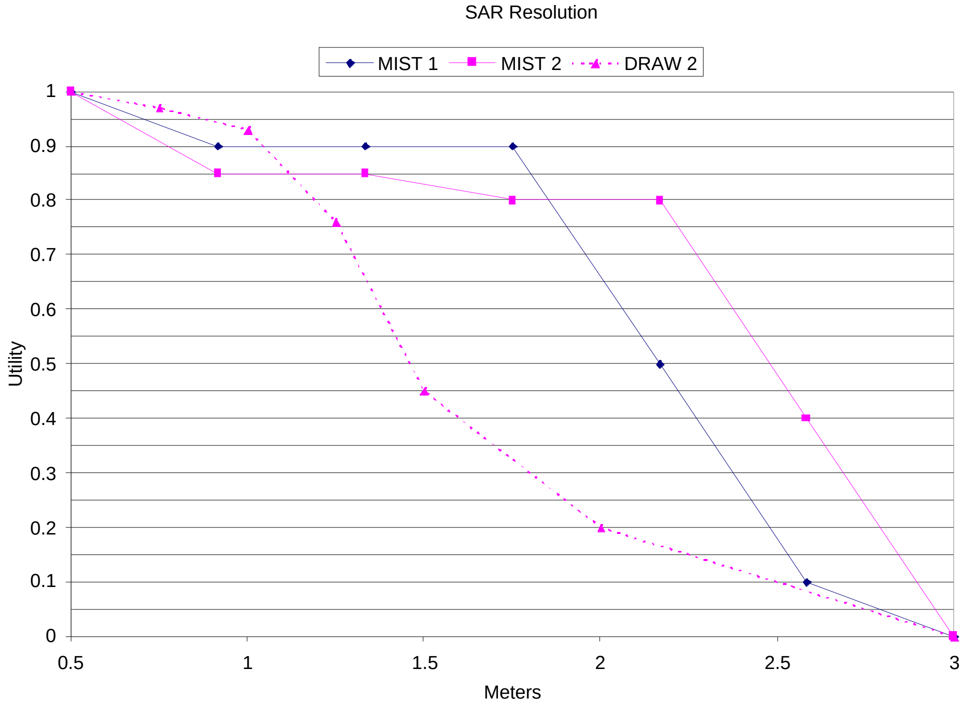


Figure 43: SAR Resolution Utility Curves

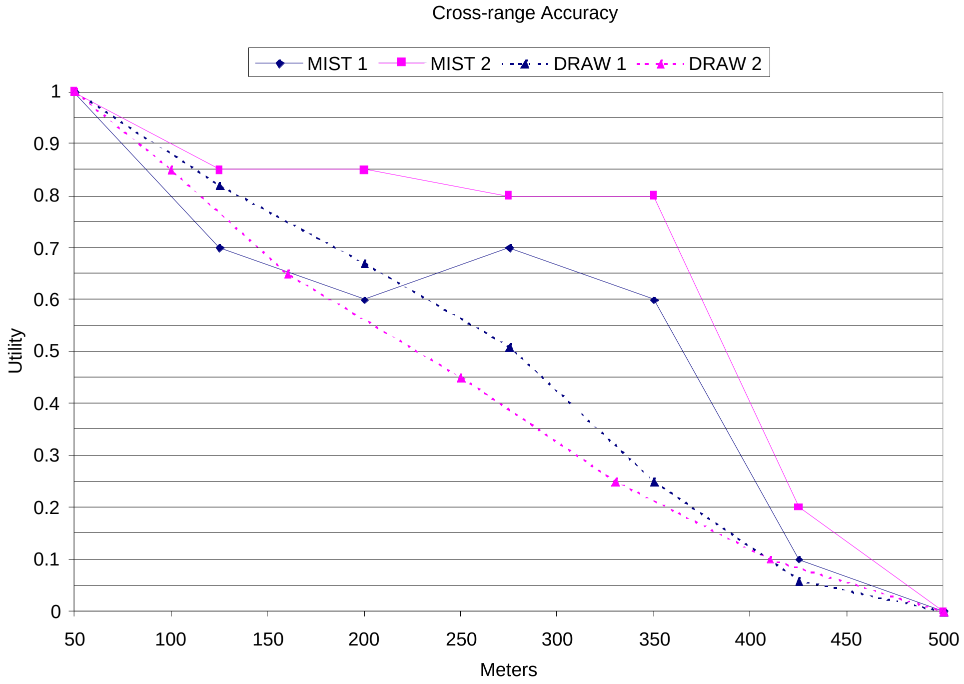


Figure 44: Cross Range Accuracy Utility Curves

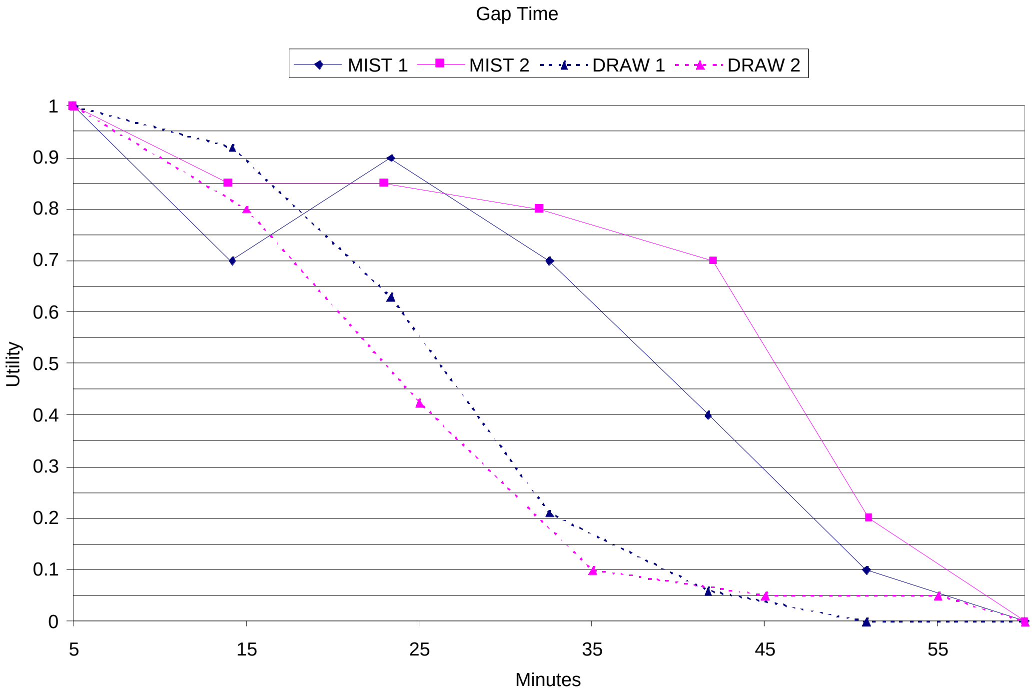


Figure 45: Gap Time Utility Curves

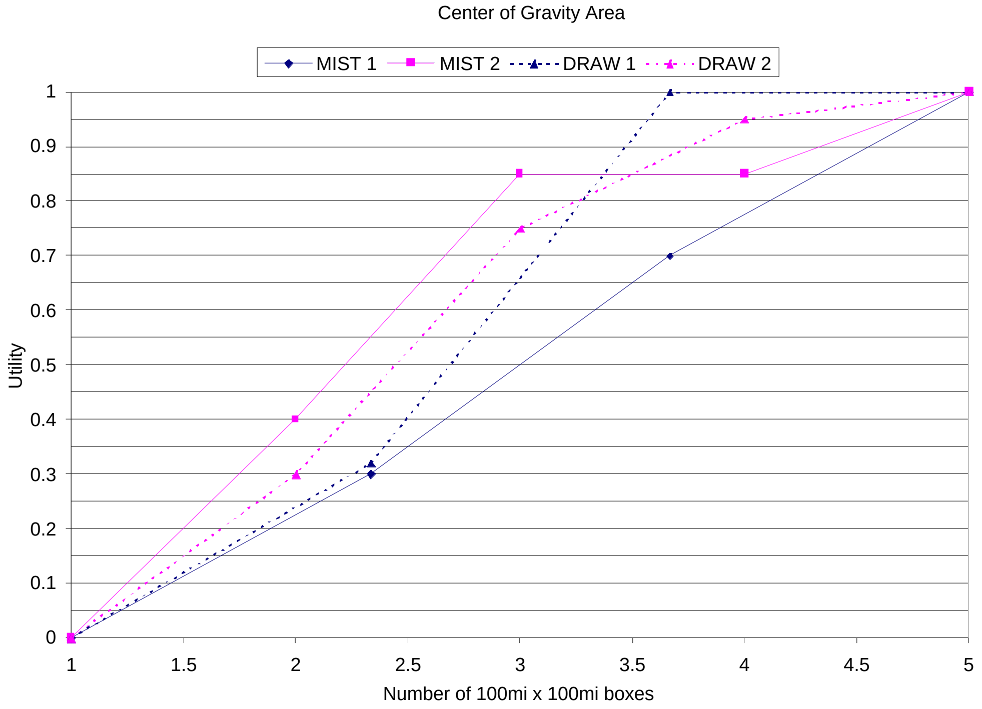


Figure 46: Center of Gravity Utility Curves

Appendix B: Model Code

B.1 Software Architecture

Before describing the modules in detail, it is helpful to see the overall flow of computation. This can be most easily seen in an N^2 diagram, where each module's interconnections can be readily seen.

	MAIN	ORBITS	MASS	COST	RADAR	UTILITY
MAIN						
ORBITS	x					
MASS	x					
COST	x		x			
RADAR	x	x				
UTILITY	x	x			x	

Figure 47: N^2 for SBR Model

This chart summarizes the dependencies in the flow of the model. Since it is lower triangular, there is no feedback between modules, which means the software will always converge on a solution.

The reality of the model is more complicated than is shown in this depiction however, since much of the analysis is completed off-line. Most of this off-line work is input directly into the Radar module. The more complete software architecture map below summarizes this input.

In general, the model is executed by first constructing a database of all possible design iterations (every combination of the design variables). Each one of these possible designs is then fed through the rest of the model in turn, in order to calculate its performance characteristics, as well is its cost. Since calculating radar performance is computationally expensive, these calculations were done offline, as discussed above.

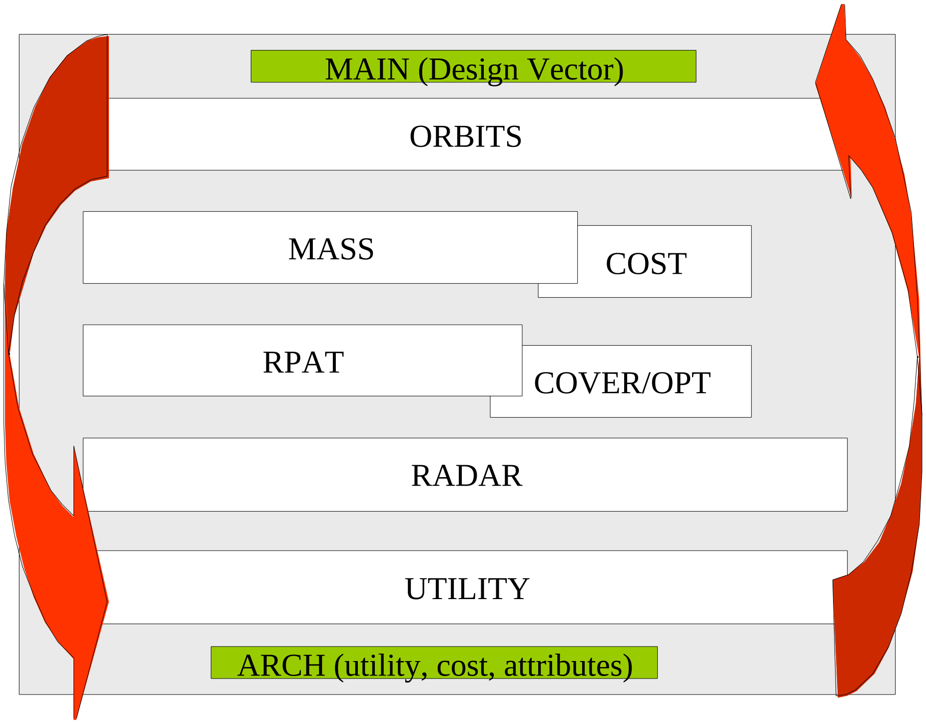


Figure 48: Software flow for SBR Model

B.2 Module Descriptions

Below each module is presented in further detail, paying special attention to the inputs and outputs, as well as the assumptions underlying each method of calculation. The actual module code follows each description.

Main Module

The main module consists of four sections. The first is the database section, which sets up and populates a database containing all possible design combinations across the design variables. This database (contained in a Matlab structure variable) will eventually contain all the information pertinent to the analysis. The structure is organized along field names, and contains a full entry for each potential architecture. For instance, to access data on the first architecture, a user would type: "arch(1).fieldname"

The second section takes the k-values from the utility interview process and calculates a multi-attribute coefficient. This coefficient is used to combine the single attribute utility values into one multi-attribute utility measure. The calculation is preformed in the sub-module "calculate K," which is shown below.

The third section calls each module in turn, saving whatever data is necessary in the arch structure. It does not call the RPAT and Coverage/Opt Time sequence, since those have been run off-line already.

The final section plots the data, with utility on the y-axis, and life cycle cost on the x. It produces several graphs colored according to various design variables.

Assumptions:

The fundamental assumption in the main module is that every possible design combination represents an actually realizable architecture. In a generic tradespace analysis, this is not necessarily so. However, the design variables were chosen such that every combination yields a physical possible solution. This assumption has an effect on the shape of the tradespace. Since only physically realizable designs are admitted, the pareto-front is likely to be less smooth, since gaps exist due to these restrictions. This has implications for how one interprets the pareto-front—there can be no interpolation between points on this front, since those hypothetical points represent non-possible solutions. Another simplifying assumption restricts the set of potential architectures to those for which all satellites are of the same physical design. One could imagine an architecture that combined several different types of satellites. Including such cases presents a difficultly in calculating utility values however, and so was not included in this analysis.

```
%.............................................................
% SPACE BASED RADAR MATE MODEL................................
% Tim Spaulding...............................................
%.............................................................
%.............................................................
% Main Module.................................................
% Origin: MIT Lincoln Lab, 22 Aug 2002.......................
% Final Revision: MIT LAI 5 March 2003........................
%.............................................................

clear all
close all


%------------------------------------------------------
% FUNCTION INPUTS
%------------------------------------------------------
% none
%-------------------------------------------------------



%------------------------------------------------------------------
% FUNCTION OUTPUTS
%------------------------------------------------------------------
% arch.num = 1;                     architecture number [#]
% arch.scan_angles = 1;             scan angle          [deg x deg]
% arch.tech_levels = 1;             tech level          [year]
% arch.apeture_sizes = 1;           aperture size       [square meters]
% arch.altitude = 1;                altitude            [kilometers]
% arch.number_of_satellites = 1;    num of sats         [#]
% arch.number_of_rings=1;           num of rings        [#]
% arch.cost = 1;                    lifecycle cost      [FY02 Billion $]
% arch.mass = 1;                    satellite mass      [kilograms]
% arch.utility = 1;                 Multi Att. Utility  [0-1]
% arch.u_min_speed = 1;             Min speed utility   [0-1]
% arch.u_tracking = 1;              Tracking Utility    [0-1]
% arch.u_sar_area = 1;              SAR utility         [0-1]
% arch.u_sar_resolution = 1;        resolution utility  [0-1]
% arch.u_geo = 1;                   Geolocat. utility   [0-1]
% arch.u_gap = 1;                   Gap time utility    [0-1]
% arch.u_cog = 1;                   COG utility         [0-1]
%------------------------------------------------------------------

% ------------------------------------------------------------------------------------
% CREATING THE DATABASE OF POTENTIAL ARCHITECTURES
% ------------------------------------------------------------------------------------

% this creates a structure containing all the design variables under consideration
database = struct('scan_angles',{{'5x5','20x5','45x15','30x15'}},...
    'tech_levels',{{'2002','2005','2010'}},...
    'apeture_sizes',[40 70 100],...
    'altitude',[800 1000 1200 1400],...
    'number_of_satellites',[8 9 10 12 15 16 17 18 19 20 21 22 24]);

counter = 1;            % this will number each architecture
arch = struct([]);      % these commands initialize the arch structure
arch = 1;
arch.num = 1;
arch.scan_angles = 1;
arch.tech_levels = 1;
arch.apeture_sizes = 1;
```

```
arch.scanning_type = 1;
arch.altitude = 1;
arch.number_of_satellites = 1;
arch.number_of_rings=1;
arch.cost = 1;
arch.mass = 1;
arch.utility = 1;
arch.u_min_speed = 1;
arch.u_tracking = 1;
arch.u_sar_area = 1;
arch.u_sar_resolution = 1;
arch.u_geo = 1;
arch.u_gap = 1;
arch.u_cog = 1;



% Each possible combinations of the design variables is enumerated and the designs stored
in the arch structure.
for loop1 = 1:length(database.scan_angles)
    for loop2 = 1:length(database.tech_levels)
        for loop3 = 1:length(database.apeture_sizes)
                for loop5 = 1:length(database.altitude)
                        for loop7 = 1:length(database.number_of_satellites)
                            arch(counter).scan_angles = database.scan_angles(loop1);
                            arch(counter).tech_levels = database.tech_levels(loop2);
                            arch(counter).apeture_sizes = database.apeture_sizes(loop3);
                            arch(counter).altitude = database.altitude(loop5);
                            arch(counter).number_of_satellites =
database.number_of_satellites(loop7);
                            arch(counter).num = counter;
                            counter=counter+1;
                        end%7
                    end%5
%             end%4
        end%3
    end%2
end%1

counter-1
disp ('architectures created')      % displays the total number of architectures
%------------------------------------------------------------------------------------



%--------------------------------------------------------------------------------------
%CALCULATIING THE MULTIPLICATIVE CONSTANT (K)
%---------------------------------------------------------------------------------------
    k_min_speed = 0.35;
    k_tracking = 0.20;
    k_sar_area = 0.30;                              % these are inputs from the MIST
interviews
    k_sar_resolution = 0.10;
    k_geo = 0.10;
    k_gap = 0.25;
    k_cog = 0.05;

k_vector = [k_min_speed;k_tracking;k_sar_area;k_sar_resolution;...
    k_geo;k_gap;k_cog];

big_k = calculate_k(k_vector);                      % Big_k calculates the multiplicitave
factor needed
                                                    % to calculate Multi Attribute
utility
%--------------------------------------------------------------------------------------

%-------------------------------------------------------------------
% EVALUATING THE ARCHITECTURES
% this section calls each submodule in turn, saving the results from the
% cost and utility modules into the arch structure
%----------------------------------------------------------------------
```

```
for loop = 1:counter-1  % loops through all of the design combinations

    % ORBITS MODULE--------------------------------
    % calculates the summary statistic for the constellation under study
    orbitdata = orbits (arch(loop).number_of_satellites,arch(loop).altitude);
    arch(loop).number_of_rings = orbitdata(1); % saves the walker stats info into the
arch
    %---------------------------------------------

    % MASS MODULE---------------------------------
    % calculates the mass of each satellite under study
    mass_summary =...

masspower(arch(loop).apeture_sizes,arch(loop).scan_angles,arch(loop).tech_levels);
    arch(loop).mass = sum(mass_summary); % saves the mass info into the arch
    %--------------------------------------------

    % COST MODULE---------------------------------
    % cacluates the lifecycle cost for each desing
    sc_life = 10; %satellite lifetime assumption
    life_cycle_cost = cost (mass_summary, sc_life,
arch(loop).number_of_rings,arch(loop).number_of_satellites);
    arch(loop).cost = life_cycle_cost; % saves the cost into the arch
    %---------------------------------------------

    % RADAR PERFORMANCE MODULE--------------------
    % This module calculates the performance of each design's radar
    attributes = radar
(arch(loop).altitude,arch(loop).tech_levels,arch(loop).apeture_sizes,...
        arch(loop).scanning_type,arch(loop).scan_angles,...
        arch(loop).number_of_satellites,arch(loop).number_of_rings,orbitdata(2));

    % this saves the attribute values in the arch
    arch(loop).min_speed = attributes(1);
    arch(loop).tracking = attributes(2);
    arch(loop).sar_area = attributes(3);
    arch(loop).sar_resolution = attributes(4);
    arch(loop).geo = attributes(5);
    arch(loop).gap = attributes(6);
    arch(loop).cog = attributes(7);
    %---------------------------------------------

    % UTILITY MODULE------------------------------
    % This calculates the utility scores for each design
    utility_vector = utility(big_k,k_vector,attributes);

    % this saves the utility values in the arch
    arch(loop).utility = utility_vector(1);
    arch(loop).u_min_speed = utility_vector(2);
    arch(loop).u_tracking = utility_vector(3);
    arch(loop).u_sar_area = utility_vector(4);
    arch(loop).u_sar_resolution = utility_vector(5);
    arch(loop).u_geo = utility_vector(6);
    arch(loop).u_gap = utility_vector(7);
    arch(loop).u_cog = utility_vector(8);
    %---------------------------------------------
end % for
%--------------------------------END OF LOOP TO CREATE TRADESPACE-------------------------
----------------



% This scales the cost numbers into Billions FY02$
for loop = 1:counter-1
    arch(loop).cost = arch(loop).cost/10^9;
end



close all
% This eliminates all NaN designs
```

```
xx=cat(arch.cost);
yy=cat(arch.utility);
zz=cat(arch.num);

[ans,I] = find(~isnan(xx));
[ans,J] = find(~isnan(yy'));
II = intersect(I,J);
 xx = xx(II);
 yy = yy(II);
 zz = zz(II);



% this will convert tech level and scanning so they can be displayed
for loop = 1:counter-1
scan_case = char(arch(loop).scan_angles);
switch scan_case
case '5x5'
    column = 1;
case '20x5'
    column = 2;
case '45x15'
    column = 3;
case '30x15'
    column = 4;
end    % switch
arch(loop).newscan = column;
arch(loop).newtech = eval(char(arch(loop).tech_levels));
end


cost = xx;
utility = yy;

% Draws the scatter plots of the tradespaces for different characteristics
figure; gscatter(cost,utility,[arch(zz).apeture_sizes])
figure; gscatter(cost,utility,[arch(zz).number_of_satellites])
figure; gscatter(cost,utility,[arch(zz).altitude])
figure; gscatter(cost,utility,[arch(zz).newscan])
figure; gscatter(cost,utility,[arch(zz).newtech])

% creates the generic tradespace picture
figure
scatter(xx,yy, SizeOfDots);
hold on;

% determines which designs are on the "pareto front"
% by finding the convex hull
CC = convhull(xx,yy);
plot(xx(CC),yy(CC),'r-')

% Labels each of the points on the hull
AA = num2str(zz(CC));
HH = text([arch(zz(CC)).cost],[arch(zz(CC)).utility],AA);
set(HH,'FontSize',6);
set(HH,'Color','cyan');
set(HH,'Color','black');
set(HH,'HorizontalAlignment','right');
set(HH,'VerticalAlignment','bottom');

%now save the information regarding the archs that are on the hull
for loop = 1:length(CC)
    specs(loop,1) = arch(zz(CC(loop))).num;
    specs(loop,2) = arch(zz(CC(loop))).cost;
    specs(loop,3) = arch(zz(CC(loop))).utility;
    specs(loop,4) = arch(zz(CC(loop))).number_of_satellites;
    specs(loop,5) = arch(zz(CC(loop))).number_of_rings;
    specs(loop,6) = arch(zz(CC(loop))).altitude;
    specs(loop,7) = arch(zz(CC(loop))).apeture_sizes;
    specs(loop,8) = arch(zz(CC(loop))).newscan;
    specs(loop,9) = arch(zz(CC(loop))).newtech;
```

```
end
% Export this information to an ASCII file
save specs.txt specs -ASCII
```

```
function [big_kay] = calculate_k (k)
%.............................................................
% SPACE BASED RADAR MATE MODEL................................
% Tim Spaulding...............................................
%.............................................................
%.............................................................
% Calculate K.................................................
% Origin: MIT Lincoln Lab, 22 Aug 2002.......................
% Final Revision: MIT Lincoln Lab 2 Aug 2003.................
%.............................................................

%------------------------------------------------------
% FUNCTION INPUTS
%------------------------------------------------------
% k                                 vector of k-values, one for each attribute
%-------------------------------------------------------


%------------------------------------------------------------------
% FUNCTION OUTPUTS
%------------------------------------------------------------------
% big_kay                           Multiplicative constant for MAUV
%------------------------------------------------------------------



num_attributes = length(k); %sets the loop variable for the next part
total_k=sum(k);
K = sym('K');   % this creates a symbolic variable 'K', which allows you to use the
'solve' function



    big_kay= solve((K*k(1)+1)*(K*k(2)+1)*(K*k(3)+1)*(K*k(4)+1)*(K*k(5)+1)*...
        (K*k(6)+1)*(K*k(7)+1)-K-1,'K'); % solves equation for 'K'

    big_kay = double(big_kay); %to use the next section, you must convert from a symbolic
arrayto a double accuracy arra



    % this is the loop that slects the correct (of the num_attributes) root
%-----------------------------------------------------------------------------------%
    actual_K=1000;                                                                  %
    if (num_attributes > 1)                                                         %
        for i=1:(num_attributes)                                                    %
            if (total_k<1)                                                          %
                if (isreal(big_kay(i)) & (big_kay(i)>0))                            %
                    actual_K=big_kay(i);                                            %
                end                                                                 %
            end                                                                     %
            if (total_k>1)                                                          %
                if(isreal(big_kay(i)) & (big_kay(i)<0) & (big_kay(i)>-1))           %
                    actual_K=big_kay(i);                                            %
                end                                                                 %
            end                                                                     %
            if (total_k==1)                                                         %
                actual_K=0;                                                         %
            end                                                                     %
        end %for                                                                    %
    end % outer if                                                                  %
%-----------------------------------------------------------------------------------%


    big_kay=actual_K;
```

Mass and Power Module

The mass and power module calculates component masses predicated on a set of the architecture configurations. The modules itself is a lookup table, composed of data taken from the Lincoln Lab summer study (specifically from work done by Robert Harvey of Lincoln Lab.)

Assumptions:

There are assumptions implied in the mass estimation performed by Robert Harvey, specifically on the mass of the various antenna sizes, and the level of technology that composes them. Also, a ten year lifecycle was used to estimate power system sizing.

```
function mass_summary = masspower(antenna_size, scan_case, tech_level);

%.............................................................
% SPACE BASED RADAR MATE MODEL................................
% Tim Spaulding...............................................
%.............................................................
%.............................................................
% Mass and Power Module.......................................
% Origin: MIT Lincoln Lab, 3 July 2002.......................
% Final Revision: MIT LAI 5 March 2003........................
% Adapted from an excel sheet by Robert Harvey, MIT LL........
%.............................................................


%-------------------------FUNCTION INPUTS---------------------------------------------
    antenna_size ;  % 40, 70, or 100                                 meters square
    scan_case= char(scan_case) ;% (5x5), (20x5), (45x15) , (30x15)   degrees
    tech_level= char(tech_level) ;% 2002,2005, or 2010               unitless
    load lookup.mat;% this the the lookup table, from excel
%---------------------------------------------------------------------------------------

% %----------------------FUNCTION OUTPUTS---------------------------------------
% % BUS SYSTEMS
%   acs_mass= mass_summary(1);             %Altitude Control System Mass            kg
%   propsys_mass= mass_summary(2);         %Propulsion System (dry) Mass            kg
%   prop_mass= mass_summary(3);            %Propellant Mass                         kg
%   power_mass= mass_summary(4);           %Conditioner and Battery Mass            kg
%   tcs_mass= mass_summary(5);             %Thermal Control System Mass             kg
%   comm_mass= mass_summary(6);            %Communication Systems Mass              kg
%   structure_mass= mass_summary(7);       %Support Structure Mass                  kg
%   harness_mass= mass_summary(8);         %Harness Mass                            kg
% %
% % PAYLOAD
%   elec_mass = mass_summary(9);           %Electronics Mass                        kg
% %
%         % ANTENNA
%         ant_structure_mass= mass_summary(10);  %Antenna Support Structure Mass    kg
%         ant_tr_mass= mass_summary(11);        %T/R module mass                    kg
%         ant_r_mass= mass_summary(12);         %R module mass                      kg
%         ant_panel_mass= mass_summary(13);     %Antenna Panels mass                kg
%         ant_electronics_mass= mass_summary(14);%Antenna electonics mass           kg
%         ant_cables_mass= mass_summary(15);    %Antennas electronics mass          kg
%
```

```
% % SOLAR ARRAY
% solar_mass= mass_summary(16);         %Solar Array Mass                           kg
% %------------------------------------------------------------------------------


%--------------------------------------------------------------------------------
% The lookup table has three indices: antenna size, scan case, and technology level
% the data is contained in a two demensional table, with antenna size as the larger
% column index, and scan case the smaller index.  Technology level is the row index.
% Each point in this three indice table is a vector of 16 component masses. These
% component masses are output to mass_summary(1:16)
%
% The following code retrives this data:

column_big = 0;
column_small = 0;
row = 0;

switch antenna_size
case 40
    column_big = 9;
case 70
    column_big = 5;
case 100
    column_big = 1;
otherwise
    disp ('ERROR IN MASSPOWER ANTENNA SIZE')
end

switch scan_case
case '5x5'
    column_small = 0;
case '20x5'
    column_small = 1;
case '45x15'
    column_small = 2;
case '30x15'
    column_small = 3;
otherwise
    disp ('ERROR IN MASSPOWER SCAN CASE')
end

switch tech_level
case '2002'
    row = 1;
case '2005'
    row = 17;
case '2010'
    row = 33;
otherwise
    disp ('ERROR IN MASSPOWER TECH LEVEL')
end

mass_summary = data(row:row+15,column_big+column_small);
total_mass = sum(mass_summary);
%--------------------------------------------------------------------------------------
```

Cost Module

The cost module is a simple adaptation of the Air Force Cost Model, 7th Edition, with minimum percent error. It takes in the mass summary from the mass and power module, using these component masses to calculate the total life-cycle cost of the system.

Assumptions:

One additional constraint beyond the typical cost model is that only Delta launch vehicles are considered. This was a restriction that Lincoln Lab used in its study, so it was mirrored here. Additionally, there is a rubber spacecraft assumption (i.e. the sizing for the launch vehicle is done entirely by mass—it is assumed that it will be able to physical fit on the launch vehicle if it is light enough. As many satellites as possible are placed on the same launch vehicle, provided they are going to the same orbital plane.

```
function life_cycle_cost = cost (mass_summary, sc_life, sc_rings,sc_num)

%..........................................................
% SPACE BASED RADAR MATE MODEL...............................
% Tim Spaulding..............................................
%..........................................................
%..........................................................
% Life Cycle Cost Module....................................
% Origin: MIT Lincoln Lab, 3 July 2002......................
% Final Revision: MIT LAI 5 March 2003.......................
% Adapted from an excel sheet by Robert Harvey...............
% CERs base on USAF UNMANNED SPACE COST MODEL, 7th ed., Min % error
%..........................................................



%----------------------FUNCTION INPUTS---------------------------------------

% BUS SYSTEMS

acs_mass = mass_summary(1) ;            %Altitude Control System Mass            [kg]

propsys_mass = mass_summary(2)  ;       %Propulsion System (dry) Mass            [kg]

prop_mass = mass_summary(3)  ;          %Propellant Mass                         [kg]

power_mass = mass_summary(4)  ;         %Conditioner and Battery Mass            [kg]
tcs_mass = mass_summary(5)   ;          %Thermal Control System Mass             [kg]
comm_mass = mass_summary(6)   ;         %Communication Systems Mass              [kg]
structure_mass = mass_summary(7) ;      %Support Structure Mass                  [kg]
harness_mass = mass_summary(8)  ;       %Harness Mass                            [kg]
%

% PAYLOAD

elec_mass = mass_summary(9) ;           %Electronics Mass                        [kg]

%

% ANTENNA

ant_structure_mass = mass_summary(10); %Antenna Support Structure Mass          [kg]
ant_tr_mass = mass_summary(11)   ;     %T/R module mass                         [kg]

ant_r_mass = mass_summary(12) ;        %R module mass                           [kg]

ant_panel_mass = mass_summary(13) ;    %Antenna Panels mass                     [kg]
```

```
ant_cables_mass = mass_summary(14)  ;  %Antenna cables mass                      [kg]

ant_elec_mass = mass_summary(15)  ;    %Antenna electronics mass                 [kg]



% SOLAR ARRAY

solar_mass = mass_summary(16)   ;         %Solar Array Mass                      [kg]

%

%PROGRAMATICS

sc_life;                          % Expected life of spacecraft           [years]
sc_rings;              % number of orbital rings                          [number]
sc_num ;               % total number of spacecraft                       [number]
sc_num_per_ring = sc_num/sc_rings;% number of spacecraft per ring         [number]

%------------------------------------------------------------------------------




%-----------------------FUNCTION CONSTANTS------------------------------------

lc_percent = 90  ;          %Learning curve percentage                     [%]

lc_average = 0.69   ;       %Learning curve cumulative average         [number]

a =1.284      ;             %First cost model constant                 [number]

b = 2.2046226  ;            %Second cost model constant                [number]

total_mass = sum(mass_summary(1:16)); % total mass                        [kg]

totalcont_mass = total_mass*1.25; % total mass with contingency           [kg]
delta_throw_weight =  5089;      % the delta throw weight LL uses         [kg]

%-------------------------------------------------------------------------------






%-----------------------FUNCTION OUTPUTS----------------------------------------

% life_cycle_cost             %total 10 year cost                        [$FY02]

%-------------------------------------------------------------------------------


%--------------------------------
% Some mass data was not available, this avoids calcuation for those designs
if sum(mass_summary) == 0
    life_cycle_cost = NaN;
    return
end
%----------------------------




%COST MODEL
%Costs are broken into three categoies:
%            xxx_non is the nonrecurring costs
```

```
%            xxx_one is the cost for the first s/c

%            xxx_rec is the recurring costs
%
% Relationships are taken from the USAF Unmanned Cost Model,
% 7th ed., using minimum percentage error.
%
% LL's assumption is that Delta launch vehicles will be used exclusively,
% and that they will take as many as possible into the same planar orbit together.
% Accordingly, launch costs are based on how many s/c will fit (rubber spacecraft
assumption)
% on a launch vehicle, and how many are going to the same orbital planes.



acs_non = a*608.289*((acs_mass*b)^0.665);

acs_one = a*165.083*((acs_mass*b)^0.757);


propsys_non = a*125.998*(((propsys_mass+prop_mass)*b)^0.733);

propsys_one = a*232.362*(((propsys_mass+prop_mass)*b)^0.575);


power_non = a*18.444*(power_mass*b);

power_one = a*16.195*((power_mass*b)^0.847);


tcs_non = a*210.753*((tcs_mass*b)^0.677);

tcs_one = a*63.166*((tcs_mass*b)^0.53);


comm_non = a*168.575*(comm_mass*b);

comm_one = a*63.904*(comm_mass*b);


structure_non = a*99.045*((structure_mass*b)^0.789);

structure_one = a*5.838*(structure_mass*b);


elec_non = a*345.781*(elec_mass*b);

elec_one = a*(-1408.508+(149.477*elec_mass*b));

ant_structure_non = a*99.045*((b*ant_structure_mass)^0.789);

ant_structure_one = a*5.838*(ant_structure_mass*b);


ant_tr_non = a*345.781*(b*ant_tr_mass);

ant_tr_one = a*(-1408.508+(149.477*ant_tr_mass*b));


ant_panel_non = a*99.045*((b*(ant_panel_mass))^0.789);
```

```
ant_panel_one = a*5.838*(b*ant_panel_mass);



ant_elec_non = a*345.781*(b*ant_elec_mass);

ant_elec_one = a*(-1408.508+(149.477*b*ant_elec_mass));



solar_non = a*34.126*(b*solar_mass);

solar_one = a*62.778*((b*solar_mass)^0.766);



total_sc_non =
solar_non+ant_elec_non+ant_panel_non+ant_tr_non+ant_structure_non+elec_non+...

    structure_non+comm_non+tcs_non+power_non+propsys_non+acs_non;

total_sc_one =
solar_one+ant_elec_one+ant_panel_one+ant_tr_one+ant_structure_one+elec_one+...

    structure_one+comm_one+tcs_one+power_one+propsys_one+acs_one;

total_sc_rec = total_sc_one*sc_num*lc_average;



total_it_non = (a*956.384)+(0.191*total_sc_non);

total_it_one = a*4.833*(totalcont_mass*b);

total_it_rec = total_it_one*sc_num*lc_average;



total_space_vehicle_non = total_sc_non + total_it_non;

total_space_vehicle_one = total_sc_one + total_it_one;

total_space_vehicle_rec = total_sc_rec + total_it_rec;



program_level_non = a*2.34*((total_space_vehicle_non/a)^0.808);

program_level_one = 0.289*total_space_vehicle_one;

program_level_rec = program_level_one*sc_num*lc_average;



ground_non = a*8.304*((total_space_vehicle_non/a)^0.638);

ops_rec = 1.284*2.212*(totalcont_mass*2.2046226)*sc_num*lc_average;


%----------------------------------------------------------------------------------------
----
% Here there is a further refinement: The LL study has decided that only Delta
% launch vehicles will be considered. The launch costs then are dependent both on
% the number of planes involved (assuming the one booster will only launch into one
% plane), as well as the weights of the spacecraft--if three will not fit on a Delta,
then
% there will have to be another delta.
%
sc_per_lv = floor(delta_throw_weight/totalcont_mass);% calculates how many s/c fit on one
delta
if sc_per_lv == 0
```

```
    %disp ('Too heavy to be launched with a delta');
    launch_rec = NaN;
else
lv_per_ring = ceil(sc_num_per_ring/sc_per_lv);       % calculates how many deltas for
each ring
total_lv = lv_per_ring*sc_rings;                     % calculates the total number of
launch vehicles needed
launch_rec = 104000*total_lv;                        % calculates total cost of all
launches

end
% ---------------------------------------------------------------------------------------
----------


life_cycle_cost = 1000*(total_space_vehicle_non+...
    program_level_non+...
    ground_non+...

    total_space_vehicle_rec+...
    program_level_rec+...
    ops_rec+...
    launch_rec);
% The 1000 converts into $FY02
```

RPAT and RPAT files module

This module is run off-line, and calls the RPAT software developed by Robert Coury at Lincoln Lab. RPAT is a group of Matlab modules that gives radar performance for both space and air based GMTI and SAR radar systems. The modules below fed the RPAT program the appropriate specifications to run analysis on each unique satellite design. There were three calls to RPAT for each design—once to get the relevant GMTI performance data, once to get the SAR data while operating at a high resolution, and once to get the SAR data while operating at a lower resolution. The details of RPAT are omitted here.

Assumptions:

In order to simplify calculation, performance characteristics were evaluated for each satellite imagining that it was performing either the SAR or the GMTI function exclusively. Therefore, performance numbers should be viewed as total potential instead of actual performance.

```
%.............................................................
% SPACE BASED RADAR MATE MODEL................................
% Tim Spaulding...............................................
%.............................................................
%.............................................................
% RPAT shell..................................................
% Origin: MIT LAI 19 December 2002............................
% Final Revision: MIT LAI 5 March 2003........................
%.............................................................


close all
clear all

% This is a list of all the filenames used as inputs to RPAT
filenames = strvcat('800_40_SARH','800_70_SARH','800_100_SARH',...
             '1000_40_SARH','1000_70_SARH','1000_100_SARH',...
             '1200_40_SARH','1200_70_SARH','1200_100_SARH',...
             '1400_40_SARH','1400_70_SARH','1400_100_SARH',...
              '800_40_SARL','800_70_SARL','800_100_SARL',...
             '1000_40_SARL','1000_70_SARL','1000_100_SARL',...
             '1200_40_SARL','1200_70_SARL','1200_100_SARL',...
             '1400_40_SARL','1400_70_SARL','1400_100_SARL',...
              '800_40_GMTI', '800_70_GMTI', '800_100_GMTI',...
             '1000_40_GMTI','1000_70_GMTI','1000_100_GMTI',...
             '1200_40_GMTI','1200_70_GMTI','1200_100_GMTI',...
             '1400_40_GMTI','1400_70_GMTI','1400_100_GMTI');


% This are the suffixs for different scanning cases
trackfilenames = strvcat('grid5','grid20','grid30','grid45');


% runs all the no scanning cases first
for loop = 1:36
     current = filenames(loop,:);
     saved = filenames(loop,:);
     RPAT (current, saved); % this is the call to the RPAT module
end


%runs all the scanning cases
for outer = 1:4
for loop = 1:36
       current = filenames(loop,:);
       saved = [filenames(loop,:) trackfilenames(outer,:)];
       RPAT (saved, saved);
end
end


radardata = struct([]);  % these commands initialize the arch structure
radardata = 1;
radardata.specs = 1;
radardata.A_im = 1;
radardata.Arate = 1;
radardata.T = 1;
radardata.mdv = 1;
radardata.rho_r = 1;
radardata.range_acc = 1;

% puts all non scanning into the MAT file
% ---------------------------------------------------------------------------------------
---
% saves all SAR data (includes high and low resolutions)
for loop  = 1:24
        current = filenames(loop,:);
        saved = filenames(loop,:);
        load (current);
        radardata(loop,1).specs = current; % specifics of design under consideration
        radardata(loop,1).A_im = A_im;     % area imaged            [km^2]
```

```
        radardata(loop,1).T = T;           % time to take an image  [seconds]
end

% saves all GMTI data
for loop  = 25:36
        current = filenames(loop,:);
        saved = filenames(loop,:);
        load (current);
        radardata(loop,1).specs = current; % specifics of design under consideration
        radardata(loop,1).Arate = Arate;   % area arate             [km^2/s]
        radardata(loop,1).mdv = mdv;       % minimum det velocity   [km/s]
        radardata(loop,1).rho_r = rho_r;   % range accuracy         [m]
        radardata(loop,1).range_acc = range_acc; % xrange accuracy  [m]
    end
%-----------------------------------------------------------------------------------------
------



% puts all the scanning cases into the MAT file
% ---------------------------------------------------------------------------------------
-------
% saves all SAR data (includes high and low resolutions)
for outer = 2:5
    for loop  = 1:24
        current = [filenames(loop,:) trackfilenames(outer-1,:)];
        load (current);
        radardata(loop,outer).specs = current; % specifics of design under consideration
        radardata(loop,outer).A_im = A_im;     % area imaged        [km^2]
        radardata(loop,outer).T = T;           % time to take image [seconds]
end

% saves all GMTI data
for loop  = 25:36
    current = [filenames(loop,:) trackfilenames(outer-1,:)];
        load (current);
        radardata(loop,outer).specs = current;  %specifics of design under consideration
        radardata(loop,outer).Arate = Arate;    % area rate             [km^2/s]
        radardata(loop,outer).mdv = mdv;        % minimum det velocity  [km/s]
        radardata(loop,outer).rho_r = rho_r;    % range accuracy        [m]
        radardata(loop,outer).range_acc = range_acc;    %xrange acc     [m]
    end
end
%-----------------------------------------------------------------------------------------
--------

save radardata.mat;
```

Coverage and Time Optimization Modules

In order to complete the analysis, an optimization was performed on this data, which proposes a generic set of targets that might be of interest to a system. Coverage statistics were calculated over a full day, given in the form of azimuth and elevation date from a satellite. By matching this coverage against the set of targets and the GMTI performance at various azimuths and elevations, overall GMTI performance could be estimated. The satellite coverage data and optimization module were provided by Dr. Ray Sedwick of MIT’s Space Systems Laboratory.

Assumptions:

In order to perform the optimization, it is imagined that an equal amount of GMTI coverage is desired at each of the 75 target sites. This assumption is valid when considering a generic set of targets, though in actual practice it would certainly be violated. The other assumption regards Ray Sedwick’s coverage profile, which was originally intended only to include the 800 and 1200 km cases, which are repeating ground tracks. A repeating ground track allows one to simplify coverage statistics, extrapolation coverage on a small piece of the earth to the rest of the area covered by the satellites’ maximum inclination. This assumption is violated for the 1200 and 1400 km cases, though for a representative set of targets, the coverage statistics are believed to be reasonably accurate.

```
%.............................................................
% SPACE BASED RADAR MATE MODEL................................
% Tim Spaulding...............................................
%.............................................................
%.............................................................
% Create GMTT data Module.....................................
% Origin: MIT LAI February 2003...............................
% Final Revision: MIT LAI 5 March 2003........................
%.............................................................


clear all
close all

% These are the indices used to calculate the radar performance
% data by RPAT
az_indices = [5 10 15 22.5 45 67.5 90];
el_indices = [6 8 12 25 40 55 70];


%[x,y,z] = size(dutypie);
for scan  = 1:5                         % will run through all scan cases
        for outerouter = 25:36             % will run through all GMTI data
                                        % (which are in rows 25-36 of radardata)
               clear duty Adot

            % chooses which altitude you are dealing with
            % and loads the right coverage file
            % The coverage file contains a duty matrix
            % which gives Az and El from a satellite to
            % a set of targets for any time in a day
```

```
            switch outerouter
              case 25:27
                load cover800.mat
              case 28:30
                load cover1000.mat
              case 31:33
                load cover1200.mat
              otherwise
                load cover1400.mat
            end

               % sub-sample duty array
               decimation = 4;
               DT = (time_vector(2)-time_vector(1))*decimation
               dutypie = duty(1:decimation:k_times,:,:);

               clear duty;
               load radardata.mat;     % radardata contains radar performance stats

        [x,y,z] = size(dutypie);
               for outer = 1:x
            for inner = 1:y
                for loop = 1:z

                    % the data in dutiepie is encoded in one number:
                    % Az: 45 El: 90 looks like 4500090
                    az = floor((dutypie (outer,inner,loop))/1000);
                    el = dutypie(outer,inner,loop)-az;

                    % Finds the closest Az El from the RPAT data
                    diffs = abs(el - el_indices);
                    mindiff = min(diffs);
                    El_index = find(diffs == mindiff);
                    diffs = abs(az - az_indices);
                    mindiff = min(diffs);
                    Az_index = find(diffs == mindiff);

                    % uses this Az El to pick out the right performance numbers
                    dutypie(outer,inner,loop) =
radardata(outerouter,scan).Arate(Az_index,El_index);
                    %call correct value
                end
            end
               end


               load targets.mat -ASCII % pulls in a list of 75 target lat and longs

               minlong = min(targets(:,1));
               minlat = min(targets(:,2));


               for loop = 1: length(targets)
            long = floor(targets(loop,1)-minlong)+1;
            lat = floor(targets(loop,2)-minlat)+1;

            for inner = 1:x
            Adot(loop,inner) = dutypie(inner,long,lat);
            end

               end

               clear dutypie radardata targets time_vector loop long lat minlong minlat;

               [xx,X] = opt_time(Adot,DT);     % Uses linear optimization to spend time
at each target and guarantee that
                                        % each target gets the same coverage
               total = xx(end);                % this is that equal area [km^2]

               GMTIresults(outerouter-24,scan) = total % saves the results
```

```
               clear X;
               clear xx;

       end  %outerouter
end %scan

save GMTIresults_new GMTIresults;       % saves this results for use in the radar module
```

Radar Module

This module takes the information produced in the preceding modules (performed off-line) and evaluates how each architecture fulfills the various attributes.

```
function attributes =...
    radar(altitude,tech_level,apeture,scanning_type,scan_case,num_s,num_r,gap)

%.............................................................
% SPACE BASED RADAR MATE MODEL...............................
% Tim Spaulding..............................................
%.............................................................
%.............................................................
% Main Module................................................
% Origin: MIT Lincoln Lab, 23 July 2002......................
% Final Revision: MIT LAI 5 March 2003.......................
% Uses data from RPAT, by R.A. Coury, MIT LL.................
%.............................................................


%----------------------FUNCTION INPUTS---------------------------------------
  altitude;             % orbit altitude      [meters]
  tech_level;           % technology level     [2002/2005/2010]
  apeture;              % antenna size        [square meters]
  scanning_type;        % scanning type       [Electronic or Mechanical]
  scan_case;            % scanning angles     [degrees (if electronic)]
  num_s;                % number of satellites [#]
  num_r;                % number of rings     [#]
  gap;                  % max median gap time  [minutes]
%----------------------------------------------------------------------------

%-----------------------FUNCTION OUTPUTS----------------------------------------
% attributes(1) = min_speed            %attribute value     [miles per hour]
% attributes(2) = tracking_area        %attribute value     [boxes]
% attributes(3) = sar_area             %attribute value     [square miles]
% attributes(4) = sar_resolution       %attribute value     [meters]
% attributes(5) = geo_accuracy         %attribute value     [meters]
% attributes(6) = gap_time             %attribute value     [minutes]
% attributes(7) = cog_area             %attribute value     [boxes]
%--------------------------------------------------------------------------------


load radardata.mat;
load GMTIresults_new.mat;
% These are the summary results from the RPAT performance calculator
% These are calcuated offline



% The data are stored in matrices, according the the altitude, aperture, and scan case
% of the system under consideration.  The column, plus, and pointer sections pick out the
% correct data from each matrix
```

```
if altitude == 800
    pointer = 1;
    elseif altitude == 1000
        pointer = 4;
        elseif altitude == 1200
            pointer = 7;
            elseif altitude == 1400
                pointer = 10;
    end


if apeture == 40
    plus = 1;
elseif apeture == 70
        plus = 2;
    elseif apeture == 100
            plus = 3;
    end

scan_case = char(scan_case);
switch scan_case
case '5x5'
    column = 2;
case '20x5'
    column = 3;
case '45x15'
    column = 4;
case '30x15'
    column = 5;
otherwise
    disp ('ERROR IN RADAR SCAN CASE')
end


attributes(1) = max(max(radardata(pointer+24,column).mdv))*(3600*.000621);
% minimum detectable velocity (MPH)
%Assumes the worst case

[x,y,new] = find(radardata(pointer,column).rho_r >= 0);
% this weeds out the -1s which represent missing data
attributes(4) = mean(new);
% SAR resolution (m)


attributes(5) = 100;
%cross range accuracy (GMTI) (m)
%The system universally maximizes this attribute

attributes(6) = gap;
% gap time (minutes)

[x,y,new] = find(radardata(pointer,column).A_im >= 0);
% this weeds out the -1s which represent missing data
attributes(3) = sum(new)*num_s*(1/4.4);
% SAR Area (square miles)

GMTIarea = GMTIresults(pointer+plus-1,column);
% total area tracked
attributes(2) = round((GMTIarea * 0.386102)/100)/75;
% GMTI Area (boxes)

[x,y,new] = find(radardata(pointer+12,column).A_im >= 0);
% this weeds out the -1s which represent missing data
attributes(7) = (((sum(new)*num_s*(1/4.4)))/100)*2;
% COG area (number of 100 square mile boxes)
```

Utility Module

This module takes the attribute values and converts them to their multi-attribute utility scores. The utility curves used for these calculations can be changed, according to which set of curves is under study.

Assumptions:

A linear interpretation is used between points.

```
function [u] = utility(big_k,k_vector,attributes);

%.............................................................
% SPACE BASED RADAR MATE MODEL................................
% Tim Spaulding...............................................
%.............................................................
%.............................................................
% Utility Module..............................................
% Origin: 16.89 Class  Date??.................................
% Final Revision: MIT LAI 5 March 2003........................
%.............................................................


%-------------------------FUNCTION INPUTS -----------------------------------------
  big_k;                            %multiplicative constant          unitless
  k_vector;                         %single attribute constants       unitless
  min_speed = attributes(1);        %minimum detectable speed         MPH
  tracking = attributes(2);         %tracking area                    boxes
  sar_area = attributes(3);         %imaging area                     sq miles
  sar_resolution =attributes(4);    %imaging resolution               meters
  geo= attributes(5);               %geolocation accuracy             meters
  gap= attributes(6);               %average gap time                 minutes
  cog= attributes(7);               %center of gravity area           boxes
  load udata.mat                    %data from MIST interviews        various
%----------------------------------------------------------------------------------

% %----------------------FUNCTION OUTPUTS---------------------------------------
%  u_min_speed;                %utilty of minimum detectable speed         utils
%  u_tracking;                 %utilty of tracking area                    utils
%  u_sar_area;                 %utilty of imaging area                     utils
%  u_sar_resolution;           %utilty of imaging resolution               utils
%  u_geo;                      %utilty of geolocation accuracy             utils
%  u_gap;                      %utilty of average gap time                 utils
%  u_cog;                      %utilty of center of gravity area           utils
%  U;                          % multiattribute utility                    utils
% %------------------------------------------------------------------------------

%---------------------------------------------------------------------------------
% LOAD UTILITY DATA
%--------------------------------------------------------------------------------
% chooses which data you want to use
% Options:
                 data = linear_data       %linear utility curves
                % data = MIST_1_data       %First MIST interview curves
                % data = MIST_2_data       %Second MIST interview curves
                % data = hand_1_data       %First hand interview curves
                % data = hand_2_data       %Second hand interview curves

% puts the data into its respective columns
```

```
x_min_speed = data(:,1);        y_min_speed = data(:,2);
x_tracking = data(:,3);         y_tracking = data(:,4);
x_sar_area = data(:,5);         y_sar_area = data(:,6);
x_sar_resolution = data(:,7);   y_sar_resolution = data(:,8);
x_geo= data(:,9);               y_geo = data(:,10);
x_gap = data(:,11);             y_gap = data(:,12);
x_cog = data(1:5,13);           y_cog = data(1:5,14);
%------------------------------------------------------------------------------------




%------------------------------------------------------------------------------------

% INTERPOLATE TO FIND SINGLE ATTRIBUUTE UTILITY VALUES
%-------------------------------------------------------------------------------------
if min_speed <= 50
    u_min_speed = interp1(x_min_speed,y_min_speed,min_speed,'linear',1);
else
    u_min_speed = NaN;
end

if 10 <= tracking
    u_tracking = interp1(x_tracking,y_tracking,tracking,'linear',1);
else
    u_tracking = NaN;
end

if 0.5 <= sar_area
    u_sar_area = interp1(x_sar_area,y_sar_area,sar_area,'linear',1);
else
    u_sar_area = NaN;
end

if 0.5 <= sar_resolution
    u_sar_resolution = 
interp1(x_sar_resolution,y_sar_resolution,sar_resolution,'linear',1);
else
    u_sar_resolution = NaN;
end

if 50 <= geo
    u_geo = interp1(x_geo,y_geo,geo,'linear',1);
else
    u_geo = NaN;
end

if 5 <= gap
u_gap = interp1(x_gap,y_gap,gap,'linear',1);
else
    u_gap = NaN;
end

if 1 <= cog
    u_cog = interp1(x_cog,y_cog,cog,'linear',1);
else
    u_cog = NaN;
end

%---------------------------------------------------------------------------------




%-----------------------------------------------------------------------------------
% CALCULATE MULTI ATTRIBUTE UTILITY VALUE
%-----------------------------------------------------------------------------------
```

```
compound_product = 1;

        compound_product = compound_product*(big_k*k_vector(1)*u_min_speed+1);

    compound_product = compound_product*(big_k*k_vector(2)*u_tracking+1);

    compound_product = compound_product*(big_k*k_vector(3)*u_sar_area+1);
    compound_product = compound_product*(big_k*k_vector(4)*u_sar_resolution+1);
    compound_product = compound_product*(big_k*k_vector(5)*u_geo+1);
    compound_product = compound_product*(big_k*k_vector(6)*u_gap+1);
    compound_product = compound_product*(big_k*k_vector(7)*u_cog+1);



U = (compound_product - 1)/big_k;

%--------------------------------------------------------------------------------

u = [U;u_min_speed;u_tracking;u_sar_area;u_sar_resolution;u_geo;u_gap;u_cog];
```

Bibliography

Alderidge, E.C., Jr. Memorandum: “Evolutionary Acquisition and Spiral Development.” Office of the Under Secretary of Defense. Department of Defense Acquisition Technology and Logistics. April 12, 2002.

Anandalingham G. and C.E. Olsson, (1989) “A Multi-Stage Multi-Attribute Decision Model for Project Selection.” European Journal of Operational Research, Dec 18, 1989.

Arrow, Kenneth J. (1970), Social Choice and Individual Values, Yale University Press.

Bedford, Tim and Roger Cooke, (1999) “A New Generic Model for Applying MAUT,” European Journal of Operational Research, 118:589-604.

Birker, John, Giles Smith, Glenn A. Kost, and Robert V. Johnson, (2000) An Acquisition Strategy, Process, and Organization for Innovative Systems, RAND, Santa Monica, CA.

Boehm, Barry (2000) “Spiral Development: Experience, Principles, and Refinements.” 9 Presentation to the Spiral Experience Workshop February, 2000.

Cantafio, Leopold, (1989) Space-Based Radar Handbook, Artech House Radar Library.

Chairman of the Joint Chiefs of Staff (CJCS), (2001), “Requirements Generation System,” Instruction, 15 April, 2001.

Chapa, Joe, (2003) Personal e-mail to the author, 7 May.

Davis, Christine C., Richard F. Deckro, and Jack A. Jackson, (2000), “A Value Focused Model for a C4 Network,” Journal of Multi-Criteria Decision Analysis, 9:138-162.

Derleth, J. E. (2003). Multi-Attribute Tradespace Exploration and its Application to Evolutionary Acquisition. Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics SM, Massachusetts Institute of Technology.

Diller, Nathan P. (2002) Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements. Aeronautics and Astronautics SM, Massachusetts Institute of Technology.

Edwards, Ward and Barron F. Hutton, (1994), "SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement," Organizational Behavior and Human Decision Processes 60: 306-325.

Farmer, T.A., (1987) "Testing the Robustness of Multi-attribute Utility Theory in an Applied Setting," Decision Sciences, 18:178-183.

Fischer, G.W. (1977) "Convergent Validation of Decomposed Multi-attribute Utility Assessment Procedures for Risky and Riskless Decision," Organizational Behavior and Human Performance, 18:295-315.

Fry, Phillip C., Dan D. Rinks and Jeffrey L. Ringuest, (1996) "Comparing the predictive validity of alternatively assessed multi-attribute preference models when relevant decision attributes are missing." European Journal of Operational Research 94: 599-609.

Ha, Vu and Peter Haddawy. "Toward Case-Based Preference Elicitation: Similarity Measures on Preference Structures." Report, University of Wisconsin-Milwaukee.

Hacker, Troy L., Raymond J. Sedwick, and David W. Miller, (2000) "Performance Analysis of a Space-Based Radar System Using Separated Spacecraft Interferometry." Thesis, Massachusetts Institute of Technology, http://ssl.mit.edu/publications/theses/SM-2000-HackerTroy.pdf

Hershey, J.C., H.C. Kunreuther, and P.J.H. Schoemaker, (1982) "Sources of Bias in Assessment Procedures for Utility Functions," Management Science, 28:926-954.

Highsmith, James. A. (2000) Adaptive Software Development: a Collaborative Approach to Managine Complex Systems. Dorset House Publishing, New York.

Keeney, Ralph L. and Howard Raiffa (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge, U.K.

Keithly, Hans. (2001) “Space Based Radar Utility Analysis,” Joint C4ISR Decision Support Center, Jan 31, 2001.

Kreps, David M, (1988), Notes of the Theory of Choice, Westview Press.

Laskey, Kathryn B. and Gregory W. Fischer, (1987) “Estimating Utility Functions in the Presence of Response Error.” Management Science 33: 985-980.

Little, Terry (2002) “Agile Acquisition and the Acquisition Center of Excellence” Presentation.

Long, Stephen and Nirav Shah, eds. (2001) “X-TOs: 16.89 Final Design Report.” Massachusetts Institute of Technology, unpublished.

Olson, David L. (1994) “Comparison of Three Multicriteria Methods to Predict Known Outcomes,” European Journal of Operational Research, 130: 576-587.

Paone, Chuck, (2002) “Acquisition Chief Discusses Transformation.” Air Force News, 5 Dec, 2002. http://www.af.mil/news/Dec2002/120502123.shtml

Roberts, C. J. (2003). Architecting Strategies Using Spiral Development for Space Based Radar. Cambridge, MA: Massachusetts Institute of Technology. Technology and Policy Program SM, Massachusetts Institute of Technology.

Ross, Adam and Nathan Diller (2002), “MIT SSPARC MATE Book,” Unpublished.

Ross, Adam, N. Diller, D. Hastings and J. Warmkessel, (2002) “Multi-Attribute Tradespace Exploration in Space System Design.” 53rd International Astronautical Congress, Houston, Texas, 10-19 Oct 2002.

Ross, A. M. (2003). Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-centric Framework for Space System Architecture and Design. Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics; Technology & Policy Program Dual-SM, Massachusetts Institute of Technology.

Scott, Michael J., and Erik K. Antonsson, (2000) “Arrow’s Theorem and Engineering Decision Making,” Research in Engineering Design, 11:218-228.

Shah, N. B. (2003). A Portfolio Based Approach to Evolutionary Acquisition. Cambridge, MA: Massachusetts Institute of Technology. Aeronautics and Astronautics SM, Massachusetts Institute of Technology.

Shaw, G.M. (1998) “The Generalized Information Network Analysis Methodology for Distributed Satellite Systems,” PhD Thesis, Massachusetts Institute of Technology.

Shaw, G. M., DW; Hastings, DE (2001), "Development of the quantitative generalized information network analysis methodology for satellite systems." Journal of Spacecraft and Rockets **38**(2): 257-269.

Skolnik, Merrill I.(1980) Introduction to Radar Systems, McGraw Hill, New York.

von Neumann, J. and O. Morgenstern, (1947). *Theory of Games and Economic Behavior*. 2nd ed. Princeton University Press, Princeton, N.J.

Wolfowitz, Paul, Memorandum: “Defense Acquisition.” Office of the Deputy Secretary of Defense, 30 Oct 2002.

Wirthlin, Rob, (2000) “Best Practices in User Needs/Requirements Generation,” Thesis, Massachusetts Institute of Technology.