NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

A SET-BASED APPROACH TO SYSTEMS DESIGN

by

Jamie M. Gumina

March 2019

Dissertation Supervisor: Clifford A. Whitcomb

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2019	**3. REPORT TYPE AND DATES COVERED** Dissertation

4. TITLE AND SUBTITLE A SET-BASED APPROACH TO SYSTEMS DESIGN	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jamie M. Gumina	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

A set-based design (SBD) approach is proposed as an alternative to traditional point-based design (PBD) methodologies. SBD is compared to other common engineering, decision-making, and optimization methods to illustrate how conventional methods do not ordinarily embrace set-based thinking (SBT) or SBD methodologies. The predominant features of Toyota's approach are summarized, leading to seven characteristics and two principles required to identify a design approach as set-based. Several Latin hypercube (LHC) sampling strategies and the distinguishing characteristics of each are described for use in creating and refining sets. Methods of set reduction and elimination are introduced, and topics related to engineering reasoning in set reduction, expectations, SBT, when to use SBD, benefits, challenges, and metrics are discussed. Improved SBD process steps are proposed and demonstrated in an unmanned air system (UAS) example. A specific type of LHC is chosen to generate points in the design space, which are then used as inputs into a simulation tool. Approaching the UAS example problem in a set-based way results in more viable options with higher system-level performance for comparable cost than if a PBD approach were used.

14. SUBJECT TERMS set-based design, SBD, systems engineering, SE, Latin hypercube sampling, LHC, LHS, DoD acquisition	**15. NUMBER OF PAGES** 265
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 | Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

A SET-BASED APPROACH TO SYSTEMS DESIGN

Jamie M. Gumina
Civilian, Department of the Navy
BS, New Mexico Institute of Mining & Technology, 2004
MS, New Mexico Institute of Mining & Technology, 2009

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2019

Approved by: Clifford A. Whitcomb
Department of
Systems Engineering
Dissertation Supervisor

Paul T. Beery
Department of
Systems Engineering

Rudolph P. Darken
Department of
Computer Science

Alejandro S. Hernandez
Department of
Systems Engineering

Eugene P. Paulo
Department of
Systems Engineering

Approved by: Ronald E. Giachetti
Chair, Department of Systems Engineering

Orrin D. Moses
Vice Provost of Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A set-based design (SBD) approach is proposed as an alternative to traditional point-based design (PBD) methodologies. SBD is compared to other common engineering, decision-making, and optimization methods to illustrate how conventional methods do not ordinarily embrace set-based thinking (SBT) or SBD methodologies. The predominant features of Toyota's approach are summarized, leading to seven characteristics and two principles required to identify a design approach as set-based. Several Latin hypercube (LHC) sampling strategies and the distinguishing characteristics of each are described for use in creating and refining sets. Methods of set reduction and elimination are introduced, and topics related to engineering reasoning in set reduction, expectations, SBT, when to use SBD, benefits, challenges, and metrics are discussed. Improved SBD process steps are proposed and demonstrated in an unmanned air system (UAS) example. A specific type of LHC is chosen to generate points in the design space, which are then used as inputs into a simulation tool. Approaching the UAS example problem in a set-based way results in more viable options with higher system-level performance for comparable cost than if a PBD approach were used.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AAO	all-at-once
ABL	allocated baseline
ABM	agent based models
ACV	amphibious combat vehicle
ADC	analog-to-digital conversion
AHP	analytical hierarchy process
AoA	analysis of alternatives
ASR	alternative system review
ASSET	advanced ship and submarine evaluation tool
BBH	Box-Behnken design
bLHD	batch sequential Latin hypercube design
bMmLHD	batch sequential maximin distance Latin hypercube design
BOSLHS	binning optimal symmetric Latin hypercube sampling
BPLHD	balanced probability-based Latin hypercube design
CAD	computer-aided design
CCD	charge-coupled device
CCD	central composite design
CDD	capability development document
CDR	critical design review
CDRL	contract data requirements list
CE	concurrent engineering
CFD	computational fluid dynamics
CMOS	complementary metal-oxide semiconductor
CONEMPS	concepts of employment
CREATE	computational research and engineering acquisition tools environment
DBT	design-build-test
DFA	design for assembly
DFE	design for environment

DFLC	design for life cycle
DFM	design for manufacture
DfX	design for “x”
DoD	Department of Defense
DOE	design of experiments
DoN	Department of Navy
EBM	equation based models
EO	electro-optical
ESE	enhanced stochastic evolutionary
FACT	feasibility assessment for cost and technology
FDD	functional design document
FEA	finite element analysis
FOV	field of view
FRD	functional requirements document
HOQ	house of quality
HPC	high performance computing
IDE	integrated development environment
IEC	International Electrochemical Commission
IEEE	Institute of Electrical and Electronics Engineers
IHEODTD	Indian Head Explosive Ordnance Disposal Technology Division
IMSE	integrated mean square error
INCOSE	International Council on Systems Engineering
IR	infrared
ISO	International Standards Organization
JIT	just-in-time
K	knowledge
KAR	knowledge and action record
LDUUV	large displacement unmanned underwater vehicle
LEAPS	leading edge architecture for prototyping systems
LHC	Latin hypercube

LHS	Latin hypercube sampling
LIB	less-is-better
LRIP	low-rate initial production
MADM	multiple attribute decision making
MBSE	model-based systems engineering
MCC	method of controlled convergence
MCDM	multiple criteria decision making
MCM	mine clearing and mine countermeasures
MDAO	multidisciplinary design, analysis, and optimization
MDD	model-driven development
MDD	materiel development decision
MDO	multidisciplinary optimization
MEASA	methodology for employing architecture in systems analysis
MIB	more-is-better
MODM	multiple objective decision making
MOE	measure of effectiveness
MOP	measure or performance
MPG	miles per gallon
NISE	Naval Innovative Science and Engineering
NLHD	nested Latin hypercube design
NOA	nearly orthogonal array
NOB	nearly orthogonal, nearly balanced
NOLH	nearly orthogonal Latin hypercube
NSWC	Naval Surface Warfare Center
OA	orthogonal array
OALHD	orthogonal array-based Latin hypercube design
OLHC	optimal Latin hypercube
OLHD	orthogonal Latin hypercube design
OMLHD	orthogonal-maximin Latin hypercube design
OMOE	overall measure of effectiveness

OSM	orchestrated simulation through modeling
PBD	point-based design
PBL	product baseline
PDR	preliminary design review
PLHD	probability-based Latin hypercube design
QFD	quality function deployment
RFP	request for proposals
RLH	random Latin hypercube
RSDE	rapid ship design environment
RSM	response surface methods
S	specifications
SBCE	set-based concurrent engineering
SBD	set-based design
SBPD	set-based product development
SBT	set-based thinking
SDLC	system development life cycle
SDS	system design specification
SEP	systems engineering plan
SET	systems engineering transformation
SETR	systems engineering technical review
sFFLHD	sliced full-factorial Latin hypercube design
SFR	system functional review
SLHD	symmetric Latin hypercube design
SLHD	sliced Latin hypercube design
SMART	Science, Mathematics, and Research for Transformation
SME	subject matter expert
SOA	strong orthogonal array-based Latin hypercube design
SoS	system of systems
SRR	system requirements review
SSC	ship to shore connector

SysML	systems modeling language
TOPSIS	technique for order of preference by similarity to ideal solution
TPS	Toyota Production System
TQC	total quality control
TSE	traditional systems engineering
U.S.	United States
UAS	unmanned air system
UML	unified modeling language
USMC	United States Marine Corps
V	variable
VOC	voice of the customer

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

A structured design approach that fits within the category of set-based design (SBD) is introduced as an alternative to traditional point-based design (PBD) methodologies and is described, compared, and demonstrated in a systems context. SBD constitutes a new way of thinking, interacting, and communicating and has advantages over PBD in terms of improved design quality, reduced development risk, and incorporation of design changes (Bernstein 1998) even though it lacks definition and consensus within a systems context and by the engineering community (Ghosh and Seering 2014).

A. COMPARISON OF SBD AND PBD

"The simplest description of SBD is design by process of elimination," i.e., it involves eliminating what are *not* solutions, including those that are infeasible or highly dominated (McKenney and Singer 2014; Singer et al. 2017, 1). SBD is fundamentally different than PBD in that designs converge with SBD rather than evolve with PBD (Liker et al. 1996) (see Figure ES-1).

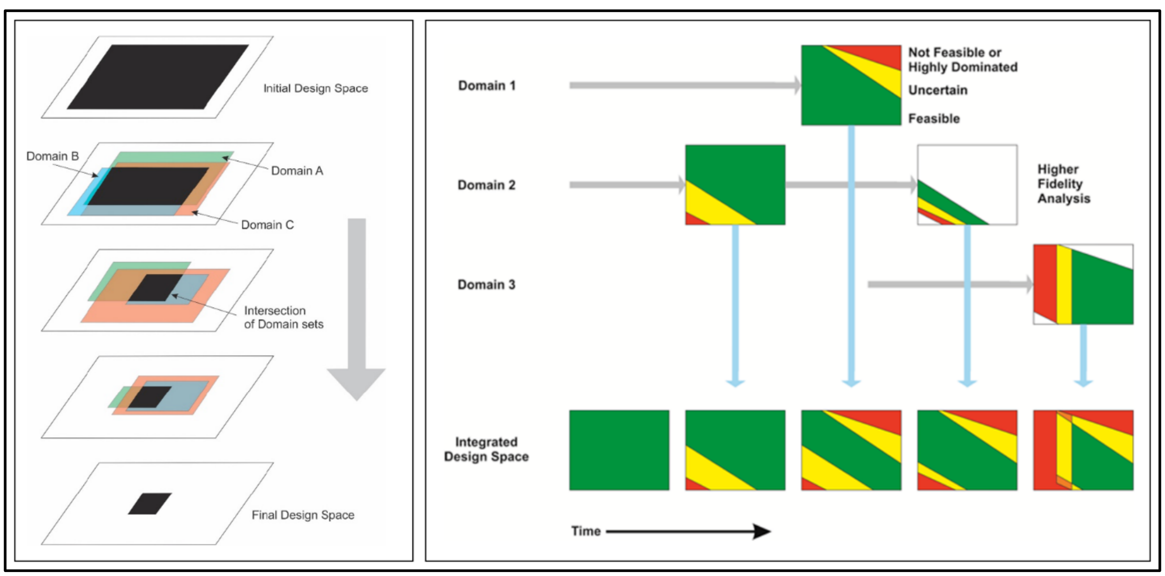


Figure ES-1. Representation of SBD within Design Space Regions.
Source: Singer (2017).

PBD solidifies requirements as early as possible, considers a number of alternatives, and selects a single, most promising point in the design space during the initial stages of development with limited understanding of the problem, which often becomes infeasible or no longer optimal and leads to rework.

SBD is a concurrent engineering (CE) approach that accounts for this lack of knowledge early on and defers the specification of requirements and postpones critical decisions until there is more information to better understand them and the tradespace. It enables further exploration of the design space by allowing each engineering domain to consider it from its own perspective and communicate its individual preferences. SBD concentrates on creating sets of design solutions and systematically eliminates them with documented evidence and rationale until converging on a final solution that achieves system-level optimization.

B. CHARACTERISTICS OF SBD

The concepts of SBD practices are derived from observations of Toyota's development process and application of CE, which do not operate on a point-to-point structure and instead follow a set-based paradigm (Liker et al. 1996; Sobek, Ward, and Liker 1999; Ward et al. 1995a; Ward et al. 1995b). Consolidating the prominent SBD principles inherent in Toyota's philosophy and culture with other examples of SBD methods and techniques found within the literature leads to seven common characteristics of set-based product development (SBPD):

1. Emphasis on frequent, low-fidelity prototyping;
2. Tolerance for under-defined system specifications;
3. More efficient communication among subsystems;
4. Emphasis on documenting lessons learned and new knowledge;
5. Support for decentralized leadership structure and distributed, non-collocated teams;
6. Supplier/subsystem exploration of optimality; and

7. Support for flow-up knowledge creation. (Ghosh and Seering 2014, 2)

Using the seven characteristics as a filter for determining what constitutes SBD, Ghosh and Seering (2014, 7) inductively arrive at two principles to describe the most common themes and influences across SBD-related work (referred to as the principles of set-based thinking [SBT]) that serve as a basis for identifying SBD efforts:

1. Considering Sets of Distinct Alternatives Concurrently
2. Delaying Convergent Decision Making

C. OTHER ENGINEERING, DECISION-MAKING, AND OPTIMIZATION METHODS

Although there are several other engineering methods that are familiar, well-accepted, and widely employed (e.g., spiral, vee, agile, systems engineering, etc.), the way they are commonly implemented does not constitute SBD. A limited number of examples are summarized in Chapter II to demonstrate how conventional methods do not ordinarily embrace SBT or SBD methodologies.

D. THE IMPROVED SBD PROCESS

The practices visually observed at Toyota leading to the vernacular of SBD surely follow a process, however, there are no formal processes (Ghosh and Seering 2014) or explicit steps for implementing SBD as it has come to evolve within the Naval ship building community and beyond. The principles encompassing Toyota practices (Sobek, Ward, and Liker 1999) synchronize with the general implementation steps for SBD (McKenney and Singer 2014; Singer et al. 2017; Singer, Doerry, and Buckley 2009; Specking et al. 2017), but are missing intermediate steps. Improved steps that better define the process are to:

1. Identify Engineering Specialties and Available Engineering Modeling Tools
2. Identify Design Factors for Each Specialty
3. Specify Design Factor Values
4. Create the Specialty Set Space

5. Explore the Specialty Set Space

6. Communicate the Specialty Set Space Preferences

7. Create the Integrated Set Space by Intersection

8. Explore the Integrated Set Space

9. Communicate the Specialty Set Space Preferences

10. Reduce the Set Space by Elimination

11. Refine the Reduced Set Space in Greater Detail

12. Explore the Refined Set Space

13. Communicate the Specialty Set Space Preferences

14. Create the Viable Set Space

15. Explore the Viable Set Space

16. Select a Design Solution

A flow chart of the improved SBD process steps introduced in this dissertation is depicted in Figure ES-2. Steps 4, 7, 11, and 14 involve set creation. Step 10 involves set elimination.

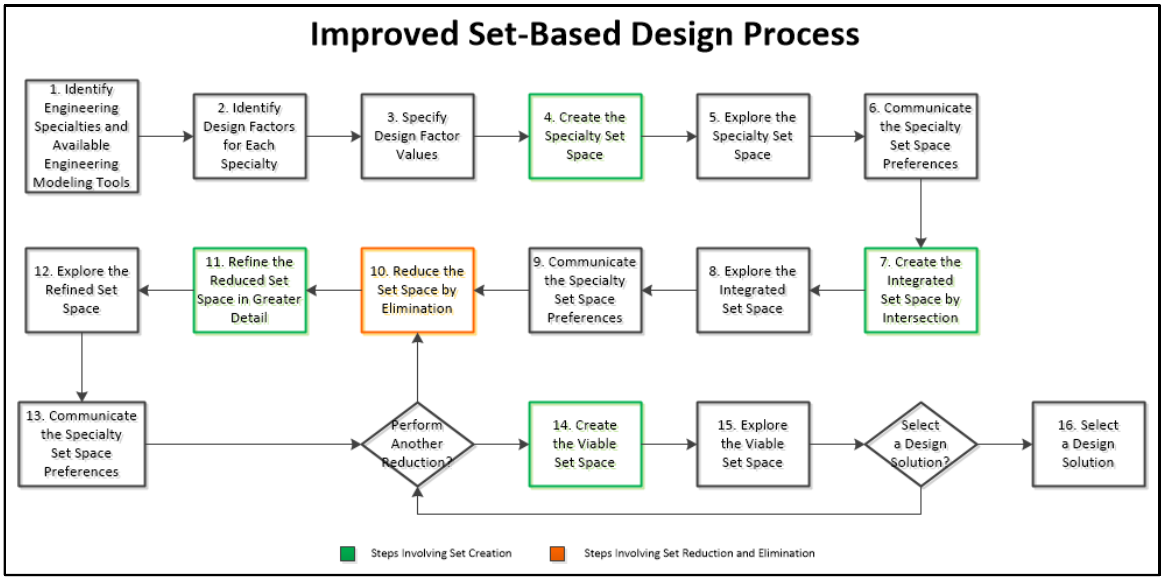


Figure ES-2. Improved SBD Process Flow

E. LATIN HYPERCUBE SAMPLING FOR SET CREATION

A design of experiments (DOE) approach is applied in a set-based manner to generate the points (i.e., design variants, alternatives, or solutions) in the design space for the steps involving set creation (steps 4, 7, 11, and 14). Latin hypercube (LHC) sampling (LHS) strategies are used to sample design solutions from the full ranges of input factors over the entire design space, including the interior and extremes. Different LHCs result in different coverage of the design space, and several LHCs and the distinguishing characteristics of each are summarized in Chapter IV.

F. METHODS OF SET ELIMINATION

Methods of set elimination for application in step 10 of the improved SBD process include: specialty-level investigations (impact of input factors on measures of performance [MOPs] or measures of effectiveness [MOEs]); system-level investigations (impact of input factors on viability or overall measure of effectiveness [OMOE]); distance to the ideal point; and visual inspection.

G. UNMANNED AIR SYSTEM EXAMPLE DEMONSTRATION

The improved SBD implementation steps introduced in this dissertation are demonstrated using an unmanned air system (UAS) example where a defense contractor has a well-established UAS product line using piston engines, but recognizes the emerging market and advancing technologies for the use of electric engines. The contractor establishes two unique design teams (electric and piston) to compare the value of a small UAS required for surveillance missions that can detect enemy activity and is transportable, survivable, and capable of maneuvering to, scanning, and dwelling at an area of interest. The teams are geographically separated and work for different divisions, but have access to the same companywide simulation and analysis tools.

The engineering specialties involved in the UAS example are: Structures where wingspan is the principle input factor; Propulsion (scoped and pre-defined to be electric or piston); Weight and Balance (total UAS weight); Sensors (electro-optical [EO, daytime] and infrared [IR, nighttime] field of view [FOV] and resolution); and Mission Assurance (operating altitude).

A design integration manager intersects all of the ranges of desired values for each specialty and samples them with LHS (Vieira 2012; Vieira et al. 2013) to generate combinations of design factors that represent unique UAS alternatives. Each combination is then inserted into a UAS simulation tool (Small 2018) adapted to fit the purposes of this dissertation to obtain performance, value, and cost.

Each specialty explores this integrated set space from its own perspective and uses its own tools, plots, and expertise to consider acceptable set reductions based on its MOPs and MOEs. The preferences and important findings for each specialty are passed on to the design integration manager for consolidation, and the resulting actions, implications, and justifications are captured in a knowledge and action record (KAR).

The design integration manager reduces the current set space by carrying out the actions on the KAR, and the cumulative reductions are documented in reduction tables. The reduced set space can be refined by generating additional design alternatives within this smaller space, or by evaluating the remaining solutions in greater detail.

Each specialty explores the refined set space from its own perspective and communicates its preferences and important information to the design integration manager who creates a KAR. Iterations of reducing, refining, exploring, and communicating occur until no further set reductions are desired.

The viable set space is created by eliminating design variants that do not meet every MOE and are unable to accomplish the mission. Each specialty explores and considers the viable solutions in a manner that leads to selection (e.g., Pareto [or other optimal] frontier, value versus cost, etc.), and a design solution is chosen to move forward with.

H. RESULTS OF APPLYING THE IMPROVED SBD APPROACH TO UAS DESIGN

Through the application of SBD and based on the current technology and desired stakeholder value, a piston variant is recommended, unless there is a threshold where the two engine types overlap. The company can identify opportune areas of technology to invest in and decide where to develop to improve electric variants and make them a more attractive option in the future. For example, efforts to reduce sensor weight or improve engine performance will mitigate limitations on payload capacity that lead to a disadvantage in sensor packages.

I. CONCLUSION

The UAS example demonstrates a complete implementation of SBD, which is proven through its ability to meet the seven characteristics of SBPD and two principles of SBT required to identify a design approach as set-based. The concept of SBD is better defined and executed in a systems context than is currently found in the literature, and methods for creating and eliminating sets are introduced. An improved definition of SBD steps that are objective and repeatable is applied that facilitates an enhanced understanding of SBD and clarifies its execution. The improved definition and demonstration of how to create and reduce sets in a systems context for the implementation of SBD as an alternative design approach to PBD is encouraging. SBD yields recognizable improvements over PBD in complex engineering problems, and the contributions made here stimulate the momentum towards this new approach and way of thinking, reasoning, and communicating about design problems.

References

Bernstein, Joshua I. 1998. “Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices.” Master’s thesis, Massachusetts Institute of Technology.

Ghosh, Sourobh, and Warren Seering. 2014. “Set-Based Thinking in the Engineering Design Community and Beyond.” DETC2014-35597. In Proceedings of the IDETC/CIE conference, Buffalo, NY.

Liker, Jeffrey K., Durward K. Sobek II, Allen C. Ward, and John J. Cristiano. 1996. “Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering.” *IEEE Transactions on Engineering Management* 43(2): 165–178.

McKenney, Thomas, and David Singer. 2014. “Set-Based Design: A Concurrent Engineering Approach with Particular Application to Complex Marine Products.” *Marine Technology, SNAME* 3, no. 1 (July): 51–55.

Singer, David J., Norbert Doerry, and Michael E. Buckley. 2009. “What is Set-Based Design?” *ASNE Naval Engineers Journal* 121(4): 31–43.

Singer, David, Jason Strickland, Norbert Doerry, Thomas KcKenney, and Cliff Whitcomb. 2017. “Set-Based Design.” T&R Bulletin 7–12. Society of Naval Architects and Marine Engineers (SNAME).

Small, Colin. 2018. “Demonstrating Set-Based Design Techniques—A UAV Case Study.” Master’s thesis, University of Arkansas.

Sobek II, Durward K., Allen C. Ward, and Jeffrey K. Liker. 1999. “Toyota’s Principles of Set-Based Concurrent Engineering.” *Sloan Management Review* 40, no. 2 (Winter): 67–83.

Specking, Eric A., Cliff Whitcomb, Gregory S. Parnell, Simon R. Goerger, Ed Pohl, and Naga Sai Achyuth Kundeti. 2017. “Trade-off Analytics for Set-Based Design.” Paper presented at the American Society of Naval Engineers (ASNE) Intelligent Ships Symposium, Philadelphia, PA.

Vieira, Jr., H. 2012. “NOB_Mixed_512DP_v2.xls design spreadsheet.” Naval Postgraduate School. Accessed September 14, 2018. http://harvest.nps.edu/.

Vieira, Jr., H., S. M. Sanchez, K. H. Kienitz, and M. C. N. Belderrain. 2013. “Efficient, Nearly Orthogonal-and-Balanced, Mixed Designs: An Effective Way to Conduct Trade-off Analyses using Simulation.” *Journal of Simulation* 7: 265–275.

Ward, Allen, Durward K. Sobek II, John J. Cristiano, and Jeffrey K. Liker. 1995a. “Toyota, Concurrent Engineering, and Set-Based Design.” In *Engineered in Japan: Japanese Technology Management Practices*, edited by Jeffrey K. Liker, John E. Ettlie, and John C. Campbell, 192–216. New York: Oxford University Press.

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II. 1995b. “The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster.” *Sloan Management Review* 36(3): 43–61.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I am tremendously thankful for the support I received along my PhD journey. The opportunity to pursue my degree would not have been possible without the scholarship funding from the Science, Mathematics, and Research for Transformation (SMART) program or the Naval Innovative Science and Engineering (NISE) program through the Naval Surface Warfare Center Indian Head Explosive Ordnance Disposal Technology Division (NSWC IHEODTD). All of my advisors are amazing and brilliant! Each one made a positive difference, challenged me to think, encouraged my creativity, kindled my curiosity and wonder, and inspired me to excel beyond familiar limits into a richer realm of learning and understanding. My incredibly devoted, loving, and patient partner and two beautiful children kept me going through imaginative incentives for adventure, quality time, laughter, and fun. My Mom, the strongest, most driven and selfless woman I know, shared this ambition alongside me on her own parallel path and loves me and believes in me unconditionally. My Dad continues to lead me in memory and is ever so proud walking with me in spirit. My siblings gave me reinforced determination through our most unbreakable bonds and steadfast faith, trust, and honor in each other. My extended family has rooted for me since the beginning and modeled the values of family, education, truth, happiness, and health that serve as a compass for my best self. My friends kept me sane and moving by listening, playing, and prompting reenergizing escapes to nature. My Tang Soo Do family has always advised me to find my center and live by a motto of humility, patience, and perseverance with courage, energy, harmony, passion, and indomitable spirit!

I am privileged to have the opportunity to participate in such a prestigious program with such a high caliber of people. I have the utmost appreciation and gratitude for all of those who have contributed to my growth and success and am extraordinarily humbled by the sincere backing I have received from everyone.

It is with great intent and enthusiasm that I thank each and every one of you!

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PREMISE

The increase in complexity of today's systems is an inevitable outcome of advances in communications, technologies, and computing power that have allowed them to be highly interconnected, in addition to the human element associated with balancing stakeholder concerns. Current systems engineering processes demand augmented methodologies to more appropriately handle these complexities as evidenced by the annual loss of billions of dollars on cancelled and failed programs and the myriad of program terminations before full operational capability in the United States (U.S.) Department of Defense (DoD) (Clowney, Dever, and Stuban 2016). The distinguishing characteristics of these complex systems as compared to traditional monolithic systems bring about new challenges for their development, management, use, sustainment, and evolution.

Conventional and iterative design methods, classified as point-based design (PBD) approaches, define the problem, generate and rank several solutions, select the most appealing solution, and analyze, assess, and modify it until a satisfactory solution is achieved. Since the selection of this single point in the design space is made in the initial stages of development with limited understanding of the problem, it often becomes infeasible or no longer optimal and leads to rework.

An alternative approach to design referred to as set-based design (SBD) can be implemented within DoD acquisition to produce superior results and mitigate some of the problems encountered with PBD methodologies. SBD accounts for the lack of knowledge early on and defers the specification of requirements and postpones critical decisions until there is more information to better understand them and the tradespace, which avoids premature commitment and promotes a mindsight of seeking enhanced capabilities and pursuing promising opportunities with informed risk. While enabling the concurrent development of the design, SBD concentrates on creating sets of solutions and systematically eliminates them with documented evidence and rationale until converging on the final solution.

B. BACKGROUND

The underpinnings of SBD as a design approach have been around for almost thirty years, dating back to 1989 when the theoretical foundation was first introduced through a description of the work performed in Allen Ward's PhD dissertation (1989). Ward and Seering (1989) developed a computer program that uses a schematic diagram, specifications, and cost or utility function as inputs to return the optimal design of a mechanical system based on a compiled selection of components identified through predefined catalog numbers. Their work represents the ethos of SBD in that it considers sets of artifacts under sets of operating conditions, explores all viable options when determining the optimal layout, and logically eliminates basic sets only when they can be proven not to work.

The actual term SBD (or set-based concurrent engineering [SBCE]) was originated by Ward et al. in 1995 (1995b) through observations of Toyota's development process and application of concurrent engineering (CE) contributing to the success of the company within the automotive industry. SBD was popularized over a ten-year span through articles and publications describing Toyota's success with SBD practices and its implementation during product development and manufacturing (Liker et al. 1996; Sobek, Ward, and Liker 1999; Ward et al. 1995a; Ward et al. 1995b). Following its prominent application at Toyota, SBD has been demonstrated in diverse areas, including the commercial aerospace industry (Bernstein 1998), military arena and Naval ship design (Burrow et al. 2014; Garner et al. 2015; McKenney, Kemink, and Singer 2011; Mebane et al. 2011; Singer, Doerry, and Buckley 2009), and DoD acquisition (Chan et al. 2016; Genta 2016), and is also of interest to researchers and academia in the engineering design and management science communities (Ghosh and Seering 2014; Specking et al. 2017).

C. PROBLEM STATEMENT

The U.S. Navy was formally exposed to SBD in 2007 as a potential approach to managing the complexities and accommodating the changes encountered in early stages of ship design (Singer et al. 2017). Within this same timeframe, despite the lack of any formal incorporation of SBD into the DoD acquisition life cycle, the U.S. Navy employed its first application of SBD; the Ship to Shore Connector (SSC) utilized SBD successfully during the preliminary and contract design phases resulting in a converged vessel design under an aggressive and demanding schedule (Mebane et al. 2011). The proven success of SBD as a feasible alternative to PBD acquisition reflected in the outcome of the SSC effort led to other instances of SBD application in the DoD, including the: hypothetical autonomous Mine Clearing and Mine Countermeasures (MCM) mission in 2011 (McKenney, Kemink, and Singer 2014); U.S. Marine Corps (USMC) Amphibious Combat Vehicle (ACV) in 2013 (Burrow et al. 2014); Small Surface Combatant in 2014 (Garner et al. 2014); and recent fiscal year 2018 Large Displacement Unmanned Underwater Vehicle (LDUUV) (Chan et al. 2016).

SBD has been applied in each of the instances carried out by the U.S. Navy to date without a consistently defined method or approach, and there is currently no specific guidance for how to execute SBD or leverage its benefits. Even so, U.S. Navy leadership is officially inspired to find alternatives to traditional PBD methods, encouraging compatibility with SBD, and emphasizing the need for evolving models and analytical support tools (Doerry 2012; Eccles 2010; Sullivan 2008). The four DoD examples and unique project challenge indicate the potential of SBD as an alternative acquisition and design method with recognizable improvements over traditional PBD methods, however, it still remains essentially undefined due to lack of explicit and specific descriptions of how to create and eliminate sets and when (or if) it is appropriate to employ SBD. SBD is expected to revamp the way engineers interact and communicate and seems to be advantageous in terms of improved design quality, reduced development risk, and faster response and accommodation of design changes (Bernstein 1998), even though—prior to the contributions of this dissertation—it is:

- Lacking definition and consensus within a systems context and by the engineering community (Ghosh and Seering 2014);
- Assumed to be no different than other optimization efforts and it is what people have already been doing (McKenney and Singer 2014);

- Criticized of being “new and untested territory” (McKenney and Singer 2014, 54);
- Doubted in that maybe a more traditional life cycle process model can produce a candidate design faster and more easily (McKenney and Singer 2014);
- Lacking in understanding of all determining factors for the successful execution of design efforts (Sobek, Ward, and Liker 1999);
- Challenged with a major obstacle of “how to facilitate, manage, and implement [it] when constraints and milestones of current acquisition policies are keyed to [PBD] practices” (Singer, Doerry, and Buckley 2009, 7);
- “Not formally defined, yet numerous authors have studied its process inspired by the example of Toyota” (Ghosh and Seering 2014, 7);
- In need of better definition on how to create and reduce sets (Specking et al. 2017);
- An open problem regarding set reduction (McKenney and Singer 2014);
- In substantial need of “execution support, especially in how decisions should be made to reduce the design space while considering total design impacts” (McKenney and Singer 2014, 55);
- Lacking standard measures (Liker et al. 1996);
- In need of determining what situations are conducive for it versus PBD (Ghosh and Seering 2014);
- Ambiguous in its explanation of steps for execution; and
- Minimal in its guidelines for integration into a formal DoD acquisition process.

D. RESEARCH FOCUS

The primary intent of this research is to generate an approach to SBD through an improved definition of the method applied using process steps, mathematically sound strategies of set creation, and objective and repeatable considerations for set reduction. A comparison of SBD to other common engineering, decision-making, and optimization methods illustrates how their typical implementation does not constitute SBD. Having personally attended some of the SBD summits hosted by the U.S. Navy, there is definitely confusion about what SBD actually is, how is it *not* systems engineering, why cannot another method be used instead, and how is it actually executed? Many presentations of SBD show an enormous number of initial design solutions (hundreds of thousands) that then dwindle down to a manageable hundred or so…but how is this done? There is never an explicit answer because each specialty is offered the freedom to implement its own analysis tools and explore the design space from its own perspective and every design problem is unique.

This dissertation brings clarity to the process in the form of more defined steps. Specifically, it defines specialty-related areas of SBD concerning input factors, measures of performance (MOPs), tradespace exploration, and communication of preferences. It defines system-related areas of SBD regarding integrated set spaces, set-narrowing, and reduced set spaces. Finally, it defines mission-related areas of SBD associated with overall measure of effectiveness (OMOE) analysis, viable set spaces, and design selection. The improved process steps are applied to an unmanned air system (UAS) example to demonstrate their application in a systems context.

There are very few references and definitive methods for set creation in the literature. This dissertation introduces a set-based way of using design of experiments (DOE) as a method of set creation that can be applied to most design problems. Once the acceptable ranges of input factors have been determined, the general approach is to synthesize the values and sample them (i.e., full factorial, orthogonal array [OA], central composite designs [CCD], Box-Behnken designs [BBH], etc.). Several stratified, mathematically proven methods of Latin hypercube (LHC) sampling (LHS) are offered to produce individual *points* of design solutions (vice curve fits), which allow each design alternative to be evaluated regardless of where it lies in the design space.

Another open area of research in SBD is how to eliminate or reduce the sets (McKenney and Singer 2014; Specking et al. 2017). This dissertation identifies specialty-level investigations, system-level investigations, viability considerations, and visual inspection methods that can be applied to narrow down the sets. Each specialty is initially concerned with its own parts at the start of a design problem, but as the individual preferences of each are integrated, they become aware of the system-level implications. Specialty-level investigations consist of looking at the design space in terms of the input factors, design variables, and MOPs and measures of effectiveness (MOEs) a certain specialty is responsible for or able to impact. System-level investigations account for how the input factors affect the OMOE and if the design solutions are capable of achieving the mission. Distance to the ideal point is a quantitative method of reducing the viable set space with the general intent of cutting it in half. Set eliminations based on visual inspection can be employed when infeasibility and dominance are glaringly apparent.

E. CONTRIBUTION

This dissertation makes three major contributions:

1. An improved definition, objective and repeatable process, and formalized steps for SBD in a systems context;
2. A specific method of set creation using LHS; and
3. Several explicit methods of set elimination.

Although several design processes have been executed in a manner that would be classified as SBD in general, there is no consistency between them, and they do not follow repeatable steps, or approach the problem in the same way. The literature has extracted the shared characteristics of these design processes to identify them as SBD, but an objective and repeatable process has not been formalized. This dissertation introduces a sequence of steps for a design approach that is based on sets and is accomplished under the category of SBD as it is distinguished by the seven common characteristics of set-based product development (SBPD) and two principles of set-based thinking (SBT) (Ghosh and Seering 2014).

The improved SBD process presented in this dissertation offers more explicit delineation between the various sets of potential solutions that are available and how each engineering specialty or integration role is involved as the design progresses. The formalized sequence of steps is beneficial in design efforts for a better sense of:

- Who is involved when (some steps involve individual specialties, while others involve all specialties together or a design integration manager);

- What the demand for resources is (some engineering modeling tools are specific to the specialties, while others are more integrated and account for inputs from all specialties);

- Where the designs are within the overall design space and how much of it remains (the part of the design space being considered at any time during the design process is dynamic and is gradually reduced);

- Whether the design alternatives being considered are feasible and viable (a dedicated step for viability ensures the selected solution is able to meet the mission requirements); and

- What the impact and level of confidence is for decisions (the amount and detail of knowledge increases as the design progresses, and knowing the history of the information generated leads to more sound and justifiable decisions).

Evidence of the improved delineation between steps is based on the addition of intermediate steps as compared to the principles encompassing Toyota practices (Sobek, Ward, and Liker 1999) and the general implementation steps for SBD (McKenney and Singer 2014; Singer et al. 2017; Singer, Doerry, and Buckley 2009; Specking et al. 2017).

A better definition of the process is exemplified through the formalized elements of each step, including the terminology (e.g., “design factor” is used to describe what was referred to as input factor, parameter, design attribute, design feature, and element in the literature; and "specialty" is used to describe what was referred to as element, team, domain, and area), the descriptor for each set space (i.e., specialty, integrated, reduced, refined, and viable set spaces), and the simplified nature of the overall process through iterations of create–explore–communicate.

The capacity of the improved SBD process presented in this dissertation as an alternative approach to systems design is demonstrated by applying each of the steps to a UAS example.

A specific method of set creation involving LHS is used in this dissertation that also serves as a major contribution because all possible permutations of input factors have typically been used in the literature, which often generates an unmanageable number of design alternatives. The method of set creation described in this dissertation applies DOE fundamentals in a set-based way to generate sets of design alternatives that are represented as a point cloud with a level of design space coverage that is not possible with traditional curve fit or response surface techniques.

Four distinct methods of performing set elimination are also introduced in this dissertation for a third contribution. SBD is not possible without a feasible way of reducing the sets quickly and efficiently.

F. CHAPTER ORGANIZATION

This dissertation is broken up into seven distinct chapters. Chapter II includes a literature review of SBD methodologies and introduces the circumstances and prior work leading up to the current enthusiasm for it. The progression of design approaches from PBD to CE to SBD emphasizes the importance of a paradigm shift. SBD is compared to other engineering, optimization, and decision-making methods, and expectations, opportune circumstances, benefits, and challenges of SBD are discussed.

Chapter III presents sixteen steps for executing SBD in an objective and repeatable manner that can be applied to most programs. Each step is described in detail to bring better delineation and clarification to the process than has ever been done to date. Four ways to perform set elimination and a quantitative, mathematically sound method of set creation are introduced.

Chapter IV elaborates on the novel method of set creation that uses a DOE approach with LHS techniques for generating design alternatives. Stratification sampling techniques are compared, and a number of different LHC designs are described to include general characteristics, strengths, limitations, and potential uses for each.

Chapter V includes a UAS example demonstration that individually addresses each of the sixteen SBD process steps. A contractor compares two UAS product-lines (electric and piston variants) and gains knowledge through the design process that leads to more informed decisions.

Chapter VI consists of the UAS example results and how they compare to the outputs of a PBD approach. A discussion of how the UAS example constitutes SBD methodologies based on the seven characteristics of SBPD and SBT is also included.

Chapter VII concludes this dissertation by summarizing the salient points and providing suggestions for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PRIOR WORK

The summarized findings for a literature review of the prior work and current research surrounding SBD describes how it arose based on visual observations of Toyota practices and the necessary progression from PBD to CE to set-based methodologies.

Concepts related to the typical execution of SBD are described to exemplify the limited avenues available for the critical elements of set creation and set elimination (which serve as two key areas of contribution in this dissertation).

A deep discussion of how common engineering, optimization, and decision-making methods compare to SBD is included not only for the purpose of showing they do not ordinarily embrace set-based characteristics, but also because personal experience suggests it is important to explain why the common methods frequently encountered in design are *not* considered SBD (in order for people to be receptive of a new method).

The expectations and benefits of implementing SBD are discussed to support that it is a useful alternative to PBD and is worth pursuing under the opportune circumstances also presented in this chapter.

Several challenges of SBD are described to roughly gleam the magnitude of definition, organization, consistency, and cultural considerations that are needed to formalize SBD methodologies within the DoD acquisition and design community (another major area of contribution in this dissertation).

A. SBD AT A GLANCE

"The simplest description of SBD is design by process of elimination," i.e., it involves eliminating what are *not* solutions (McKenney and Singer 2014; Singer et al. 2017, 1). These non-solutions are referred to as infeasible or highly dominated solutions in SBD. Infeasible solutions are those where there is high confidence the solution cannot be achieved or does not exist. Highly dominated solutions are those that are evaluated as inferior to other solutions in every metric of interest. SBD as a design method provides the underlying principles for how to understand and define, analyze, and select the various design alternatives. In general, the process for implementing SBD entails the following steps (McKenney and Singer 2014; Singer et al. 2017; Singer, Doerry, and Buckley 2009; Specking et al. 2017):

1. Communicate broad sets of design values for each domain

   Each domain, specialty, or area of expertise communicates its preferred range of design values, requirements, or MOPs. It explicitly states what it would ideally like to see in the design from its own perspective and in the form of sets instead of points.

2. Create the design space by developing sets of design solutions

   The total design space is generated by synthesizing the sets of design values communicated by each domain. All of the sets of design values from each domain are intersected (or overlapped) with the other domains to create sets of design solutions.

3. Evaluate the design solutions to identify the feasible and preferred regions

   Each of the synthesized sets of solutions is evaluated for feasibility and considered against the preferences of each domain.

4. Reduce the sets by eliminating the infeasible and highly dominated design solutions

   Sets are eliminated based on the information currently available. Design decisions to eliminate sets are delayed until there is solid justification for doing so.

5. Gather additional information about the remaining sets

   New knowledge and supporting information can be obtained through additional development efforts and testing, higher fidelity analysis, follow-on discussions with other specialties, modeling and simulation, and more in depth research.

6. Converge on a design solution

   The sets are gradually and incrementally reduced as new knowledge is learned and additional information is incorporated until designers converge on a final solution.

7. Document the rationale for eliminating particular design solutions or regions of the design space

Substantiating why a solution is *not* a solution is easier to agree on and typically more scientifically, data, and fact-based than trying to convince someone of or defend why any one particular solution is correct or optimal. The reasoning behind the elimination of a set and the accumulation of new knowledge is traceable and well-documented.

SBD is a systems engineering methodology that utilizes concurrent design efforts to expand the design space, explores design combinations by integrating the intersections, systematically narrows the set by eliminating infeasible or highly dominated design solutions, delays critical decisions until sufficient information is known, converges on a globally optimal design, and documents the rationale for reaching the resultant solution (Sobek, Ward, and Liker 1999). Figure 1 shows a graphical representation of SBD as it applies to regions within the design space. Note that each region can contain a multitude of design points and that only the overlapping regions are feasible.

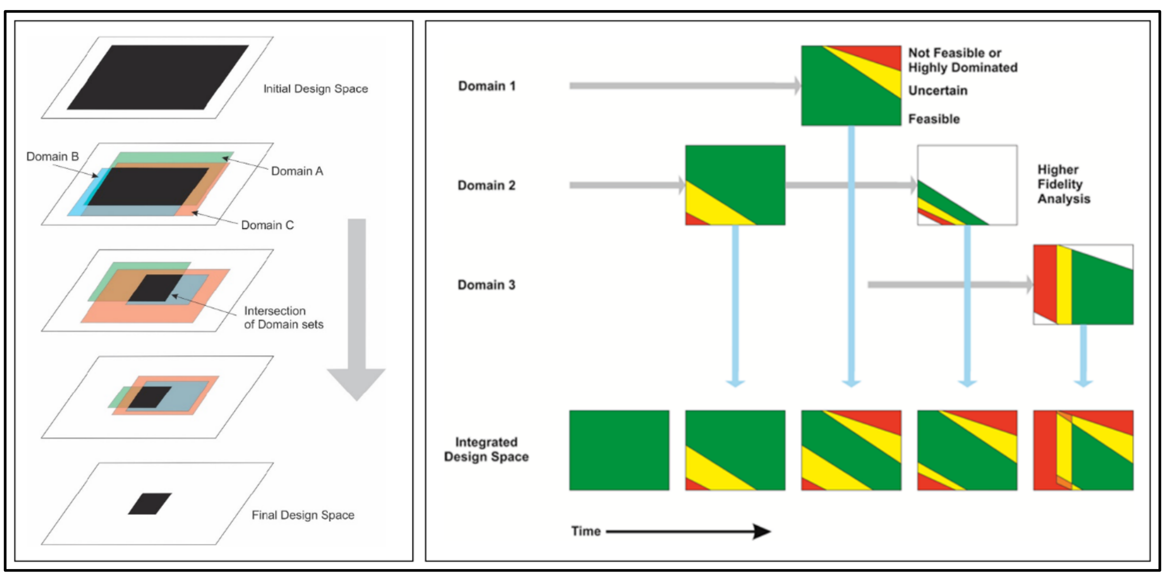


Figure 1. Representation of SBD within Design Space Regions. Source: Singer (2017).

B. SBD CONTRASTED WITH PBD

SBD is fundamentally different than PBD in that the designs converge with SBD rather than evolve with PBD (Liker et al. 1996). In SBD, the final selection is determined by eliminating infeasible and dominated solutions over time as more knowledge and understanding are obtained versus selecting what appears to be the single best solution very early on with PBD. SBD explores concepts in parallel while the solution space is gradually narrowed to a single solution. In PBD, commitment is established early on to one alternative that receives singular focus and further development; it moves from point-to-point in the design space through iterations of modifications and improvements until it is satisfactory (Ward et al. 1995a; Ward et al. 1995b). Table 1 further captures the unique ways each task is accomplished for the PBD and SBD design processes (Singer, Doerry, and Buckley 2009):

Table 1. Comparison of Point-Based and Set-Based Design. Source: Singer, Doerry, and Buckley (2009).

Task	Point-Based Design (PBD)	Set-Based Design (SBD)
Search: How to find solution ideas?	• Iterate an existing idea by modifying it to achieve objectives and improve performance • Brainstorm new ideas	• Define a feasible design space, then constrict it by removing regions where solutions are proven to be inferior
Communication: Which ideas are communicated?	• Communicate the best idea	• Communicate sets of possibilities that are not Pareto dominated
Integration: How to integrate the system?	• Provide teams design budgets and constraints • If a team can't meet budget or constraints, reallocate to other teams	• Look for intersections that meet total system requirements
Selection: How to identify the best idea?	• Formal schemes for selecting the best alternative • Simulate or make prototypes to confirm the solution works	• Design alternatives in parallel • Eliminate those proven inferior to others • Use low cost tests to prove infeasibility or identify Pareto dominance
Optimization: How to optimize the design?	• Analyze and test the design • Modify the design to achieve objectives and improve performance	• Design alternatives in parallel • Eliminate those proven inferior to others
Specification: How to constrain others with respect to your subsystem design?	• Maximize constraints in specifications to assure functionality and interface fit	• Use minimum control specifications to allow optimization and mutual adjustment
Decision Risk: How to minimize risk of "going down the wrong path?"	• Establish feedback channels • Communicate often • Respond quickly to changes	• Establish feasibility before commitment • Pursue options in parallel • Seek solutions robust to physical, market, and design variations
Risk Control: How to control and minimize damage from unreliable communications?	• Establish feedback channels • Communicate often • Respond quickly to changes • Review designs and manage information at transition points	• Stay within sets once committed • Manage uncertainty at process gates

The overall intent of iterative approaches and PBD strategies is to identify the best solution as early as possible and avoid devoting additional time and resources by considering other options unnecessarily (Bernstein 1998). If the selected solution fails to meet requirements or falls short of customer needs, deviations are applied, or it is discarded to pursue a new, more favorable concept. The general steps involved with iterative approaches and PBD strategies include (Singer, Doerry, and Buckley 2009, 4):

1. Define the problem

   Understand the customer's needs and establish product requirements.

2. Generate a large number of alternative, point design concepts

   Engineers and designers have brainstorming sessions to come up with several possible design concepts and solutions. A limited understanding of the tradespace is developed.

3. Conduct preliminary analyses and select a single concept for further development

   A single design solution is chosen early on for further development and made to work if possible. This single design solution represents a particular point in the design space and serves as the starting point for iterative development.

4. Analyze and modify the remaining concept

   The selected design solution is incrementally changed, modified, and improved upon until all of the desired outcomes and requirements are met. Ideally an optimized solution is found.

5. Repeat the process until a solution is found

   The process begins again if the chosen alternative fails to meet the requirements or proves to be infeasible.

The steps above suggest a *hill-climbing* analogy because one organization begins each iteration with what looks like the best alternative and passes it on to different parts for critique and modification, moving up the hill towards total optimality; each new and successive solution is another step closer to the best possible design at the top of the hill (Liker et al. 1996; Ward et al. 1995a; Ward et al. 1995b). Another way to describe this approach is *over-the-wall* because the design solution is thrown over the wall through serial communication between the various functional groups or engineering specialties as decisions are passed to the next in line (Sobek, Ward, and Liker 1999).

One of the major shortcomings of serial engineering lies in the delayed feedback loops. Often times the feedback from downstream functions (the groups on the receiving side of the wall) comes later and perhaps after the upstream functions (groups on the delivery side of the wall) have already committed to a particular solution. By then, each feedback loop is typically associated with reworking or backtracking because important design considerations were omitted early on, or the design did not suffice the next group in line (Bernstein 1998). A domino effect of changes and analysis occurs propagating even more rework and demand for communication. With this potentially perpetual cycle, the process may never reach a state of convergence, and without a clear picture of the entire design space, the final design is likely to be far from optimal (Sobek, Ward, and Liker 1999).

C. CONCURRENT ENGINEERING

CE is offered as an attempt to shift away from this serial *over-the-wall* approach to a parallel process that obtains feedback sooner (Sobek, Ward, and Liker 1999). CE refers to the simultaneous design of a system that has traditionally been developed sequentially, and it can be portrayed as *overlapping problem solving* (Ward et al. 1995a), especially between product development and manufacturing. It is inherently about horizontal coordination across the functions of a system hierarchy and its subsystem interfaces and vertical coordination across the development stages of the system life cycle (Ward et al. 1995b). CE is one step beyond PBD in that it fosters enhanced communication in the engineering community and cross-functional design teams enabled through collocation (Singer, Doerry, and Buckey 2009). The various functional groups and engineering specialties are physically and organizationally closer, which facilitates better upstream and downstream interaction, shortens design processes, feedback cycles, and iterative loops, and mitigates communication errors caused by distance (Liker et al. 1996; Singer, Doerry, and Buckley 2009). According to Smith (1997, 67), CE can be defined in terms of four principles:

1. “The increased role of manufacturing process design in product design decisions”

Manufacturing and functional design constraints are considered simultaneously. Ideally, functional barriers within an organization between design and manufacturing have been removed. The influence of manufacturing processes on design concerns is addressed early on to create the opportunity for reduced manufacturing costs and improved product quality.

2. “The formation of cross-functional teams to accomplish the development process”

   The development of new products and processes is achieved jointly. Cross-functional teams include people with various functional backgrounds to combine different knowledge bases and areas of expertise.

3. “A focus on the customer during the development process”

   Engineers account for customer preferences during the design process and are more responsive to customer desires, leading to a more successful product.

4. “The use of lead time as a source of competitive advantage “

   Time to market is an important factor of modern competition and indicator of potential market success. Reducing the lead time creates a market advantage and enables a faster incorporation of new technologies and response to market trends.

While CE still depends on the establishment of requirements and commitment to decisions as early as possible, it bolsters the ability of downstream groups to influence the design decisions of upstream groups, which results in the improved quality of these critical decisions early on. Ideally, CE intends for downstream groups to work with upstream groups simultaneously, so the design is more likely to be developed “right the first time” (Bernstein 1998; Ward et al. 1995b, 48).

1. CE as a Point-Based Approach

The concept of CE is widely researched and implemented, especially in the midst of today’s complex systems that have become so prevalent. While CE has grossly improved the design of these complex systems, the nature and fundamentals of the point design process itself and the interactions between the upstream and downstream groups have not changed (Bernstein 2018; Liker et al. 1996; Singer, Doerry, and Buckley 2009). The formation of cross-functional teams does not automatically imply the process is set-based. In the case of CE, the initial requirements are established by a certain group, the design solution is proposed by another, and feedback, comments, and recommendations are made by yet another number of groups; a single solution still moves iteratively from point-to-point in the design space until an acceptable solution is reached.

PBD is executed either in series or concurrently (Sobek, Ward, and Liker 1999), but both still require early design decisions with multiple iterations on a single solution. Serial PBD involves passing finalized parts of the design onto the next group in line downstream with minimal feedback loops; the *over-the-wall* approach. Concurrent PBD involves selecting an initial, baseline, perceived best solution, and iterating it with increasing fidelity, while incorporating feedback from other functional groups in parallel until the final design emerges. Concurrent PBD essentially *lowers-the-wall* by obtaining feedback earlier and enabling upstream groups to receive input from downstream groups more frequently (Bernstein 2018).

2. Problems with CE

The essence of CE is supposed to be parallel design, yet the challenges associated with communicating and integrating information across various design groups while being constrained to the establishment of early requirements negates simultaneous design and parallelism.

Conventional, iterative approaches suggest the establishment of firm requirements early on to simplify the interactions among the various design groups and support the "right the first time" philosophy (Ward et al. 1995b). The logic is that each of the design groups does not necessarily understand the requirements and constraints of every other group, so each group must thoroughly specify its part to ensure the proper interface and functionality with others. The problem is that this presents a catch-22 in that designers are encouraged to specify requirements early on, but the very nature of an iterative process means those requirements will change over each succession. Moreover, because each group does not know the limits and needs of another group, any change or decision made by one group can invalidate the previous decisions of another group and also any resultant work hinged on those decisions (Liker et al. 1996). Downstream groups must be able to operate on information presented early on from upstream groups. If this information has not been finalized, then any changes will propagate from the upstream originating group through to any downstream groups using it, which is exacerbated as the more coupling and interdependency exists between groups.

This highly probable potential for wasted effort and work to become moot creates reluctance for some groups to act on communication from other groups and serves as incentive to wait until decisions have been frozen. Contrary to simultaneous design, consecutive decision making and execution of work that could be conducted in parallel revert to serial and sequential, and, as with serial PBD, there is no theoretical guarantee the process will ever converge on a final solution due to the potential perpetual cycle of changes.

3. Beyond CE

Shifting away from a point-based paradigm is needed for truly effective concurrent design to come to fruition (Singer, Doerry, and Buckley 2009), and this underlying paradigm of design must be revolutionized to fully implement CE (Sobek, Ward, and Liker 1999). New tools and approaches are needed to facilitate information transfer and cross-functional communication to allow simultaneous development and to relieve the primary bottleneck of CE that is the communication and decision-making ability of designers (Liker et al. 1996). New ways of thinking, approaching problems, minimizing barriers between functions, and reasoning and communicating about design ideas and solution alternatives are necessary to excel beyond traditional PBD philosophies and stimulate a team-oriented culture with open flow of information, including a central consideration of the complex characteristics of people and behaviors in any new systems engineering process.

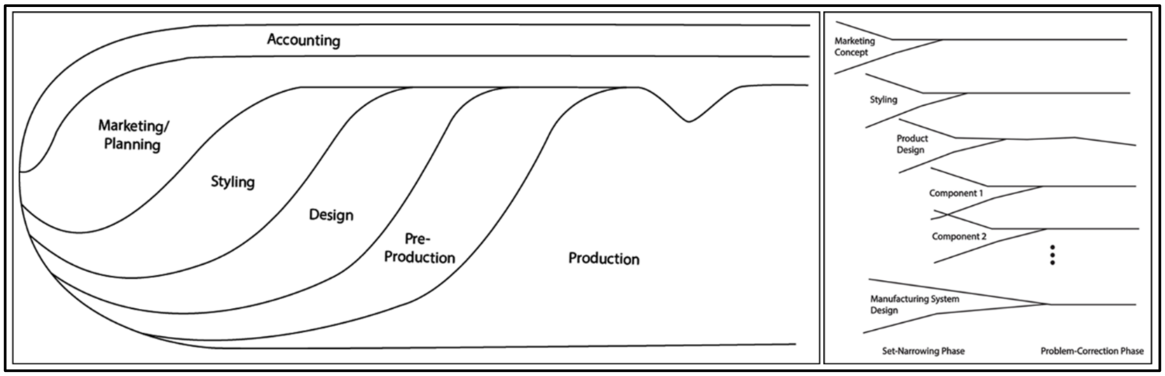
Set-based theories, such as SBD or SBCE, have the potential to substantiate this revolutionary paradigm shift and build the necessary tools and approaches by getting team members to think, reason, and discuss in sets of designs as opposed to individual design ideas (and with stakeholder input). Some implications of SBCE for the design process include:

- Going slow early in the process and considering [a] larger number of alternatives may facilitate faster downstream design, with less need for backtracking;
- Engineers may need to develop new vocabularies for communicating in sets;
- A different [mindsight] is needed to think in terms of set-narrowing as opposed to iteration;
- [Developing firm requirements and full specifications early on (e.g., a complete computer-aided design (CAD) database] may actually inhibit effective design as it forces commitment to many specific design decisions before enough information is known; and,
- [SBD] may reduce the need for frequent face-to-face communication. (Liker et al. 1996, 177)

D. TOYOTA PRACTICES

1. CE Practices at Toyota

Toyota practices are often advertised as models of CE in the automotive industry and are generally viewed as a major source of its competitive advantage (Liker et al. 1996; Sobek, Ward, and Liker 1999; Ward et al. 1995a; Ward et al. 1995b). Toyota describes its design process as one where the focus on certain tasks varies over time, and each group is responsible for its own schedule and acquisition of information to meet major, overall deadlines (see Figure 2).



Based on a sketch by Toyota's general manager of body engineering in 1993

Figure 2. Toyota Design Process. Source: Ward et al. (1995a).

Without viewing Toyota's methods in a SBD paradigm, the impression is they are clumsy, wasteful and grossly inefficient, irrational, and counterintuitive (Ward et al. 1995a). From the perspective of a poorly executed PBD paradigm, it is questionable why Toyota needs to go through so many iterations to generate so many different designs; and why it takes so long to synthesize the first solution in order to provide a starting point to build off of and modify. In fact, Toyota hardly employs any of the commonly proclaimed practices deemed critical for successful CE, such as collocated, dedicated, cross-functional teams, highly structured development process, and frequent meetings with suppliers. Additional deviations from the typical CE elements of freezing specifications early on and reducing the number of prototypes are also observed at Toyota.

Ward, Liker, Sobek, and Cristiano (Liker et al. 1996; Sobek, Ward, and Liker 1999; Ward et al. 1995a; Ward et al. 1995b) describe a second Toyota paradox where delaying decisions, excessive prototyping, providing hard specifications late in the process, and communicating ambiguously enables Toyota to design better cars faster and cheaper (the first is their Toyota Production System [TPS] or just-in-time [JIT] manufacturing). The drastic contrast between the practices Toyota pursues to achieve best-in-class quality in CE and what are purported as essential CE practices suggests misguided conceptions about what is known about CE. Ward et al. assert:

> CE communications can be both very effective and quite efficient, and most work can be done independently. Consensus-oriented, highly reliable decisions can be combined with powerful, creative

> design leadership. A thorough exploration of the design space, leading to highly optimized designs, can be combined with a very fast and efficient process: You *can* have it "cheap, good, and fast," but you must change the conceptual paradigm under which you conduct the development process (Ward et al. 1995a, 215).

2. SBCE Practices at Toyota

Toyota does not operate on a point-to-point structure and instead follows a version of the SBCE paradigm (recall Figure 2). The general approach Toyota uses is (Ward et al. 1995a; Ward et al. 1995b):

1. A set of solutions is defined at the system level, rather than a single solution.

2. The sets of possible solutions are defined for each subsystem.

3. The sets of possible solutions for each subsystem are explored in parallel, and the corresponding design spaces are characterized through analysis, design rules, and experiments.

4. The sets of solutions are gradually narrowed down based on the characterization results in order to converge slowly towards a single solution.

5. Once a single solution for any part of the design has been established, it does not change unless it is absolutely necessary, i.e., there is no hill-climbing to an optimal solution at the top.

The four predominant features of Toyota's approach as summarized by Singer, Doerry, and Buckley (2009) are:

1. The values for each design parameter are defined in broad sets of values to initiate concurrent design efforts;

2. Set-narrowing is delayed until the "last responsible moment" (Ghosh and Seering 2014, 8) to allow more information to become available;

3. The sets are gradually narrowed as the design improves towards becoming more globally optimal; and

4. The design fidelity and level of detail increase as the sets narrow.

According to Sobek, Ward, and Liker (1999, 73), three SBD principals – and guiding sub-principles—encompass these features and Toyota's approach:

1. Map the Design Space

> Toyota maps the design space by developing and characterizing sets of alternatives to use in the convergence process. Engineers and individual specialty groups explore and communicate several alternatives to map out the possibilities, feasibilities, costs, and benefits and gain a better understanding of the design space.

i) Define Feasible Regions

> Each specialty group defines the feasible regions and primary design constraints for its own functions from its own perspective and communicates them to the other specialties, including updates to what is possible, new technologies, and new problems.

ii) Explore Tradeoff by Designing Multiple Alternatives

> Tradeoffs are explored by designing, prototyping, or simulating the various alternatives until "best guesses" can be made "based on judgment and experience only when the decision is obvious, unimportant, or subjective; otherwise they invest as required to gather quantifiable data."

iii) Communicate Sets of Possibilities

> Sets of ideas and regions of the design space are communicated instead of singular ones to better understand the implications of choosing certain alternatives over others and the impact on the overall system. Discrete alternatives, lists of ideas, drawings, models, options, constraints, bounded or open-ended intervals, tradeoff curves, performance charts, and best estimates with design tolerances are examples of ways to communicate about sets of possibilities.

2. Integrate by Intersection

> Solutions acceptable to all specialties are identified after each one understands the possibilities from its own perspective and others'.

i) Look for the Intersection of Feasible Sets

Engineers look for intersections of different specialties and identify where feasible regions overlap to find a solution acceptable to all that optimizes overall system performance. Specialties are given the opportunity to study, critique, and provide feedback on solutions to achieve consensus before a formal course of action is proposed.

ii) Impose Minimum Constraint

Minimum constraint is imposed by “making decisions in their own time” so further exploration or adjustments can be made (as opposed to constraining the design with early decisions).

iii) Seek Conceptual Robustness

Design alternatives that remain functional and maintain performance despite input variations (as defined by Taguchi (1993)) or that remain valid regardless of which part of the design decisions are made are strived for.

3. Establish Feasibility Before Commitment

Late problems are avoided and designs that enable overall system optimization are realized by understanding the possibilities and interactions prior to committing to any one particular solution.

i) Narrow Sets Gradually while Increasing Detail

The most important alternatives are considered more thoroughly as ideas are eliminated gradually.

ii) Stay within Sets Once Committed

Decisions must remain inside the narrowed sets so other specialties can proceed without concern for changes.

iii) Control by Managing Uncertainty at Process Gates

The design process is viewed as a continuous flow through a series of gates pertaining to an integrating event. Information is exchanged as needed and the level of uncertainty is controlled at each gate and reduced successively.

E. TYPICAL SBD EXECUTION

While the principles encompassing Toyota practices (Sobek, Ward, and Liker 1999) exhibit some agreement with the general implementation steps for SBD (McKenney and Singer 2014; Singer et al. 2017; Singer, Doerry, and Buckley 2009; Specking et al. 2017), more delineation is needed to describe the process with less confusion and is discussed in Chapter III as part of the contribution offered by this dissertation. Concepts and methods found in the literature related to set creation and elimination are described here, although better definition is required for both to develop SBD more fully (Specking et al. 2017). Novel methods of set creation and set elimination as they manifest a contribution are also introduced in Chapter III.

1. Set Creation

a. System-Level Solutions

The analysis based preferences and feasible regions from each specialty are integrated into the larger context that includes all specialties. By intersecting all feasible regions across all specialties, a smaller set of unified global concepts representing total system solutions is created. In order to truly have a system-level solution, there must be intersecting points between specialties; if not, some ranges must be expanded to create them.

b. Methods of Set Creation

There is no right way to create or define sets in the integrated set space, and it can be accomplished in a number of ways. Through a structured literature review, Specking et al. (2017) found set creation mentioned in work surrounding four unique categories:

1. *Design Space*—e.g., using all possible permutations of specific design factors, parametric design, designing space tree, kernel-based support vector domains and clustering methods, design space mapping, function to form mapping;

2. *Performance or Preference Modeling*—e.g., data driven predictive modeling, preference generation;

3. *Component Selection*—e.g., catalog numbers, search tree algorithms; and

4. *Set Theoretic*—e.g., abstracting design alternatives into a set, first-order predicate logic expressions and quantified relations.

2. Set Elimination

a. Mutually Feasible Sets

All new information and preferences are consolidated and considered (typically by an integration role) to determine the infeasible and highly dominated solutions. Set reduction is the process of eliminating these infeasible or dominated solutions under the assumption that no further knowledge will change the reduction decision (Singer et al. 2017). The solutions that are not eliminated and prevail to be investigated further are those that are mutually feasible regions; which implies only those regions that intersect.

b. Pareto Front

While the principle intent of most PBD methods is to identify and select an optimal solution along the Pareto front directly, SBD considers it differently. First, points within sets that lie on the Pareto frontier can be identified better in SBD than they can as standalone points in PBD (see Figure 3). Secondly, depicted graphically, the infeasible and highly dominated regions are quite distant from the Pareto front (see Figure 4).

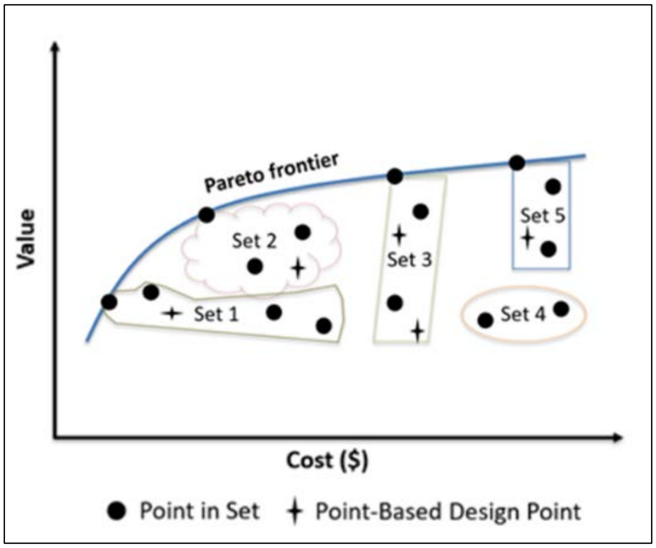


Figure 3. Sets along the Pareto Front. Source: Wade et al. (2018).

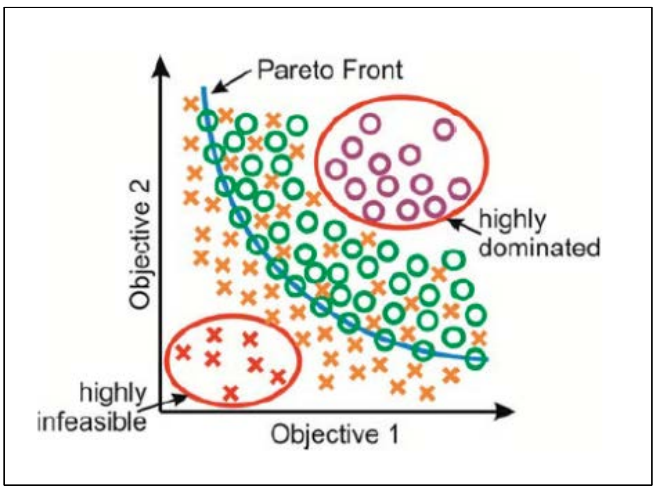


Figure 4. Distance from the Pareto Front. Source: Singer et al. (2017).

SBD seeks to eliminate these infeasible and highly dominated regions as they become apparent with the current information available. Even with a better understanding of the problem as more knowledge is acquired through future analysis, refined requirements, and evolved customer preferences, the infeasible and highly dominated solutions are not likely to change. The set reduction process allows the colossal number of potential material solutions to be converged as Pareto improvements, while producing globally preferred solutions. Eliminating the worst solutions rather than selecting the best solution allows more time to hone in on one by increasing the quality and level of detail, learning new information, strategically responding to requirements as they solidify, and gradually progressing through the design process.

c. Methods of Set Reduction

As with set creation, there is no right way to eliminate sets from the successive iterations of the integrated set space, and it can be accomplished in a number of ways. In the same structured literature review referenced for set creation, Specking et al. (2017) also found set elimination mentioned in work surrounding four unique categories:

1. *Infeasibility*—e.g., specification conflict, inability to meet constraint requirements, specific elimination criteria, component incompatibility, proving alternatives to be inferior;

2. *Preference*—e.g., cost and weight, difference in performance, magnitude of main effects plots, designer preference, decision rules with real options, wellbeing indicators;

3. *Pareto Optimality*—e.g., low cost test to prove Pareto dominance, dominance analysis, Pareto frontier and tradeoff; and

4. *Set Theoretic*—e.g., set-based reasoning as a basis for exploring or not exploring alternatives.

d. Engineering Reasoning in Set Reduction

Engineering reasoning must be used to determine when it is appropriate to eliminate solutions (and ultimately when to finally converge on one). Whitcomb and Hernandez (2017) describe three basic reasoning approaches—deductive, inductive, and abductive—and a hybrid class, retroductive, that combines all three into a design process. The difference between approaches is based on what is initially given and what is derived as the reasoning process is carried out, i.e., design variables (V), knowledge (K), or specifications (S) (see Table 2).

Table 2. Types of Reasoning. Source: Whitcomb and Hernandez (2017).

Reasoning Type	Given	Derived
Deductive	V, K	S
Inductive	V, S	K
Abductive	K, S	V

Traditional systems engineering and PBD methods resemble the path of deductive reasoning, as does set creation in SBD. The set reduction phase of SBD, however, is a process of discovery through iterations of inductive and abductive reasoning, which makes SBD retroductive overall. The introduction of new knowledge and variables can lead to updated specifications and refined sets of solutions initiating an ensuing cycle of retroduction. Figure 5 shows this retroductive reasoning cycle superimposed on a decision timeline for SBD as it occurs in the set reduction phase.

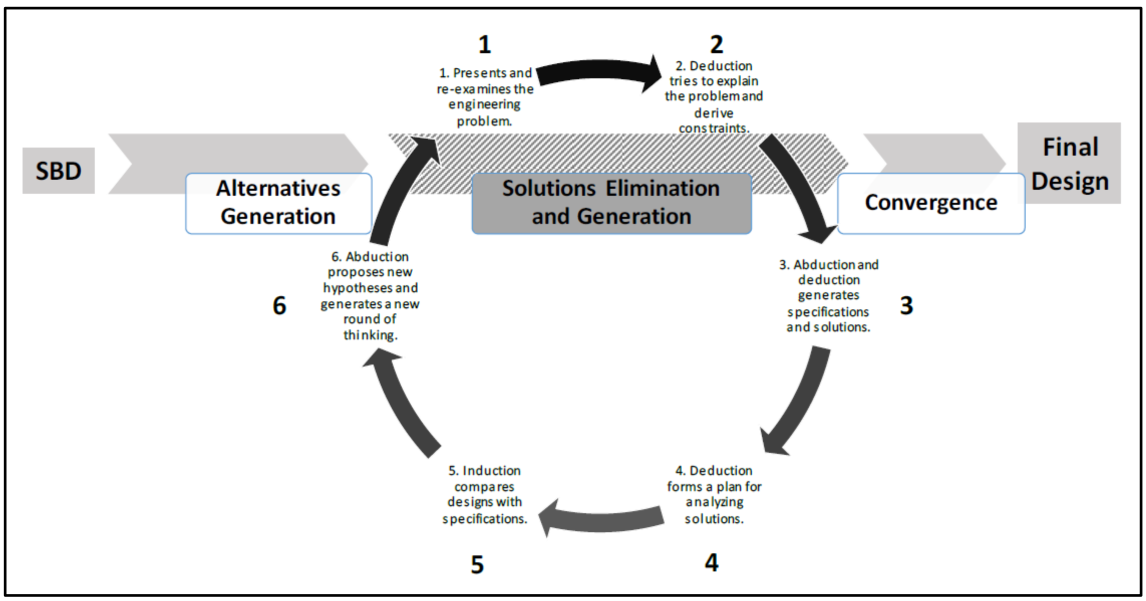


Figure 5. Reasoning in SBD. Source: Whitcomb and Hernandez (2017).

F. EXPECTATIONS OF SBD

Although there is a plethora of ways to achieve convergence on a final solution in SBD, "thorough characterization of the design space, maintaining flexibility throughout set reductions, tracking convergence, documenting reduction decisions, continuous communication [and feedback], and proactive leadership during execution are all important elements of a productive SBD process" (Singer et al. 2017, 43). The bearing of SBD is captured by three basic tenets as is evidenced above and reminiscent of Toyota practices:

1. Consider a large number of design alternatives by understanding the design space, [i.e., sets];

2. Allow specialists to independently review and consider a design from their own perspective; and

3. Use the intersections between individual sets to optimize a design and establish feasibility [and viability] before commitment. (Singer, Doerry, and Buckley 2009, 10)

SBD can be implemented in any number of ways, and, at a minimum, the application of these principles "enables the development of conceptually robust designs and promises a capacity to adapt quickly to changing requirements and design discoveries" (Singer, Doerry, and Buckley 2009, 11). Additional expectations and intended outcomes of SBD include:

- Identifying principle design factors and manageable sets of values essential to achieving maximum effectiveness;

- Recognizing which sets are of highest priority;

- Understanding the sensitivity of the varying design combinations to certain design factors and sets of values, i.e., which design factors and values are the most important to compare when distinguishing between the most promising synthesized designs;

- Comparatively assessing the total value of the most promising designs based on current knowledge;
- Examining how a shift in design factor priorities impacts the final design recommendation;
- Substantiating decisions to eliminate particular design solutions by documenting the design space evaluation and analysis; and
- Providing a resource for design flexibility if changes in operational requirements, anticipated technologies, program funding, or execution schedules are introduced in the future.

G. SET-BASED THINKING

By consolidating the prominent SBD principles inherent in Toyota's philosophy and culture with other examples of SBD methods and techniques found within the literature, Ghosh and Seering identify seven common characteristics of SBPD:

1. Emphasis on frequent, low-fidelity prototyping;
2. Tolerance for under-defined system specifications;
3. More efficient communication among subsystems;
4. Emphasis on documenting lessons learned and new knowledge;
5. Support for decentralized leadership structure and distributed, non-collocated teams;
6. Supplier/subsystem exploration of optimality; and
7. Support for flow-up knowledge creation. (Ghosh and Seering 2014, 2)

Using the seven characteristics as a filter for determining what constitutes SBD, Ghosh and Seering (2014) inductively arrive at two principles to describe the most common themes and influences across SBD-related work (referred to as the principles of SBT):

1. Considering Sets of Distinct Alternatives Concurrently

2. Delaying Convergent Decision Making

These two principles of SBT serve as a basis for defining SBD, and combining them with the three SBD tenets helps in identifying what is actually SBD.

H. OTHER ENGINEERING METHODS

Although there are several other engineering methods that are familiar, well-accepted, and widely employed, the way they are commonly implemented does not constitute SBD. A limited number of examples are summarized in Table 3 and are described and addressed in the following sections to demonstrate how conventional methods do not ordinarily embrace SBT or SBD methodologies. It is important to distinguish these common methods from SBD in order to show that SBD is *not* being performed and to encourage an open-mindedness about it as an alternative design method (especially for the more familiar spiral, vee, and agile life cycle process models and systems engineering).

Table 3. Comparison of SBD to other Methods

	Name	Description	Reason it is not SBD
Pre-Specified and Sequential Life Cycle Process Models	Historical Code-and-Fix	• Essentially a trial-and-error model • Tries a solution first, then considers requirements, design, test	• Does not consider sets of distinct alternatives and instead brute forces a solution • Does not delay decisions and instead picks one immediately
	Stagewise	• Addresses the code-and-fix downfalls • Stipulates successive stages	• Essentially over-the-wall processes • Lacks efficient communication among subsystems • Lacks support for flow-up knowledge creation • Lacks tolerance for under-defined system specifications; specifies early on • Do not consider sets of distinct alternatives
	Waterfall	• Enhances the stagewise model • Recognizes feedback loops between stages	
	Evolutionary and Transform	• Address propensity of waterfall models to pursue stages in the wrong order • Evolutionary provides quick initial capability and plan for subsequent improvements; Transform model transforms initial capability into specification and optimizes design from this new starting point	
Evolutionary and Concurrent Life Cycle Process Models	Spiral	• Classic approach for ship design • Design factors considered in sequence around the spiral in increasing detail upon convergence • Win-Win variation accounts for stakeholder win conditions	• Seeks a base design for use as a starting point to develop further • Does not consider entire design space; looks at a subset of alternatives instead • May choose not to select alternatives based on risk, but does not specifically eliminate the infeasible or highly dominated regions • Lacks support for flow-up knowledge creation
	Vee	• Left side follows waterfall model as decomposition descends into design details • Right side describes integration and verification flowing up to a user-validated system	• Attempts to identify the best solution based on user requirements and advances that solution • Communications between subsystems is not the overarching culture • Does not necessarily use the objective and threshold values to synthesize the entire design space
Interpersonal and Emergent Life Cycle Process Models	Agile (Extreme program., dynamic systems dev., scrum, etc.)	• Shifts away from heavy plans and specifications • Values individuals and interactions over processes and tools; products over documentation; collaboration over contract negotiation; responding to change over following a plan	• Prefers minimal documentation, which negates the emphasis on documenting lessons learned and new knowledge • Does not explore the entire design space or eliminate options • Not typically executed across multiple specialties and does not account for intersection of preferences to achieve global optimization • Does not delay decisions to avoid refactoring
Other Engineering Methods	Systems Engineering	• Primarily considered a discipline, but can also be a perspective (interdisciplinary approach) or process (top-down iterative design, bottom-up realization process) when applied by systems engineers	• SBD is the methodology that systems engineers would employ • Much more to systems engineering than what is captured in SBD, such as business or mission analysis, architecture, validation, etc. • SBD utilized during activities that require generating and comparing options
	Design for "X" (DfX)	• Optimizes a specific aspect of the design (represented by X)	• Lacks subsystem exploration of optimality and does not converge to a globally optimal solution • Does not consider multiple specialty areas concurrently • Seeks to optimize a specific X instead of eliminating infeasible and highly dominated regions
	Design-Build-Test (DBT)	• Iterative process where single cycle provides information to the next iteration over the three phases of design, build, and test	• No strong element of allowing the specialties to consider the design space from their own perspectives • Uses new learnings from test phase as basis for whether to consider new options or pursue current ones, instead of as justification for eliminating regions of the design space • Does not consider the entire design space
	Method of Controlled Convergence (MCC)	• Defines selection criteria for evaluating concepts, generates large number of design alternatives, and converges based on eliminating weakest concepts or those that do not meet requirements	• Capable of specifying the evaluation criteria in sets, but does not explicitly synthesize them to create a total design space or intersect them for feasibility • Decisions can only be made with respect to individual alternatives and not grander regions of the design space
Decision-Making and Optimization Methods	Multiple Attribute Decision Making (MADM)	• Used for prioritizing and sorting alternatives by rank according to several criteria (e.g., quality functional deployment (QFD) and analytical hierarchy process (AHP))	• Same argument as MCC in that although there is ability to define sets of design factors, they are never explicitly synthesize to create a total design space • Highest ranking or most appealing option is selected after evaluation as opposed to converging on a solution by elimination
	Multiple Objective Decision Making (MODM)	• Methods involving multiple criteria where the optimal solution cannot be obtained without accounting for preferences	• Rely on integrated computational tools to identify optimal solutions, whereas SBD allows each specialty to use its own tools and communicate its preferences without needing to combine all of the computational tools for each specialty • Considers multiple points and uses algorithms to converge on a solution, whereas SBD explores sets of possibilities and uses elimination for convergence
Design of Experiments and Response Surface Methods	Design of Experiments (DOE) and Response Surface Methods (RSM)	• Organized method of creating and sampling the design space, curve-fitting the observed data with a response surface approximation, and selecting a solution based on Pareto optimality	• Utilizes curve fits instead of point clouds used in SBD; SBD involves the points without the curve fits • Commonly uses exterior points and limited interior points, whereas SBD samples the entire design space, especially interior points • Decisions are generally focused on a Pareto (or some other optimal) front, whereas they are based on the actual points in SBD • Makes tradeoffs visible at the highest level, but not during entire design process

1. Life Cycle Process Models

Every system has a life cycle comprised of various stages it must go through, even if not formally defined. Progress through the life cycle for man-made systems is a result of people in organizations performing and managing the execution of actions through processes (ISO/IEC/IEEE 2015). The generic life cycle stages of concept, development, production, utilization, support, and retirement (ISO/IEC 2010) are separated by decision gates (or control gates, milestones, or reviews) with typical options of acceptable (proceed to the next stage, or continue with action items); unacceptable (do not proceed, go back to a previous stage, or hold); or unsalvageable (terminate the project) (INCOSE 2015). Although the progression of these stages often occurs in a linear time sequence as illustrated in Figure 6, they do not necessarily have to as shown in Figure 7.

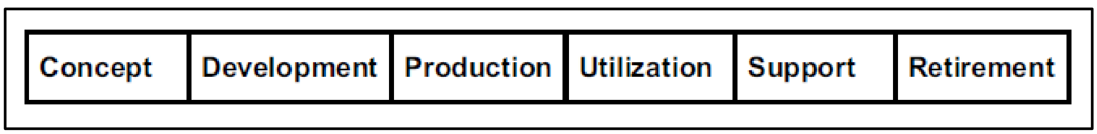


Figure 6. Linear Progression of Life Cycle Stages. Source: ISO/IEC (2010).

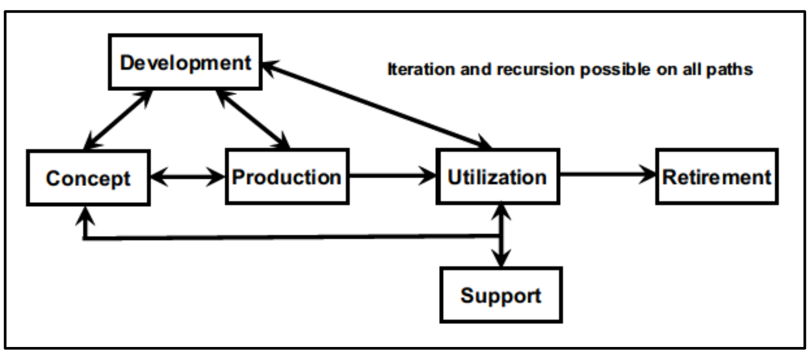


Figure 7. Non-linear Progression of Life Cycle Stages. Source: ISO/IEC (2010).

Several life cycle process models are useful in defining the process activities at each stage and the associated entry and exit criteria. ISO/IEC/IEEE (2015) describes four distinct process groups related to agreement, organizational project-enabling, technical management, and technical processes. Forseberg, Mooz, and Cotterman (2005) have further categorized the technical processes as: pre-specified and sequential (e.g., waterfall); evolutionary and concurrent (e.g., spiral and vee); or interpersonal and emergent (e.g., agile development).

a. Pre-specified and Sequential

(1) Description of Historical Methods, Stagewise, and Waterfall Models

Many of the familiar life cycle models have roots in software design problems and have evolved over the years in response to certain limitations. The basic model used in the earliest days of software development is the code-and-fix model (Boehm 1988), which can be interpreted as a sort of trial-and-error model outside the software realm. This method of trying a solution first and then thinking about the requirements, design, test, and maintenance later has obvious downfalls, including:

- After so many trials, the solution configuration becomes so poorly structured and managed that subsequent changes and rework are very expensive;
- Often times the final solution does not meet the needs of the user so it is either rejected or redeveloped; and
- Solutions are usually expensive due to poor preparation for testing and modification.

The stagewise model addresses these downfalls by stipulating successive stages (or, steps or phases), such as a requirements phase before design, a design phase before trying a solution, and planning and preparation stages for considering test and evaluation phases early on. The waterfall model described by Royce (1970) enhances the stagewise model by recognizing feedback loops between stages and including a prototyping (build-it-twice, implementation, or low-rate initial production [LRIP]) phase. Figure 8 shows ideally how each phase is executed and completed in sequence until the final product is delivered from a general systems engineering perspective.

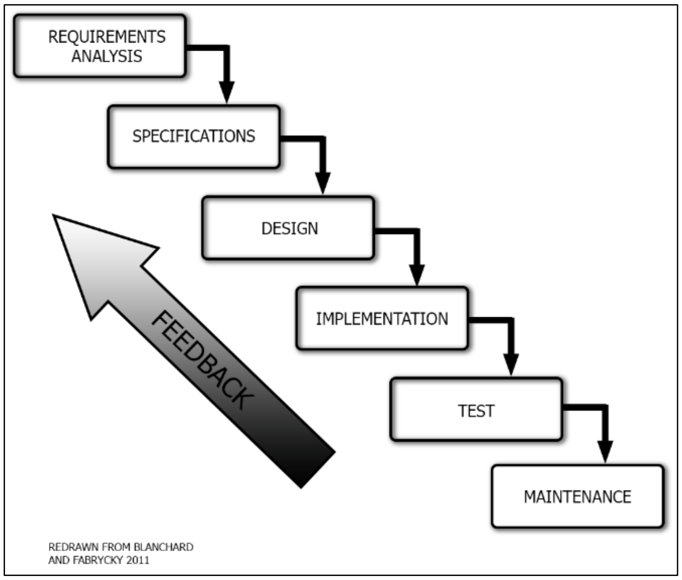


Figure 8. Waterfall Model. Source: Adapted from Blanchard and Fabrycky (2011).

Boehm (1988) more realistically describes how the iterative interaction between the various phases is never confined to successive steps, and feedback loops should be present between testing, design, and requirements to minimize development risk late in the process and ensure traceability. It also includes five additional features that constitute an effective waterfall approach as presented in Royce (1970):

1. Program Design Comes First (facilitated by the addition of a preliminary design phase);
2. Document the Design;
3. Do It Twice (through a mini process or smaller implementation for verification purposes);
4. Plan, Control, and Monitor Testing; and
5. Involve the Customer (in a formal way so there is commitment early on before final delivery).

Although less familiar outside of software, the evolutionary development and transform models address the propensity of waterfall models to pursue stages in the wrong order (Boehm 1988). The evolutionary development model is beneficial when stakeholders are unsure of what they want, but will know it when they see it because it gives them a quick initial capability along with a plan for subsequent improvements. The transform model goes a step further and transforms the initial capability into specification and optimizes the design from this new starting point. Circumstances where the transform model are of value include, when multiple independently evolved systems must be closely integrated, when temporary workarounds become solid constraints, or when a larger legacy system is being incrementally replaced by new systems and decent methods of bridging the new and existing systems are required.

(2) Discussion of Historical Methods, Stagewise, and Waterfall Life Cycle Models as They Compare to SBD

The code-and-fix model definitely does not contain elements of SBT or SBD. It does not consider sets of distinct alternatives and instead practically brute forces specific solutions right from the beginning in hopes of success. Although it might end up considering several design alternatives by default from trying the wrong thing over and over, it is by no means on the order of the large combinatorial explosion of the entire design space, nor is the exploration performed in a systematic or strategic way. The timeline for decisions is quite the opposite of delayed and occurs immediately as a starting point of the process. The various derivations of the waterfall model are all iterative and point-based. Although there may be feedback loops, they are essentially *over-the-wall* processes that lack efficient communication among subsystems, support for flow-up knowledge creation, and tolerance for under-defined systems specifications (as considered from the seven characteristics of SBPD).

b. Evolutionary and Concurrent

Many DoD design processes follow evolutionary and concurrent life cycle process models, such as the spiral and vee. The Naval ship building community primarily utilizes a spiral approach, while most other DoD programs have typically used the vee.

(1) Description of the Spiral Model

Before the SSC, MCM, ACV, Small Surface Combatant, and LDUUV efforts that used SBD, the classic approach for ship design was the Design Spiral shown in Figure 9.

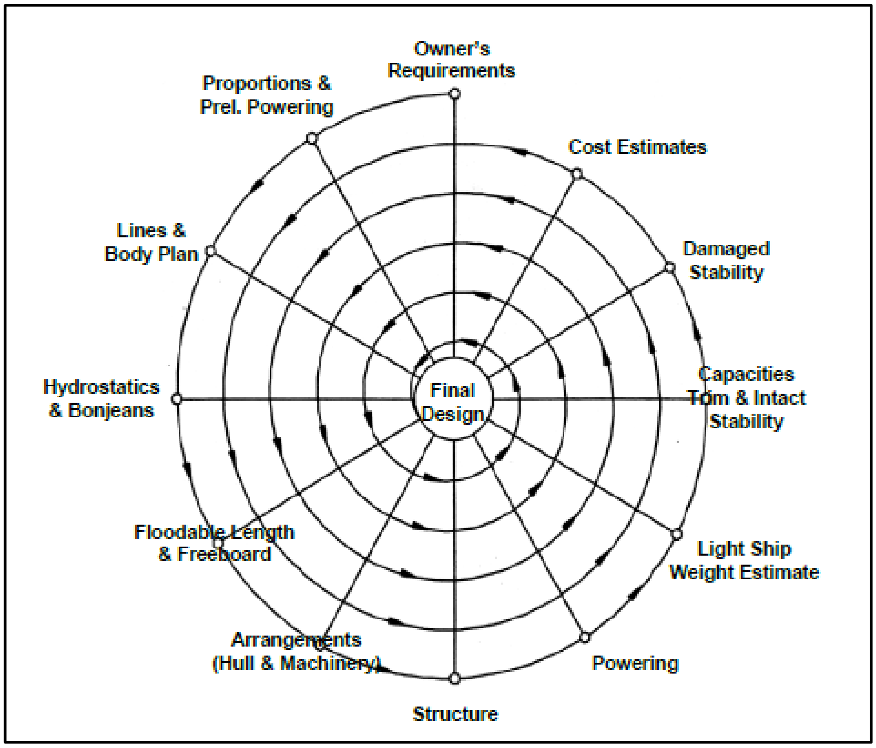


Figure 9. Design Spiral. Source: NAVSEASYSCOM (2012).

The spiral model is intended as a risk-driven approach to development and can lead to the utilization of other process models as a result of the risk patterns identified around each pass of the spiral. Each design factor within the various specialties is considered in sequence as the design progresses counter-clockwise around each cycle of the spiral with increasing detail and decreasing risk. The design is tested for convergence after completing each cycle, which involve the same sequence of steps but different levels of elaboration. A particular cycle is repeated at the same level of fidelity if convergence is not achieved, but progresses to the next cycle at a higher fidelity if it is. The cycles continue until an acceptable and satisficing design is reached. This final design can also be further developed and refined following convergence (Evans 1959; NAVSEASYSCOM 2012; Singer, Doerry, and Buckley 2009).

Perhaps Hall (1969) stimulated the development of the spiral model with his discussion of a morphological analysis technique related to the three dimensions of systems engineering:

1. Time dimension segmented by major decision milestones;

2. Problem solving dimension with procedures and steps to be performed; and

3. Subject matter expertise (SME) dimension that "refers to the body of facts, models, and procedures which define discipline, professions, or technology" (Hall 1969, 156).

Within Hall (1969), a cornucopia shape (formed when the problem solving steps modeled as a linear dimension overemphasize the temporal features) resembles an iterative and converging spiral as shown in Figure 10. When the cornucopia is stretched out and viewed from the larger diameter opening, the counter-clockwise progression of design and bi-directional feedback paths appear in the two-dimensional morphology (see Figure 11).

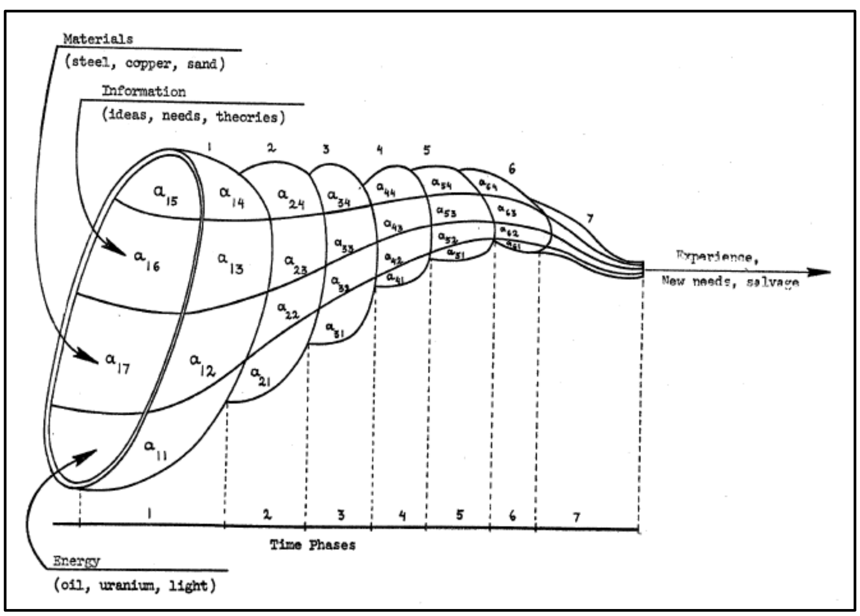


Figure 10. Cornucopia Model. Source: Hall (1969).

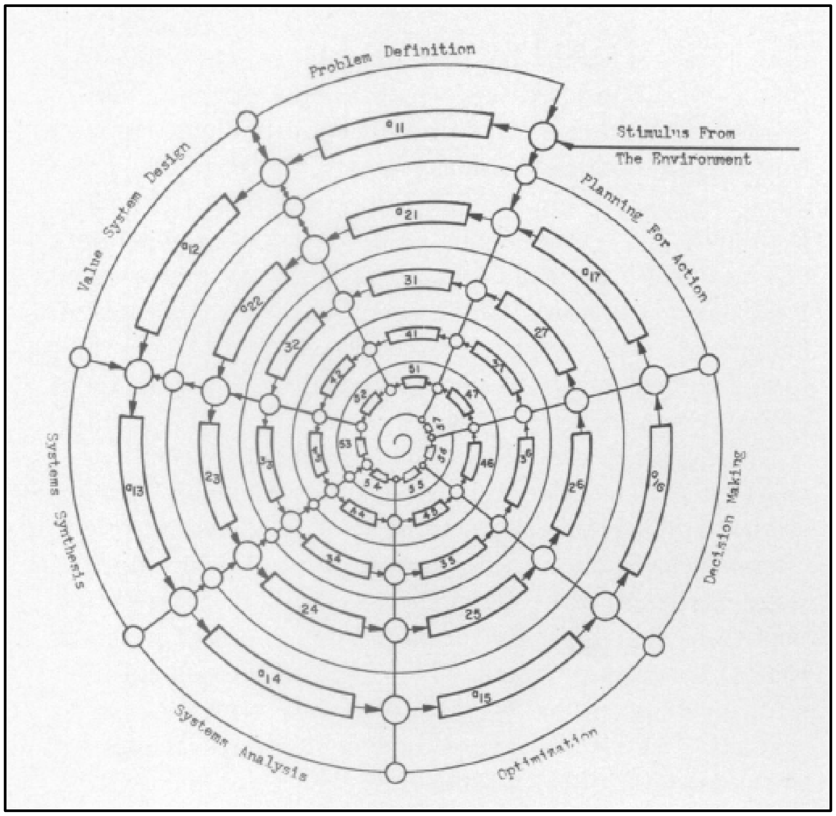


Figure 11. Cornucopia Model End Perspective. Source: Hall (1969).

The evolution of the spiral model can also be attributed to the various refinements of the waterfall model in terms of software and the design process in general. An original diagram of spiral development from Boehm (2000) in Figure 12 shows the design steps progressing clockwise along the angular dimension in addition to the cumulative costs incurred and how they become more detailed and exact in the radial dimension through the accomplishment of each step. A later rendition is the WinWin spiral model in Figure 13 that extends the original spiral model by identifying stakeholders and their win conditions; determining the circumstances under which the system will produce what outcomes (win-win, win-lose, or lose-lose); evaluating, negotiating, and synthesizing the candidate win-win alternatives; analyzing, assessing, and resolving the risks of win-lose and lose-lose solutions; obtaining and accumulating commitments; and repeating the cycle through the spiral.

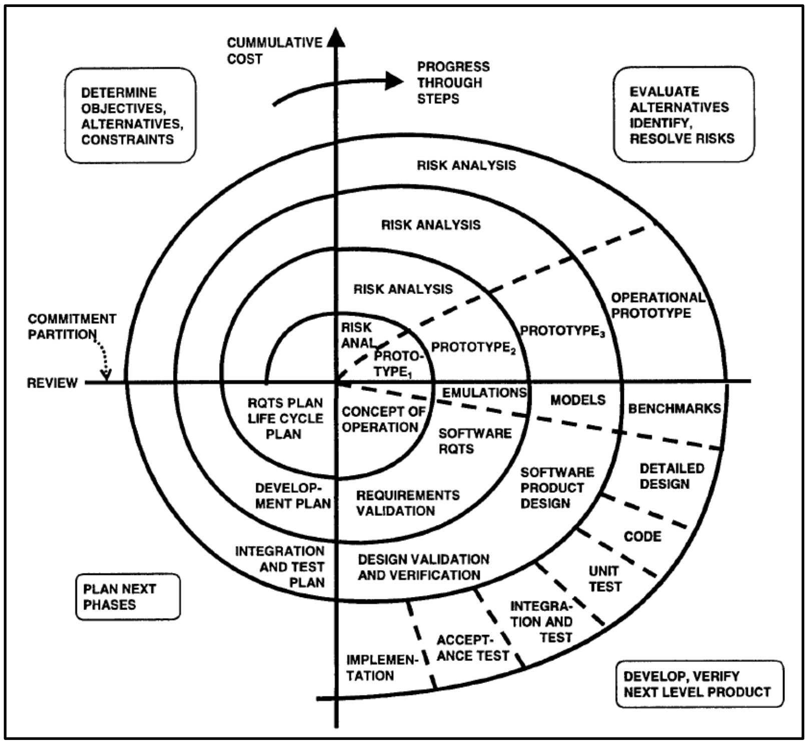


Figure 12. Spiral Development Model. Source: Boehm (2000).

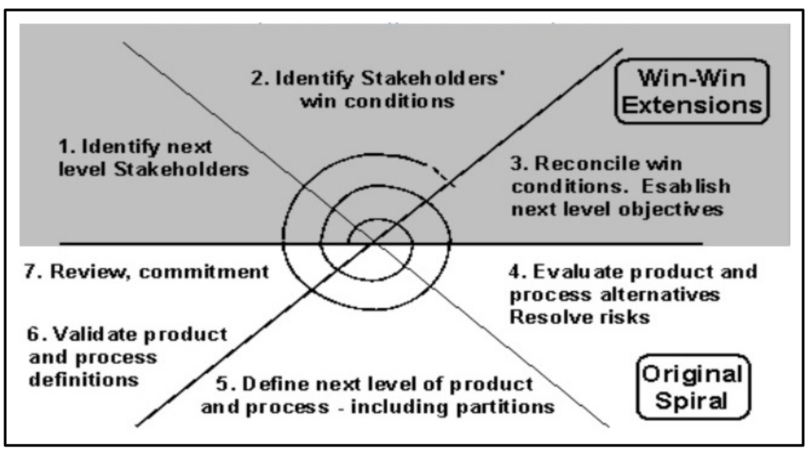


Figure 13. WinWin Spiral Model. Source: Boehm (2000).

Boehm (2000) characterized the spiral model further by defining six attributes that invariably must be incorporated:

1. *Concurrent Rather than Sequential Determination of Design Factors and Values* to avoid constraining the design prematurely;

2. *Consideration of the Requirements, Constraints, Alternatives, Risks, and Stakeholder Commitment in Each Spiral Cycle* to avoid "commitment to unacceptable or overly risky alternatives, [or] wasted effort elaborating on unsatisfactory alternatives" (Boehm 2000, 9);

3. *Level of Effort Driven by Risk Considerations* by answering the question "how much is enough" of each activity (specialty engineering, prototyping, testing, etc.) to avoid overkill and diminishing returns or belated risk resolution;

4. *Degree of Detail Driven by Risk Considerations* by answering the question "how much is enough" of each design factor (requirements, specifications, plans, design, etc.) to avoid over- or under-specifying requirements and to recognize when looser constraints or tighter tolerances are appropriate;

5. *Use of Anchor Point Milestones* to serve as stakeholder commitment points, gates, or checkpoints to avoid "analysis paralysis" (Boehm 2000, 14), unrealistic expectations, requirements creep, and unsustainable alternatives; and

6. *Emphasis on System Life Cycle Activities and Design Factors* to drive system solutions based on requirements, milestones, and risk that enable system-level optimization.

(2) Discussion of the Spiral Life Cycle Model as it Compares to SBD

The spiral model is considered PBD because it seeks a base design for use as a starting point to develop further. It does not consider the entire design space and instead looks at a subset of alternatives up front and moves towards a pointed selection as more detail is obtained and risks are resolved. The WinWin spiral model is an improvement in that it involves “using negotiation processes to determine a mutually satisfactory set of objectives, constraints, and alternatives for the stakeholders” (Boehm 2000, 23) based on their win criteria. It also begins to synthesize these various conditions deemed acceptable to stakeholders and considers those that are less favorable. What the WinWin spiral model does not do, however, is systematically eliminate infeasible and highly dominated regions from the total design space generated during synthesis. Certain design solutions may be inferior based on risk factors and may not be selected as a result, but they are not deliberately eliminated to converge on an optimal solution. Overall, the process still continues to limit the evaluation of the design space to a very small subset of promising design alternatives.

Neither of the models explicitly allow specialists to collectively or independently consider designs from their own perspectives. The focus of the sets in the WinWin spiral model are attributed to stakeholder requirements, and are not necessarily translated or refined into specialist requirements. While there may be negotiation of the win conditions, there is no negotiation of sets of design factor values as specified by the specialists.

Hall’s (1969) two-dimensional morphological representation of systems engineering in Figure 11 only accounts for the temporal and process dimensions and not the SME dimension. Even if designers were perfectly versed in the tools, models, and attitudes essential in the execution of all the activities on the two-dimensional morphological matrix, it would still be insufficient for producing anything comprehensive or viable because the application of specific knowledge and technology is lacking. This third dimension of SME is missing from the spiral models and is necessary for more well-rounded systems engineering as shown in Figure 14 (Hall 1969). The morphological box in three dimensions is more representative of SBD because it captures the entire design space and includes regions of SME input.

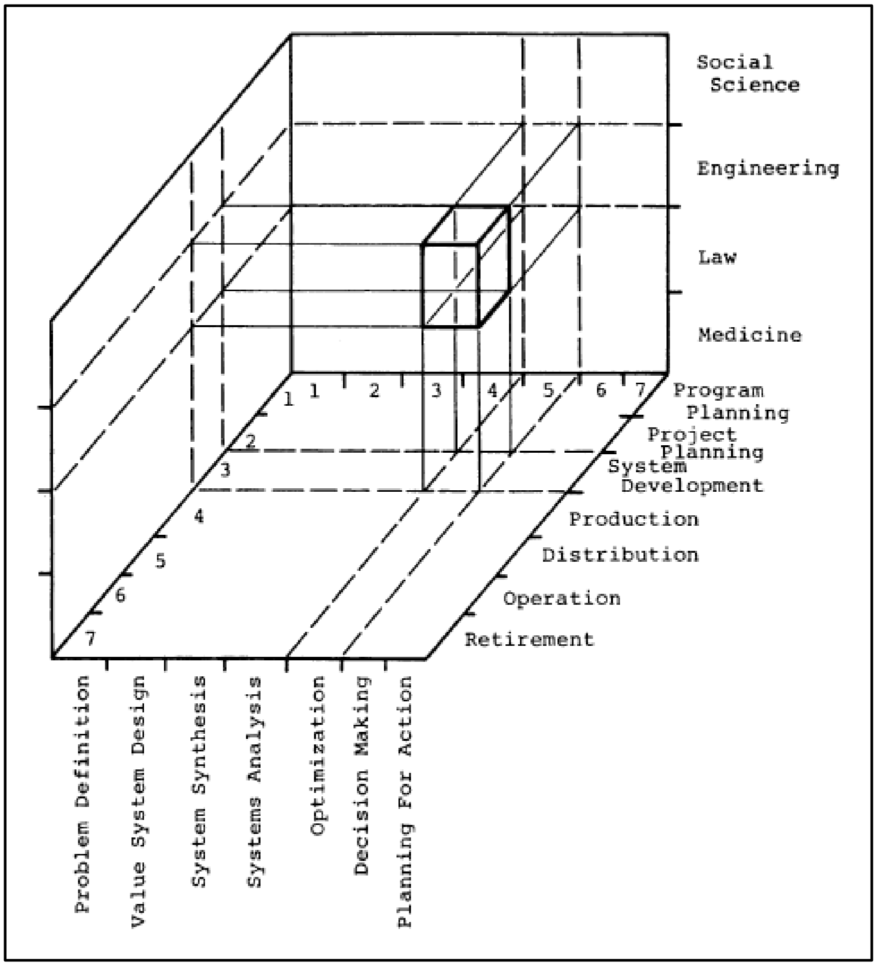


Figure 14. Three Dimensions of Systems Engineering. Source: Hall (1969).

There is also no set number of cycles around the spiral. While decision making is delayed by allowing progression through each cycle in greater detail, there is a tendency to declare a solution prematurely based on time and schedule limitations. The solution is typically not optimized at the system level either, since it did not go through a true set-based process. An alternative view of the spiral model in Figure 15 highlights the serial nature of the process and better shows why designs are “done” when time runs out and not necessarily when it is converged or optimal.

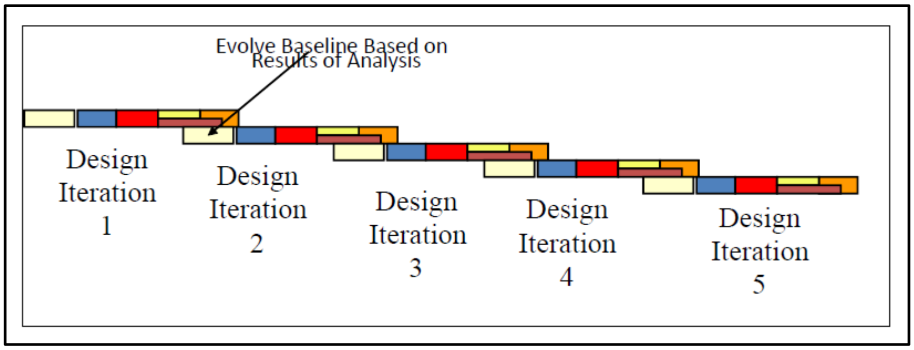


Figure 15. Iterative View of the Spiral Model. Source: NAVSEASYSCOM (2012).

(3) Description of the Vee Model

Forsberg and Mooz (1991) capture the technical aspect of the project cycle with their description of the vee model. The left side of the vee follows the waterfall model as decomposition and definition activities descend into the system architecture and design details, while the integration and verification activities on the right side flow upward ending with a user-validated system (see Figure 16).

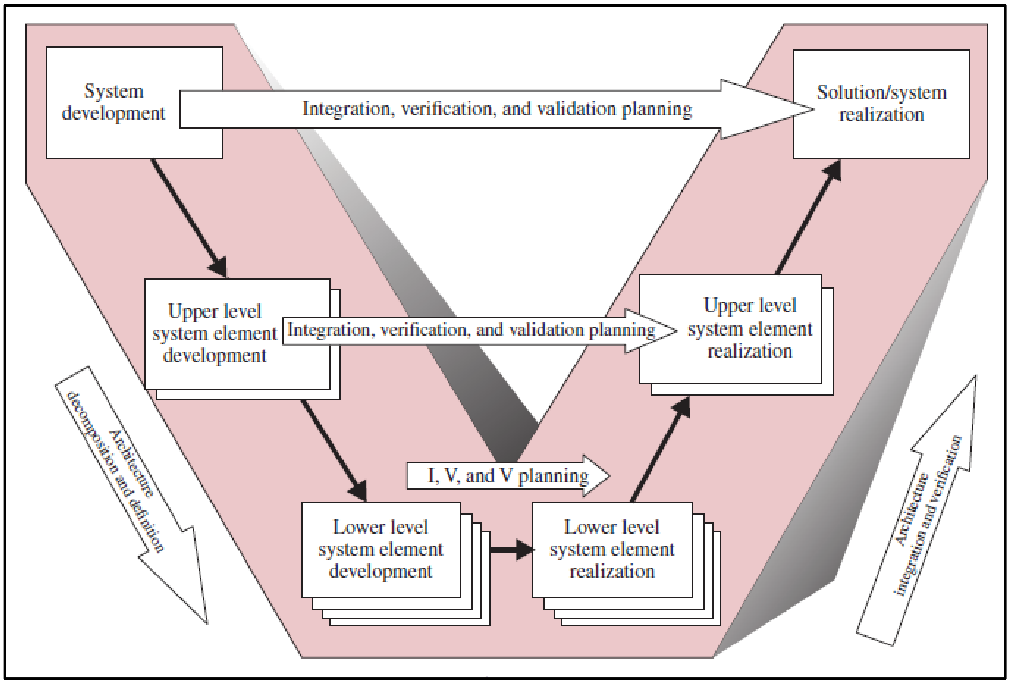


Figure 16. Vee Model. Source: Forsberg, Mooz, and Cotterman (2005).

The vee model represents a pre-specified and sequential process of project events that involves plans, specifications, and products that are baselined and configuration controlled. As the design matures through time from left to right, it progresses from the analysis and agreement of stakeholder requirements to the exploration of system concepts and definition of final system elements. Although the process is sequential, iterations down the left side within each phase help to establish feasibility, identify and quantify risks, enable continuous in-process validation and stakeholder discussion, analyze opportunities, and initiate alternate concept studies in more detail to determine the best approach. Iterations upwards, however, might be an indication of ill-defined requirements, premature advancement of the project, or lack of SME involvement.

(4) Discussion of the Vee Life Cycle Model as it Compares to SBD

Typical explanations of the vee model describe a point-based process that attempts to identify the best solution based on user requirements (which makes sense since the start of the vee model follows the waterfall process). Although the early application of involved technical and support disciplines is emphasized, it is not the over-arching culture of the process. Forsberg and Mooz (1991, 6) say no dedicated team is required to “ensure that appropriate expert advice and detailed assistance is applied to all areas of project risk,” which implies the various specialties are not specifically set out to explore tradeoffs from their own perspectives and then negotiate.

Most implementations of the vee model define objective and threshold values for design factors (which are actually sets) and rank various alternatives against those values to determine the best solution to advance. These ranges are not necessarily synthesized to reveal a complete design space, nor are they purposely refined throughout the design process to reduce the number of alternatives. Even though there are iterative considerations of stakeholder acceptance, risk, and feasibility, typical implementation of the vee model intends to select a single alternative as soon as possible with the least amount of information.

c. Interpersonal and Emergent

Although current DoD acquisition strategies tend to favor the design spiral for ships and vee model for other systems, agile methods are becoming popular for design efforts beyond software.

(1) Description of Agile Methods

Interpersonal and emergent life cycle process models came about in response to more closely integrated and interactive systems exhibiting traits of complexity. The trend towards rapid development in an accelerated pace of change environment prompted a desire to shift away from “heavyweight plans, specifications, and other documentation imposed by contractual inertia and maturity model compliance” (Boehm 2006, 19). Agile methods, such as extreme programming, dynamic systems development, scrum, and innovation-based processes, came about on the premise of the agile manifesto that values:

> Individuals and interactions over processes and tools; Working software [or products] over comprehensive documentation; Customer collaboration over contract negotiation; and Responding to change over following a plan. (“Agile Manifesto” 2001)

Agile methods generally use a "combination of customer collocation, short development increments, simple design … refactoring, and continuous integration to reduce the [cost of change]" (Boehm 2006, 19). They include divergent phases (discover and design) where the broadest possible insight is sought and convergent phases (develop and test) where insight is narrowed into a useable deliverable (see Figure 17).

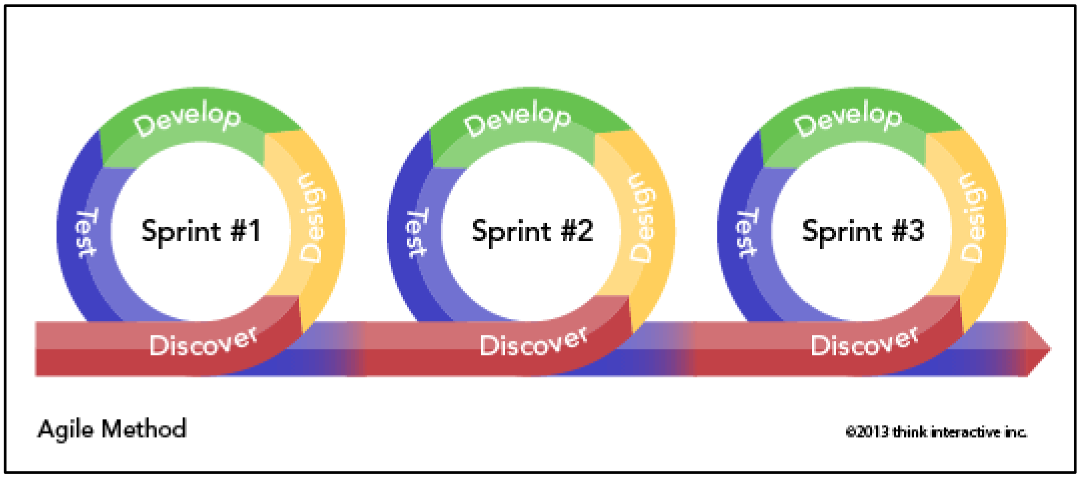


Figure 17. Agile Method. Source: Hallman (2013).

(2) Discussion of Agile Methods as They Compare to SBD

There are elements of agile methods that agree with SBD, such as customer collaboration, responding to change, and the use of tools; but others, like refactoring and minimal documentation, are a complete dichotomy (also see Chapter VII Section B.6 for a related discussion on future work). Agile methods are intended to reduce the cost of change over time, but data shows they do not for larger projects, which require more explicit plans, controls, and high-level architecture representations that counteract the original benefits (Boehm 2006). Furthermore, agile methods are "most workable on small projects with relatively low [risk] outcomes, highly capable personnel, rapidly changing requirements, and a culture [that thrives] on chaos versus order" (Boehm 2006, 19).

Although there are divergent and convergent phases in agile methods, they do not explore the entire design space or eliminate options. Instead, they generate tests (or prototypes, typically of high quality) prior to implementation, continuously evolve the design for onward development, and refactor or modify them to improve performance, increase maintainability, and make them more aesthetically pleasing (Black et al. 2009).

From a point of view specific to software (as agile methods usually are), the direction towards model-driven development (MDD) to capitalize on the integration of application and software domain models supports the broad notion of SBD to include collaboration, CE, and the independent design exploration by each specialty. The push to MDD also underscores the importance of integrating systems and software engineering, as evidenced by statistics showing more than half of software project failures are attributed to insufficient consideration of systems engineering-related activities, or systems engineers belatedly discover they need access to more software skills (Boehm 2006). Two common failure modes occur when systems engineering and teambuilding are not applied to agile scale-up efforts using teams of teams from the very beginning: it is hard to find team leaders who can mutually satisfy one team's preferences along with other teams' constraints; and quality requirements for the overall system are treated as an afterthought and accounted for in later increments, leading to the inability to compensate for early decisions regardless of refactoring attempts (Boehm 2006). This disconnect between software and systems engineering highlights how agile methods stray from SBD principles because they do not typically execute design across multiple specialties (or teams of teams), do not account for the intersection of preferences and constraints in order to achieve global optimization, and do not delay decisions with the intent of making more informed ones to avoid refactoring.

2. Systems Engineering

a. Description of Systems Engineering

From experience, when SBD is first introduced as a new concept, the tendency seems to be an adamant stance that systems engineering is superior to, or actually is, SBD. ISO/IEC/IEEE (2015) formally recognizes systems engineering as a discipline, and INCOSE (2015) defines systems engineering as a perspective (interdisciplinary approach), process (top-down iterative design and bottom-up realization process), and profession (discipline). The role of the systems engineer is to orchestrate the development of a solution through the various life cycle stages by assuring the proper involvement of SMEs, the pursuit of all advantageous opportunities, and the identification and mitigation of all significant risks (INCOSE 2015). Systems engineering tasks are usually focused on the beginning of the life cycle during concept and development, but it is an important part of all life cycle stages.

For the most part, the DoD acquisition life cycle in Figure 18 (referred to as defense acquisition program model in Department of Defense [2017]) is erroneously considered to be the "systems engineering process," but, in reality, systems engineering is simply applied to the DoD acquisition life cycle. Various life cycle models (e.g., waterfall, spiral, vee, etc.) are approaches to addressing certain stages of a particular life cycle. For example, the spiral model might be selected as the approach for executing the concept stage of the generic life cycle or the materiel solution analysis phase of the DoD acquisition life cycle. Systems engineering can be applied with every model during every stage for every life cycle.

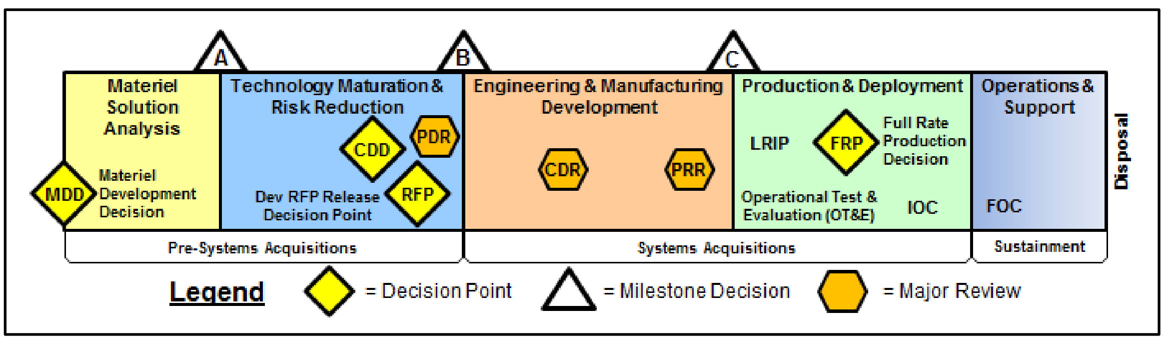


Figure 18. DoD Acquisition Life Cycle. Source: Department of Defense (2017).

b. Discussion of Systems Engineering as it Compares to SBD

Systems engineering is a discipline that is considered a process or approach when applied by systems engineers. Systems engineers and designers can, therefore, employ SBD when developing solutions. In other words, SBD is the methodology that can be used to create sets of alternatives and narrow down the options until one is selected.

There is also more to systems engineering than what is captured by SBD. Some of the activities performed or coordinated and managed through systems engineering of technical processes include: "business or mission analysis, stakeholder needs and requirements definition, system requirements definition, architecture definition, design definition, system analysis, implementation, integration, verification, transition, validation, operation, maintenance, and disposal" (ISO/IEC/IEEE 2015, 16). SBD, on the other hand, is ideally used for generating options and using data driven analysis, SME knowledge, and refined information to evaluate and eliminate infeasible and highly dominated alternatives to converge on a solution, which means it can be utilized during any activity that requires generating and comparing options, such as the definition of requirements and design and system analysis.

3. Design for "X" (DfX)

a. Description of DfX

DfX optimizes a specific aspect of the design (represented by X), which can be a certain area of focus, particular life phase (manufacture, assembly, disposal, etc.), or virtue that the product should possess (design factor, cost, quality, environmental impact, etc.) (Holt and Barnes 2010). It stems from design for assembly (DFA) efforts, expanded into design for manufacture (DFM), and is most recently focused on total life cycle considerations (design for life cycle [DFLC]), environmental concerns (design for environment [DFE]), disassembly, and recyclability (Kuo, Huang, and Zhang 2001).

b. Discussion of DfX as it Compares to SBD

The most outstanding difference between SBD and DfX is that SBD creates a holistic view of product development, whereas DfX is applied in isolation and is reductionist in nature. SBD encompasses all perspectives, while DfX focuses on one specific virtue or life phase at the exclusion of others. Each DfX technique is developed individually and improves the design from one perspective and does not relate decisions to the product on a system-level basis. It does, however, have the elements of generating awareness about important factors to consider and evaluating designs from a certain point of view, but it lacks the extension into multiple specialty areas.

Recent studies on DfX (Holt and Barnes 2010) recognize the difficulty in comparing designs across various DfX techniques simultaneously and that research efforts are limited regarding how to integrate them. The need for tradeoffs is acknowledged as is the concept of preferences pointing to the most desirable features, properties to include, and which have the highest priority. Although the consideration of multiple DfX techniques is a step in the right direction, it is not an established part of DfX like it is with SBD. There is also no mention of defining these preferences in the beginning of the design process nor in the form of sets. Furthermore, the entire design space cannot be defined without concurrently accounting for each specialty, which means designers seek to optimize a specific X instead of eliminating infeasible and highly dominated regions.

Holt and Barnes (2010) discuss how DfX techniques aim to improve the definition (generating design alternatives) and evaluation (identifying which design alternatives are acceptable or most promising) activities of the design process through qualitative guidelines, metrics, and feasibility checks, and that various design tools and software support this process by attempting to generate better candidate design solutions, provide more accurate evaluations, or incorporate feedback from evaluation to produce new and improved design alternatives. While DfX techniques help determine the expected behavior of the various design solutions in relation to a certain performance characteristic, the evaluation is limited to identifying the most promising design from those that have already been generated. Poor designs cannot be improved without using findings from evaluation as a spur to generating new design alternatives, which implies exploring modifications or improvements before deciding which design to move forward with. A "cycle of definition, evaluation, and redefinition" (Holt and Barnes 2010, 129) is created that makes the current implementation of DfX a PBD approach.

4. Design-Build-Test (DBT)

a. Description of DBT

Wheelwright and Clark (1994) recognize that problem solving during development is a learning process, and several iterations of comprehending the problem and understanding the solution alternatives are necessary before converging on a final design and detailed specifications. DBT is an iterative process where a single cycle provides information to the next iteration of the cycle over the course of three unique phases: design, build, and test (see Figure 19). The problem is framed and candidate solutions are generated in the design phase; prototypes or working models – either physical or computer-based – of the design alternatives are constructed during the build phase; and appropriate tests are conducted in the test phase. The prototypes and models serve as a focal point for studying issues, exploring customer requirements and responses, and evolving the design, while also forcing communication and feedback, testing, and increased specification.

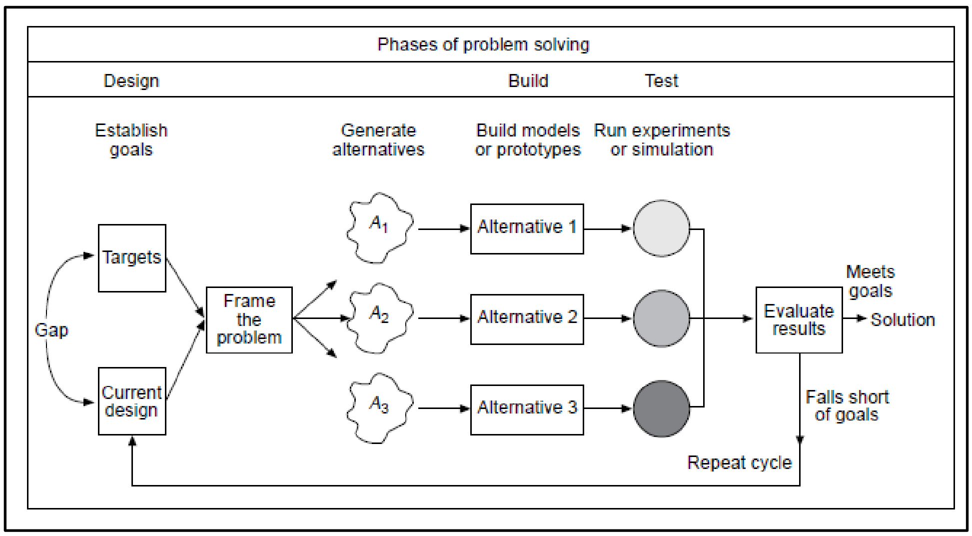


Figure 19. Design-Build-Test Method. Source: Wheelwright and Clark (1994).

b. Discussion of DBT as it Compares to SBD

DBT is a convergent method that suggests carrying multiple design alternatives for a protracted period of time, as is SBD. Both methods generate solutions to gain insight about how different design parameters impact customer attributes and user requirements, however, SBD goes one step further by working with the options in a set-based manner that allows each specialty to explore the design space independently.

Wheelwright and Clark (1994, 39) acknowledge the importance of "integrated problem solving" where upstream and downstream groups engage in a rich, open, and trusting pattern of communication linked in time; a mode where the downstream groups "not only participate in ongoing dialogue with their upstream counterparts, but use that information and insight to get a flying start on their own work." This type of mindset in DBT has definitely graduated from *over-the-wall* practices, but does not necessarily emphasize the consideration of all design aspects from the perspective of each individual specialty that is inherent with SBD. Each specialty is encouraged to discuss with other specialties and share ideas, but there is no strong element of considering each option from its own perspective.

Both DBT and SBD have strategic times when prototypes and models are built that reinforce the fundamental nature of problem solving as a learning process. Also, the detail and fidelity of these builds improves as the design progresses (if necessary). The test phase of DBT is very similar to the periodic test phases of SBD where the general intent in both cases is to gain a better understanding of the problem, evaluate design solutions, increase knowledge, and acquire new information. In DBT, all of the new learnings become the basis for a new DBT cycle and support the determination of whether or not to consider new design options or pursue the development of those already identified. In stark contrast, SBD uses the new learnings as justification for eliminating certain options or entire regions of the design space.

Although DBT is a convergent method that considers sets of alternatives and emphasizes the importance of "face-to-face communication, direct observation, interaction with physical prototypes, and computer-based representations" (Wheelright and Clark 2010, 39), it does not stress the independent evaluation of design alternatives by each specialty, the deliberate elimination of solutions for corroborated reasons, or the consideration of the entire design space.

5. Method of Controlled Convergence (MCC)

a. Description of MCC

Pugh (1981) describes a concept selection process for controlled convergence to the best possible design solution based on eliminating (or minimizing) conceptual vulnerability (i.e., design concepts that are considered weak). Referred to as MCC, it starts with defining the selection criteria against which all concepts are evaluated and generates a large number of design alternatives for comparison. A concept comparison and evaluation matrix is established where each concept is ranked with respect to a datum (or baseline) concept in terms of better than, worse than, or same as (see Figure 20).

Criteria \ Concept	1	2	3	4	5	6	7	8	9
A	+	-	+	-	+	-	DATUM	-	+
B	+	S	+	S	-	-		+	-
C	-	+	-	-	S	S		+	S
D	-	+	+	-	S	+		S	-
E	+	-	+	-	S	+		S	+
F	-	-	S	+	+	-		+	-
Sum +	3	2	4	1	2	2		3	2
Sum -	3	3	1	4	1	3		1	3

Figure 20. MCC Matrix. Source: Pugh (1981).

The highest ranking and strongest concepts are retained and all others are either eliminated or modified and combined with the reduced set, which might also include the addition of new alternatives. The process of expanding the set with new options and contracting it with the removal of weak concepts repeats as the sets are narrowed further until only one option remains (see Figure 21).

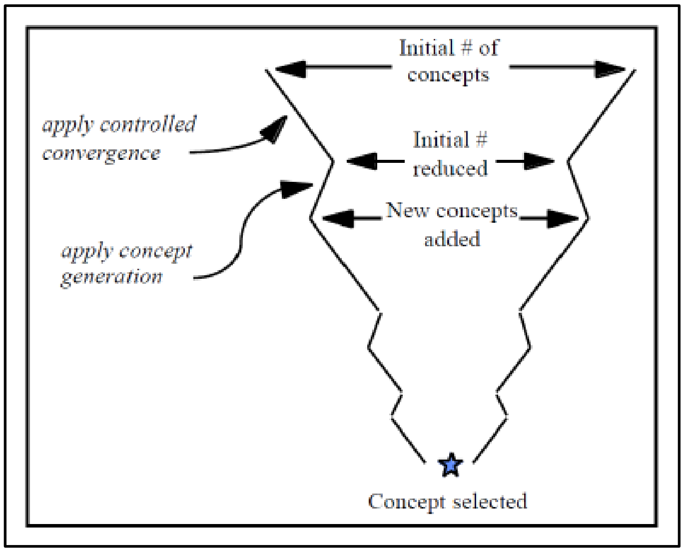


Figure 21. MCC. Source: Bernstein (2018).

b. *Discussion of MCC as it Compares to SBD*

MCC has the potential to consider sets of design factors by specifying the evaluation criteria in ranges of values, however, it never explicitly synthesizes all of these sets to create a total design space nor does it necessarily intersect them all to produce feasibility. In other words, MCC only applies to individual alternatives and not the design space as whole, which means decisions can only be made with respect to individual alternatives and not to grander regions of the design space.

MCC may surpass SBD in that it allows for the inclusion of additional design alternatives and accepts the reality of design where it is hardly possible to identify every design option in the beginning (Bernstein 2018). By providing a means of introducing new concepts for consideration and evaluation, MCC violates the SBD principle of remaining within a set. SBD stipulates design alternatives cannot be revisited after they are eliminated in order to build trust within the various specialties so they can proceed with development and pursue certain solutions with confidence that these efforts will not be negated by re-opening options that were already deemed infeasible or highly dominated. Arguably, SBD does allow for new options to be presented, especially when advances in technology merit new alternatives or changes in requirements are incurred. Since SBD systematically eliminates regions of the design space and documents the rationale for all decisions, it is amenable to new alternatives as is MCC.

MCC is very similar to SBD in that it has the capacity to consider sets, does not restrict individual specialties from considering concepts independently, increases detail as solutions are narrowed, and is convergent, however, it does not consider the design space in its entirety, which is a key element of SBD.

6. Decision Making and Optimization Methods

Most engineering methods include an element or phase dedicated to the selection of a design solution to pursue. While engineering methods encompass an entire process that extends beyond simply selecting an alternative, decision making and optimization methods isolate this function and serve a purpose of choosing the best solution without elaborating on the successive steps in the design process. The way these decision making and optimization methods are currently implemented does not exhibit SBT or SBD methodologies as is described through the following examples.

a. Multiple Criteria Decision Making (MCDM)

MCDM comprises a broad set of methodologies for decision making in design problems where there are several conflicting criteria. The overall intent of MCDM is to select the best alternative, individually or from a set, that is the most attractive when considered over all criteria. MCDM is broadly divided into two categories: Multiple Attribute Decision Making (MADM) and Multiple Objective Decision Making (MODM).

(1) MADM

MADM is used for prioritizing and sorting alternatives by rank according to several criteria. It typically deals with discrete decision spaces involving a finite number of predetermined alternatives and attributes (or goals or decision criteria) that are each assigned a weight of relative importance reflective of its impact on the decision (Bhushan and Rai 2004). Some common MADM methods are included in Table 4.

Table 4. Examples of MCDM Methods. Source: Velasquez and Hester (2013).

Method	Advantages	Disadvantages	Areas of Application
Multi-Attribute Utility Theory (MAUT)	Takes uncertainty into account; can incorporate preferences.	Needs a lot of input; preferences need to be precise.	Economics, finance, actuarial, water management, agriculture.
Analytic Hierarchy Process (AHP)	Easy to use; scalable; hierarchy structure can easily adjust to fit many sized problems; not data intensive.	Problems due to interdependence between criteria and alternatives; can lead to inconsistencies between judgement and ranking criteria; rank reversal.	Performance-type problems, resource management, corporate policy and strategy, public policy, political strategy, and planning.
Case-Based Reasoning (CBR)	Not data intensive; requires little maintenance; can improve over time; can adapt to changes in environment	Sensitive to inconsistent data; requires many cases.	Businesses, vehicle insurance, medicine, and engineering design.
Data Envelopment Analysis (DEA)	Capable of handling multiple inputs and outputs; efficiency can be analyzed and quantified.	Does not deal with imprecise data; assumes that all input and output are exactly known.	Economics, medicine, utilities, road safety, agriculture, retail, and business problems.
Fuzzy Set Theory	Allows for imprecise input; takes into account insufficient information.	Difficult to develop; can require numerous simulations before use.	Engineering, economics, environmental, social, medical, and management.
Simple Multi-Attribute Rating Technique (SMART)	Simple; allows for any type of weight assignment technique; less effort by decision makers.	Procedure may not be convenient considering the framework.	Environmental, construction, transportation and logistics, military, manufacturing and assembly problems.
Goal Programming (GP)	Capable of handling large-scale problems; can produce infinite alternatives.	Its ability to weight coefficients; typically needs to be used in combination with other MCDM methods to weight coefficients.	Production planning, scheduling, health care, portfolio selection, distribution systems, energy planning, water reservoir management, scheduling, wildlife management.
ELECTRE	Takes uncertainty and vagueness into account.	Its process and outcome can be difficult to explain in layman's terms; outranking causes the strengths and weaknesses of the alternatives to not be directly identified.	Energy, economics, environmental, water management, and transportation problems.
PROMETHEE	Easy to use; does not require assumption that criteria are proportionate.	Does not provide a clear method by which to assign weights.	Environmental, hydrology, water management, business and finance, chemistry, logistics and transportation, manufacturing and assembly, energy, agriculture.
Simple Additive Weighting (SAW)	Ability to compensate among criteria; intuitive to decision makers; calculation is simple and does not require complex computer programs.	Estimates revealed do not always reflect the real situation; result obtained may not be logical.	Water management, business, and financial management.
Technique for Order Preferences by Similarity to Ideal Solutions (TOPSIS)	Has a simple process; easy to use and program; the number of steps remains the same regardless of the number of attributes.	Its use of Euclidean Distance does not consider the correlation of attributes; difficult to weight and keep consistency of judgement.	Supply chain management and logistics, engineering, manufacturing systems, business an marketing, environmental, human resources, and water resources management.

Quality Function Deployment (QFD) and Analytic Hierarchy Process (AHP) are two examples of MADM methods that are frequently used in design decision making and optimization problems. QFD was born under the umbrella of Total Quality Control (TQC) as a method for converting user demands into quality characteristics and deliverable actions prior to production (Akao 2004). Although QFD has been modernized to use many small tools for speed and efficiency, the classic core is the House of Quality (HOQ) matrix that captures the voice of the customer (VOC) to gain an understanding of the true requirements and meaning of value, determine which features to include and what level of performance to deliver, benchmark the competition, and rank and compare alternatives (see Figure 22). The customer requirements and associated importance values are typically derived through pairwise comparison and then followed by the ranking of each alternative against them.

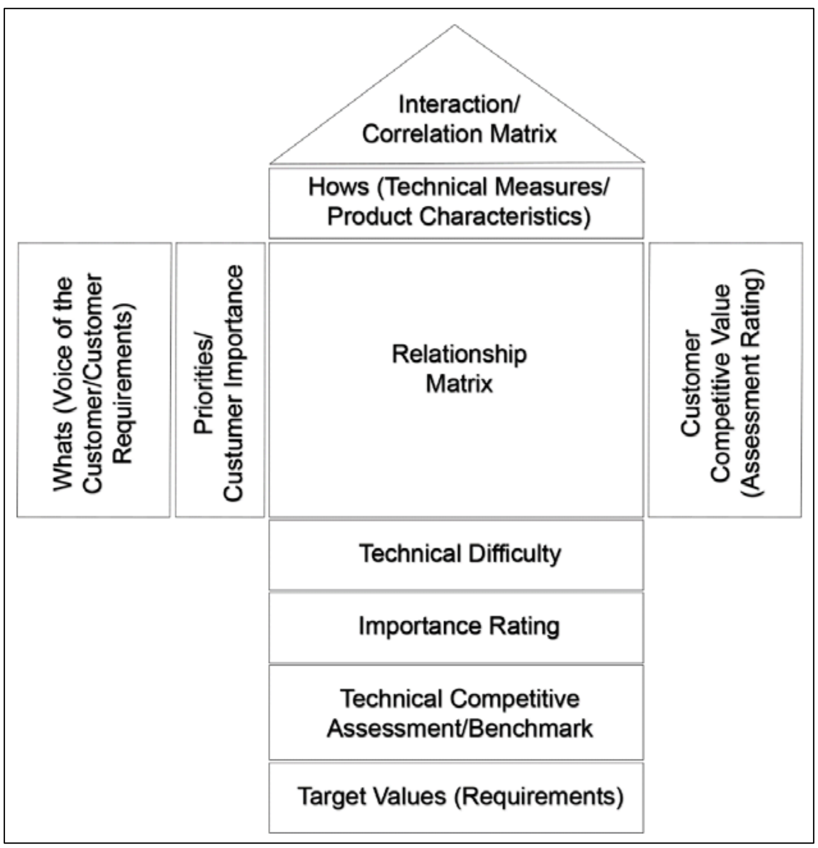


Figure 22. QFD HOQ. Adapted from ASQ (2018).

AHP is a systematic approach to decomposing a problem into a hierarchical structure of goals, criteria, sub-criteria, and alternatives that can be more easily understood and evaluated by technical specialties and SMEs (see Figure 23) (Bhushan and Rai 2004).

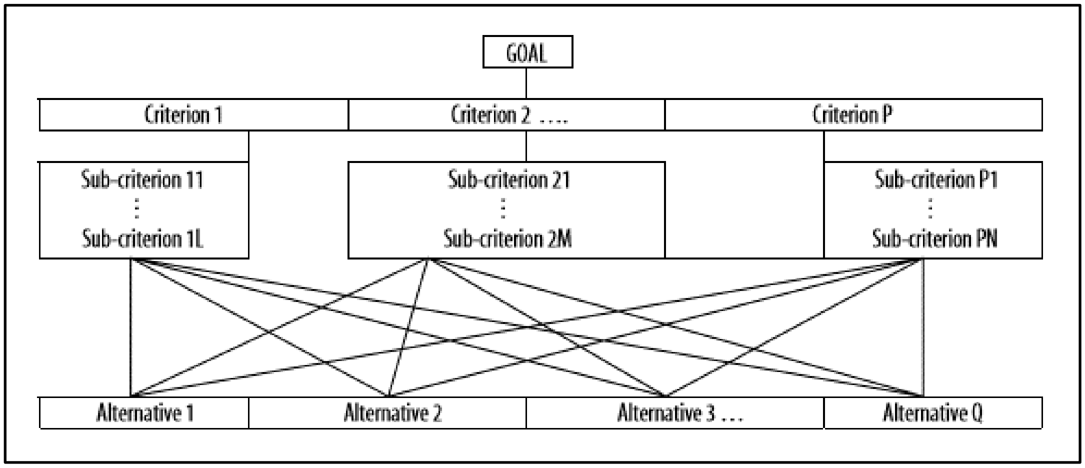


Figure 23. AHP Decomposition. Source: Bhushan and Rai (2004).

The process starts by deriving weights or priority rankings for each criterion based on pairwise comparisons. The local priority (or preference) of each alternative is then derived with respect to each criterion separately, also through pairwise comparison. Finally, all of the local priorities are combined for each alternative to account for each criterion and achieve a global score and consequent ranking.

(2) Discussion of MADM Methods as They Compare to SBD

MADM methods are very similar to the initial setup of the MCC engineering method in that selection criteria are defined for which design alternatives are compared against. As with SBD (although limited to only a portion of the design space with MADM), the best option that emerges through MADM methods will presumably achieve the most suitable tradeoff among the different criteria and present a global optimum as opposed to optimizing a single criterion. SMEs and individual specialties can also provide data and insight corresponding to the pairwise comparisons of alternatives and independently judge the importance of various alternatives with respect to a common criterion based on experience. As with MCC, MADM methods only apply to individual alternatives because —although they have the potential to consider sets of design factors—they are never explicitly synthesized to create a total design space. The highest ranking or most appealing option is selected after an evaluation of comparison versus converging on a solution by elimination, which rules MADM methods as point-based.

(3) MODM

MODM techniques, such as mathematical programming problems with multiple objective functions, are used when the decision space is vector-based or continuous. Decomposition-based optimization is formed when MODM problems are transformed into multiple single-objective optimization problems by decomposing the system into a set of reduced order subsystems; this is typical for managing the complexity encountered when attempting to study the system as whole. Often times, the decomposition is performed based on the analysis capabilities of the engineers along multidisciplinary lines; hence, they are referred to as multidisciplinary optimization (MDO) problems. Whitcomb and Hernandez (2017) cites further classifications of MDO problems as either hierarchical or non-hierarchical based on the coupling between disciplines resulting from the decomposition and also identifies the all-at-once (AAO) or multi-level solution approaches.

(4) Discussion of MODM Methods as They Compare to SBD

Although there are some examples of analytical formulations for SBD available in the literature (Hannapel, Vlahopoulos, and Singer 2012), in general, SBD has not been mathematically defined for integrated computational application in ways that are comparable to decomposition-based design methods (Ghosh and Seering 2014); nor does it need to be. SBD utilizes teams of people in a decomposed system sense by making each specialty responsible for certain subsystem design functions. Each specialty can construct mathematical representations of its own design space and communicate important information to a system-level adjudicator of design variables that leads the process to convergence. While MODM methods typically rely upon integrated computational tools to identify optimal solutions, SBD has no overall requirement to integrate the subsystem models into a single computational environment for the purpose of tracking design space exploration to a final solution (Singer et al. 2017).

MODM methods are also implemented computationally using *a priori* defined objective functions, constraints, and design variables that are coordinated, assigned, or adjudicated until a consistent set is found at the final converged solution (Whitcomb and Hernandez 2017); they consider multiple points within a design space and use algorithms to guide the convergence to a single solution. SBD uses sets, or regions, of possibilities within the total design space and elimination for convergence. If the principles of SBD were employed in MODM, the design space would change through the optimization process, and the algorithms would produce results in sets as opposed to pointed values. For example, in the case of the MDO application in Hannapel, Vlahopoulos, and Singer (2012, 1), "the algorithm returns the optimal choice for the reduced design space instead of a single, specific value for each design variable."

7. DOE and Response Surface Methods (RSM)

a. Description of DOE and RSM

The field of statistical DOE was largely developed in the 1920s through Fisher's work related to agriculture (Fisher 1971). Classical DOE refers to a structured and organized way of measuring and evaluating the effects of input variables (factors) on response variables (responses) through the principles of randomization, replication, blocking, orthogonality, and factorial experimentation. By executing controlled tests with purposeful changes, statistically valid inferences and definitive conclusions about the behavior of a system can be made with the minimum use of resources (Telford 2007). Enabling optimization and gaining insight into the behavior of complex engineering systems are major areas of DOE utilization.

Although DOE has its roots in physical experiments, physical experimentation is not always possible or cost effective and can be prohibitively time consuming. Computer experiments are a typical alternative because of the advances in computing power combined with the fact many physical systems can be described by mathematical equations with numerical solutions (e.g., Navier-Stokes). Computer experiments use deterministic computer models, such as finite element analysis (FEA) or computational fluid dynamics (CFD), where the output is not subject to random variations; which make blocking, randomization, and replication irrelevant and yield identical outputs for each run with the same inputs. The time it takes to evaluate a single run of some computer models can be extensive or costly, especially in the case of high-fidelity models or for exploring large input spaces when little is known about the system *a priori*. Since direct optimization is often infeasible, metamodels (surrogate models, models of models) can be constructed to replace the complex and expensive computer models and allow for what-if analyses.

Metamodels are generated by sampling a set of design points from the experimental space of synthesized input variables and then fitting a model to the data observed in the response; objectives include response prediction, exploring numerous factors simultaneously, and identifying significant input variables and relationships. Although there is no standard and little is known about which linear, nonlinear, parametric, nonparametric, or semiparametric model will fit the data beforehand, several statistical methods and multivariate analysis techniques are used to build metamodels, including: parametric polynomial response surface approximation, splines, neural networks, spatial correlation models (like kriging and Gaussian processes), frequency domain methods, additive models, radial basis function, and support vector machines (MacCalman, Vieira, and Lucas 2017; Viana 2016).

Response surface approximations use low-order polynomials (first or second-order) to determine curvature, detect interactions among factors, optimize the process, and establish operating conditions (Montgomery 2001) after the important factors have been revealed through a screening or factorial experiment. The second-order model is the most favored because it includes main effect, quadratic, and two-way interaction terms capable of distinguishing the greatest influential factors and revealing their interdependencies and nonlinearities towards the response in addition to global and local maximums and minimums (MacCalman, Vieira, and Lucas 2017). Response surfaces are typically used to optimize a response that is influenced by several input variables and can be visualized graphically as contour plots or in three dimensions. Response surface curves define the system design space and offer opportunities for selecting a solution based on Pareto (or other) optimality.

b. Discussion of DOE and RSM as They Compare to SBD

DOE by itself is not considered SBD, but this dissertation is applying DOE in a set-based way to generate the design solutions, which are represented as a point cloud that covers the design space instead of response surfaces and curve fits with RSM.

Typically, central composite designs (CCD) and Box-Behnken designs (BBH) are used to generate the points that bound the design space, and then RSM is used to approximate these bounds as a second-order curve fit (Box and Draper 1987). Although not universal, RSM commonly uses outside points and one interior point (two extremes and a midpoint) and a second-order equation; the experimenter does not see a point cloud, nor do they see interior points like they do with SBD. SBD involves the points without the curve fits. The DOE part of SBD for generating the points in this dissertation is performed using LHS to populate points inside the design space.

Design decisions are based on the actual points in SBD while covering the entire design space, as opposed to RSM where they are limited to the Pareto limits and do not get the full understanding of the design space. RSM uses a second-order curve fit because it is guaranteed to be smooth and convex, which offers a global optimum when it is optimized. The disadvantage is there could be some non-linear points outside the response surface that are better; this global optimum could be used, or a SBD approach could be applied based on regions of feasibility and viability to further explore the design space and perhaps identify better solutions.

Both RSM and SBD explore the same design space, but in different ways. The power of SBD is in gaining and applying new information for decision making. If designers get no new information, then RSM and SBD are essentially the same, but the real advantage of SBD is engaging the different specialty domains to get better exploration of the design space to gain more information about it before making decisions. RSM gives visibility of the tradeoffs at the highest level, but does not reveal the ramifications of each decision throughout the design. SBD makes the data visible during the design process and allows each specialty to study it in detail, which makes for a lot of learning and leads to an improved design and better knowledge about the design.

I. OPPORTUNE CIRCUMSTANCES FOR SBD

1. Complexity

It is evident through the literature (Norman and Kuras 2006; White 2016) that traditional systems engineering (TSE) processes are limited when it comes to addressing system of systems (SoS) and accounting for complexity. McKenney and Singer (2014) question how effective current and previous design methods will be with increasing complexity and rapidly changing environments, and Singer, Doerry, and Buckley (2009) assert these methods often fail due to complexities typical of design projects on a grander scale. Sobek, Ward, and Liker (1999, 71) emphasize “that ‘flexibility’ [during] development is an important factor to success, particularly in unpredictable and rapidly changing environments” (which are certainly associated with complex systems).

Kennedy, Sobek, and Kennedy (2014) suggest the TSE approach of narrowing and fixing a design early needs to be shifted in order to reduce rework and produce complex systems more efficiently. Augmenting TSE with SBD methodologies should be considered for complex systems to offer the flexibility, collective problem solving, and later refinement of specifications necessary for dealing with the multipurpose and distinct solution space, ambiguous boundaries, and evolving requirements characteristic of these systems.

2. Interdependency and Coupling

The level of interdependency (or coupling) between components is important when considering the implementation of SBD methodologies. Case studies in Liker et al. (1996) suggest a proportional relationship between the interdependency and degree of SBD. Tighter coupling between components implies changes to one component have higher impact on other components and more rapid and pervasive propagation through the system. Higher interdependence means the need for effective communication and coordinated decisions is increasingly critical and difficult, which suggests SBD methodologies are better suited for these types of problems.

3. Other Factors

Many factors influence the effectiveness of SBD practices as it structures how design tradeoffs are negotiated with respect to uncertain technology and requirements. SBD will be most useful when implemented in design projects with attributes of:

- A large number of design variables;
- [Tight coupling among] design variables;
- Conflicting requirements;
- [Flexibility in requirements] and many possible design tradeoffs; or
- Unknown technologies and [design problems that are not well understood]. (McKenney and Singer 2014, 54)

SBD is not the proper method if a product is well understood, has requirements for specific technologies, when tradeoffs and optimization are dependent on only one or two dimensions, or when design decisions can be made independently with little need for coordination (Liker et al. 1996); a traditional PBD approach is reasonable in these instances (McKenney and Singer 2014). Situations where decisions require a precise definition of the design and detailed engineering analysis may also be poorly suited for SBD (Specking et al. 2017).

Programmatic factors, such as cost and schedule, and other relative driving factors, such as analysis tools and prototyping, are also important considerations when determining if SBD can be employed effectively. SBD advocates for the incorporation of multiple prototypes into the program baseline to accelerate understanding, expand the knowledge base, and reduce technical risk. Prototyping increases the research and development costs and should be budgeted for to facilitate the set reduction process. Ideally, prototyping is synergistic with the design analysis tools, where the tools are used to generate low-fidelity prototypes to create and narrow the design space. Funding to allocate, analyze, identify, and employ SBD enabling tools is necessary to promote tradeoffs and design space exploration.

4. Development Stage

An important consideration for deciding whether SBD is appropriate or not is where the system is at in its development along the acquisition life cycle. So far, the best use of SBD is during the initial phases of design where the tradespace is still available (e.g., PDR) and the focus is on grasping requirements versus completing detailed design (e.g., CDR) (Chan et al. 2016; Singer et al. 2017). Gaining information and insight and learning about and understanding the problem are primary objectives of the early stages of design that lead to the definition of reasonable and obtainable requirements for justifying design decisions. A characteristic strength of SBD is the capacity to gauge the nature of a design problem, understand it in terms of potential requirements, and explore actual solution alternatives; the unstable, flexible, and imprecise description of the information and requirements during early design stages is what enables and maximizes the design space exploration that is so integral to SBD. By later stages, the design is largely fixed and committed to, meaning the number of feasible alternatives has already been reduced before the benefits of SBD were leveraged.

5. Defense Acquisition

a. Factors

Several factors affect the successful application of SBD in defense acquisition, including the (Chan et al. 2016, 59):

- Applicable defense acquisition directives, instructions, and policies;
- Type of product (or system) being developed, such as hardware or software dominant, and the timeline for design;
- Defense acquisition life cycle that will be used (accelerated acquisition, hardware or software intensive, etc.);

- Engineering modeling tools available to the program, such as Feasibility Assessment for Cost and Technology (FACT), Leading Edge Architecture for Prototyping Systems (LEAPS), Advanced Ship and Submarine Evaluation Tool (ASSET), etc.;
- Budgeting and availability of tools to support SBD execution within the program; and
- Programmatic factors, such as the schedule impact of delaying decisions, culturally shifting the design approach, and communicating in sets.

b. Department of Navy (DoN) Two-Pass/Six-Gate Process

"SBD can be accommodated within the existing instructions without revision" (Chan et al. 2016, 97). The current DoN two-pass/six-gate review process establishes "a disciplined and integrated process for requirements and acquisition decision-making" (Department of Navy 2011, 1-52) that is designed to meet the milestone goals of the DoD acquisition life cycle in Department of Defense (2017) (see Figure 24). SBD can be implemented into the two-pass/six-gate process by tailoring the gate entrance and exit criteria to encourage SBD concepts.

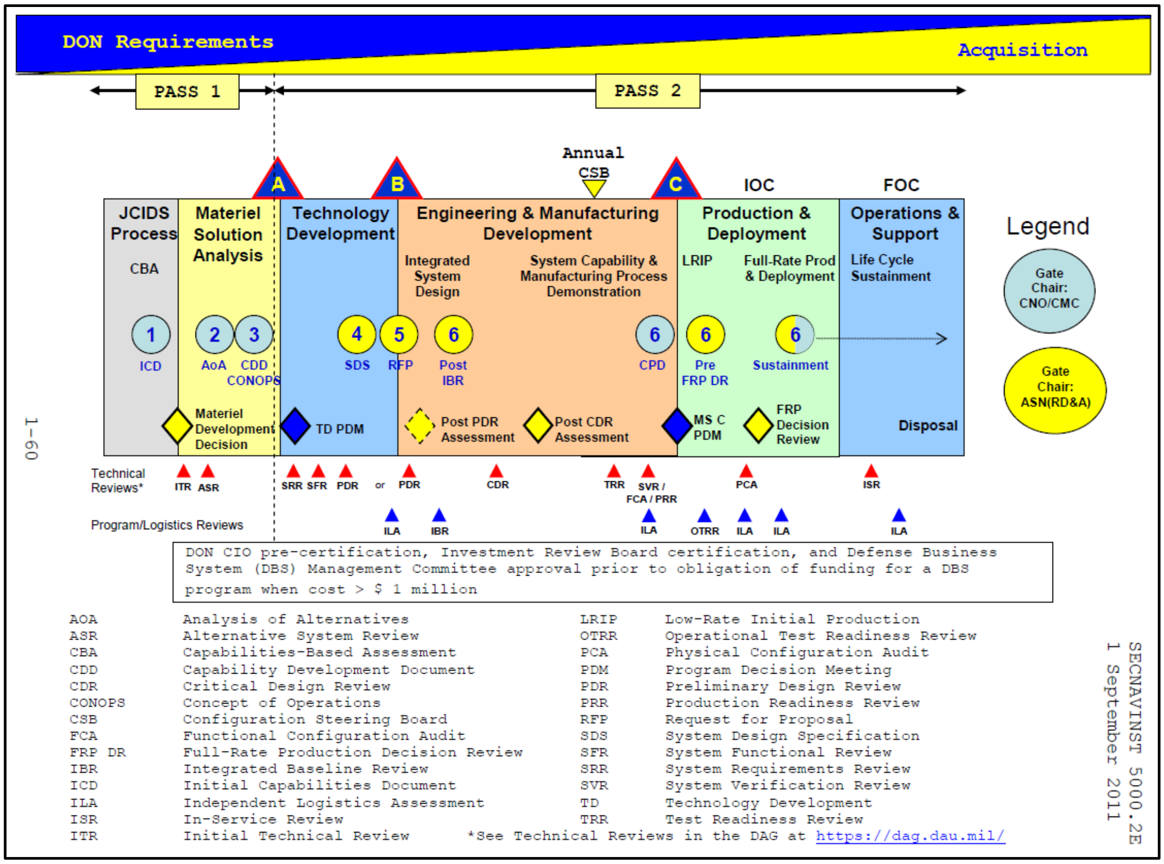


Figure 24. Two-Pass/Six-Gate Process. Source: Department of the Navy (2011).

Using lessons learned from DoD instances of SBD and two theoretical scenarios, Chan et al. (2016) provides guidelines for tailoring the DoD two-pass/six-gate process to accept SBD. One of the most prominent is the identification of system specifications in a Functional Design Document (FDD) allocated to subsystems in Functional Requirements Documents (FRDs), which essentially provides requirements analogous to those achieved by combining the System Requirements Review (SRR) and System Functional Review (SFR), but in a set-based manner. Other suggestions extracted from Chan et al. (2016) include:

- Document the desired use of SBD in the Service review of the Analysis of Alternatives (AoA) Guidance;
- Begin applying SBD after the AoA Guidance is approved at Gate 1;

- Draft the Systems Engineering Plan (SEP) prior to Gate 2 in order to provide a basis for informing stakeholders about the intent to execute SBD;
- Identify sets of preferred capability concepts instead of a preferred alternative at the Alternative System Review (ASR) for Gate 2;
- Also focus on the sets of operational requirements with corresponding performance parameters and relative importance to facilitate tradeoff curve development and evaluate sets of possible system configurations at Gate 2;
- Refine the assumptions, requirements, and constraints for each subsystem (or specialty) and increase the maturity of the FRDs to reduce the specialty tradespace at Gate 3;
- Intersect the subsystem requirements (FRDs) and trace the various configurations to system-level performance (FDD);
- Reduce the set of system configuration alternatives (FDD) to better solidify the design and map the requirements back to each subsystem (FRDs) to complete the System Design Specification (SDS) for Gate 4, i.e., replace the SDS development plan with the FDD;
- Combine the SRR and SFR to select the preferred subsystem configuration;
- Mature the FRDs as the final subsystem solutions emerge and approve them at Preliminary Design review (PDR);
- After PDR completion and allocated baseline (ABL) approval, explore sets of possible component configurations using SBD principles;

- Mandate the use of SBD practices in the request for proposals (RFP) at Gate 5 if contracting out the detailed design; otherwise, align Gate 5 with Milestone B if maintaining a government-led detailed design; and

- Between Gate 5 and Gate 6, perform more tradespace exploration and reduce the set of acceptable component configurations until the product baseline (PBL) is approved at Critical Design Review (CDR).

c. DoD Acquisition Program Models

Of the six defense acquisition program models (Department of Defense 2017), Chan et al. (2016) demonstrate that SBD complements both the hardware intensive and accelerated acquisition program models and explains how its use is limited for software intensive systems. Singer, Doerry, and Buckley (2009) anticipate SBD will provide the greatest benefit to the DoN between the Materiel Development Decision (MDD) and Milestone A. The SSC program utilized SBD during preliminary design (from Milestone A to Gate 4) (Mebane et al. 2011). The ACV and Small Surface Combatant utilized SBD for AoA (from MDD to Gate 2) (Garner et al. 2015; Burrow et al. 2014). The LDUUV plans to use SBD up to Gate 4 (Chan et al. 2016). Chan et al. describe how SBD can be utilized up to CDR prior to Gate 6, but cautions that conducting SBD and leaving the tradespace open post-CDR may be costly and inefficient, since the design has largely been defined.

d. Life Cycle Process Models

Most of the life cycle process models (waterfall, spiral, vee, etc.) can be tailored to accommodate SBD, since SBD is simply a methodology to execute the specific step (requirements definition, specification, evaluation, AoA, etc.) within a certain stage of the model (concept, development, etc.). Kennedy, Sobek, and Kennedy (2014, 291) propose "a set of concrete changes to the front end of the vee model … concentrated in the 'off-core' activities associated with the left side" (see Figure 25). Specific, set-based learning activities augment the traditional vee and move to converge by the initial stages of detailed design where specifications start to solidify. Off-core activities are connected horizontally, "indicating the ongoing evolution of the set-based knowledge at each level as the design decisions continue to converge at each level, influenced by the new knowledge learned from below and the new tradeoffs made from above" (Kennedy, Sobek, and Kennedy 2014, 292).

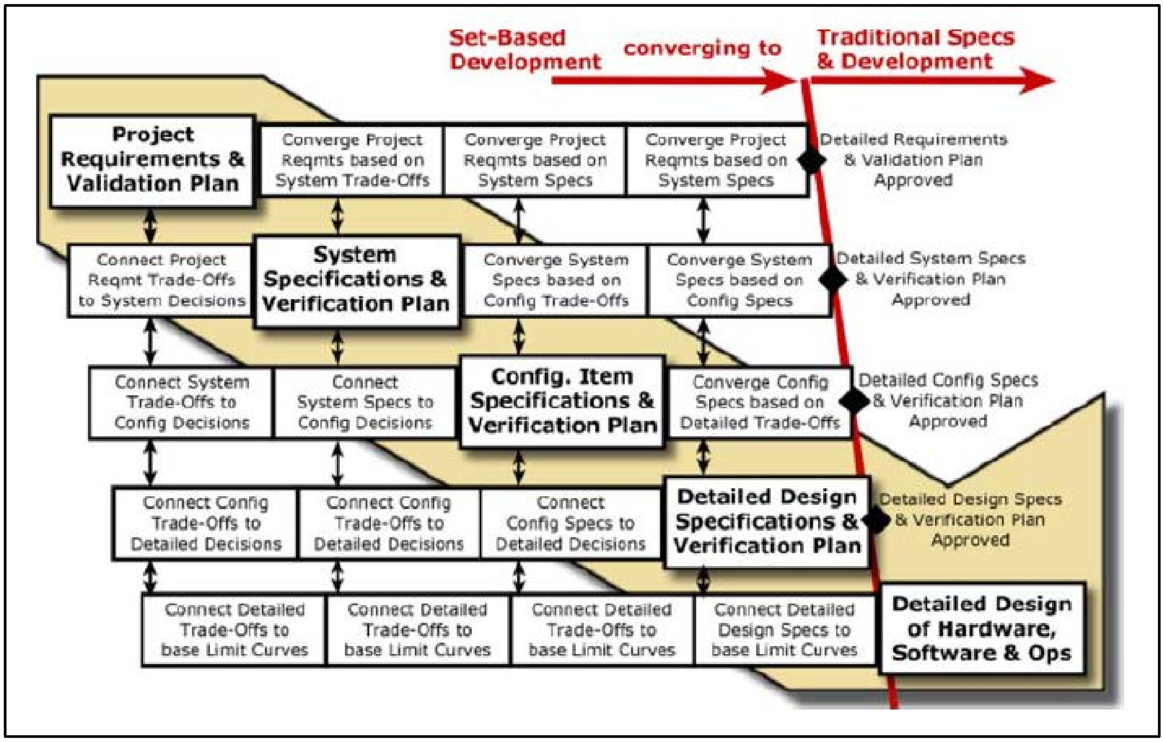


Figure 25. Vee Model with Set-Based Front End. Source: Kennedy, Sobek, and Kennedy (2014).

J. BENEFITS OF SBD

Some of the summarized benefits of SBD include giving designers the opportunity to: explore the entire design space; achieve more globally optimal solutions; maintain flexibility when handling uncertainty; and "develop in-depth knowledge about the technical problem and potential solution set, a risk-based understanding of what is feasible and unfeasible, and high confidence cost estimates based on technical feasibility and diversity of solutions" (Burrow et al. 2014, 15). Critical early-stage design decisions are based on data in SBD and it allows for: fixing requirements and specifying the design later on, which leads to greater exploration of solutions, ensures feasibility before commitment, and results in designs that meet and achieve customer expectations and satisfaction (Liker et al. 1996; Ward et al. 1995a); better ability to adapt to and accommodate changes during later stages of design (McKenney, Kemink, and Singer 2014); greater parallelism (Singer et al. 2017; Ward et al. 1995b); geographically dispersed teams; diverse teams; consideration of the design from the own perspective of each specialty; and reliable, effective, efficient (Ward et al. 1995a), rich, bilateral, and frequent communication (Sobek, Ward, and Liker 1999). SBD also promotes institutional learning and harmonious and trusting working relationships between specialties (Ward et al. 1995a; Ward et al. 1995b) and facilitates a flexible and robust knowledge structure (Singer et al. 2017).

1. Design Space Exploration

In SBD, decisions are made based on a good understanding and consideration of the entire design space and not just the analysis of one or two options (or points) from somewhere within it (Singer et al. 2017; Singer, Doerry, and Buckley 2009). Going through the SBD process and communicating about sets of ideas creates an opportunity to explore a far greater range of options than conventional PBD approaches, which facilitates the ability to obtain more robust and globally optimal solutions (Mebane 2011). Rather than limiting new design efforts based on existing systems or legacy efforts, SBD encourages the evaluation of many different concepts and accounts for possible alternatives beyond the Pareto front. SBD provides a structured approach that enables forming mental maps of the design space and rationally reducing it by narrowing sets and eliminating regions leading to more accurate and informed decisions (McKenney, Kemink, and Singer 2011; Sobek, Ward, and Liker 1999; Ward et al. 1995b).

2. Delaying Decisions

One of the key principles of SBD is deferring requirements and design decisions. Ward et al. (1995a, 204) believe "the earliest decisions about designs have the largest impact on the ultimate quality and cost but that such decisions also are made with the smallest amount of data." Figure 26 shows how making decisions as late as possible can be advantageous when considered with respect to three factors influencing design: "the evolution of a product's cost, management's ability to affect these costs, and the evolution of designers' knowledge about a design problem" (Bernstein 1998, 41).

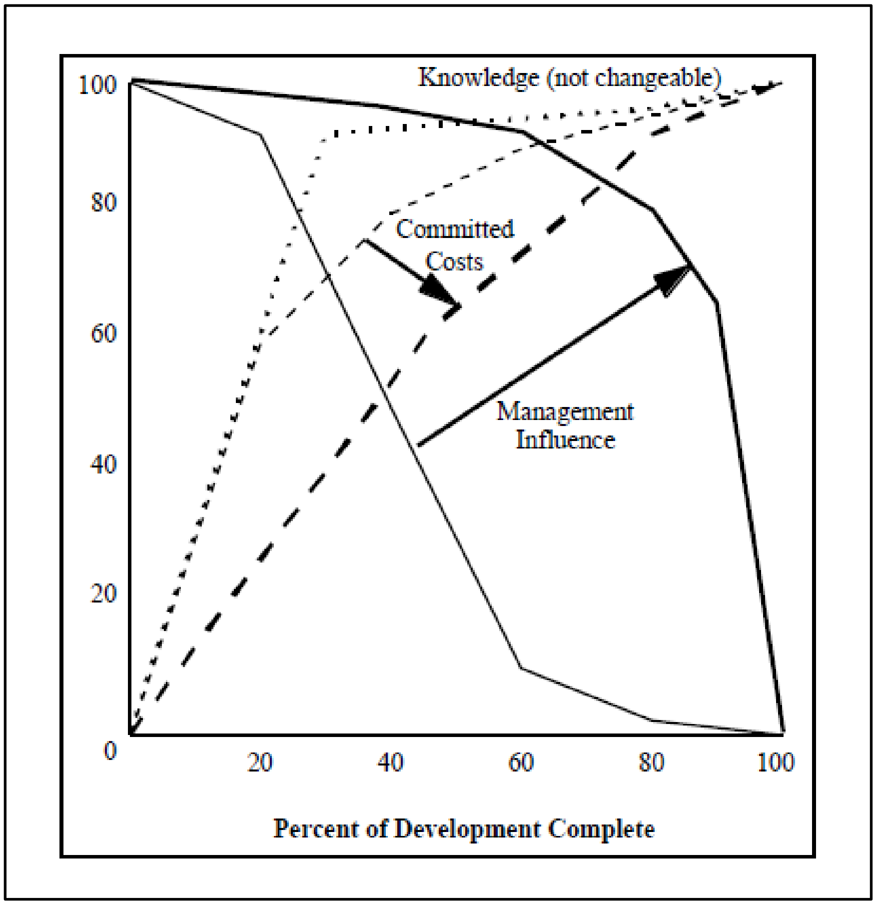


Figure 26. Impact of Delaying Decisions. Source: Bernstein (1998).

A goal of SBD is to "reduce the committed costs to more closely follow the incurred costs" (Singer, Doerry, and Buckley 2009, 7). Traditionally, in early stages of design, predictions are made about the final cost and matched to a budget, but the costs are not actually incurred until later stages of development even though they were committed early on. These decisions on cost have long-lasting consequences on the total cost of the system and are difficult to reverse later. The power of management to influence design costs quickly declines as the product develops. By delaying design decisions and deferring cost commitments, both stakeholders and management have greater flexibility to influence the design (Bernstein 1998; Chan et al. 2016; Singer, Doerry, and Buckley 2009).

SBD also seeks to gain knowledge and understanding about the details of the design problem before making decisions. Traditionally, early stage design decisions are made with incomplete data and the reliant information is subject to change (Ward et al. 1995a). In contrast, SBD defines variables in ranges so the design can proceed until a confident and informed decision is made to limit the design space based on solid and defendable data (Ward et al. 1995b). Instead of forcing decisions early on, SBD delays them based on the logical fact that more information is available and new knowledge is gained as the design progresses with time, and further exploration and additional analyses occur that instill a better understanding of the requirements (Hannapel, Vlahopoulos, and Singer 2012; Singer, Doerry, and Buckley 2009).

Bernstein (1998) cites additional benefits associated with delaying decisions include: achieving better balance between customer wants and technical feasibility; the opportunity to include the latest technology; and better traceability and tracking of competitive products and changes in customer preferences.

3. Design Discovery

SBD replaces PBD with design discovery (Singer, Doerry, and Buckley 2009) by allowing the various specialties to participate in more of the design efforts concurrently, finalizing stakeholder needs later on, and deferring detailed specifications until the design space and tradeoffs are more fully understood. Stakeholder preferences often evolve, so fixing requirements too early may result in considerable rework, design changes, delay, or inability to meet the needs of the customer (Kennedy, Sobek, and Kennedy 2014). By delaying decisions on exact tolerances and dimensions, SBD enables greater product-process design overlap, exploration of numerous possible designs, realization of feasible solutions, and meeting and achieving customer expectations and satisfaction (Liker et al. 1996; Ward et al. 1995a).

4. Adaptability to Change and Handling Uncertainty

McKenney, Kemink, and Singer (2011) demonstrate how the characteristic of delaying decisions leads to a better ability to adapt to and accommodate changes during later stages of design based on the opportunity to compare the tradeoffs among opposing design factors and explore the tradespace of design alternatives (Ward et al. 1995a) before deciding upon a detailed, finalized alternative. A study at Carderock (Chan et al. 2016) shows multiple design options remain open for consideration in SBD, and stakeholders can provide feedback and influence the design as it develops, which enables designers to adapt quickly to requirements changes without a significant cost increase. Furthermore, "working with sets and delaying decisions [in SBD] allows for better handling of uncertainty during the design process" (Hannapel, Vlahopoulos, and Singer 2012, 1), and "small changes due to uncertainty do not necessarily push the design into an unfeasible region or require rework" (Hannapel, Vlahopoulos, and Singer 2012, 2).

a. Rework

Sobek, Ward, and Liker (1999, 71) theorize SBD "can be conducted with no back-tracking or redoing at all," and, "in practice, the costs of eliminating all back-tracking could probably not be justified, but a focus on convergence, rather than on tweaking a good idea to optimize it, can dramatically reduce the amount of back-tracking in the process." PBD requires constant rework and re-analysis because modifications are made to a single design of interest through iterations where the properties change and require repeat analysis each time (Hannapel, Vlahopoulos, and Singer 2012). In SBD, earlier analyses of the design space prior to any particular elimination are still of benefit to the reduced space that remains, and only the well understood regions are developed further. Also, the probability that more detailed information or uncertainties will cause rework in the future is reduced because regions of the design space are only eliminated if they are robust to new information (Singer et al. 2017).

SBD can prove more affordable than traditional PBD approaches based on the reduction of rework (Chan et al. 2016). Kennedy, Sobek, and Kennedy (2014, 278, 281) explain "rework that occurs late in the product life cycle is dramatically more expensive than design work performed early in the cycle," and "fewer and less severe rework efforts mean more reliable on-time performance, faster time to market, lower development costs, fewer manufacturing problems, more innovation, better quality and customer satisfaction, and higher returns on investment." It also suggests reducing or eliminating rework by focusing on and addressing the three culprit situations where:

> 1. The development team learns something critical late in the development process that invalidates prior assumptions or otherwise causes the team to revisit a prior decision;
>
> 2. The development team makes critical decisions too early in the project, before they have the knowledge needed to make a reliable decision; and
>
> 3. Development team members with one expertise inadvertently make decisions that overly constrain those of another expertise. (Kennedy, Sobek, and Kennedy 2014, 281)

Kennedy, Sobek, and Kennedy (2014) also offer ways in which SBD can be used to remedy these three causes of rework through:

- *Accelerated Learning*—Using limit curves and set-based knowledge; systematic, innovative testing before design; and proving "success is assured" as early as possible;

- *Delaying Critical Decisions Until Knowledge is Learned*—Set-based definition of requirements to explore tradeoffs, set-based definition of specifications early on and finalized after learning through the convergence process; and set-based management of major alternative concepts by investigating them in parallel; and

- *SBCE*—Communicate key areas between specialties, break down the walls, minimize design restrictions and maximize design windows.

b. Institutional Learning and Knowledge Structure

The ability to facilitate institutional learning is another advantage of SBD even though it may be intangible and hard to quantify (Singer et al. 2017; Ward et al. 1995a; Ward et al. 1995b). Institutional learning is a process with the capacity to "change behavior and improve performance by reflecting on and reframing the lessons learned during the [design] process" (Watts et al. 2007, 4); a process where "management teams change their shared mental models of their company, markets, and competitors" (De Geus 1988, 70). In SBD, designers can gain insight about the problem from different perspectives as the various specialties communicate, and lessons learned can serve as a reference and provide a jump start on future design efforts (Ward et al. 1995b). Additionally, the emphasis on thorough documentation practices in SBD enables new information to be readily incorporated into the design process and accelerates the understanding of its impact (Singer et al. 2017). Ward et al. (1995b, 59) assume "designers are notoriously resistant to documenting their work" and explains "one reason may be the sense that documentation is generally useless." Documentation efforts in SBD will absolutely serve a solid purpose, so designers may be more inclined to participate.

In addition to institutional learning, SBD also facilitates a robust and flexible knowledge structure interpreted as the evidence, concept ideas, and decisions used by designers and the relationships between them (Singer et al. 2017). The effects on the design knowledge structure promulgated by SBD include: consolidating knowledge and information early on, productive communication throughout the design activity, progressive development of well-characterized regions of the design space, and parallel development of mutually feasible regions. The knowledge structure in an SBD environment is more impervious to change and uncertainty and enables designs to better account for new information and adapt, which implies less rework and a lower probability of needing to reconstruct the knowledge structure as a result of such changes and uncertainty.

c. Specialty Domains and CE

A powerful benefit of SBD is its built-in emphasis on providing each individual specialty an opportunity to consider the design from its own perspective, affording greater parallelism in the design process and more effective use of sub-teams and domain expertise (Singer et al. 2017; Ward et al. 1995a; Ward et al. 1995b). Rather than holding a series of meetings to critique and modify a point design several times in succession, each specialty presents sets of possibilities and integrates them with other specialties to find intersections of feasibility (Liker et al. 1996). The opportunity for executing the design as a parallel process exists since upstream and downstream decisions are mutually discerned and compatible, and future decisions will not invalidate previous ones. Preferences and tradeoffs from each specialty are explored with increased bandwidth and greater consideration and are communicated sooner in the design process when decisions can be felt the most.

Decisions can also be made on partial information, and infeasible regions can be identified independent of what other specialties discover, which means the different domains of expertise can work semi-autonomously (Singer et al. 2017). SBD allows significant concurrency without incurring costs typically associated with collocated teams and frequent meetings (Ward et al. 1995b). It enables the ability to produce designs with distributed, geographically dispersed, and remotely located teams (Mebane et al. 2011) and supports diverse teams across government, defense, acquisition, operational, technical, industry, and academic communities (Burrow et al. 2014).

SBD also promotes harmonious and trusting working relationships (Ward et al. 1995a; Ward et al. 1995b). The productive and meaningful communication created by default through the integration of knowledge and domain interdependencies describes the whole set of possibilities and remains valid as the sets are narrowed and supplemented with additional, more precise information. Communicating the sets of possibilities early on, involving all key entities from inception, and simultaneous product and process development helps build trust in the partnerships between specialties, stakeholders, and suppliers.

K. CHALLENGES OF SBD

Some of the summarized challenges associated with SBD include: perception issues; a reluctance to shift ones paradigm; no instruction on how to execute SBD, no guidelines for how to integrate SBD into DoD acquisition, limited training, experience, and exposure to SBD for reference when trying to apply it, additional commitment of resources needed to build and integrate SBD and model-based systems engineering (MBSE) tools

(Chan et al. 2016); poor definition of SBD in general, how to assess design solutions for viability, how to assess mission effects on a large number of alternatives, time-sensitive design synergies and how to handle dynamic interdependencies throughout the design effort, difficulty determining when and where to make set reductions, identification of robust decision paths—especially for complex problems typically encountered with SBD, set communication and negotiation among disparate design teams, high fidelity tools (McKenney and Singer 2014); no standard metrics for SBD; and no explicit guidelines for when SBD should be used.

1. Perception Issues and Paradigm Shift

The widespread adoption of SBD is hindered by perception issues and a reluctance to adopt a new paradigm. Two major criticisms surrounding SBD are: a more traditional life cycle process model can produce a candidate design faster and more easily; and SBD is assumed to be no different than other optimization efforts and it is what people have already been doing (McKenney and Singer 2014). It is important to consider the reasoning for SBD and the aspects it offers in order to truly realize these criticisms can be rebuked. Although a candidate design could quite possibly be conceived more efficiently, SBD produces a more defensible design with greater resilience to requirements changes. SBD also distinguishes itself from other approaches by eliminating infeasible and highly dominated solutions, establishing a sense of belonging, trust, and communication between team members, and delivering a feasible, viable, system-level optimized design.

Although there is the criticism that SBD is not unique, there is also the opposite misconception – that SBD is "new and untested territory" (McKenney and Singer 2014, 54). There are examples confirming that eliminating design alternatives based on feasibility and set-reduction based on dominance are not novel (Ghosh and Seering 2014; McKenney and Singer 2014; Specking et al. 2017; game theory; utility theory), however, there is still the impression that SBD is introducing a brand new paradigm that is difficult to adopt. There is resistance in leaving the comfort of understandable, familiar, conventional, and proven ideas and methods for one that is seemingly unknown and original; which applies to the established tools and other technical hurdles as well.

2. Implementation Issues

It is a difficult feat for certain types of organizations, such as government-related ones, to rearrange the structure, values, culture, and personnel necessary to embrace SBD practices (McKenney and Singer 2014). There is no concrete, prescriptive, step-by-step, detailed direction (recipe) for implementing SBD; instead, its principles are applied differently for each design project as indirect and guiding systems-based methodologies in a looser formulation of more generalized steps (regimen). Sobek, Ward, and Liker (1999) acknowledge all the determining factors for successful SBD design efforts are not fully understood, and caution that attempts to implement independent piece parts of SBD will likely fail because of the integrated nature of its elements. It also reiterates a necessary shift from the ingrained responses derived from education and normal way of doing things and believes such organizations mentioned above either can or already have exhibited the capacity to effectively practice SBD.

3. DoD Acquisition Integration Issues

According to Singer, Doerry, and Buckley (2009, 7), “the major obstacle to SBD in Naval design is how to facilitate, manage, and implement SBD when the constraints and milestones of current acquisition policies are keyed to [PBD] practices.” The administrative burdens leading up to SBD are the biggest challenge so far, in addition to the lack of guidance for tailoring current acquisition strategies to include SBD for managing cost growth and schedule delays caused by changing requirements and resultant design volatility (Chan et al. 2016). The SSC, ACV, and Small Surface Combatant examples of SBD display a lack of similarity in the implementation process, which is good in that it allows for tailoring SBD to the specific needs of a program, but bad in that there is inconsistency and obscurity in how to actually implement SBD. A thorough evaluation of how, when, and if SBD should be applied to potential programs or platforms, an analysis regarding the incorporation of SBD into targeted acquisition strategies, considerations for Systems Engineering Technical Review (SETR) checklists, and specific guidelines for each unique acquisition program model would be useful.

Instruction on how to execute and leverage SBD in general is needed to boost experience and warrant proficiency. Additionally, education and training for how to employ existing tools to facilitate SBD are needed, along with considerations for standardizing SBD tools and mechanisms. Examples of tools and templates to build certain SBD products include research databases, capability concept wheels, and documentation applications for tracking and applying lessons learned across the DoD community. Growing and enabling other engineering modeling and MBSE tools to perform SBD analysis is also an area of improvement.

4. Relationships, Experience, and Skill Set Issues

Singer, Doerry, and Buckley (2009, 1) voice a growing concern about "a serious shortage of engineers and a loss of critical skills due to attrition in the experienced design community," which is also seconded by Mebane et al. (2011) in the recap of challenges associated with the SSC program. As more practiced engineers leave the workforce and get backfilled with younger, less experienced engineers, there is a more urgent demand for innovative ways of communicating and negotiating about design preferences and sharing new knowledge and information. SBD can serve as a new method of Naval ship design, but it will still be faced with the challenges of young leaders, along with its own challenges regarding ill-defined execution and implementation steps. It does seem likely to require more skill and judgment than traditional PBD approaches, and, while Toyota has had decades to teach SBD, learning and adopting SBD by other organizations might be slow and error-prone, since there is no proven methodology (Ward et al. 1995b).

SBD may "require experience and a strong working relationship between the customer and supplier" (Liker et al. 1996, 169). A certain level of trust by the customer and intent to creatively explore options by the supplier are required when sets of specifications are provided with implied or explicit tolerances for variation. Experience in the relationship also means customers and suppliers are more familiar with each other's preferences, incentives, and modus operandi and struggle less with miscommunication and dispute. The receptiveness for SBD methodologies increases when the relationship becomes stronger through interaction and experience, as does set-based, informal, ambiguous, and free-flowing communication. The value of long-standing relationships and set-based principles over contracts that is so prevalent in successful occurrences of SBD may not be so plausible in certain organizations attempting to adopt SBD.

Designers must be mindful of how the set-based characteristics of identifying sets of values, delaying decisions, generating a large number of prototypes, and gradual convergence on a final solution can lead to problems based on impressions of seemingly ambiguous, unstable, and late communication of requirements (Liker et al. 1996). This can be portrayed as slow, indecisive, and difficult setting of requirements vice the deliberate process of exploring the design space and systematically narrowing sets. Better working relationships will lead to knowing whether suppliers find ambiguous targets confusing and frustrating or whether they prefer the flexibility and ambiguity.

5. Definition Issues

Ghosh and Seering (2014, 10) declare "SBD has never been formally defined, despite many authors having studied its process inspired by the example of Toyota." The two key aspects that require better definition in order to develop SBD methods more fully are set creation and set reduction (or elimination) (Specking et al. 2017).

Barriers to set creation stem from the initial assignment of design values for the factors (MOPs) specific to each specialty and the interdependencies between them. Although not trivial, synthesizing the ranges across each specialty results in the creation of an integrated design space for further exploration and reduction. One problem is assessing these potential design solutions for viability. Diligent definition of the factors with the greatest impact on *system* performance and implementation (MOEs) and the relationships between them up front is key to developing and assessing the second and third order impacts of designs (Specking et al. 2017). Another problem is the ability to assess mission effects on large sets of alternatives, which is compounded by the exponential expansion of alternatives during synthesis. Methods of reducing operational simulation runs to provide general inferences about a solution's impact on mission effectiveness would be beneficial. "[The development of] more automated methods and tools for conducting modified designs and incorporating more feasibility elements in new designs to reduce the probability that a feasible configuration will prove not viable in future stages of design" are also desirable improvements in the set-creation process of SBD (Garner 2015, 9).

According to McKenney and Singer (2014, 54, 55), "the guidance of set reduction is a critical element of SBD execution for large-scale, team-based design efforts and remains an open problem." Furthermore, "there is still a substantial need for SBD execution support, especially in how decisions should be made to reduce the design space while considering total design process impacts." Time-sensitive design synergies and the determination of robust decision paths are two topics of concern related to set elimination that influence the outcome of design; they are associated with five major problem areas (McKenney and Singer (2014):

1. *Time-sensitive design synergies and how to handle dynamic interdependencies throughout the design effort.* How do changing dependencies affect the outcome as the design evolves? Do new relationships emerge? Are old ones broken?

2. *Difficulty determining when and where to make set reductions.* Knowing design is a time-dependent problem, when should decisions be made?

3. *Determination of robust decision paths, especially for complex problems typically encountered with SBD.* How are potential circumstances and areas of concern that present a failure risk avoided when the current set ranges are adversely affected by changes associated with particular design dependencies?

4. *Set communication and negotiation among disparate design teams.* How are the sets communicated? How is SBT extended and applied over several specialties that are solving the same problem and making similar decisions as any one individual specialty?

5. *High fidelity tools.* If the amount of higher fidelity analysis increases throughout the set reduction process, how and when are decisions made to

use certain tools that take a long time to set up and run amid an environment that prioritizes delaying decisions?

6. Metrics and Characteristics Issues

Liker et al. (1996, 171, 176) points out “there are no standard measures of set-based design” and “future research is needed to more carefully develop measures of point-based versus set-based design.” It develops its own set-based indicators in the form of dependent variables representing the level of set-based communication and design, but admits it is difficult to translate such an abstract concept into quantifiable constructs or identify objective measures. Additionally, Specking et al. (2017, 11) stipulate, “SBD must use a sound mathematical foundation [to express design preferences and uncertainty] and have a clear set of methods to define sets, assess the value of points within the sets, explore the design tradespace, and eliminate sets” for it to realistically become a viable alternative to PBD.

a. Set-Based/Point-Based Process Spectrum

According to Ghosh and Seering (2014), a future area of research includes determining what situations are conducive for SBD versus PBD. It attempts to describe when it is appropriate to use SBD or PBD techniques and defines PBD relative to the two principles of SBT on a spectrum as shown in Figure 27.

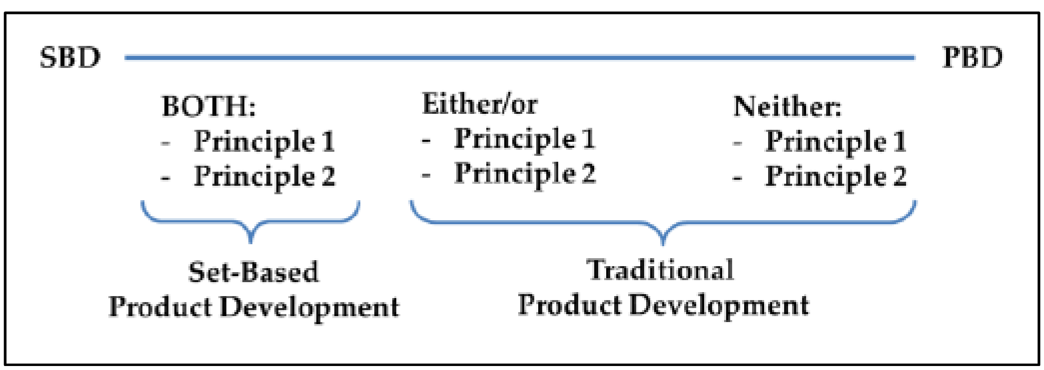


Figure 27. Product Development Spectrum. Source: Ghosh and Seering (2014).

Both principles of SBT are required for an ideal case of SBD, and the lack of both principles constitutes pure PBD. The two principles of SBT can act independently, and although processes might exhibit some characteristics of SBT, they may not necessarily represent pure SBD. The spectrum helps classify the degree to which a certain process represents a complete implementation of SBD based on the SBT principles, but does not extend into specifically identifying when SBD should be used and under what circumstances. Specking et al. (2017) agree a definition of key SBD characteristics will help clarify SBD for comparison to PBD, pinpoint where design efforts fall on the spectrum, implement the most suitable design method, and define prudent tradeoff analytics.

b. SBD Rigor Metric

McKenney (2013) describes a rigor metric used to evaluate the extent to which a design activity is set-based. It focuses on the general regimen of SBD principles as opposed to concrete rules because SBD can be executed in varying degrees and requires design-specific tailoring. The SBD rigor metric shown in Table 5 is intended to assess the level of set-based principles present in a design process before it begins: the goal implies the most rigorous level of SBD applied where infeasibility and dominance steer set eliminations leading to the gradual convergence on a single, feasible, and viable solution. Level 1 describes the barebones support needed within each element for the lowest rigor of SBD and increases to a more fulfilling and structured level 3 rigor.

Table 5. SBD Rigor Metric. Source: McKenney (2013).

SBD Element	Process Goal to Achieve SBD	Success Level
Characterization	Define, bound, partition, describe, and document the design space, while also including specialty preferences, uncertainty, and infeasibility criteria	1) Design space characterized heuristically with little formal data 2) Formal declaration and documentation of parameters, bounds, and partitioning into specialties 3) Supports external review and approval
Flexibility	Facilitate, review, track, and document reduction decisions with detailed justifications, while maintaining the flexibility to accommodate errors and adapt to changing requirements and conditions	1) Concurrent evaluation of alternatives across specialties 2) Tracking, documentation, and review of set reduction decisions and rationale 3) Supports external review and approval of reduction decisions and a protocol for re-opening design space with good reason
Convergence	Move towards optimal and robust solutions by supporting set convergence and staying within previously defined sets (only expand them for special exceptions and legitimate reasons)	1) Design space sizing strategy is provided to estimate the relative size of the design space and track reduction progress 2) Measures for tracking convergence are defined and progress and projected completion time are documented 3) Supports tracking, documentation, and external review of deviations outside previous set ranges
Communication	Communicate design space preferences and feasibility and design factor importance and influence to capture tradeoffs and promote information transfer and integration	1) Defined grouping strategy to facilitate communication 2) Formal communication protocol established 3) Facilities provided for tracking, documentation, and external review of negotiations
Facilitation	Guide the set reduction process by establishing set reduction strategies and convergence rate goals and support communication across specialties	1) Simple integration protocol provided where integration lead resolves conflicts 2) Integration protocol provided that supports convergence strategy and uses preferences to eliminate infeasible and inferior regions 3) Facilitation, tracking, documentation, and external review of negotiations involving competing and conflicting preferences across specialties are provided

c. *SBD Diversity Metric*

Doerry (2015) describes a diversity metric that reflects the number of different component options that would fit the set of feasible configurations and evaluates how different system design configurations are from one another. In essence, the diversity metric measures the risk associated with feasible solutions becoming not viable as the design progresses and more knowledge is gained (such as with parts obsolescence and single points of failure). A higher diversity metric, which is associated with a greater number of configurations that can achieve a certain capability, reduces this risk and implies component substitutions are more likely available, while still maintaining a representative cost.

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE IMPROVED SBD PROCESS

The SBD process is more thoroughly explained in this chapter because more delineation is needed to describe it with less confusion. The principles encompassing Toyota practices (Sobek, Ward, and Liker 1999) sync with the general implementation steps for SBD (McKenney and Singer 2014; Singer et al. 2017; Singer, Doerry, and Buckley 2009; Specking et al. 2017), but some of the intermediate steps are missing.

The improved SBD process introduced in this dissertation, as depicted in the flow chart in Figure 28, involves steps with just the individual specialties, steps where all the specialties communicate and overlap, and steps where decisions are made, plus documentation throughout. There are conceivably several design spaces, but certain ones pertain to certain steps within the process. There are also several sets created, and a clear description is needed to determine which set is being considered when. The mention of feasible regions is also obscure, since they too occur differently depending on where the design is in the process. This chapter describes the improved steps that better define the SBD process and account for these considerations.

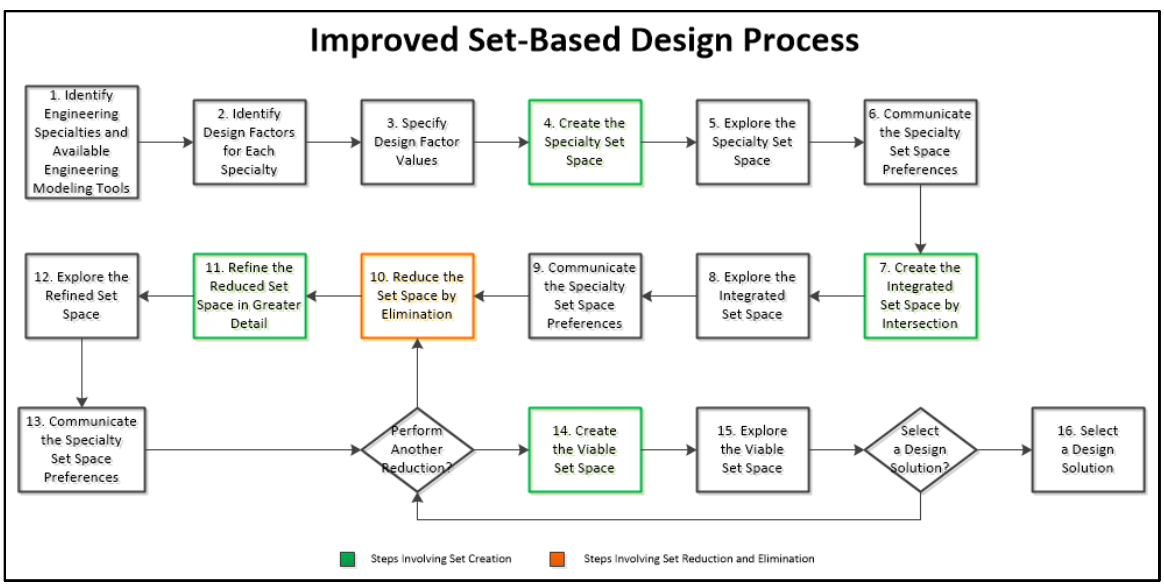


Figure 28. Improved SBD Process Flow

Although there is no explicit step regarding documentation, documenting the assumptions, justification, and rationale for set reductions is critically important in SBD. Learning new information and acquiring new knowledge are an inherent part of SBD, and thorough documentation practices are necessary to accommodate it. Requirements modifications and scope changes are inevitable during the design process, and good documentation enables a quick evaluation of where the impacts are, on which set reductions, and in what decisions. Documentation throughout the entire design process is also helpful during program reviews and for traceability in general (to requirements, decisions, the accumulation of new knowledge, etc.).

A. IDENTIFY ENGINEERING SPECIALTIES AND AVAILABLE ENGINEERING MODELING TOOLS

The execution of SBD starts with identifying individual engineering specialties (or specialty groups, domains of experts, functional groups, functional areas, etc.) after the required capabilities and functions have been determined based on gap or need. Most SBD approaches described to date use teams in a decomposed systems sense where the specialty teams are responsible for the subsystem design functions, an integration team facilitates communication between the various specialties, and a chief engineer (or lead integration manager, high-level program manager, *shusa* (Ward et al. 1995a, etc.) serves as a spur to consensus (Ward et al. 1995a) or authority for system-level decisions leading to convergence on a final solution (Whitcomb and Hernandez 2017). Specialties can also be represented by disciplines (such as structures, propulsion, and value determination) and contribute technical expertise for evaluating design factors and candidate solutions. The engineering modeling tools available for design are most likely specific to the specialties, and it is important to know upfront what each one (and the overall program) have to work with, especially in the sense of grander, system-level simulation tools.

B. IDENTIFY DESIGN FACTORS FOR EACH SPECIALTY

Design factors are solution parameters, characteristics, or relationships that influence the design at a system level, such as length, weight, engine size, fuel capacity, etc. (Singer et al. 2017). Identifying the factors that influence the overall design and recognizing how they impact other specialties is a very important part of SBD because it indicates the definite communication and feedback interfaces needed between specialties later on. The proposed design factors from each specialty are considered from the point of view of all other specialties to determine the magnitude of influence. For example, some design factors might simply be informational with no specific preference (e.g., those used as inputs to a modeling tool); some might conflict or have direct influence and need to be negotiated (e.g., the type of metal used which impacts the weight which impacts the transit speed, fuel capacity, or engine size).

C. SPECIFY DESIGN FACTOR VALUES

Each specialty defines the acceptable range of values for the design factors important to it or a discrete state as appropriate (turbo or prop for example). It individually identifies "the primary design constraints on its subsystems—what can or cannot be done or should or should not be done—based on past experience, analysis, experimentation and testing, outside information," engineering checklists, and lessons learned (Sobek, Ward, and Liker 1999, 73). A set of values is identified as opposed to a single one, which helps the other specialties understand the possibilities, tradeoff options, and what they have to work with. Offering a range of values also promotes system-level optimization instead of optimizing one part at the expense of the others. It is the responsibility of each specialty to understand how the options within the sets of every other specialty influence the materiel solution from its own perspective.

D. CREATE THE SPECIALTY SET SPACE

The specialty set space is created by synthesizing every option within the acceptable range of values for each design factor within each specialty. The options in the acceptable sets for each design factor imply a potential solution exists for every combination, which can be extremely abundant depending on the size of the sets and number of design factors.

E. EXPLORE THE SPECIALTY SET SPACE

Each specialty explores its synthesized variants and determines what is feasible and preferred. Regions in the specialty set space are evaluated for feasibility and further examined to learn about tradeoffs and design factor sensitivity. Multiple alternatives and design solutions are explored in slightly more detail (maybe through sketches, simple spreadsheets, quick calculations, or low-fidelity modeling) to gain a better understanding of what is possible.

F. COMMUNICATE THE SPECIALTY SET SPACE PREFERENCES

Each specialty identifies the sets of design solutions it determines to be infeasible or highly dominated at the current level of detail. By default with SBD, solutions are kept if there is no apparent reason to conclude they are infeasible. When the infeasible and highly dominated regions of the specialty set space are removed, each specialty communicates the remaining regions of design solutions and any other important information to all other specialties. This is typically accomplished by a chief engineer or design integration manager role to fully reach all relevant specialties and ensure no new information has become available that might alter the outcome.

G. CREATE THE INTEGRATED SET SPACE BY INTERSECTION

The analysis based preferences and feasible regions from each specialty are integrated by intersection to create a smaller set of unified global concepts representing total system solutions. This dissertation demonstrates and contributes a novel method of set creation using a DOE approach applied in a set-based manner to generate points (design variants or solutions) in the design space for input into a simulation tool to obtain responses—because decisions are based on outputs and responses, not inputs—value, and cost. Of the methods of set creation found in Specking et al. (2017), it fits within the *design space* category because it samples the design space.

The integrated design space can be created by a full permutation of all input factors, but will most certainly lead to an egregious number of design solutions that may be prohibitively time consuming or non-cost effective to explore. Sampling strategies using stratification techniques that are space-filling with sample coverage over the experimental region and that distinguish the effects of each input factor can be used to maximize the amount of information gained while minimizing the amount of data to collect, such as tensor product sampling, Monte Carlo sampling, jittered sampling (a hybrid of tensor product and Monte Carlo), and LHS. The example in this dissertation uses LHS and is further described in Chapter IV.

H. EXPLORE THE INTEGRATED SET SPACE

The integrated design solution sets incorporate the feasible and preferred regions from all the specialties and are bounded in an integrated set space that accounts for inputs from everyone. Each specialty now has the opportunity to evaluate the new integrated set space from within its own expertise and perspective to determine what works and what does not. Each specialty can utilize its own methods and tools to evaluate its aspects of the design.

During this evaluation phase, specialties can confirm the information they need has been provided and that they are not missing anything critical. Specialties can also verify the initial ranges of design factor values and preferences are reasonable. Knowing the appropriate information is available and the specialty design solution sets are reasonable leads to a proper and complete exploration of the integrated set space and indicates the direction the design should or should not go…and the reasons why (McKenney, Kemink, and Singer 2011).

Each specialty takes the opportunity to explore the integrated set space from its own perspective by:

- Looking at the input factors (MOPs) it is responsible for and the operational requirements (MOEs) that are a function of these input factors;
- Considering what part of the input factor ranges can be eliminated; and
- Identifying important information and new discoveries.

I. COMMUNICATE THE SPECIALTY SET SPACE PREFERENCES

Having had a chance to consider all of the combined information and explore the integrated set space from its own perspective, each specialty communicates its preferences and any other pertinent information and important findings to the design integration manager. Each specialty can provide feedback, comments, and recommendations; communicate what it would like to see; and open the dialogue for negotiation with other specialties, while also maintaining a mindsight of accommodating as many of the other specialties' options as possible. Additionally, a specialty can communicate its preferences graphically in the form of preference curves that show the weight of its preferences on a scale for a given value in a range.

J. REDUCE THE SET BY ELIMINATION

After all new information and preferences are consolidated by the design integration manager, regions of the design space considered to be infeasible or highly dominated are eliminated, leaving only mutually feasible sets to explore further. The methods of set elimination demonstrated in this dissertation pertain to the *infeasibility*, *preference*, and *set-theoretic* categories described by Specking et al. (2017). Set reduction and elimination can occur through specialty-level investigations, system-level investigations, distance to the ideal point, and visual inspection.

1. Specialty-Level Investigations

The most auspicious application of SBD occurs at the specialty level, where each specialty learns about its impact on the system and the overall implications of its decisions. Set reduction at the specialty level can occur by each specialty considering only the input factors, design variables, and MOPs it is responsible for, or by considering the input factors and the MOEs that are a function of those input factors. For example, an engine specialty for an automobile design team might have fuel type as an input factor, which is part of an efficiency MOE measured in miles per gallon (MPG); the engine specialty can look at the effects of fuel type on the design, or it can look at the effects of fuel type on the design and on the MPG specifically. Set reductions related to input factors aside from their MOEs are generally based on dominance, i.e., a certain subset of the range of a given input factor is highly dominated (be it through negation, preference, performance, etc.). Set reductions where the input factor is considered against its MOEs are generally based on viability, i.e., a certain subset of the input factors is incapable of achieving the desired threshold value for a particular MOE.

A specialty is responsible for the input factors it defines acceptable values for and passes on to the other specialties. Input factors are only assigned to one specialty, whereas MOEs can be shared across several specialties. For example, the efficiency (MPG) MOE is important to both the engine specialty in terms of fuel type and to the drive train specialty in terms of tire size, so both specialties would consider it. After each specialty has explored its input factors and MOEs, it determines what subset of the range of each input factor can be eliminated. Set reductions are performed by eliminating portions of the input factor only and not by changing the limits of the MOEs; specialties have no control over the desired threshold values for the MOEs as those are defined by the stakeholders.

Specialties make valuable contributions to the design process by identifying important information and sharing new discoveries, especially those that impact the way other specialties look at their own parts. New knowledge communicated by one specialty can impact the other specialties by constraining them and limiting their ranges of input factors, enabling them and allowing full flexibility of their ranges of input factors, or neutrally doing nothing one way or another at the present state of the design space. A brilliant facet of SBD is that none of the communicated information is acted on until every specialty has had a chance to discuss the new knowledge and consider the ramifications from its own perspective, which promotes and encourages shared learning and communication throughout the entire design process.

2. System-Level Investigations

System-level investigations account for how the input factors affect the OMOE and whether or not the design solutions are capable of achieving the mission. At the system level, the sets are considered holistically with input from all the specialties and reduced based on viability and ability to meet the desired OMOE.

3. Distance to the Ideal Point

Set reductions based on distance to the ideal point aim to eliminate points that are furthest away from the ideal point (i.e., typically lowest OMOE and highest cost options are removed). Putting a viability index such as this on the points removes a portion of the variants and keeps more open for consideration, as opposed to potentially eliminating too many design solutions and keeping too few by constantly looking for the Pareto boundary. Three different measures for distance to the ideal point include the: L1-norm, or city block distance; L2-norm, or Euclidean distance based on the basic Cartesian distance between two points; and the technique for order of preference by similarity to ideal solution (TOPSIS) method that simultaneously computes the sum of the distance to ideal and the distance from non-ideal.

4. Visual Inspection

Set eliminations based on visual inspection may be appropriate when design solutions are highly dominated and stand out from the other alternatives.

K. REFINE THE REDUCED SET SPACE IN GREATER DETAIL

Improving the design towards becoming more globally optimal and increasing the design fidelity and level of detail as the sets narrow are distinguishing features of SBD (Singer, Doerry, and Buckley 2009). The set space can be refined by generating additional points within the reduced area (smaller space), or by further specifying the remaining solutions within the reduced sets. Although generating additional points seems counterintuitive, it works because the solution space is still being narrowed in terms of the portion of the design space, but the fidelity within the remaining regions is being increased. Eventually the remaining regions of the design space will be small enough that refining them will lead to design solutions that are essentially the same, and any point can be selected.

L. EXPLORE THE REFINED SET SPACE

Reducing and refining the set space results in smaller regions where only feasible solutions remain. Ongoing and planned analysis focused on these regions is enabled by communicating the results of the set reduction and refinement across all specialties. New learnings occur when each specialty takes the opportunity to explore the refined set space from its own perspective.

Additional knowledge augments this smaller set space through increasing levels of detail and design fidelity during the design process. Since design knowledge is cumulative and not iterative (Singer et al. 2017), new knowledge is combined with previous knowledge to determine subsequent set reductions based on additional information instead of arbitrary decisions (Singer, Doerry, and Buckley 2009).

Set reductions occur as specialties develop new knowledge and acquire new information, which reduces uncertainty and leads to improved decisions with less probability of rework. Specialties can also continue to explore and develop reduced design space solution sets knowing the investment will result in new knowledge that will be relevant and beneficial to future reduction decisions without negating prior ones.

M. COMMUNICATE THE SPECIALTY SET SPACE PREFERENCES

The newly learned preferences and important findings from each specialty are communicated to the design integration manager. Repeating the process of set reduction, refinement, exploration, and communication over several iterations may be desired until the design space has been sufficiently reduced to a manageable number of design solutions that can be confidently compared and no further set reductions are desired. All regions within the remaining design space are feasible because the infeasible and highly dominated regions have explicitly been eliminated during each preceding iteration of set reduction. The intersections of feasible regions will describe at least one materiel solution even if these feasible regions are not contiguous within the boundaries of the reduced design space.

N. CREATE THE VIABLE SET SPACE

Early in the design process, integration by intersection will not result in a specific design, but rather an integrated set space of feasible configurations for system-level solutions. It is important to distinguish that although the sets created are deemed feasible (meaning they can be built), they may not necessarily be viable (they are not capable of meeting the mission requirements if built). Further exploration and more in-depth research may validate these solutions as viable, but it might also prove them not viable. Ideally, the configurations are diverse enough so more detailed analysis does indeed validate the subset of feasible configurations remains viable (Garner et al. 2015). This implies the sets of feasible configurations should differ in failure mechanism (or that the likelihood of failure is low), so the risk of finding a feasible solution not viable is minimized. Furthermore, the set creation process should consider as many unique options and distinct concepts as possible as opposed to simply including variants of a single concept (Ghosh and Seering 2014).

The viable set space is created by:

- Verifying all the set reduction criteria have been applied and all input factors (MOPs) are within the reduced and agreed upon ranges;
- Eliminating design variants that do not meet every MOE; and
- Checking that the viable design variants do indeed spread across the whole range of acceptable input factor (MOP) values.

Although design variants can be outside the reduced and agreed upon ranges of input factors and still be viable, checking that they adhere to the ranges is important for credibility between the specialties because they are following through with the justifiable and agreed upon reductions; going backwards will break trust and negate a benefit of SBD unless it has been mutually agreed upon to do so. If there are no design variants reaching out to the extremes of the acceptable values, then it may be worth considering another set reduction to tighten the range of those particular input factors and further refine the design space. Design variants that fail to meet every MOE must be eliminated in order to create the viable set space because the stakeholder MOEs are indicators of achieving the mission needs; design variants must meet all MOEs to be viable. If the mission changes or the stakeholder preferences and MOEs are adjusted, it is possible in SBD—and facilitated by thorough documentation – to return to a previous state of the design space when a reduction occurred that impacted a particular MOE of interest. If this should happen, the design process can start again from there with new iterations of set reductions and gathering new knowledge, or the design solutions can be checked for and eliminated based on viability.

O. EXPLORE THE VIABLE SET SPACE

The viable solutions are explored and considered in a manner that leads to selection (e.g., Pareto (or other) optimality, value versus cost, etc.). Ideal implementation of SBD uses several set reductions to hone the design space and optimize at the system level, but the most straightforward way—that also results in the least amount of specialty learning and new knowledge—is to eliminate variants based on whether or not they meet the threshold requirements for all MOEs and then make a selection. Eliminating solutions that fail to meet the MOEs and then making a selection is similar to how PBD is done and does not get the SBD benefit of increased design space fidelity. The number of variants available to consider is also limited, since no more are added to the design space with tighter ranges achieved through subsequent set reductions and refinement. It could end up that none of the design choices meet the operational requirements (i.e., none are viable). Should this occur, most likely the ranges of input factor values need to be relaxed, or less granularity is needed in sampling.

P. SELECT A DESIGN SOLUTION

Convergence on a design solution is possible when "decisions are systematically made with an ever-increasing amount of knowledge and detail" (Singer, Doerry, and Buckey 2009, 11). The best possible design is selected at a point when all sets are feasible and viable and all tradeoffs have been explored (McKenney, Kemink, and Singer 2011) (there may not even be much difference between the design solutions at this point). The beauty of the set reduction process is whence the design space has been finalized to include only feasible and viable regions (i.e., the viable set space), any remaining solution can conceivably be picked at random. The methods warranted by this flexibility for determining a final solution include (Singer et al. 2017):

- Letting the source selection process determine the final solution based on specifications that describe the viable design space;

- Empowering stakeholders to negotiate the final solution based on inputs describing the characteristics of the viable design space;

- Soliciting stakeholder priorities to partition the viable set space according to preference (i.e., highly preferred, preferred, and not preferred) and then exploring it for regions of overlapping levels of preference or negotiating until agreement is reached;

- Using stakeholder preferences and utility goals as inputs into traditional optimization tools;

- Pursuing a robust subset of feasible solutions to explore and resolve risk before deciding on a final solution; and

- Filtering out sources that cannot support the cost or schedule requirements based on the solution presenting the maximum and most aggressive demand; essentially the bounds of the design space offer bounds on the best and worst case scenarios.

Q. SBD CONCEPT GRAPHIC

An overarching concept graphic for SBD is shown in Figure 29 that illustrates the role of each process step and how the individual specialties fit within the process as a whole. An updated concept graphic from the original representations of SBD (recall Figure 1) is necessary to emphasize that:

- Regions of the design space are eliminated for each domain as the design progresses (the left portion of Figure 1 focuses on where the regions intersect for each specialty, but it is less direct about what is being eliminated);

- It is possible for the initial design space to be narrowed down to multiple, disparate regions that are not necessarily continuous or rectangular (the left portion of Figure 1 converges on a single black rectangular area, and the right portion converges on a single oddly shaped green area);

- A design solution is selected from the remaining spaces where the preferences of each domain overlap after infeasible and highly dominated areas have been eliminated (this is not as easily understood in the left portion of Figure 1, but is depicted well in the right portion through the red and green regions); and

- SBD is concerned with the actual points in the design space as they represent unique design alternatives (neither portion of Figure 1 depicts this).

In Figure 29, everything to the left of the thick, dotted line occurs *before* the knowledge and preferences from each specialty have been integrated, i.e., the specialties are unaware of the information and needs of other specialties. Everything *after* and to the right of the thick, dotted line incorporates knowledge and preferences from all specialties combined, and each specialty can see its impact on other specialties.

Starting with the system space, three unique specialties are identified in step 1: blue, red, and green. Although the graphic is conceptual and the x-axis and y-axis can be a lot of things—never exceeding the size of the design space initially defined by the system space—in step 1, they can represent MOP versus MOE, so each specialty sees how its MOPs fit within the MOEs.

Steps 2 and 3 then flow upwards out of each colored specialty into three separate blocks of matching color that contain different symbols (triangles, squares, and stars) to represent the various input factors specific to each specialty. Here, the x-axis and y-axis can represent input factor versus MOP.

The input factors are combined in step 4. While this step could be shown with dots that rollup and encompass synthesized values of each input factor, it is depicted as an overlap in the graphic to maintain that all of the values of each input factor are being crossed and accounted for. This block can represent MOP versus MOE again. The large, dotted arrow connecting steps 1 and 4 shows how the specialty set space is now populated within the overall system space.

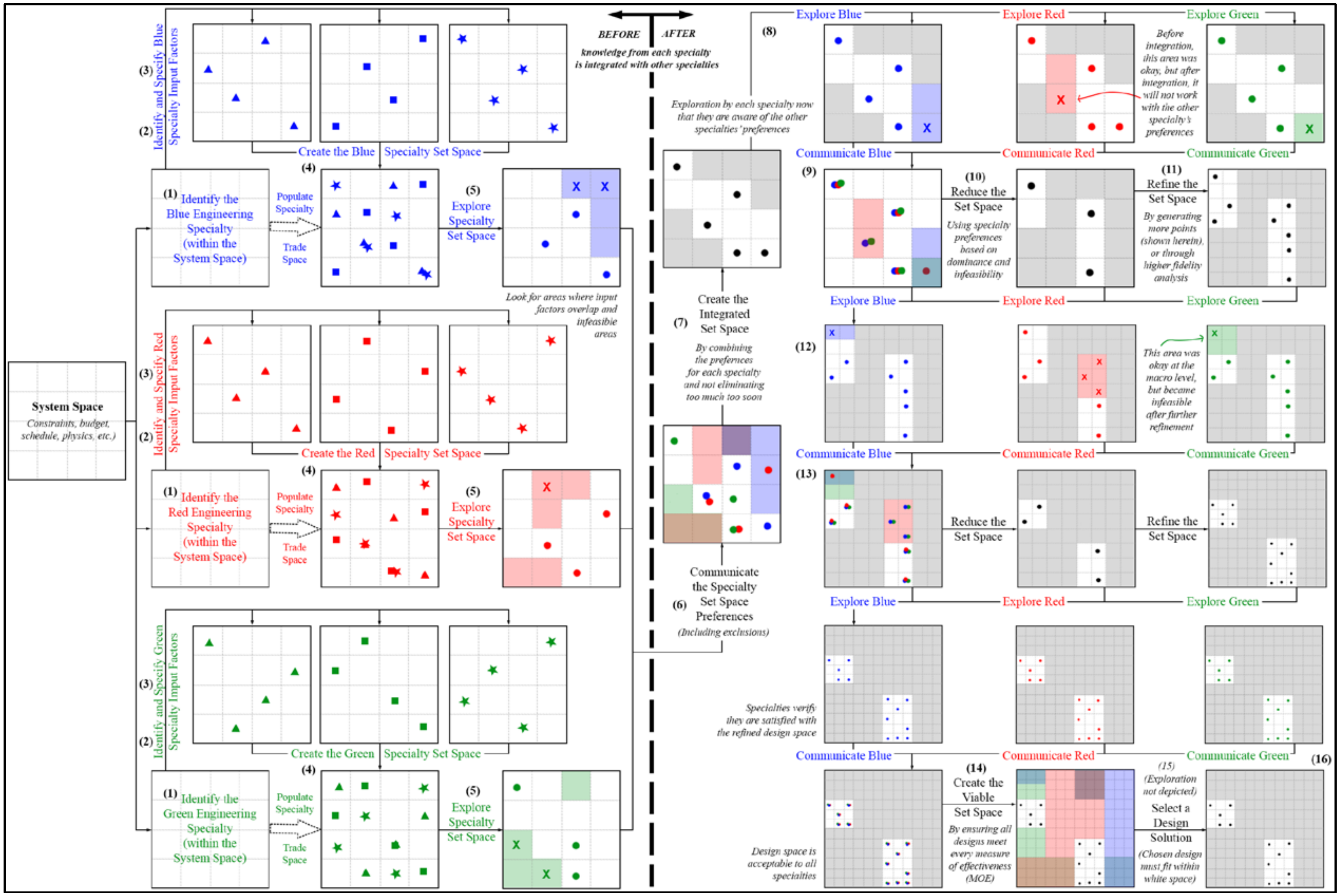


Figure 29. SBD Concept Graphic

In step 5, each specialty looks for areas where input factors overlap (feasible areas), and areas where they do not (infeasible). Often times, areas at the extremes of the design space will be infeasible (e.g., the largest tire size and smallest rim, or highest rigidity and most ductile material). Other times there is insufficient data (cells with no symbol in them), and specialties have to use their expertise to infer meaning and make determinations. The graphic portrays promising areas as dots and infeasible areas x's. The dots come from areas where several symbols overlap, or if the symbols trend together. The x's come from areas of no overlapping symbols and symbols trending apart.

Step 6 is the first time all of the knowledge and preferences from each specialty come together. Each specialty has passed its information forward to a design integration manager for consolidation.

The design integration manager creates the integrated design space in step 7 by synthesizing all desired ranges of input factors from each specialty and shading the infeasible regions. Some of the preferences were not applied yet so as not to eliminate too much of the design space too soon (e.g., one of the red squares in the interior).

Step 8 allows each specialty to consider the consolidated information and integrated design space from its own perspective, which is represented by the three separate exploration blocks on the top right of the graphic. One area of interest is the cell containing a red **x**: this area looked appealing to the red specialty prior to integration, but after integration, the red specialty learned that it was not feasible with the preferences and constraints of the other specialties.

The specialties communicate their knowledge and preferences to a design integration manager in step 9, and then the reduced set space is created in step 10. The refined set space is created in step 11 by generating additional points in the reduced space.

Each specialty gets an opportunity to explore the new, reduced and refined set space from its own perspective in step 12 and communicates its preferences in step 13, which leads into another iteration of reduction, refinement, exploration, and communication. An additional note of interest is how the top-left corner area within the green specialty looked acceptable from the macro-level view in previous exploration phases, but ended up being infeasible in more refined views; turns out the acceptable areas are actually a bit below the top-left corner in the white space containing the two green dots.

The last exploration phase depicted shows three blocks with no change. This is an opportunity for the specialties to verify the last reduction and refinement is satisfactory. The fact there is no change implies there are no further reduction or change recommendations by the specialties.

The viable set space is created in step 14 by ensuring all designs meet every MOE and that they fit within the acceptable and agreed upon ranges. In this graphic, all designs must lie within the white regions of the design space, as the colored regions represent the cumulative infeasibilities (or highly dominated regions) for each specialty. Exploration of the viable set space (step 15) is omitted from the graphic because it is similar to previous exploration phases that are depicted on the graphic. A design solution is selected in step 16 and must be selected from the white space.

IV. LHS AND SIMULATION FOR SET CREATION

This dissertation explicitly demonstrates and contributes a novel method of set creation using a DOE approach applied in a set-based manner to generate the points (design variants or solutions) in the design space for input into a simulation tool to obtain responses, value, and cost. SBD deals with points in the design space as opposed to curve fits familiar in RSM, which makes how they are generated important. A general introduction and comparison of stratified sampling techniques are included in this chapter, which lead to the justification and appeal of LHS strategies for set creation in SBD. The characteristics of LHS are useful because design solutions can be sampled from the full ranges of input factors over the entire design space, including the interior and extremes. Several LHC design methods are described in this chapter and result in different coverage of the design space.

A. SAMPLING AND STRATIFICATION TECHNIQUES

LHCs have been studied mathematically to construct distinct design solutions as described by Ghosh and Seering (2014) as input for set creation. OAs and factorial designs are common sampling strategies for computer experimentation (Montgomery 2001; Sacks et al. 1989), but stratification techniques are often used to obtain a more uniform selection of samples.

Tensor product sampling grids are binning optimal (each equally spaced cell contains only one sample point) but have large discrepancies (non-uniformity measure comparing the total number of samples to the total cell volume) (Dalbey and Karystinos 2010).

Monte Carlo sampling refers to the purely random generation of design points. Monte Carlo techniques are robust, universally applicable to uncertainty quantification, and especially useful for high-dimensional integration, however, the standard error in the estimate of the mean decreases relatively slowly as the number of points increases, which requires over a million samples to get an accuracy of three significant digits and is not repeatable. Improved (quasi-Monte Carlo) methods control the random sampling to improve equal distribution and exhibit low discrepancy (a metric of uniformity where higher uniformity equates to lower discrepancy; Dalbey and Karystinos 2010, 3) for a faster rate of convergence.

Jittered sampling perturbs each point in a tensor product grid by a random amount. It is binning optimal (a space-filling property; Dalbey and Karystinos 2010, 5) and decreases discrepancy faster than Monte Carlo, but is limited in that the number of samples must be exponential in the number of input directions (Dalbey and Karystinos 2010).

B. LATIN HYPERCUBE SAMPLING

LHS was introduced by McKay, Beckman, and Conover (1979) for numerical evaluation of multiple integrals. Since then, LHCs have been used extensively in the literature for achieving more accurate estimates obtained from computer experiments (Santner, Williams, and Notz 2003). LHCs have found wide applications in DOE and numerical integration (Tang 1998), and are considered "good general-purpose designs for exploring complex simulation models when little is known about the response surfaces" (Sanchez and Wan 2015, 1799). LHCs are constructed as balanced designs where each level of each variable is sampled an equivalent number of times (Montgomery 2001).

When compared to other sampling methods, LHCs correspond to an OA of strength 1 for an index (number of repeats) equal to 1. They cover the bounds of the design space like fine grid factorial designs, but with a subset of the information in the interior and orders of magnitude less sampling. LHS is also very flexible due to its non-collapsing property (i.e., good projective properties of the sampling points with no duplicate coordinate values; also referred to as low-discrepancy). Viana (2013) explains that if some of the dimensions are eliminated in LHS making the domain smaller, the existing data can still be reused without reducing the sample size, which is in contrast to factorial designs that are collapsing.

Dalbey and Karystinos (2010) distinguish between LHS, Monte Carlo, and jittered sampling for large sample sizes: LHS shows a lower variance than traditional Monte Carlo sampling and improves convergence like jittered sampling through better uniformity. They further point out differences in dimensional behavior: LHS and jittered sampling are the same in one dimension where random selections are made within a stratified distribution of random variables. For the multidimensional case, however, while jittered sampling is a "perturbation on a full tensor product grid of samples, in LHS, each dimension is stratified separately and then values from each dimension are paired to form coordinates of points" (Dalbey and Karystinos 2010, 8).

Additional LHS characteristics that contribute to it being the predominant and well-accepted DOE method for computer experiments (Hernandez, Lucas, and Carlyle 2012) are that it:

- Is available and obtainable through many software packages;
- Has optimizable space-filling properties without replication or projection redundancy on any single dimension;
- Has few restrictions on the number of factors and design points;
- Can be fit to many diverse metamodels; and
- Has reduced discrepancy with good uniformity in low dimensional projections of the sample space, allowing for all input factors to be represented in a fully stratified manner regardless of dominance.

There are also challenges with LHS, such as those also encountered in other designs with high dimensions where the design space becomes more difficult to fill as the number of variables becomes large (sparse sampling of the design space). Other open areas of research are optimization of LHCs; mixing discrete and continuous variables where there is misalignment between the number of levels in the design and the number of discrete variables; incorporation of global sensitivity information by controlling the number of levels in each dimension based on the rate of how the response varies; and sequential sampling for adapting and improving the metamodel based on the performance of the previous set of points (Viana 2013).

C. LHC OPTIMIZATION

Finding the optimal LHC is challenging because of the combinatorial nature of the problem over an exponentially increasing design space. In practice, random LHCs are often generated and the best is selected based on a certain objective function criterion targeting space-filling or orthogonality characteristics (Hernandez et al. 2012). Model-specific alphabetic optimality criteria, such as D-optimal which minimizes the determinant of the covariance matrix (good for determining non-significant factors), or I-optimal which minimizes the average prediction variance (when improved prediction is desired), work well when the input and output relationships can be assumed. For broader experimental regions with more complex responses, pre-specified models are insufficient, and alternative approaches focused on minimizing the bias part of the mean squared error are sought based on the input space.

Space-filling designs aim to sample and spread the design points uniformly across the experimental region. They are more appropriate for identifying unknown response surfaces and improve interpolation methods and prediction precision, which make them better able to detect localized effects as well. Examples of space-filling criteria include minimum integrated mean square error (IMSE), maximum entropy, minimum discrepancy, minimum Audze-Eglajs potential energy (1977), maximin distance (maximizing the minimum distance between points), and minimax distance (minimizing the maximum distance between any non-design point and the closest design point in the design space).

Orthogonality considerations in LHC designs are obtained by controlling the correlations among columns of input variables (primarily linear effects, but also higher-order polynomial effects). Ideally, correlation is minimized so the degree of influence for each term in the main effects and interaction effects can be isolated. Highly correlated designs also leave large areas of the experimental region unexplored, leading to poor prediction in those areas. The reduction or elimination of column-wise correlations is typically achieved through transformation procedures that change the original design.

Specialized algorithms are used to accomplish optimization of LHC designs; most use an exchange method for searching the design space (e.g., swapping elements within a randomly chosen column to generate a new design). Coordinate exchange (Audze and Eglajs 1977), columnwise-pairwise (Cioppa and Lucas 2007; Li and Wu 1997; Liefvendahl and Stocki 2006; Ye, Li, and Sujianto 2000), enhanced stochastic evolutionary (ESE; Chen and Xiong 2017; Jin, Chen, and Sudjianto 2005), simulated annealing (Morris and Mitchell 1995), threshold accepting (Fang and Lin 2003), particle swarm (Chen et al. 2013), genetic algorithms (Bates, Sienz, and Toropov 2004; MacCalman, Vieira, and Lucas 2017) and others have been used to construct optimal LHC designs.

Table 6 shows some of these optimization strategies and associated objective functions for LHC designs. Although most seek to improve on either space-filling or correlation characteristics, optimization is not necessarily achieved with respect to the other, and a good LHC design is not guaranteed. Current research efforts intend to develop algorithms to explore both characteristics of a good design and generate space-filling and uncorrelated experimental designs (Cioppa and Lucas 2007; Hernandez, Lucas, and Sanchez 2012; Joseph and Hung 2008; MacCalman, Vieira, and Lucas 2017).

Table 6. LHC Optimization Strategies. Adapted from Chen and Xiong (2017) and Viana (2013).

Researchers	Year	Algorithm	Objective Functions
Audze and Eglajs (1977)	1977	Coordinates exchange	Potential energy
Park (1994)	1994	2-stage exchange- and Newton-type	Integrated mean-squared error (IMSE) and entropy criterion
Morris and Mitchell (1995)	1995	Simulated annealing	ϕ_p criterion
Li and Wu (1997)	1997	Columnwise-pairwise	ϕ_p criterion
Kennedy and Eberhart (1995)	1995	Particle swarm	ϕ_p criterion
Ye, Li, and Sudjianto (2000)	2000	Columnwise-pairwise	ϕ_p criterion and entropy
Fang and Lin (2003)	2003	Threshold accepting	Centered L2-discrepancy
Bates, Sienz, and Toropov (2004)	2004	Genetic algorithm	Potential energy
Jin, Chen, and Sudjianto (2005)	2005	Enhanced Stochastic Evolutionary (ESE)	ϕ_p criterion, entropy, and centered L2-discrepancy
Liefvendahl and Stocki (2006)	2006	Columnwise-pairwise and genetic algorithm	Minimum distance and potential energy
Van Dam et al. (2007)	2007	Branch-and-bound	l-norm and infinite norm distances
Cioppa and Lucas (2007)	2007	Columnwise-pairwise	Correlation and distance
Grosso, Jamali, and Locatelli (2008)	2008	Iterated local search and simulated annealing	ϕ_p criterion
Ranjan, Bingham, and Michailidis (2008)	2008	Branch-and-bound and genetic algorithm	Contour estimation
Joseph and Hung (2008)	2008	Smart-swap method (modified simulated anneal)	ϕ_p criterion and correlation
Dalbey and Karystinos (2010)	2010	Locality preserving	L2-discrepancy
Viana, Venter, and Balabanov (2010)	2010	Translational propagation	ϕ_p criterion
Vieria et al. (2011)	2011	Mixed integer programming	Correlation and D-optimality
Hernandez, Lucas, and Sanchez (2012)	2012	Mixed integer programming	Maximum absolute pairwise correlation
Zhu et al. (2012)	2012	Successive local enumeration	ϕ_p criterion
Chen et al. (2013)	2013	Particle swarm	ϕ_p criterion, Hamming distance
MacCalman, Vieira, and Lucas (2017)	2017	Genetic algorithm	Correlation (quadratic)
Chen and Xiong (2017)	2017	Multi-layer ESE	ϕ_p criterion

D. TYPES OF LHC DESIGNS

A direct contribution of this dissertation is the use of LHC designs for set creation and refinement. The way in which the design points are created (or sampled) is important because SBD considers each design point instead of relying on curve fits. Addressing the point cloud in SBD as opposed to the curve fit in RSM allows for further exploration of the design space and a better understanding of the impact of decisions (recall Chapter II Section H.7). Sampling of the design space is different for various types of LHCs and so yield different points. Table 7 summarizes several LHCs and lists general characteristics, strengths, limitations, and potential uses for each. Written descriptions with more detail are also included in this chapter.

Table 7. Comparison of LHC Designs

LHC Design	Short	Description	Strengths and Limitations	Uses
Random (Sacks, Schiller, and Welch 1989)	RLH	Original LHC design	- Easy to construct - Less expensive - Without replications - Good space-filling properties only for *n*>>*k* - Poor prediction if *n*~*k* - High probability of obtaining highly correlated designs	As a quick and inexpensive method for large sample sizes compared to the number of factors
Optimal (Morris and Mitchell 1995; Park 1994)	OLHC	Combines LHC with optimal design	- Well spread over the experimental region - Without replications - Often symmetric in low dimensions - Nearly optimal - Smaller prediction error than RLH - May not be globally optimal because restricted to optimal midpoint LHCs - Computationally expensive - More difficult to construct than RLH	If looking to optimize uniformity according to some space-filling criteria

LHC Design	Short	Description	Strengths and Limitations	Uses
Symmetric (Ye, Li, and Sudjianto 2000)	SLHD	Combines built-in design optimality with reasonable computing effort	- Each sample point is reflected through the center - Advantageous over RLHs in terms of entropy and minimum intersite distances - Orthogonality properties and reduced correlation - Flexible run size - Less computing time - Not all OLHCs are symmetric	Quick and simple design method with built-in space-filling properties and reduced correlation
Optimal Symmetric	Optimal SLHD	Improves SLHD according to distance or entropy criterion	- Same as SLHD except better selection - Not all optimal SLHDs are symmetric	Quick and simple design method with reduced correlation and improved space-filling properties
Binning Optimal Symmetric (Dalbey and Karystinos 2010)	BOSLHS	Combines jittered sampling with LHS New metric of space-filling	- Space-filling in full dimensional space - Space-filling in each individual dimension - Shares SLHD characteristics - Lower discrepancy, faster decay of discrepancy - Can have sample points in any power of two ≥ twice the number of input dimensions - Suggest discarding excess dimensions from the next higher power of two design if high quality designs for non-power of two values are needed - Binning optimality considered a “rather weak sort of optimality” (Dalbey and Karystinos 2010, 6) - Can potentially improve discrepancy by “making low dimensional projections binning optimal and [making] better selection of orientations of adjacent bins” (Dalbey and Karystinos 2010, 22)	Similar applications as SLHD; obtains well-distributed points in each bin and then reflects them; doubles the number of points for each iteration as compared to SLHD

LHC Design	Short	Description	Strengths and Limitations	Uses
Orthogonal (Ye 1998)	OLHD	LHC designs with zero correlation	- Zero correlation - Less flexible run sizes - Number of runs confined to powers of 2 - Lack of methods addressing sample sizes of *n*=18-24 - Scarcity of combinations of number of runs and factors in practice	When zero correlation is desired; not necessarily space-filling
Optimal Orthogonal (Ye 1998) Orthogonal Maximum (Joseph and Hung 2008)	OMLHD	Minimizes correlation while maximizing intersite distances	- Minimizes correlation while maximizing space-filling properties - More general than OLH designs - OMLHD exist for all *n* and *k* - Might not necessarily be zero correlation - OLH exist for only certain *n* and *k*	If space-filling desired in zero correlation designs, e.g., injection molding cooling system
Orthogonal Array-Based (Owen 1992; Tang 1993) Strong OALHD (He and Tang 2013) Star-based OALHD (Ranjan and Spencer 2014)	OALHD NOA (Nearly Orthogonal OALHD) SOA (Strong OALHD)	LHC designs based on orthogonal arrays	- Uniformity in *t*-dimensional margins as opposed to only one-dimensional projections - Better space-filling properties than RLH - Built-in multivariate uniformity - SOAs are even better-space-filling - Star-based LHC designs or geometric NOAs have less stringent existence conditions - Fairly simple to construct - Can be symmetric - Arbitrary numbers of points and dimensions may not exist	If *r*-variate uniformity is desired; Geometric NOAs good for multi-stage factorial experiments with randomization restrictions
Nearly Orthogonal (Cioppa and Lucas 2007; Hernandez 2008; Hernandez, Lucas, and Carlyle 2012; MacCalman, Vieira, and Lucas 2017)	NOLH	Absolute maximum pairwise correlation no greater than 0.05	- Minimum pairwise correlation between factors - Number of factors can be increased without increasing the number of runs - Exist for most run-variable combinations - Near orthogonality between all main, quadratic, and two-way interaction terms - Only for numerical factors - Rounding errors when discrete-valued factors are used	Experimental settings where users precisely choose input variables and outputs exactly determined, e.g., training in peacekeeping/ stability ops

LHC Design	Short	Description	Strengths and Limitations	Uses
Nearly Orthogonal and Balanced (Vieira, Sanchez, and Kienitz 2011; Vieira 2012; Vieira et al. 2013)	NOB	Absolute maximum pairwise correlation no greater than 0.05 Number of objects in each of the levels of each column in equal	- Same as NOLH except also balanced - Can be used for mixed designs with categorical, discrete, and continuous factors - Orders of magnitude more efficient with an acceptable number of design points compared to OA or full-factorial designs - "Allows correct analysis of non-normal heteroscedastic experiments" (Vieria, Sanchez, and Kientitz 2011, 3606) - High-order aliasing effects	Design problems with qualitative and quantitative inputs, e.g., defense and national security design problems (life cycle mgt., sub launched unmanned air systems)
Sequential and Batch (Duan et al. 2017; Loeppky, Moore, and Williams 2010)	bLHD (batch LHD) bMmLHD (batch maximum distance LHD)	Inputs chosen sequentially based on learnings from previous runs; augmented runs based on each batch stage	- Ability to terminate experiments early - Good space-filling and orthogonality properties - Qualitative or quantitative factors	General stage, sequential, and batch operations
Probability-Based (Hung, Amemiya, and Wu 2010; Hung 2011)	PLHD BPLHD	Provide space-filling designs in slid-rectangular regions	- Applies to non-rectangular (slid rectangular) experimental regions - Quantitative or qualitative factors - Adaptive based on some criterion	Good for hot spot applications, clusters, ecologic studies, environmental pollution studies, sparse vegetation ecosystems, and population density estimates

LHC Design	Short	Description	Strengths and Limitations	Uses
Sliced (Qian and Wu 2009; Qian 2012)	SLHD	LHC design that can be partitioned into smaller LHC designs	- Accommodates both quantitative and qualitative factors - Can be partitioned into smaller LHC designs - Each slice has good uniformity - Overall design has good uniformity when collapsed - Run size must be a multiple of that of each slice	Good for batch operations, evaluating computer models, cross-validation, data pooling, and stochastic optimization
Sliced Orthogonal and Nearly Orthogonal (Wang, Yang, and Liu 2017)	---	Minimize correlation in sliced LHC designs	- First- and second-order orthogonal and nearly orthogonal variants - Nearly orthogonal can accommodate an equal number of runs and factors - Small correlation between columns of slices - Nearly orthogonal SLHDs need to have fold-over structure	Batch operations and other sliced applications when control of correlation is desired
Sliced Full Factorial (Duan et al. 2017)	sFFLHD	Extension to the concept of sliced OALHDs	- "Good univariate stratification achieved at all stages of the sequential design" (Duan et al. 2017, 11) - Orthogonality achieved at certain stages pertaining to big and intermediate grid levels - Favorable sampling and fitting qualities - A priori selection of the total number of runs is not required - Batches can be added indefinitely - Can terminate early - Certain golden stages where the sFFLHD becomes a FFLHD	Good for batch operations and anything with stages, e.g., water flow through a borehole
Nested (Qian 2009) Nested OALHD (He and Qian 2011) Flexible NLHD (Chen and Xiong 2017)	NLHD	Multiple layers of LHCs nested together	- Stratification in the univariate and bivariate margins - Flexible NLHDs can be constructed with any number of factors, layers, and run sizes - Each layer has good one-dimensional projection and uniformity properties	Suitable for many multi-level (or multi-fidelity) sequential computer experiments

1. Random

LHCs in which the columns are randomly constructed as was originally proposed in LHC design are referred to as random LHCs (RLH) (Sacks, Schiller, and Welch 1989). The random determination of point locations results in decent space-filling properties and orthogonal behavior if the number of points is much larger than the number of input factors ($n >> k$), but can act poorly in estimation and prediction otherwise. Hernandez, Lucas, and Sanchez (2012) describe the degree of non-orthogonality expected from this randomness in terms of correlation and explain the high probability of obtaining highly correlated designs – for 1,000 4x3 RLH design matrices, over 77% had correlations greater than 0.8 or less than -0.8, and almost 25% had at least one pair of columns with perfect correlation.

2. Optimal

Regular LHC designs are not related to any optimality of design and can be poor estimators of the response, especially at the untried input sites. Optimal LHC designs (OLHC) combine the advantages of optimal design and LHC design. Park (1994) considers optimal LHC designs by minimizing IMSE or maximizing entropy and finds they have good coverage of the experimental space without replications, are often symmetric in low dimensions, and are nearly optimal with smaller prediction error than regular LHC or factorial designs. The ease of construction is lost and the computational cost is expensive for OLHCs as compared to regular LHCs, especially for large sample sizes or high number of input values, hence the development of a fast algorithm for finding the OLHC becomes critical. Park (1994) uses a two-stage exchange and Newton-type algorithm that first restricts the interest to optimal midpoint LHCs, which makes it unlikely that the resulting OLHC will be globally optimal, although Park states it is not far from the true optimal LHC when $n >> k$. The OLHCs described by Park (1994) and Morris and Mitchell (1995) offer a compromise between the good projective properties of LHCs and an objective criterion for optimization.

3. Symmetric

The purpose of symmetric LHC designs (SLHD) is an LHC with inherent design optimality and reasonable computing effort. Each sample point is reflected through the center of the design in SLHDs, which makes them advantageous over regular LHCs in terms of entropy and also statistically significantly better with respect to minimum intersite distances (Ye, Li, and Sudjianto 2000). The symmetry of SLHDs helps reduce the correlation between input dimensions and provides some orthogonality properties; “the estimation of quadratic effects and bi-linear interactions is uncorrelated with the estimation of the linear effects” for each variable (Ye, Li, and Sudjianto 2000, 156). SLHDs can be considered generalizations of orthogonal LHC designs (OLHD) with flexible run size and reduced search time. The globally optimal LHC is not always an SLHD, however “foldover designs” have been previously recognized by Morris and Mitchell (1995) as those that are both optimal and symmetric.

4. Optimal Symmetric

Optimal SLHDs improve upon the selection of an SLHD by optimizing the entropy or minimum distance criterion through a searching algorithm. Optimal SLHDs still carry the benefits of the LHC structure and orthogonality properties inherited from SLHDs and are expected to have reduced search times in comparison to regular LHCs.

5. Binning Optimal Symmetric

Binning optimal symmetric LHS designs (BOSLHS) combine the appealing aspects of jittered sampling and regular LHS in that they are space-filling in the full dimensional space and in each dimension individually. They also share the symmetric structure of SLHDs and describe the binning optimality metric for quantifying the space-filling property. Dalbey and Karystinos (2010) demonstrate that BOSLHS designs are superior to regular LHS, tensor product, Monte Carlo, and jittered sampling with regard to its space-filling characteristics, orthogonality, t-quality metric (a measure of the computing cost), and degree of non-binning optimality (a measure used to compare designs that are not binning optimal). BOSLHS designs also achieve a lower discrepancy and faster decay of discrepancy and can have sample points in any power of two greater than or equal to twice the number of input dimensions. Dalbey and Karystinos (2010) suggest discarding excess dimensions if high quality designs for non-power of two values are needed. They also admit “binning optimality is a rather weak sort of optimality” that verifies the right number of bins but not that the point location within them is optimal (Dalbey and Karystinos 2010, 6). BOSLHS designs can potentially improve in discrepancy by making low dimensional projections binning optimal and better selecting the orientations of adjacent bins.

6. Orthogonal

OLHDs have zero pairwise correlation between any two factors (Ye 1998), which is desirable in regression analysis for giving uncorrelated estimates of the coefficients, avoiding partial confounding, and enhancing the performance of other procedures (e.g., regression tree) (Hernandez, Lucas, and Carlyle 2012). In the recent past, the application of OLHDs was limited due to having less flexible run sizes and being confined to a power of two number of runs. This is especially problematic when a large number of runs is needed to handle a large number of factors, however, the library of OLHDs continues to grow through continued studies on their construction. Additionally, OLHDs are often hindered by the algorithms used to create them in practice through a scarcity of available design dimensions (combinations of number of runs and factors); Hernandez, Lucas, and Carlyle (2012) illustrate the constraints of the current algorithms by showing a lack of methods available to address sample sizes of eighteen to twenty-four for any number of factors.

7. Optimal Orthogonal

LHC optimization is generally achieved by improving space-filling properties or eliminating or reducing correlation. Although the columns in OLHDs are constructed with zero correlation, good space-filling properties are not necessarily obtained. Optimization of OLHDs can be accomplished by pursuing a space-filling criterion. Maximum distance criteria are advantageous because they do not have the problem of choosing the correlation functions that is present with model-based criteria (i.e., IMSE, entropy). Joseph and Hung (2008) describe a multi-objective optimization approach for orthogonal-maximin LHC designs (OMLHD) that combines correlation and distance performance measures.

8. Nearly Orthogonal

Cioppa and Lucas (2007) construct efficient, space-filling nearly orthogonal LHCs (NOLH) with a maximum absolute pairwise correlation less than or equal to 0.03. Hernandez (2008) defines its NOLH designs to be no greater than 0.05 between factors; although arbitrary, there are no adverse multicollinearity effects observed for designs with such measures. By allowing small amounts of correlation (as opposed to zero for OLHDs), the number of factors can be increased without increasing the number of runs. All of the NOLHs described are constructed for continuous-valued factors, and NOLH application to discrete-valued factors requires rounding. Although some rounding is acceptable, the near-orthogonality of the design can be ruined if there are too many factors that have only a few value levels. Hernandez, Lucas, and Carlyle (2012) develop a mixed integer programming algorithm that generates NOLHs for most determinate run-variable combinations, including fully saturated designs, and can accommodate changing experimental conditions and runs. MacCalman, Vieira, and Lucas (2017) use a genetic algorithm to construct second-order NOLHs with near orthogonality between all arrangements of the main, quadratic, and two-way interaction terms.

Vieira et al. (2011) and (2013) describe improved efficient, nearly orthogonal, nearly balanced (NOB) mixed designs where the maximum absolute pairwise correlation between any two design columns is minimal (less than 0.05), and the number of occurrences of each distinct factor level is nearly equal (less than 20% imbalance, but problem-specific). NOB mixed designs allow for different factor types (categorical, discrete, and continuous) and different numbers of levels across the various factors. Efficient designs have an acceptable number of design points, which is also problem-specific.

9. Orthogonal-Array-Based

Owen (1992) and Tang (1993) propose using OAs for constructing LHC designs that achieve stratification and uniformity in t-dimensional margins when OAs of strength t are used, as opposed to only one-dimensional projections for regular LHCs (recall that an LHC is an OA of strength 1). OA-based LHC designs (OALHD) have better space-filling properties than RLHs and can also be further optimized according to some criterion for expanding coverage of the experimental region. The multivariate uniformity properties are inherent in OALHDs, while other LHC designs must often strive for better uniformity through supplemental control of correlation. OALHDs can also be symmetric (equal factor levels) or asymmetric (mixed levels). OALHDs are simple to construct as long as the OAs exist; they may not exist for arbitrary numbers of points and dimensions. Ranjan and Spencer (2014) present a space-filling LHD based on nearly OAs (NOA) derived from geometric star structures, sometimes referred to as star-based LHDs or geometric NOAs, that are generalizations of OALHDs with less stringent existence conditions.

Strong OALHDs (SOA) of strength t exhibit even better space-filling properties than comparable OALHDs in all g-dimensional projections for any $2 \leq g \leq t$-1 (He and Tang 2013). SOAs are motivated by quasi-Monte Carlo methods, but differ in that they stress low-dimensional margins, achieve uniformity in each t-dimensional margin, have uniformity properties that can be considered finite sample properties (because each division contains only one sample point), and are more general in terms of run sizes.

10. Sequential and Batch

Sequential design strategies are desirable because they offer the ability to terminate experiments early if certain stopping criteria are reached or to cease collecting data if some measure of prediction precision (variance) is achieved. Loeppky, Moore, and Williams (2010) introduce batch sequential designs where additional batches of design points can successively be added to computer experiments. Bin-based sequential design is described that yields aggregate LHS designs based on the use of bins to construct augmented sets of runs. Ideally, the resulting design has good space-filling and orthogonality properties after the addition of each batch. Duan et al. (2017) compare the performance of some common LHS designs as they are implemented in a batch sequential manner, including a batch sequential maximin distance LHC design (bMmLHD) and a RLH batch sequential LHC design (bLHD).

11. Probability-Based

Hung, Amemiya, and Wu (2010) introduce probability-based LHC designs (PLHD) to address non-rectangular experimental regions (called slid-rectangular) where the "desirable range of one factor depends on the level of another factor" (qualitative or quantitative factors) (Hung 2011, 1). Slid-rectangular experimental regions can occur in computer experiments if manufacturing limitations prohibit some combinations of variables. They are frequently encountered in ecologic studies, environmental pollution studies, sparse vegetation ecosystems, hot-spot applications, and population density estimates. Most existing space-filling designs are not applicable to slid-rectangular regions because they assume rectangular experimental regions. PLHDs account for the slid-rectangular structure during design construction so desirable space-filling properties are maintained with respect to the one-dimensional balance property that uniformly spreads the design points (by equally spacing the number of observations along the interval).

PLHDs have been improved by incorporating a proportional balance property that addresses the problem of how the total number of observations in any one experimental interval is not necessarily proportional to the length of that interval (Hung 2011). These balanced PLHDs (BPLHD) are constrained PLHDs that can achieve both one-dimensional balance and proportional allocation properties if the inclusion probabilities are highly correlated.

Adaptive PLHDs and BPLHDs (Hung 2011) can be obtained for the efficient identification of hot spots typically clustered in a few locations. Based on a number of initial observations, if a particular design point yields an acceptable response with respect to a given criterion, additional points from within the general area (neighborhood) of that particular point are added to the sample. The addition of points continues until none meet the criterion, and a final design containing every point satisfying the condition in a particular neighborhood within the slid-rectangular region results.

12. Sliced

Qian and Wu (2009) present sliced LHC designs (SLHD) to accommodate both quantitative and qualitative factors. SLHDs can be partitioned into slices of smaller LHC designs where: 1) the quantitative factors are spread evenly over each slice for any qualitative factor level combination (one dimensional uniformity); and 2) the whole design is space-filling when the quantitative points are collapsed over all slices (higher-dimensional uniformity). The approach to constructing SLHDs starts with generating a sliced OALHD for the quantitative factors (using field-to-field projection methods, Bush's method, Rao-Hamming method, difference matrices, etc.) and then partitioning it into groups according to the different level combinations of the qualitative factors. SLHDs are easy to construct, can accommodate any number of factors, lead to reduced variance, are flexible in run size if it is a multiple of that of each slice, are desirable for batch operations, and are useful for evaluating computer models, cross-validation, data pooling, and stochastic optimization (Qian 2012; Qian and Wu 2009).

Wang et al. (2017) propose the systematic construction of first- and second-order orthogonal and nearly orthogonal SLHDs. First-order orthogonal SLHDs are defined by those that have orthogonality between any two columns of each slice, which ensures independent estimates of linear effects when fitting first-order models. For an SLHD to be considered second-order orthogonal, each slice must satisfy the first-order condition and also exhibit orthogonality between any column and the "elementwise product of any two columns, identical and distinct" (Wang et al. (2017, 111). The second-order criterion ensures orthogonality between all linear and quadratic effects when fit to a second-order model. Nearly orthogonal SLHDs can be constructed by adding columns to an original nearly orthogonal SLHD with foldover structure and are beneficial when the number of factors is too high for orthogonal SLHDs. Nearly orthogonal SLHDs can accommodate an equal number of factors and runs and show small correlation between columns of slices.

13. Sliced Full Factorial

Duan et al. (2017) present a batch sequential experiment design method that combines SLHD (Qian and Wu 2009) and batch (Loeppky, Moore, and Williams 2010) concepts. Sliced full-factorial LHC designs (sFFLHD) are constructed based on the concept of sliced OALHDs. Good univariate stratification and orthogonality are achieved at all stages of the sequential design, favorable sampling and fitting qualities are observed (variance reduction similar to ordinary LHCs, or OALHDs when formed), *a priori* selection of the total number of runs is not required, and batches can be added indefinitely.

An L-level FFLHD is defined by Duan et al. (2017, 12) as one in which: "1) every dimension of X is partitioned into L evenly spaced levels; and 2) when X is projected onto any dimension, precisely one point falls within *n* equally spaced levels." One slice from the FFLHD is sampled at each batch stage and constructed using a series of OAs. The sequential sampling process creates a big grid (preserves orthogonality), intermediate grid (allows orthogonality on more than L levels to ensure it is a fractional factorial design), and small grid (preserves LHC projection properties). Batches continue to be added until a stopping criterion is reached or it is at its best space-filling ability as an FFLHD (referred to as the golden stage).

14. Nested

Qian (2009) introduces a nested LHC design (NLHD) that contains multiple layers of LHCs nested together with uniformity in one-dimensional projection. He and Qian (2011) introduce nested OALHDs to improve upon NLHDs by achieving stratification in the univariate and bivariate margins of the large design (as opposed to just the univariate margin). NLHDs can be constructed with any number of layers, but face limitations in practice because the run size in each larger LHC design must be a multiple of that in a smaller one, which can lead to such large run sizes that are infeasible to conduct with current computational resources. Chen and Xiong (2017) propose flexible NLHDs that avoid the restrictions of NLHDs. Every layer of a flexible NLHD simultaneously possesses good one-dimensional projection and space-filling properties and maximum flexibility in the number of factors, layers, and run sizes. These features of flexible NLHDs make them suitable for many multi-level (or multi-fidelity) sequential computer experiments. Potential ways of extending flexible NLHDs include constructing flexible OALHDs to achieve higher-dimensional projection properties and constructing flexible SLHDs to allow for quantitative and qualitative factors, both of which would also rid the sample size restrictions.

E. SIMULATION

For any design problem, selection decisions are based on the output values (responses or y-values); not the input values (factors or x-values). For example, traditional approaches using RSM are based on curve-fitting the y-values, and point selection in PBD is based on OMOE, or the ability of the design to meet the required MOEs, which are outputs based on input factors. Selection decisions in SBD are also made on the response variables, and a simulation tool certainly simplifies the revelation of these responses.

In this dissertation, an NOB LHS technique (Vieira 2012; Vieira et al. 2013) is applied to synthesize the design points. An NOB LHC design was selected because of its ability to accommodate both discrete (engine type and sensor EO and IR FOVs and resolutions) and continuous variables (wingspan and operating altitude) in the UAS example engineering problem and different numbers of levels across the various factors (e.g., six levels for both sensor FOVs, nine levels for both sensor resolutions, and two levels for engine type). It also has minimal correlation and near-orthogonality based on 512 design points, which is an adequate number of points for understanding the UAS design space, especially for the refined set spaces when an additional 512 design alternatives are generated in a reduced space. The space-filling properties of the NOB LHC design guarantee adequate sampling of the experimental region for any given specialty and the system design space. It also provides distinct solutions as is required by the first principle of SBT (Ghosh and Seering 2014). Another advantageous reason for selecting the NOB LHC design is its free, available, user-friendly tool (Vieira 2012).

The design points generated through the NOB LHS tool serve as inputs to a simulation tool (Small 2018) to determine feasibility and produce the responses required to check for viability. The simulation tool is capable of using built-in physics equations to check each possible design solution for feasibility. In other words, the NOB LHS tool makes space-filling combinations of input factors that represent design solutions, but it has no way of knowing if these design solutions are feasible or not; hence, the design solutions are inputted into a simulation tool, and it does the checking.

Just because design solutions are feasible does not automatically mean they are viable. The simulation tool generates the performance, value (OMOE), and cost for every feasible design solution. SBD is applied to work towards a selection; whether it be the most straightforward way of eliminating variants that fall short of achieving every MOE, or whether it be through specialty iterations, expertise, and new knowledge, the ultimate decision is based on the responses that result from the input factors.

THIS PAGE INTENTIONALLY LEFT BLANK

V. UNMANNED AIR SYSTEM EXAMPLE DEMONSTRATION

The improved SBD implementation steps introduced in this dissertation are demonstrated using a UAS example where a defense contractor has a well-established UAS product line using piston engines, but recognizes the emerging market and advancing technologies for the use of electric engines. The contractor establishes two unique design teams (electric and piston) to compare the value of a small UAS required for surveillance missions that can detect enemy activity and is transportable, survivable, and capable of maneuvering to, scanning, and dwelling at an area of interest. The teams are geographically separated and work for different divisions, but have access to the same companywide simulation and analysis tools. Table 8 summarizes the functional and operational requirements along with threshold and objective values, including less-is-better (LIB) or more-is-better (MIB) annotations.

The progression of the UAS example is shown in Figure 30. For simplicity of example, the information in steps 1 through 7 applies to both the electric and piston teams. Beginning in step 8, the example continues with the execution of SBD by the electric team separately from the piston team (for three iterations). After the viable set space has been generated for the electric team in step 14, the example picks up with the execution of SBD by the piston team as it continues from step 8 (for two iterations). The viable set space is generated for the piston team in step 14, and then the viable set spaces for both teams are combined. The example continues as if the combined viable set spaces are an initial creation (step 7) and flows through to final selection (step 16).

Table 8. UAS Requirements

Functional Objective	Stakeholder Measure of Effectiveness (MOE)	Threshold Value	Objective Value	Swing Weight
Be Transportable	Portage Load (LIB) (Air Vehicle Weight+Sensor/Comms)	50 lbs	40 lbs	0.10
Maneuver to, Scan Across, and Dwell at Area of Interest	Time Required to Fly 10 km (LIB)	10 minutes	9 minutes	0.08
	Dwell Time (MIB)	30 minutes	45 minutes	0.07
	Time to Scan a 5 km x 5 km Search Box During the Day (LIB)	240 minutes	225 minutes	0.08
	Time to Scan a 5 km x 5 km Search Box at Night (LIB)	240 minutes	225 minutes	0.09
Be Survivable	Perceived Area of UAS at Operating Altitude (LIB)	12 ft^2	10 ft^2	0.04
	Difference between Operating Altitude and an Attack Helicopter at 1,000 m Altitude (MIB)	300 m	400 m	0.06
Detect Enemy Activity	Probability of Detecting a Human During the Day (MIB)	80%	82%	0.10
	Probability of Detecting a Human at Night (MIB)	60%	70%	0.12
	Probability of Detecting a Vehicle During the Day (MIB)	80%	82%	0.12
	Probability of Detecting a Vehicle at Night (MIB)	60%	70%	0.13

Note: The probabilities of detecting a vehicle during the day and night are omitted from further example because they always offer values around 80%.

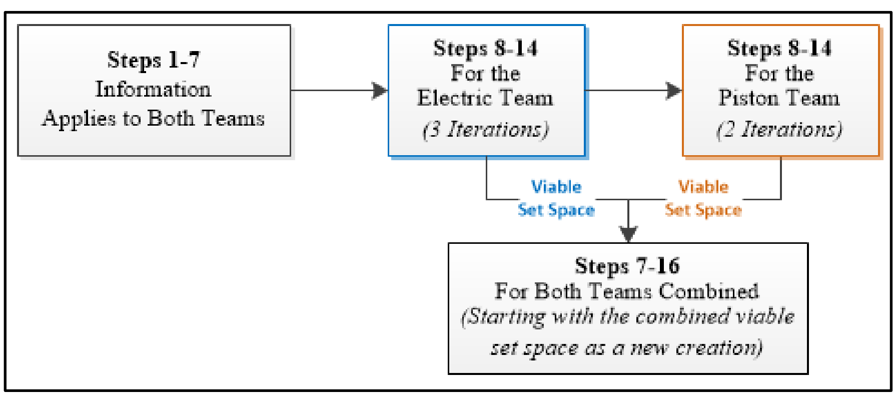


Figure 30. Progression of UAS Example

A. STEP 1: IDENTIFY ENGINEERING SPECIALTIES AND AVAILABLE ENGINEERING MODELING TOOLS

The seven unique specialties for this example are Structures, Propulsion, Sensors, Weight and Balance, Software, Systems Engineering, and Mission Assurance. Software and Systems Engineering are not elaborated on in this example, but are mentioned so as not to overlook the significance of both when it comes to ensuring net- and data-centricity, hardware compatibility, and requirements traceability.

B. STEP 2: IDENTIFY DESIGN FACTORS FOR EACH SPECIALTY

Each specialty considers the design problem from its own perspective by exploring various design factors within its expertise. The design factors that are determined by each specialty indicate the communication and feedback interfaces with other specialties that will be necessary throughout the design process. A table of dependencies is generated to show the magnitude of influence of each design factor as it applies to each specialty (see Table 9). It is created by:

- Assigning each MOP, MOE, and other important objectives or variables to its appropriate specialty and placing them in rows;
- Making a column for each input variable and every dependent variable (or response) that is derived from these input variables; and
- Categorizing each cell based on which specialty it belongs to and whether it is critical or informational.

Critical elements generally correspond to input variables and are needed to compute the response variables. Informational elements generally correspond to response variables and can be found if the critical (input variables) are known. If the input variables are unknown to a certain specialty, then the response variable of interest becomes critical. For example, Structures needs the sensor weight for its payload capacity considerations, but all of the relevant input variables belong to Sensors. Rather than make all these input variables critical to Structures, Sensors calculates the sensor weight and passes it onto Structures.

Table 9. Table of Dependencies

Eng. Spec-ialty	Perform. Function Reqts.	Operational Requirement (or Objective)	Wing-span	Engine Type	Oper-ating Alt.	UAS Wt.	Len-gth	Air-speed	Endur-ance	EO FOV	EO Res Horiz	EO Res Vert	EO_GSD_h	EO_GSD_v	EO_Human	EO_Vehicle	EO Ground Cover Rate	EO Ground Swath	IR FOV	IR Res Horiz	IR Res Vert	IR_GSD_h	IR_GSD_v	IR_Human	IR_Vehicle	IR Ground Cover Rate	IR Ground Swath	Dc_Human	Dc_Vehicle	N_50	Sen-sor Wt.	Sen-sor Ball Dia.	Dist. To 1000 m
Weight and Balance	Trans-port-table	Portage Load	*	*		*																											
Struc-tures		Payload Capacity	*	*		*																									*		
		Length	*	*			*																										
Prop-ulsion		Engine Type		*																													
Sensors	Scan Across	Time to Scan 5 km x 5 km search box during the day	*	*	*	*		*		*							*	*															
		Time to Scan 5 km x 5 km search box at night	*	*	*	*		*											*							*	*						
	Detect Enemy Activity	Probability of detecting a human during the day			*					*	*	*	*	*	*													*		*			
		Probability of detecting a vehicle during the day			*					*	*	*	*	*		*													*	*			
		Probability of detecting a human at night			*														*	*	*	*	*	*				*		*			
		Probability of detecting a vehicle at night			*														*	*	*	*	*		*				*	*			
		Sensor and comms weight									*	*								*	*										*	*	
Mission		Operating altitude			*																												
	Surviv-able	Perceived area of UAS at operating altitude	*	*	*		*																										
		Difference from attack helicopter at 1000 m			*																												*
	Move to and dwell	Time required to fly 10 km	*	*		*		*																									
		Dwell time	*	*		*			*																								

Engineering Specialty
User Input
Calculated Value
Critical
Belongs to that Engineering Specialty
Informational

Note: Critical elements generally correspond to User Input elements and are needed to compute the Calculated Value elements. Informational elements generally correspond to Calculated Value elements and can be found if the Critical elements are known.

1. Structures

The Structures specialty looks at design factors related to the fuselage, wings, and tail, such as wingspan, truss style, and material type. It also considers specific characteristics such as modulus, tensile strength, lift to drag forces, weight, air density, angle of attack, surface area, and payload capacity (see Figure 31).

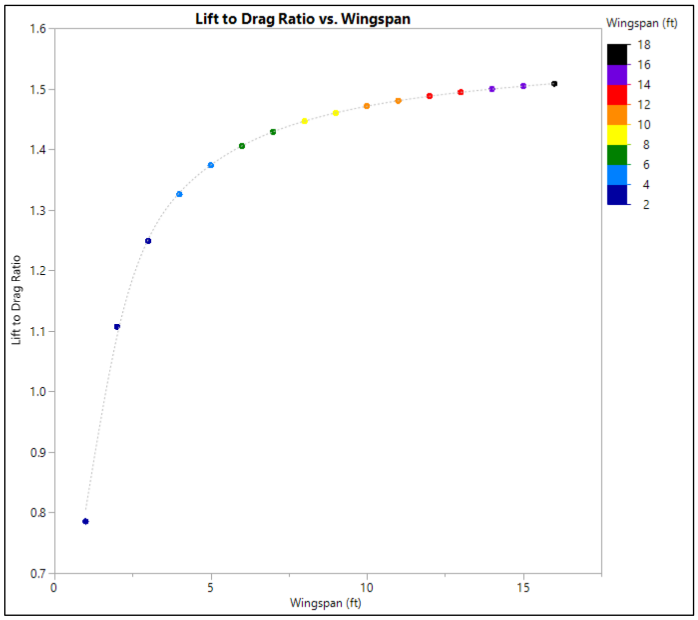


Figure 31. Example of a Structures Specialty Design Factor

2. Propulsion

The Propulsion specialty discusses a variety of possible propulsion systems (such as pulse jet, solar, advanced fuel cell, etc.), while also considering appropriate factors related to altitude limits, g-forces, fuel economy, power-to-weight ratio, size (to limit drag), reliability, in-flight shutdown rate, stealth, and energy and power densities (see Figure 32).

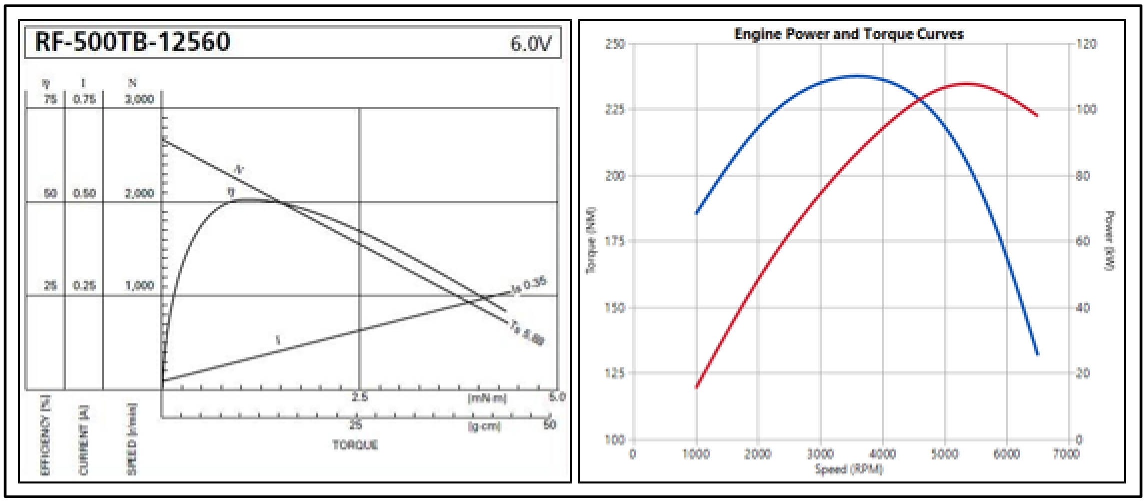


Figure 32. Propulsion Design Factors. Source (left): Pozmantir (2018); Adapted from (right) Sarvaiya (2016).

3. Sensors

The Sensors specialty looks at specifics related to charge-coupled device (CCD) and complementary metal-oxide semiconductor (CMOS) chip designs, image processing and analog-to-digital conversion (ADC), power requirements, image quality, light sensitivity, signal-to-noise ratio, dynamic range, chip size, sensor ball diameter, weight, individual pixel size, resolution, electromagnetic spectrum, and field of view (FOV). Important objectives to the Sensors specialty are the time required to scan a 5 kilometer (km) x 5 km search box using raster scan flight pattern at the proposed operating altitude and slant angle from normal to zero (day and night) and the probability of human and vehicle detection (day and night) (see Figure 33).

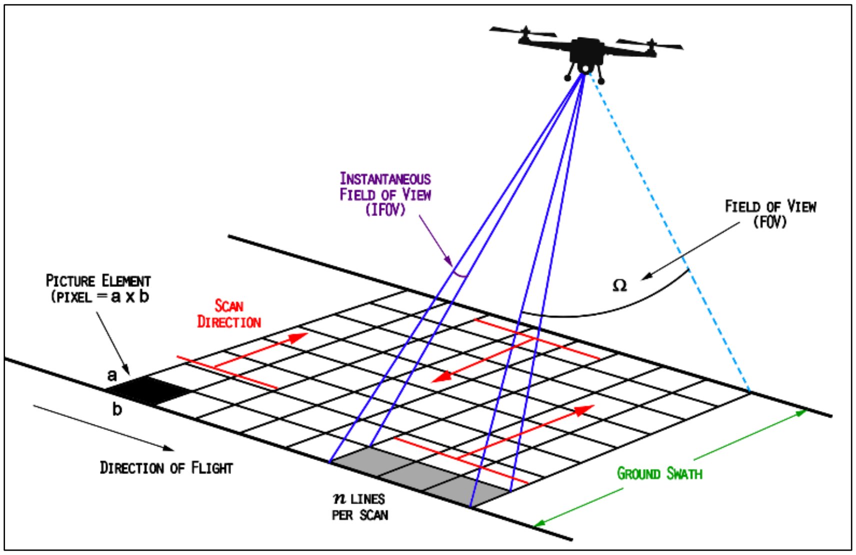


Figure 33. Sensors Specialty Design Factors. Adapted from Kumar (2015).

4. Weight and Balance

The Weight and Balance specialty keeps track of the total UAS weight to ensure the portage load requirement is met. For this example, the portage load includes the combined weight of the payload (which is solely sensor/comms in this example) and air vehicle, but neglects any additional factors, such as packaging.

5. Mission Assurance

The Mission Assurance (Mission) specialty looks at operational requirements and relevant factors to ensure the mission can be completed successfully. Several system-level objectives (measures of effectiveness, MOEs) fall under the Mission specialty, including survivability metrics like the perceived area of the UAS at operating altitude and the distance between the UAS operating altitude and an attack helicopter at an altitude of 1,000 meters (m); and maneuver-to and dwell metrics like the time required to fly 10 km and dwell time. Operating altitude, endurance, and airspeed important design factors for the Mission specialty.

C. STEP 3: SPECIFY DESIGN FACTOR VALUES

Specification of every design factor considered by the specialties is beyond the scope of this example, but each specialty explores the possible ranges of values for each design factor of interest. Simplified to convey the intent of this process step, Structures specifies wingspans from 1–18 ft and plastic, aluminum, and titanium materials. The context of this example has narrowed the engine type down to either electric or piston. Sensors looks at CCD and CMOS chips, FOVs from 15–180 degrees, ball diameters from 0–12 inches, and resolutions from 200–4,000 pixels. Mission considers operating altitudes from 100–1,000 m, endurances of 1–240 minutes, and airspeeds from 10–80 knots.

D. STEP 4: CREATE THE SPECIALTY SET SPACE

Each specialty synthesizes its design factors over the full ranges to create the specialty set space. For example, Structures makes all combinations of wingspans and materials (e.g., 1 ft wingspan made of plastic; 1 ft wingspan made of aluminum; 1 ft wingspan made of titanium; 2 ft wingspan made of plastic; etc.). Sensors makes all combinations of ball diameters, FOVs, and resolutions for both CCD and CMOS chips. Mission makes all combinations of operating altitude, endurance, and airspeed values.

E. STEP 5: EXPLORE THE SPECIALTY SET SPACE

Each specialty explores its synthesized variants and determines what is feasible and preferred. For example, Structures finds the variants with the largest wingspans are too heavy to be considered man-portable. Sensors discovers variants with highest FOV and lowest ball diameter are infeasible. Mission also notices infeasibilities at the extremes; the highest endurance values cannot be achieved at the highest airspeeds and the lowest operating altitudes are unrealistic.

F. STEP 6: COMMUNICATE THE SPECIALTY SET SPACE PREFERENCES

Each individual specialty passes on the sets (or ranges) of its desired design factors and any other important information to the lead systems integrator for consolidation. Table 10 includes the combined information: the discrete and continuous design factors represent measures of performance (MOPs) that serve as inputs to the simulation tool; and the awareness design factors represent important design variables or constraints.

Table 10. UAS Design Factors

Engineering Specialty	Design Variable or Measure of Performance (MOP)	Options
Structures	Wingspan (continuous)	● 1 ft - 16 ft
Propulsion	Engine (discrete)	● Electric ● Piston
Sensors	Electro-Optical (EO) Sensor Resolution (discrete)	● 200 Pixels x 200 Pixels ● 400 Pixels x 400 Pixels ● 600 Pixels x 600 Pixels ● 800 Pixels x 800 Pixels ● 1,000 Pixels x 1,000 Pixels ● 1,200 Pixels x 1,200 Pixels ● 1,400 Pixels x 1,400 Pixels ● 1,600 Pixels x 1,600 Pixels ● 1,800 Pixels x 1,800 Pixels
	EO Sensor Field of View (discrete)	● 15 Degrees ● 30 Degrees ● 45 Degrees ● 60 Degrees ● 75 Degrees ● 90 Degrees
	Infrared (IR) Sensor Resolution (discrete)	● 200 Pixels x 200 Pixels ● 400 Pixels x 400 Pixels ● 600 Pixels x 600 Pixels ● 800 Pixels x 800 Pixels ● 1,000 Pixels x 1,000 Pixels ● 1,200 Pixels x 1,200 Pixels ● 1,400 Pixels x 1,400 Pixels ● 1,600 Pixels x 1,600 Pixels ● 1,800 Pixels x 1,800 Pixels
	IR Sensor Field of View (discrete)	● 15 Degrees ● 30 Degrees ● 45 Degrees ● 60 Degrees ● 75 Degrees ● 90 Degrees
Weight and Balance	UAS Weight (awareness)	● 50 lbs maximum
Mission	Operating Altitude (continuous)	● 300 m - 1000 m
	Airspeed (awareness)	● 60 knots maximum
	Endurance (awareness)	● 15 hours maximum

G. STEP 7: CREATE THE INTEGRATED SET SPACE BY INTERSECTION

The lead systems integrator creates the integrated set space by inputting all of the MOP options into an NOB LHC sampling tool (Vieira 2012; Vieira et al. 2013) to generate space-filling combinations of design factors that represent unique UAS alternatives (see Table 11). Each combination is then inserted into a UAS simulation tool (Small 2018) adapted to fit the purposes of this dissertation. Various response variables and MOEs are outputted through the simulation tool based on physics equations in addition to value—represented as OMOE—and cost (see Tables 12 and 13). Actual simulation of the design alternatives is necessary to ensure the viability of the potential design solution. Simply generating the design space only exposes the possible alternatives that can be later assessed for feasibility based on physics, but does not address the capacity of each alternative to accomplish the mission goals associated with the MOEs. To examine the ability of an alternative to achieve a mission requires a simulation model. The integrated set space represents the widest aperture of all input variables and is the bounded design space.

Table 11. NOB LHC Inputs

lo	0	15	15	200	200	1	300
hi	1	90	90	1800	1800	16	1000
decimals	0	0	0	0	0	0	0
discrete levels	2	6	6	9	9	continuous	continuous
Scenario	**Engine**	**EO FOV**	**IR FOV**	**EO Res**	**IR Res**	**Wingspan**	**Altitude**
1	0	60	60	400	1200	2	960
2	1	30	45	400	400	8	552
3	0	30	90	400	600	6	911
4	1	15	90	200	200	8	323
5	0	45	45	800	800	13	989
6	1	75	90	1200	1000	16	510
7	0	60	90	200	600	14	607
8	0	90	30	800	200	8	974
9	1	30	45	1200	200	3	458
10	1	60	90	400	400	7	451
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
510	1	30	15	800	1200	14	334
511	1	75	90	1200	400	4	959
512	0	90	90	1200	800	7	463

Note: For simplicity of example, the initial inputs into the simulation tool are the same for the electric and piston teams, but engine type is specified as either all electric (all zeros) or all piston (all ones).

Table 12. Example Simulation Output (Electric Team)

	Design Variable or MOP							Value and Cost		Example of Responses		Example of MOEs	
Index	Eng-ine	EO FOV (deg)	IR FOV (deg)	EO Res (deg)	IR Res (deg)	Wing-span (ft)	Altitude (m)	Value (OMOE)	Cost ($K)	Air-speed (knots)	UAS Weight (lbs)	Scan Time Day (min)	Time to Fly 10 km (min)
1	E	60	60	400	1200	2	960	49.43	140.23	27.43	3.51	87.41	11.81
2	E	30	45	400	400	8	552	44.68	138.11	35.93	11.31	250.04	9.02
3	E	30	90	400	600	6	911	44.16	138.17	33.09	8.71	164.48	9.79
4	E	15	90	200	200	8	323	43.39	137.69	35.93	11.31	869.69	9.02
5	E	45	45	800	800	13	989	56.58	142.35	43.01	17.81	75.41	7.53
6	E	75	90	1200	1000	16	510	57.12	148.87	47.26	21.71	71.84	6.85
7	E	60	90	200	600	14	607	46.93	139.27	44.43	19.11	85.33	7.29
8	E	90	30	800	200	8	974	42.37	138.56	35.93	11.31	37.97	9.02
9	E	30	45	1200	200	3	458	40.84	139.19	28.84	4.81	375.38	11.23
10	E	60	90	400	400	7	451	44.80	137.92	34.51	10.01	147.86	9.39
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
510	E	30	15	800	1200	14	334	44.09	146.09	44.43	19.11	334.16	7.29
511	E	75	90	1200	400	4	959	51.27	140.61	30.26	6.11	59.67	10.71
512	E	90	90	1200	800	7	463	59.20	144.73	34.51	10.01	83.16	9.39

Table 13. Example Simulation Output (Piston Team)

	Design Variable or MOP							Value and Cost		Example of Responses		Example of MOEs	
Index	Eng-ine	EO FOV (deg)	IR FOV (deg)	EO Res (deg)	IR Res (deg)	Wing-span (ft)	Altitude (m)	Value (OMOE)	Cost ($K)	Air-speed (knots)	UAS Weight (lbs)	Scan Time Day (min)	Time to Fly 10 km (min)
1	P	60	60	400	1200	2	960	59.89	141.23	31.03	10.28	77.26	10.44
2	P	30	45	400	400	8	552	51.28	140.35	47.59	26.36	188.76	6.81
3	P	30	90	400	600	6	911	51.37	140.00	42.07	21.00	129.38	7.70
4	P	15	90	200	200	8	323	51.46	139.93	47.59	26.36	656.56	6.81
5	P	45	45	800	800	13	989	58.61	145.62	61.39	39.76	52.83	5.28
6	P	75	90	1200	1000	16	510	57.41	143.72	69.67	47.80	48.73	4.65
7	P	60	90	200	600	14	607	49.25	142.74	64.15	42.44	59.10	5.05
8	P	90	30	800	200	8	974	50.51	140.80	47.59	26.36	28.66	6.81
9	P	30	45	1200	200	3	458	48.48	140.40	33.79	12.96	320.43	9.59
10	P	60	90	400	400	7	451	53.61	139.95	44.83	23.68	113.83	7.23
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
510	P	30	15	800	1200	14	334	46.78	149.56	64.15	42.44	231.42	5.05
511	P	75	90	1200	400	4	959	58.65	142.03	36.55	15.64	49.40	8.86
512	P	90	90	1200	800	7	463	66.21	146.77	44.83	23.68	64.01	7.23

H. SBD CONTINUED FOR THE ELECTRIC TEAM

1. Step 8: Explore the Integrated Set Space

With the integrated set space created, each specialty now takes the opportunity to explore it from its own perspective by:

- Looking at the input factors (MOPs) it is responsible for and the operational requirements (MOEs) that are a function of these input factors;

- Considering what part of the input factor ranges can be eliminated; and
- Identifying important information and new discoveries.

a. Structures

Knowing the engine type is electric, Structures gains knowledge about how wingspan relates to length, endurance, airspeed, air vehicle weight, and max payload as shown in Figure 34. In short, the engine type serves as a constraint on wingspan. From the Structures perspective, the whole range of wingspans it put forward to the team is acceptable, but payload capacity is more limited at lower wingspans and is less than a half a pound at 1 ft wingspans.

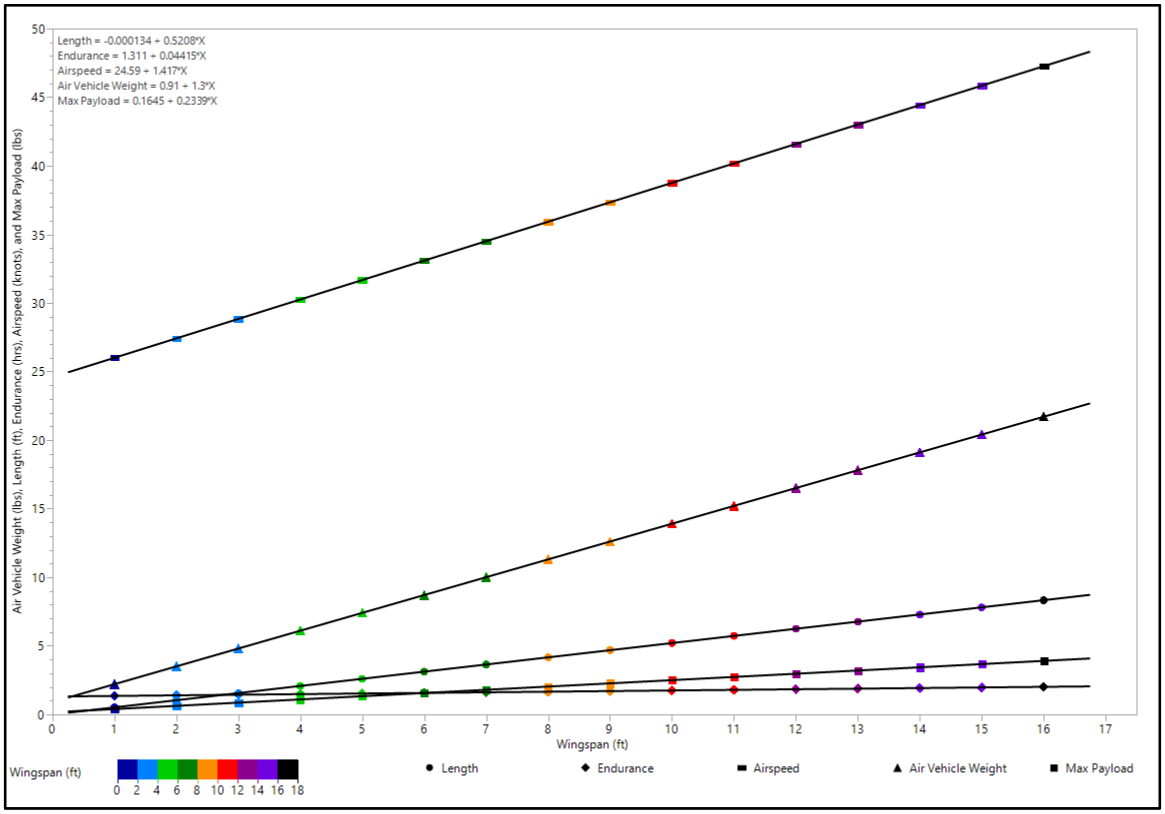


Figure 34. Integrated Set Space 1: Structures Perspective on Wingspan (Electric Team)

b. *Weight and Balance*

Weight and Balance identifies the same relationships as Structures does for max payload and air vehicle weight versus wingspan. It also finds that nearly 80% of the design variants are infeasible because the max payload is exceeded.

c. *Sensors*

Sensors gets a better understanding of its suite of options based on wingspan by looking at the max payload for each wingspan as compared to the possibilities it presented to the team. Sensors uses the integrated information to generate Figure 35, which teaches them the total (EO+IR) resolutions cannot be greater than 1,000 pixels. Figure 35 is used as follows:

- On the left plot, start at the max payload axis (x-axis) and trace the vertical line for the wingspan of interest until it intersects the horizontal line of equal value on the sensor/comms weight axis (y-axis on the left plot);

  e.g., for an 11 ft wingspan, find the second red line on the x-axis of the left plot (approximately 2.75 lbs max payload) and trace it upwards until it intersects the second red line, which corresponds to an equivalent sensor/ comms weight of 2.75 lbs on the y-axis of the left plot;

- Follow that horizontal line to the right until it intersects the curve on the right plot;

  e.g., follow the red line over from the y-axis of the left plot to the right plot until it intersects the black curve (this is a triple point with two red lines and one black curve);

- Once at the intersection point, follow the vertical line downward to the total (EO+IR) resolution axis;

  e.g., follow the red line down until it intersects the x-axis of the right plot at a total (EO+IR) resolution of around 985 pixels;

- This point represents the maximum combined (EO+IR) resolution allowable for the wingspan of interest.

i.e., 985 pixels is the total (EO+IR) resolution allowed for an 11 ft wingspan, however, since resolutions are in increments of 200 pixels, the maximum total (EO+IR) resolution allowed is 800 pixels.

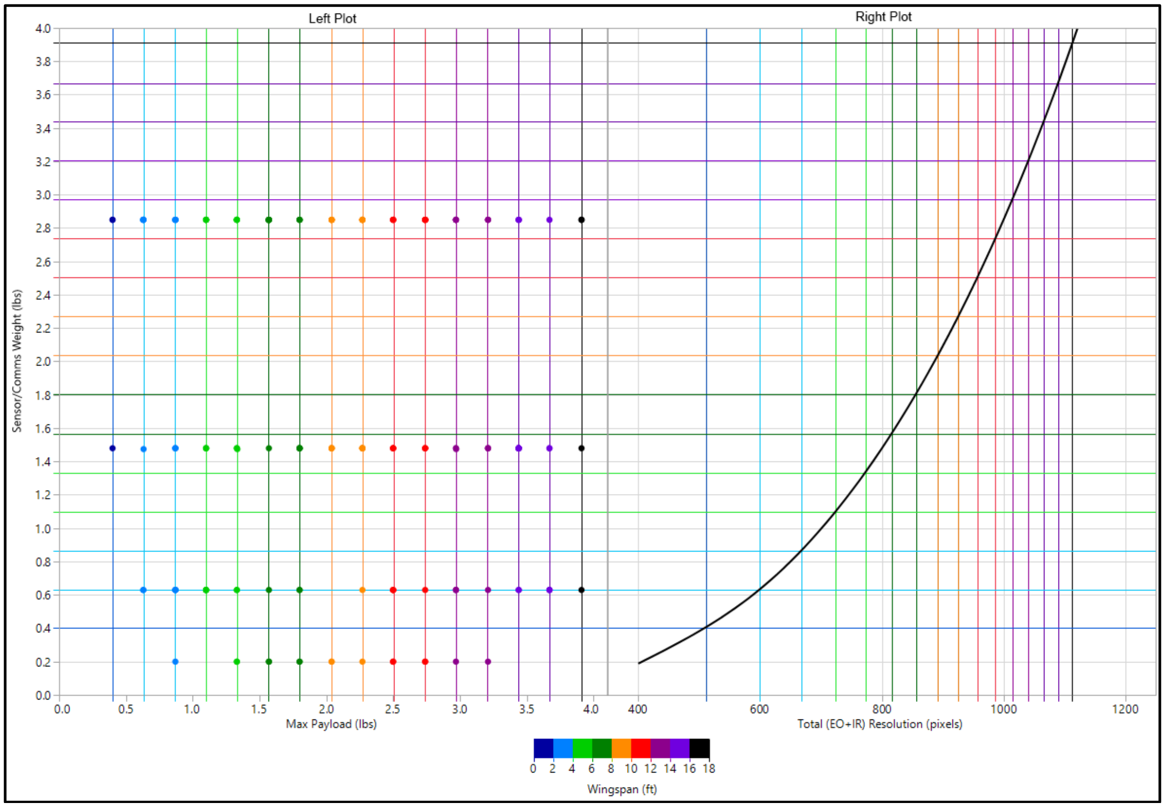


Figure 35. Integrated Set Space 1: Sensors Perspective on Resolution (Electric Team)

d. Mission

Mission first looks at the operating altitude MOP it is responsible for and recognizes operating altitude is inversely proportional to the scan time, detection, perceived UAS area, and difference from attack helicopter MOEs (see Figure 36). Mission identifies a potential tradeoff negotiation that may be required for stakeholders concerning the perceived UAS area and difference from attack helicopter MOEs, since they work opposite of each other. To be survivable, stakeholders require a distance of at least 300 m from an enemy attack helicopter operating at 1,000 m and a perceived UAS area of less than 12 ft^2; in order to achieve less perceived UAS area, higher altitude is needed, while achieving greater distance from an attack helicopter requires less altitude.

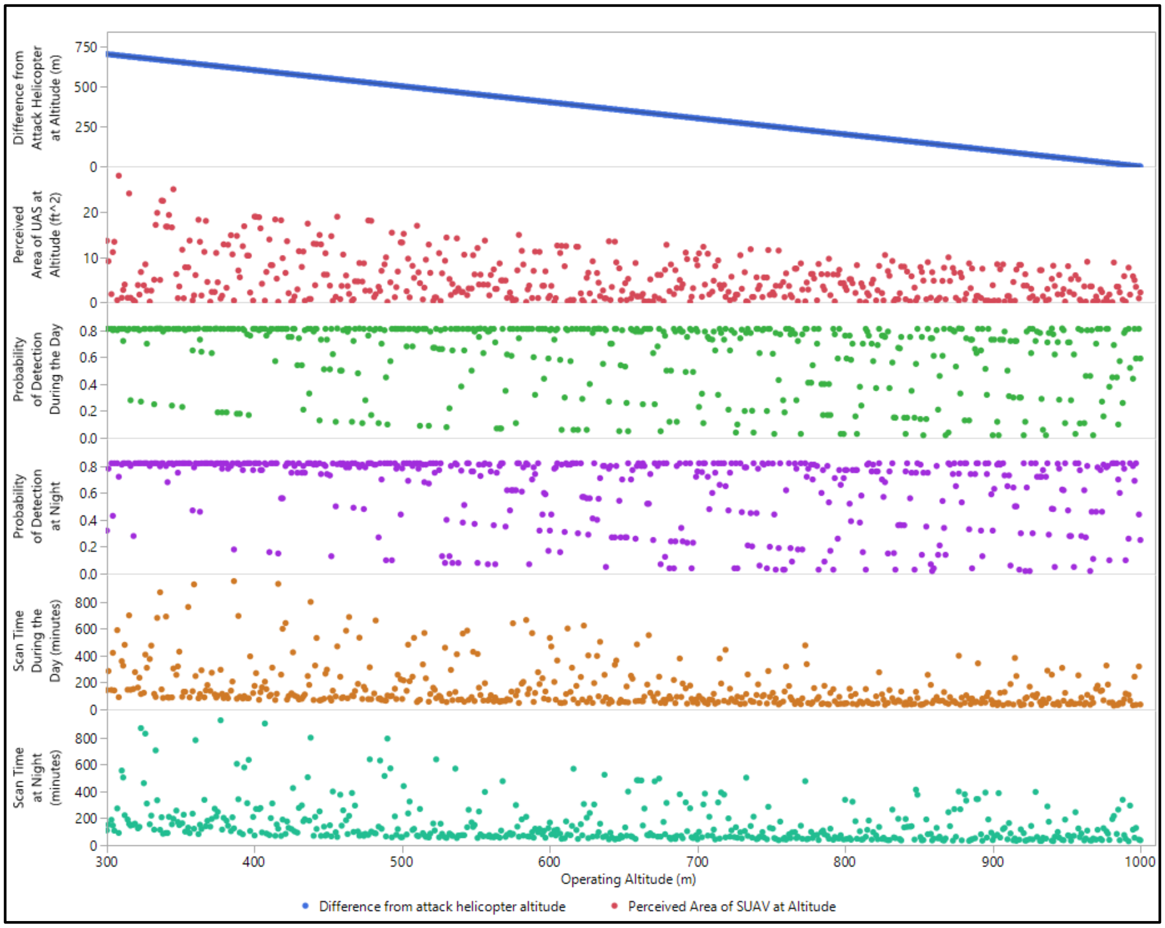


Figure 36. Integrated Set Space 1: Mission Perspective on Operating Altitude (Electric Team)

Mission also considers the ability of the different design variants to meet the stakeholder objectives it is either responsible for or influences through operating altitude. It finds that none of the variants meet all of these particular MOEs simultaneously. Of the feasible solutions in the design space, 13–17% do not meet the day and night scan times, time to fly 10 km, or perceived UAS area MOEs; 45% do not meet the difference from attack helicopter; 64% do not meet the probability of detection at night; and 83% do not meet the probability of detection during the day.

2. Step 9: Communicate the Specialty Set Space Preferences

Having had a chance to consider all of the combined information and explore the integrated set space from its own perspective, each specialty communicates its preferences and important findings to the design integration manager. This information received, implications, resulting actions, justification, cumulative design space description, and knowledge passed on to the specialties following the design integration manager's interpretation of the information is summarized on a knowledge and action record (KAR) in Table 14.

Table 14. Knowledge and Action Record 1 (Electric Team)

Specialty Information Received	Implication
Structures	
Max Payload = 0.1645+0.2339*Wingspan, which is only 0.4 lbs for 1 ft wingspans All wingspans from 1 ft to 16 ft are still accepatble in terms of endurance, airspeed, and air vehicle weight Endurance = 1.311+0.04415*Wingspan Airspeed=24.59+1.417*Wingspan Air Vehicle Weight=0.91+1.3*Wingspan Integrated Set Space 1: Structures Perspective on Wingspan plot provided	Smaller wingspan means less payload capacity means less sensor resolution
Weight and Balance	
Air Vehicle Weight=0.91+1.3*Wingspan Max Payload = 0.1645+0.2339*Wingspan 80% of the design variants exceed max payload	Sensors are too heavy for some wingspans
Sensors	
Total (EO+IR) resolution cannot exceed 1,000 pixels in general, varies by wingspan Maximum (EO or IR) resolution is 800 pixels in general, varies by wingspan Integrated Set Space 1: Sensors Perspective on Resolution plot provided	Max total (EO+IR) resolution for 12-16 ft wingspans is 1,000 pixels Max total (EO+IR) resolution for 6-11 ft wingspans is 800 pixels Max total (EO+IR) resolution for 3-5 ft wingspans is 600 pixels Max total (EO+IR) resolution for 1 ft and 2 ft wingspans is 400 pixels
Mission	
No variants are viable (13-17% because ofday and night scan times, time to fly 10 km, or perceived UAS area; 45% because of difference from attack helicopter; 64% becase of probabaility of detection at night; and 83% because of probability of detection during the day)	Difference from attack helicopter can be addressed by capping the operating altitude May need to consider the design space with some of the operational requirements relaxed to explore more viable options

Action	Justification
Reduce EO resolution range from 200-1,800 pixels to 200-800 pixels	Several variants exceeded max payload due to sensor/comms weight Integrated Set Space 1: Sensor Perspective on Resolution plot
Reduce IR resolution range from 200-1,800 pixels to 200-800 pixels	Several variants exceeded max payload due to sensor/comms weight Integrated Set Space 1: Sensor Perspective on Resolution plot
Reduce operating altitude range from 300-1,000 m to 300-700 m	Difference from attack helicopter operating at 1,000 m altitude must be at least 300 m

Cumulative Design Space Description
Integrated Set Space 1 Wingspan (1-16 ft); EO and IR Resolution (200-1,800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-1,000 m) 512 Design Choices -- None are Viable
Reduction 1 Eliminate 1,000-1,800 pixel EO resolutions; Eliminate 1,000-1,800 pixel IR resolutions; Eliminate Operating Altitudes > 700 m 54 Design Choices -- None are Viable
Integrated Set Space 2 Wingspan (1-16 ft); EO and IR Resolution (200-800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-700 m) 566 Design Choices -- 10 are Viable -- Value (OMOE) Range: 40.87-48.30 -- Cost Range: $137.72K-$139.91K

Information Communicated to the Specialties	
Integrated Set Space 1: Structures Perspective on Wingspan plot Integrated Set Space 1: Sensors Perspective on Resolution plot New Integrated Set Space 2	New EO Resolution Range: 200 pixels to 800 pixels New IR Resolution Range: 200 pixels to 800 pixels New Operating Altitude Range: 300 m to 700 m

3. Step 10a: Reduce the Set Space by Elimination

The design integration manager reduces the current set space (Integrated Set Space 1) by carrying out the actions from Table 14, including: eliminating EO resolutions from 1,000 to 1,800 pixels; eliminating IR resolutions from 1,000 to 1,800 pixels; and eliminating operating altitudes above 700 m. These eliminations are represented by the greyed out values in Table 15.

Table 15. Reduction Table 1 (Electric Team)

Reduction 1 (Electric Team)					
Structures	Sensors				Mission
Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)
1	200	200	15	15	300
2	400	400	30	30	350
3	600	600	45	45	400
4	800	800	60	60	450
5	1000	1000	75	75	500
6	1200	1200	90	90	550
7	1400	1400			600
8	1600	1600			650
9	1800	1800			700
10					750
11					800
12					850
13					900
14					950
15					1000
16					

4. Step 11a: Refine the Reduced Set Space in Greater Detail

The reduced set space is refined by performing NOB LHC sampling with the new MOP ranges and running the UAS simulation tool again with these new LHC samples to generate the new integrated set space (Integrated Set Space 2).

5. Step 12a: Explore the Refined Set Space

With the refined set space created, each specialty now takes the opportunity to explore it from its own perspective by:

- Looking at the input factors (MOPs) it is responsible for and the operational requirements (MOEs) that are a function of these input factors;
- Considering what part of the input factor ranges can be eliminated; and
- Identifying important information and new discoveries.

a. *Structures*

With the payload weights under control, Structures looks at the day and night scan times, time to fly 10 km, perceived UAS area, and dwell time MOEs that are impacted by wingspan. There are design variants that meet the dwell time and scan time requirements for the whole range of wingspans from 1–16 ft, but those with 1–5 ft wingspans do not meet the time to fly 10 km requirement, and those with 16 ft wingspans fail to meet the minimum perceived UAS area requirement (see Figure 37).

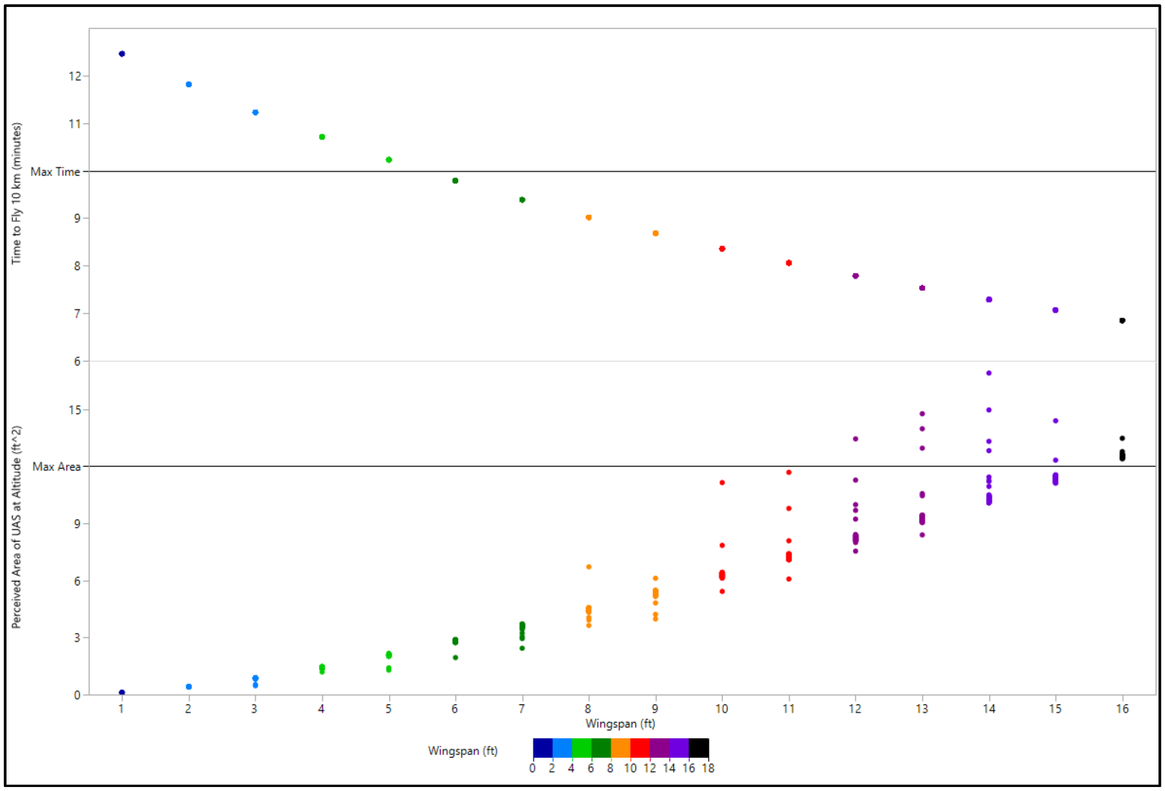


Figure 37. Integrated Set Space 2: Structures Perspective on Wingspan (Electric Team)

b. *Weight and Balance*

Weight and Balance confirms that all design variants have appropriate payloads and do not exceed 50 lbs total.

c. *Sensors*

Sensors explores the relationship between scan time and FOV and finds all variants with a 15 degree FOV exceed 240 minutes (see Figure 38).

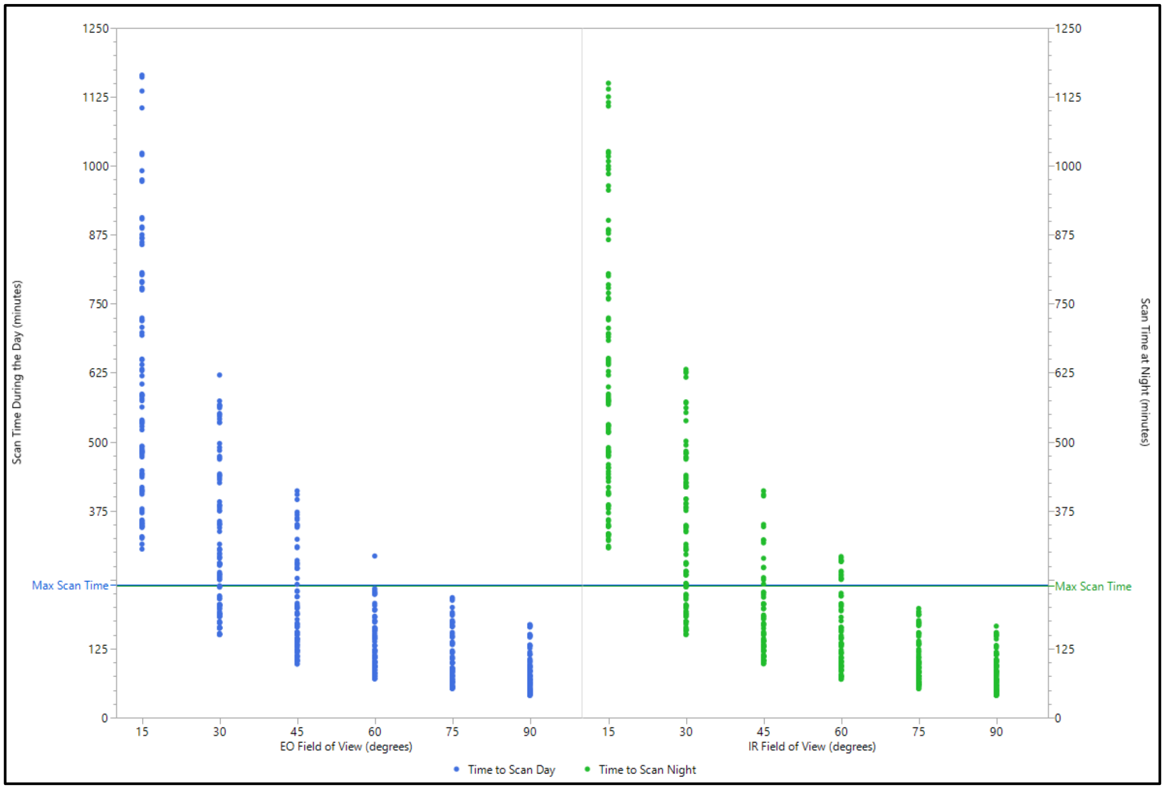


Figure 38. Integrated Set Space 2: Sensors Perspective on FOV (Electric Team)

d. *Mission*

Mission looks at the new dataset in terms of operating altitude, and it reflects all impacted MOEs are achievable over the full range of operating altitudes.

6. Step 13a: Communicate the Specialty Set Space Preferences

Each specialty communicates its preferences and important findings to the design integration manager after it has had the chance to explore the refined set space from its own perspective. Table 16 shows the KAR containing the information received, implications, resulting actions, justification, cumulative design space description, and knowledge passed on to the specialties following the design integration manager's interpretation of the information.

Table 16. Knowledge and Action Record 2 (Electric Team)

Specialty Information Received	Implication
Structures	
1 ft to 5 ft wingspans do not meet the time to fly 10 km requirement 16 ft wingspans do not meet the perceived UAS area requirement Integrated Set Space 2: Structures Perspective on Wingspan plot provided	Airspeed is a function of UAS weight which is a function of wingspan Airspeed is limited by engine performance Perceived area is a function of length, which is a function of wingspan
Weight and Balance	
All design variants have appropriate payloads and do not exceed 50 lbs	No issues or set reductions at this time
Sensors	
All variants with 15 degree EO and IR FOV exceed scan time requirements Integrated Set Space 2: Sensors Perspective on FOV plot provided	Ground swath is a tangential function of FOV Smaller FOV, longer scan time
Mission	
All MOEs that are impacted by operating altitude are achievable across the full range of 300-700 m	No issues or set reductions at this time
Action	**Justification**
Reduce the wingspan range from 1-16 ft to 6-15 ft	Minimum time to fly 10 km unachievable for 1-5 ft wingspans Perceived UAS area too large for 16 ft wingspans Integrated Set Space 2: Structures Perspective on Wingspan plot
Reduce EO FOV range from 15-90 degrees to 30-90 degrees	Minimum scan time during the day unachievable for all 15 degree variants Integrated Set Space 2: Sensors Perspective on FOV plot
Reduce IR FOV range from 15-90 degrees to 30-90 degrees	Minimum scan time at night unachievable for all 15 degree variants Integrated Set Space 2: Sensors Perspective on FOV plot
Cumulative Design Space Description	
Integrated Set Space 1 Wingspan (1-16 ft); EO and IR Resolution (200-1,800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-1,000 m) 512 Design Choices -- None are Viable	
Reduction 1 Eliminate 1,000-1,800 pixel EO resolutions; Eliminate 1,000-1,800 pixel IR resolutions; Eliminate Operating Altitudes > 700 m 54 Design Choices -- None are Viable	
Integrated Set Space 2 Wingspan (1-16 ft); EO and IR Resolution (200-800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-700 m) 566 Design Choices -- 10 are Viable -- Value (OMOE) Range: 40.87-48.30 -- Cost Range: $137.72K-$139.91K	
Reduction 2 Eliminate 1-5 ft wingspans; Eliminate 16 ft wingspans; Eliminate 15 degree EO FOV; Eliminate 15 degree IR FOV 267 Design Choices -- 10 are Viable -- Value (OMOE) Range: 40.87-48.30 -- Cost Range: $137.72K-$139.91K	
Integrated Set Space 3 Wingspan (6-15 ft); EO and IR Resolution (200-800 pixels); EO and IR FOV (30-90 degrees); Operating Altitude (300-700 m) 779 Design Choices -- 36 are Viable -- Value (OMOE) Range: 40.82-48.30 -- Cost Range: $137.23K-$139.91K	
Information Communicated to the Specialties	
New EO Field of View Range: 30 degrees to 90 degrees New IR Field of View Range: 30 degrees to 90 degrees	New Wingspan Range: 6 ft to 15 ft Integrated Set Space 3

7. Step 10b: Reduce the Set Space by Elimination

The design integration manager reduces the current set space (Integrated Set Space 2) by carrying out the actions from Table 16, including: eliminating 1–15 ft wingspans; eliminating 16 ft wingspans; eliminating 15 degree EO FOVs; and eliminating 15 degree IR FOVs (see Table 17).

Table 17. Reduction Table 2 (Electric Team)

Reduction 2 (Electric Team)					
Structures	Sensors				Mission
Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)
1	200	200	15	15	300
2	400	400	30	30	350
3	600	600	45	45	400
4	800	800	60	60	450
5	1000	1000	75	75	500
6	1200	1200	90	90	550
7	1400	1400			600
8	1600	1600			650
9	1800	1800			700
10					750
11					800
12					850
13					900
14					950
15					1000
16					

8. Step 11b: Refine the Reduced Set Space in Greater Detail

The reduced set space is refined by performing NOB LHC sampling with the new MOP ranges and running the UAS simulation tool again with these new LHC samples to generate the new integrated set space (Integrated Set Space 3).

9. Step 12b: Explore the Refined Set Space

With the refined set space created, each specialty now takes the opportunity to explore it from its own perspective by:

- Looking at the input factors (MOPs) it is responsible for and the operational requirements (MOEs) that are a function of these input factors;
- Considering what part of the input factor ranges can be eliminated; and
- Identifying important information and new discoveries.

a. Structures, Weight and Balance, and Mission

The Structures, Weight and Balance, and Mission specialties confirm all of their MOP values can lead to satisfying the MOE requirements and that there are no more justifiable set reductions to make at this time.

b. Sensors

Sensors considers the combined impact of resolution and FOV on the probabilities of detection. It finds almost all design variants with 200 pixel EO resolution are unable to meet the daytime detection requirement across all EO FOVs (see Figure 39). It also finds almost all design variants with 90 degree EO FOV are unable to meet the daytime detection requirement across all EO resolutions (see Figure 39).

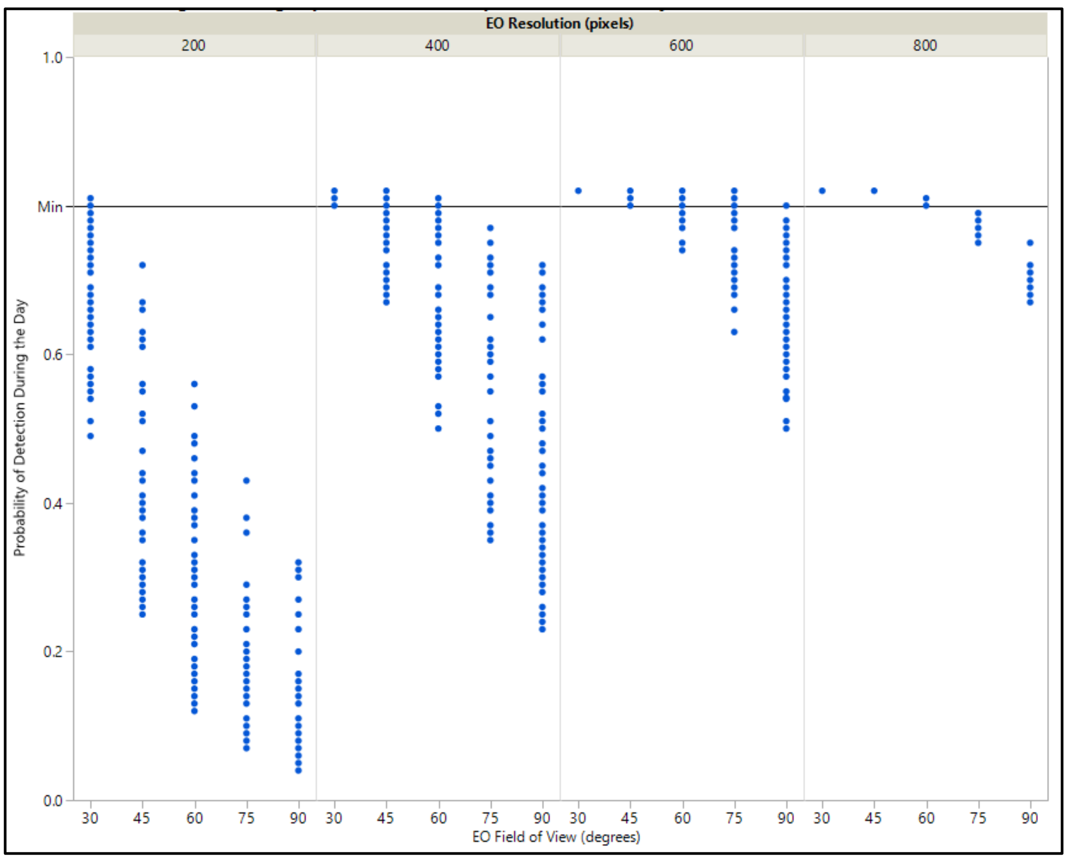


Figure 39. Integrated Set Space 3: Sensors Perspective on Daytime Probability of Detection (Electric Team)

10. Step 13b: Communicate the Specialty Set Space Preferences

Each specialty communicates its preferences and important findings to the design integration manager after it has had the chance to explore the refined set space from its own perspective. Table 18 shows the KAR containing the information received, implications, resulting actions, justification, cumulative design space description, and knowledge passed on to the specialties following the design integration manager's interpretation of the information.

Table 18. Knowledge and Action Record 3 (Electric Team)

Specialty Information Received	Implication
Structures	
All MOEs that are impacted by wingspan are achievable across the full range of 5-16 ft	No issues or set reductions at this time
Weight and Balance	
All design variants have appropriate payloads and do not exceed 50 lbs	No issues or set reductions at this time
Sensors	
Almost all variants with 200 pixel EO resolution fail to achieve the probability of detection required during the day Almost all variants with 90 degree EO FOV fail to achieve the probability of detection required during the day Integrated Set Space 3: Sensors Perspective on Probability of Detection plot provided	Higher resolutions are needed to meet a higher probability of detection Field of view and probability of detection are inversely proportional
Mission	
All MOEs that are impacted by operating altitude are achievable across the full range of 300-700 m	No issues or set reductions at this time
Action	**Justification**
Reduce the EO resolution from 200-800 pixels to 400-800 pixels	Cannot achieve 80% Probability of detection during the day at 200 pixel EO resolutions Integrated Set Space 3: Sensors Perspective on Probability of Detection plot
Reduce the EO FOV from 30-90 degrees to 30-75 degrees	Cannot achieve 80% Probability of detection during the day with 90 degree FOV Integrated Set Space 3: Sensors Perspective on Probability of Detection plot
Cumulative Design Space Description	
Integrated Set Space 1 Wingspan (1-16 ft); EO and IR Resolution (200-1,800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-1,000 m) 512 Design Choices -- None are Viable	
Reduction 1 Eliminate 1,000-1,800 pixel EO resolutions; Eliminate 1,000-1,800 pixel IR resolutions; Eliminate Operating Altitudes > 700 m 54 Design Choices -- None are Viable	
Integrated Set Space 2 Wingspan (1-16 ft); EO and IR Resolution (200-800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-700 m) 566 Design Choices -- 10 are Viable -- Value (OMOE) Range: 40.87-48.30 -- Cost Range: $137.72K-$139.91K	
Reduction 2 Eliminate 1-5 ft wingspans; Eliminate 16 ft wingspans; Eliminate 15 degree EO FOV; Eliminate 15 degree IR FOV 267 Design Choices -- 10 are Viable -- Value (OMOE) Range: 40.87-48.30 -- Cost Range: $137.72K-$139.91K	
Integrated Set Space 3 Wingspan (6-15 ft); EO and IR Resolution (200-800 pixels); EO and IR FOV (30-90 degrees); Operating Altitude (300-700 m) 779 Design Choices -- 36 are Viable -- Value (OMOE) Range: 40.82-48.30 -- Cost Range: $137.23K-$139.91K	
Reduction 3 Eliminate 200 pixel EO resolutions; Eliminate 90 degree EO FOV 410 Design Choices -- 36 are Viable -- Value (OMOE) Range: 40.82-48.30 -- Cost Range: $137.23K-$139.91K	
Integrated Set Space 4 Wingspan (6-15 ft); EO Resolution (400-800 pixels); IR Resolution (200-800 pixels); EO FOV (30-75 degrees); IR FOV (30-90 degrees); Operating Altitude (300-700 m) 922 Design Choices -- 86 are Viable -- Value (OMOE) Range: 40.82-48.45 -- Cost Range: $137.23K-$139.91K	
Information Communicated to the Specialties	
New EO Resolution Range: 200 pixels to 800 pixels New EO FOV Range: 30 degrees to 75 degrees	Integrated Set Space 4

11. Step 10c: Reduce the Set Space by Elimination

The design integration manager reduces the current set space (Integrated Set Space 3) by carrying out the actions from Table 18, including: eliminating 200 pixel EO resolutions; and eliminating 90 degree EO FOVs (see Table 19).

Table 19. Reduction Table 3 (Electric Team)

Reduction 3 (Electric Team)					
Structures	Sensors				Mission
Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)
1	200	200	15	15	300
2	400	400	30	30	350
3	600	600	45	45	400
4	800	800	60	60	450
5	1000	1000	75	75	500
6	1200	1200	90	90	550
7	1400	1400			600
8	1600	1600			650
9	1800	1800			700
10					750
11					800
12					850
13					900
14					950
15					1000
16					

12. Step 11c: Refine the Reduced Set Space in Greater Detail

The reduced set space is refined by performing NOB LHC sampling with the new MOP ranges and running the UAS simulation tool again with these new LHC samples to generate the new integrated set space (Integrated Set Space 4).

13. Step 12c: Explore the Refined Set Space

Each specialty explores the refined set space from its own perspective.

14. Step 13c: Communicate the Specialty Set Space Preferences

Each specialty communicates to the design integration manager that it is satisfied with the current set space and has no further reduction recommendations.

15. Step 14: Create the Viable Set Space

The viable set space is created by:

- Verifying all the set reduction criteria have been applied and all input factors (MOPs) are within the reduced and agreed upon ranges;

  Although design variants can be outside these ranges and still be viable, this step is important for credibility between the specialties because they are following through with the justifiable and agreed upon reductions, and going backwards will break trust and negate a benefit of SBD unless it has been mutually agreed upon to do so.

- Eliminating design variants that do not meet every MOE; and

  The stakeholder MOEs in this example are indicators of achieving the mission needs; design variants must meet all MOEs to be viable.

- Checking that the viable design variants do indeed spread across the whole range of acceptable input factor (MOP) values.

  If there are no design variants reaching out to the extremes of acceptable values, then it may be worth considering another set reduction to tighten the range of those particular variables and further refine the design space.

Using the most current integrated set space (Integrated Set Space 4), the design integration manager ensures all of the MOPs for each variant lie within the agreed upon ranges (see Figure 40). Vertical lines illustrate the bounds of each color-coordinated MOP with corresponding arrows to cover the range of permissible values. Every colored marker must fit within the vertical bands of matching color. For example, the black wingspan markers must lie between the two vertical black lines coinciding with 6 ft and 15 ft wingspans. The ends of the normalized x-axis relate to the initial ranges of preferred MOP values put forth by the specialties (e.g., from 1 ft to 16 ft for wingspan, 200 pixels to 1,800 pixels for resolution, etc.). All of the variants fit within the colored bounds appropriately, and there is also another potential set reduction to consider for the 75 degree EO FOV if desired (because no EO FOVs reach out to 75 degrees).

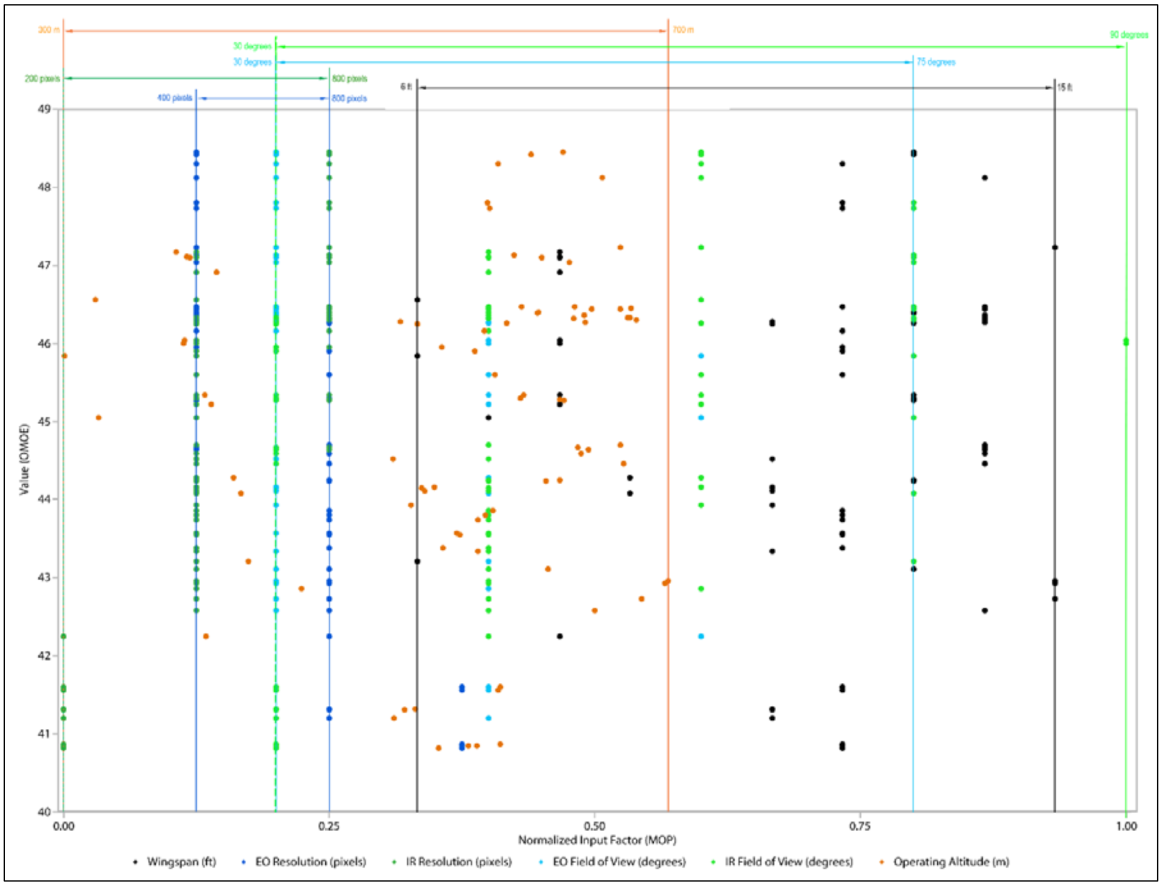


Figure 40. Viable Set Space: Value vs. Input Factor (Electric Team)

Figure 41 demonstrates the concept of viability by ensuring all design variants meet (or exceed) stakeholder MOEs. The ends of the normalized x-axis relate to the range of responses produced for each MOE across all of the simulation runs (e.g., scan times are from 20 minutes to 1,200 minutes, difference from attack helicopter is from 0 m to 700 m, etc.). All of the variants are viable, since none escape their respective colored bounds.

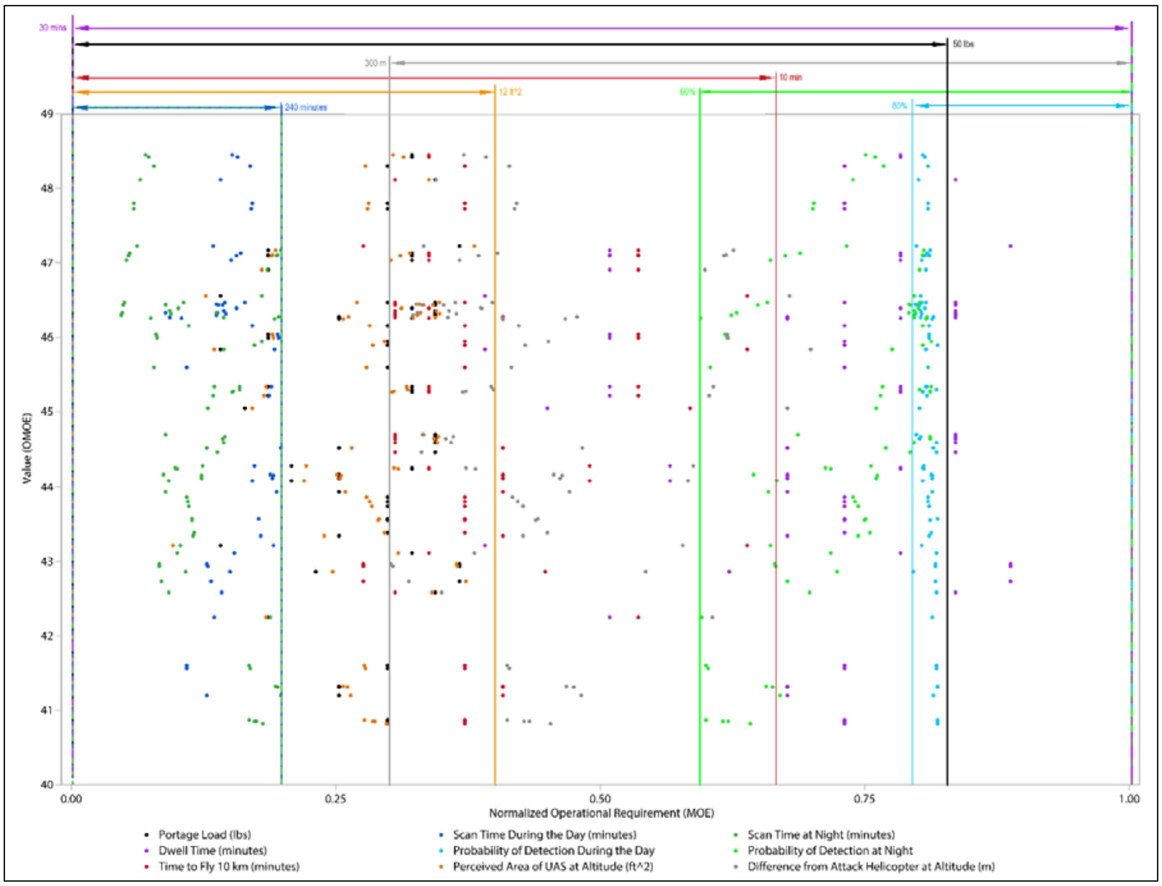


Figure 41. Viable Set Space: Value vs. Operational Requirement (Electric Team)

I. SBD CONTINUED FOR THE PISTON TEAM

1. Step 8: Explore the Integrated Set Space

At this point, since the piston engine UASs are already a well-established product line for the company and the operational requirements are given up-front, traditionally, the company might simply look at the viable solutions and select the best one based on value and cost (see Figure 42, where the point labels show the MOPs for each design choice in the [scenario number, EO FOV, IR FOV, EO resolution, IR resolution, operating altitude] format).

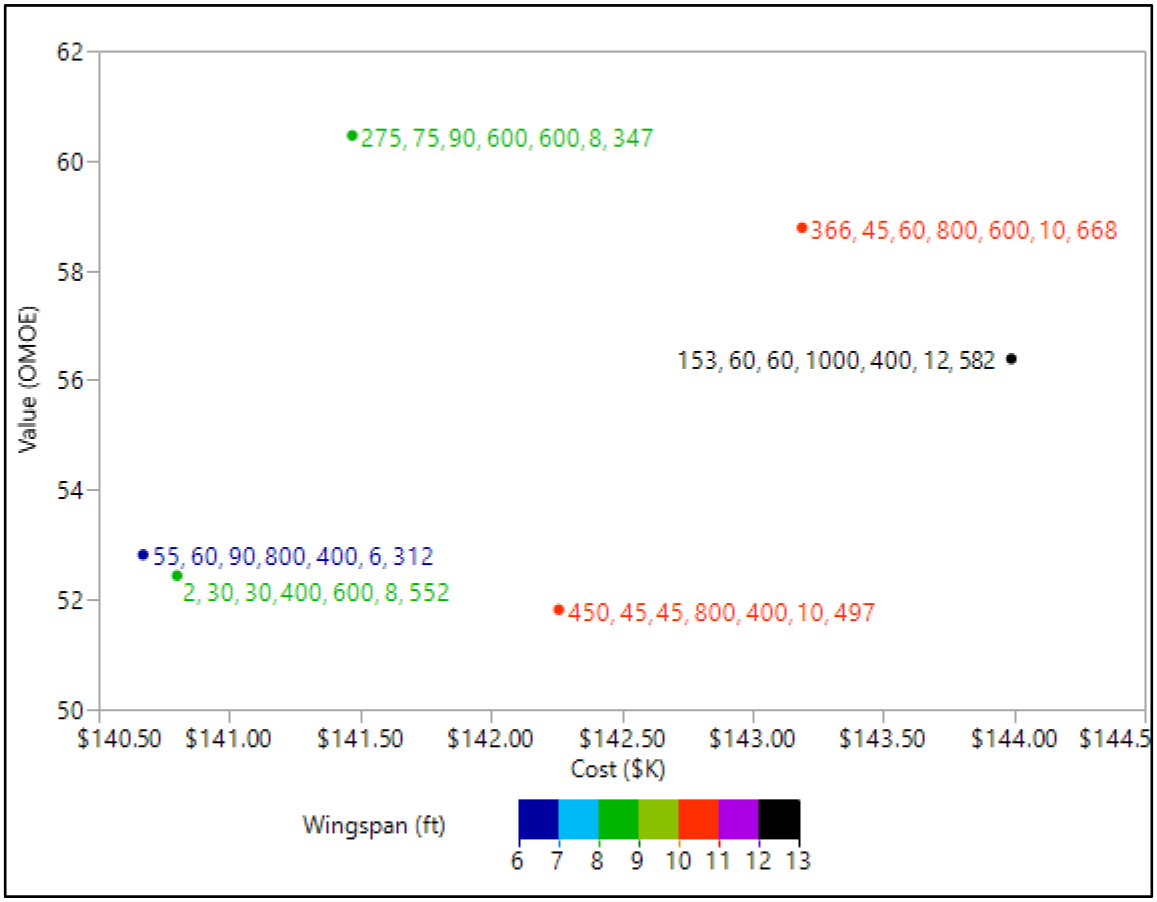


Figure 42. Integrated Set Space 1: Value vs. Cost (Piston Team)

In order for the company to make informed decisions about where to invest for the most successful introduction and growth of a new electric engine product line, it wants to characterize its current piston engine product line and learn more through design.

Using a SBD approach, each individual specialty for the piston team takes the opportunity to explore the integrated design space from its own perspective by:

- Looking at the input factors (MOPs) it is responsible for and the operational requirements (MOEs) that are a function of these input factors;
- Considering what part of the input factor ranges can be eliminated; and
- Identifying important information and new discoveries.

a. *Structures*

Structures is responsible for wingspan and begins exploring the integrated set space by identifying relationships between design and performance characteristics calculated from wingspan (see Figure 43). Variants with 15 ft and 16 ft wingspans sustain an endurance greater than 15 hours, and those with 13–16 ft wingspans achieve airspeeds greater than 60 knots, which means they over-perform unnecessarily for the stakeholder constraints in this example.

In terms of MOEs as a function of wingspan, Structures observes the scan times during the day and night can be achieved by every wingspan, that all wingspans are less than 50 lbs without any payload, and that all dwell times far exceed the 30 minute requirement by about ten-fold. Figure 43 shows 1 ft and 2 ft wingspan variants take too long to fly 10 km, and 16 ft wingspan variants have perceived areas that are too large.

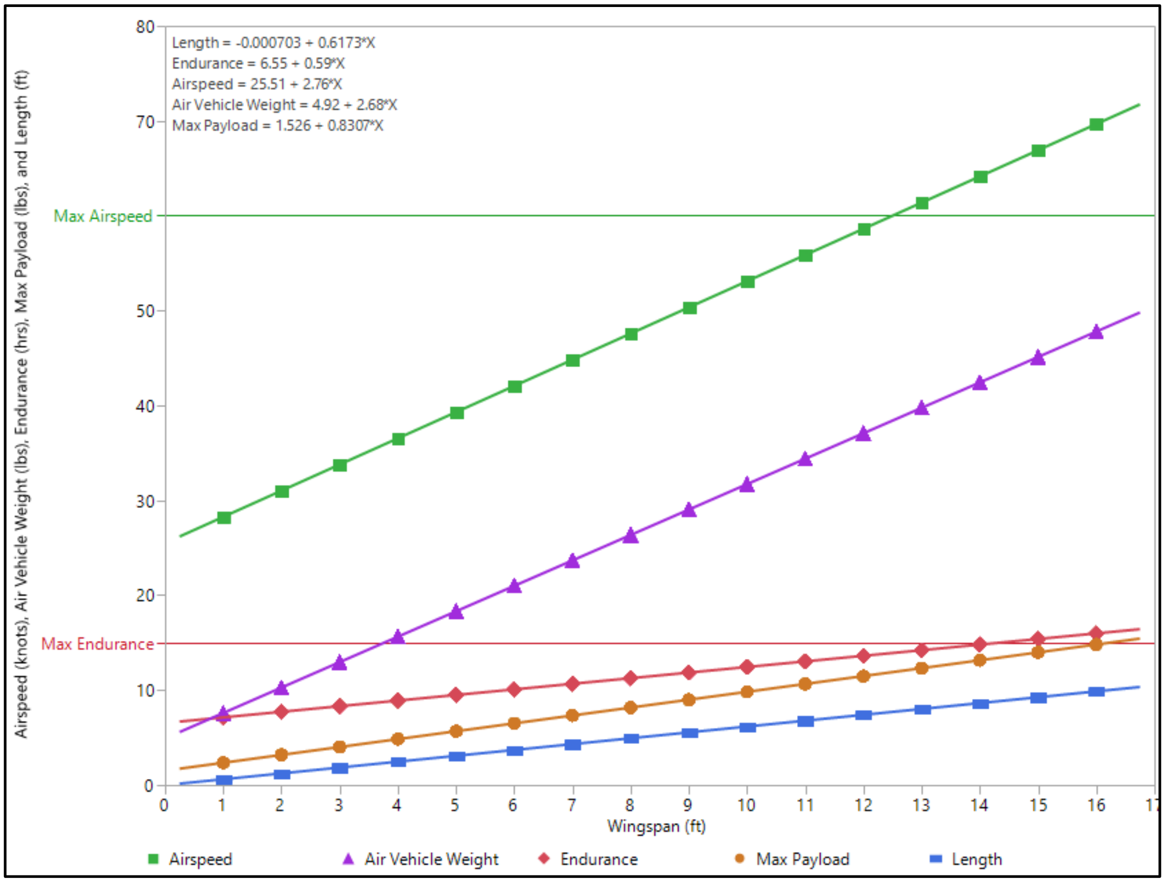


Figure 43. Integrated Set Space 1: Structures Perspective on Wingspan (Piston Team)

b. *Weight and Balance*

Weight and Balance considers the portage load MOE and the individual weights that contribute to it, plus the payload limits (see Figure 44). Several variants, represented by X markers, exceed the max payload, and some with appropriate payloads surpass the 50 lbs max threshold when combined with air vehicle weight. Weight and Balance learns the sensor/comms weight cannot exceed the max payload weight; the sensor/comms weight and air vehicle weight combined cannot exceed 50 lbs; and larger wingspans have higher payload capacity, but also higher air vehicle weight.

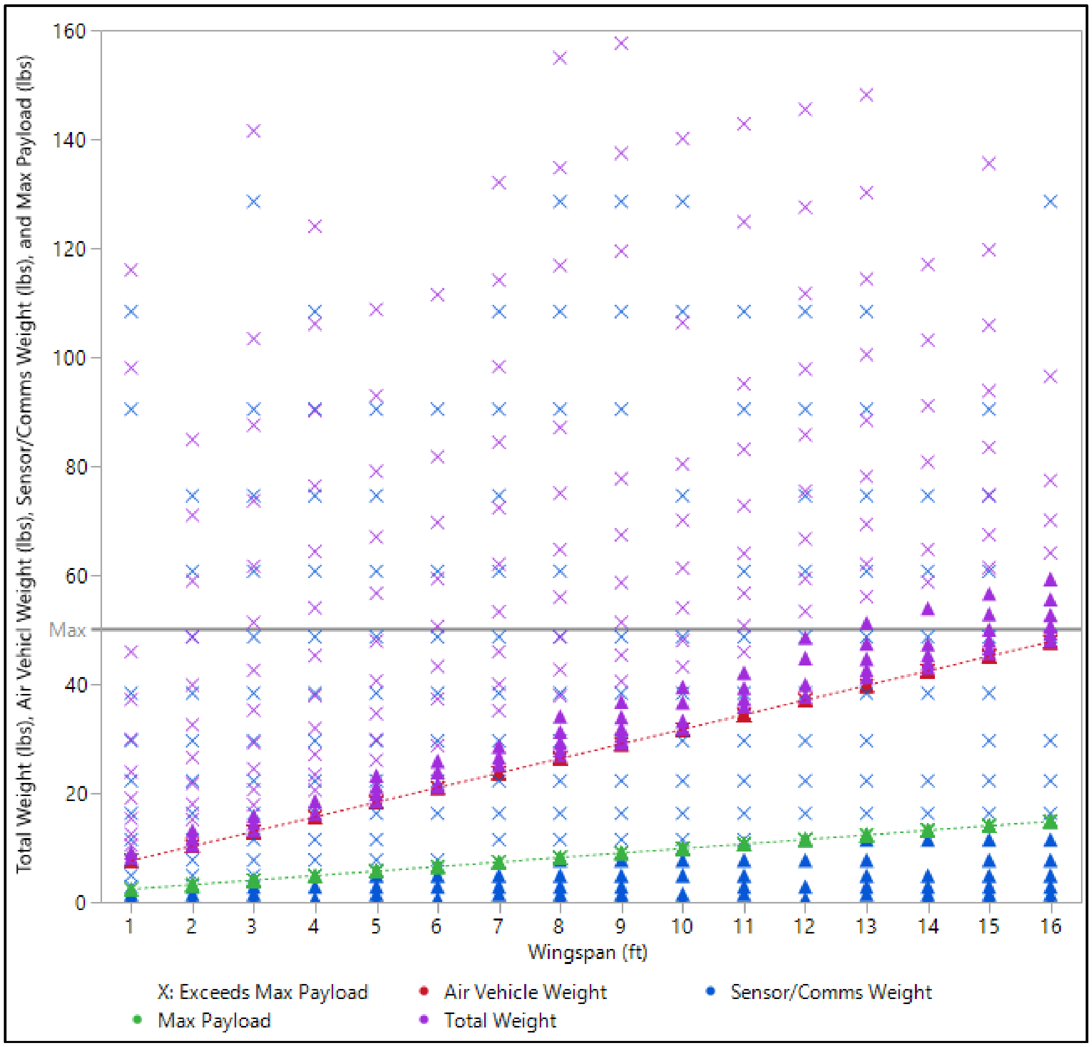


Figure 44. Integrated Set Space 1: Weight and Balance Perspective on Payload (Piston Team)

c. *Sensors*

Sensors starts exploring the integrated set space by learning how much weight it can afford for a maximum portage load of 50 lbs and what this means in terms of sensor ball diameter, and, ultimately, sensor resolution (a function of ball diameter) for each wingspan. The data from the integrated set space offers Sensors the ability to identify the relationships between air vehicle weight, max payload, and wingspan so it can determine the limits on sensor/comms weight (recall payload is solely sensor/comms in this example).

Table 20 shows the max payloads translate to combined EO and IR resolutions no greater than 1,600 pixels, which are achievable for wingspans as low as 12 ft.

Table 20. Max Sensor Resolution for Piston Variants

	Piston Engine				Total
Wingspan (ft)	Air Vehicle Weight (lbs)	Max Payload (lbs)	Sensor Ball Diameter (in)	Resolution (pixels)	Piston Engine Max Resolution, EO+IR (pixels)
16	47.80	14.82	8.45	1744.03	1600
15	45.12	13.99	8.28	1710.52	1600
14	42.44	13.16	8.12	1675.64	1600
13	39.76	12.33	7.94	1639.27	1600
12	37.08	11.49	7.76	1601.23	1600
11	34.40	10.66	7.57	1561.30	1400
10	31.72	9.83	7.37	1519.24	1400
9	29.04	9.00	7.15	1474.74	1400
8	26.36	8.17	6.93	1427.42	1400
7	23.68	7.34	6.68	1376.76	1200
6	21.00	6.51	6.42	1322.13	1200
5	18.32	5.68	6.13	1262.62	1200
4	15.64	4.85	5.82	1196.99	1000
3	12.96	4.02	5.47	1123.35	1000
2	10.28	3.19	5.06	1038.72	1000
1	7.60	2.36	4.58	937.76	800

The MOEs Sensors is concerned with include scan time and probability of detection during the day and night. Sensors ensures the thresholds for these operational requirements can still be met with lower weight options (see Figures 45 and 46).

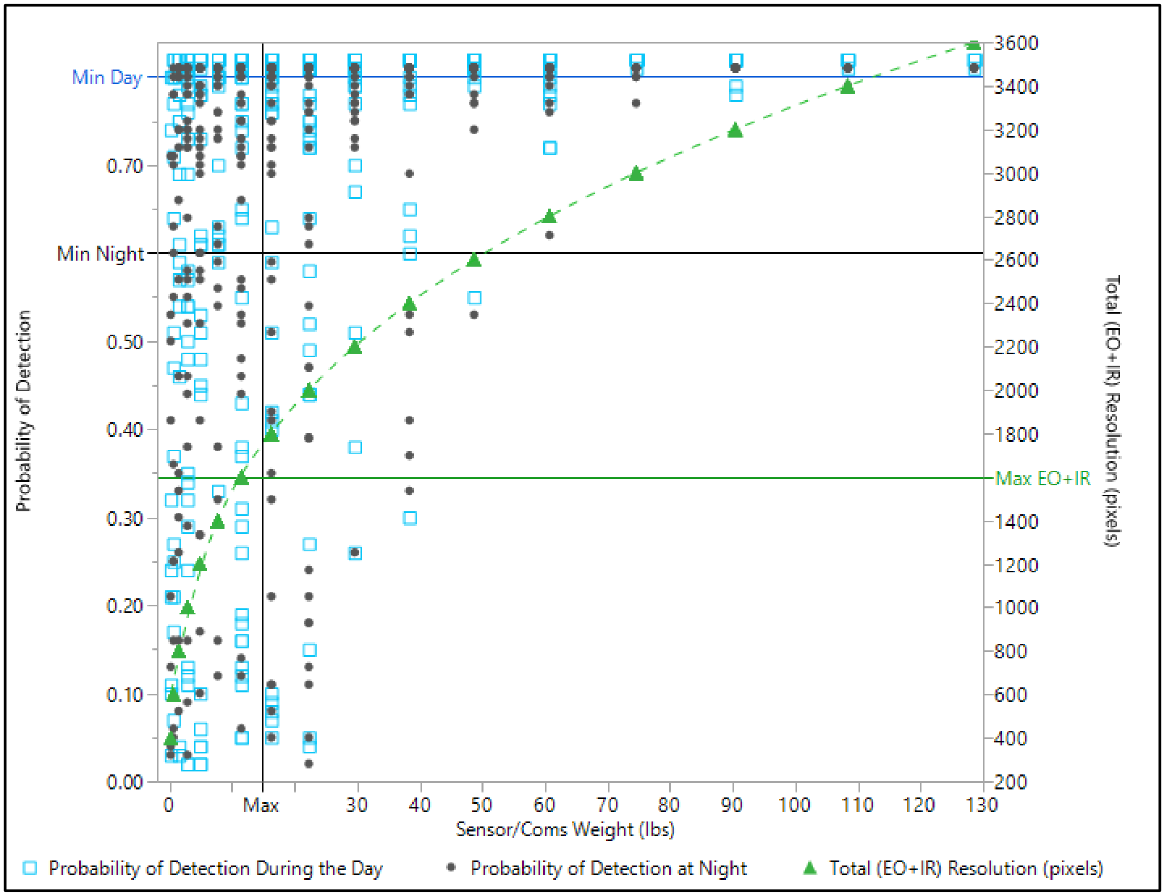


Figure 45. Integrated Set Space 1: Sensors Perspective on Detection and Sensor/Comms Weight (Piston Team)

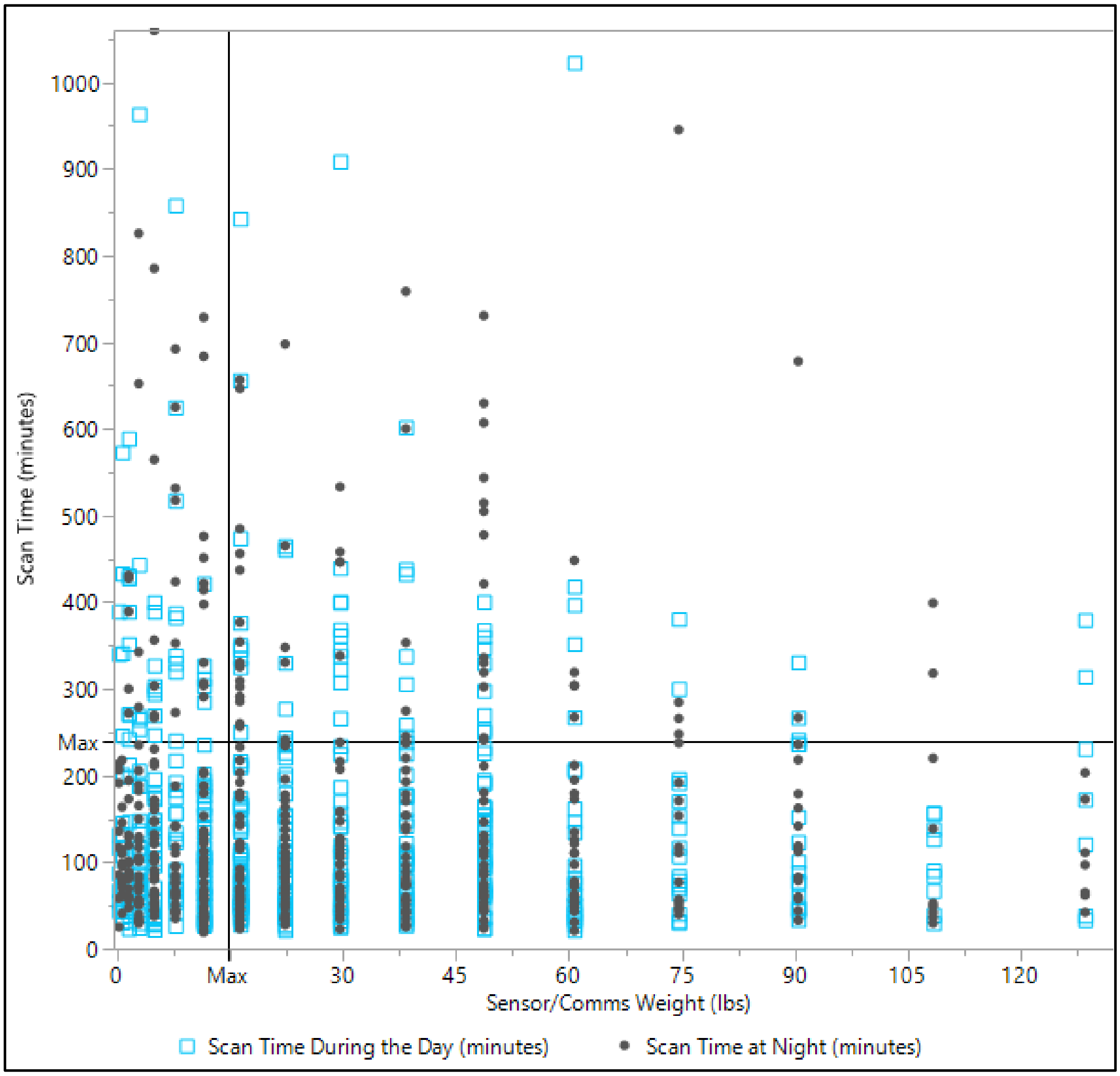


Figure 46. Integrated Set Space 1: Sensors Perspective on Scan Time and Sensor/Comms Weight (Piston Team)

d. Mission

Mission's look at operating altitude shows there are several options across the full range of operating altitudes that meet the day and night scan time and time to fly 10 km requirements (see Figure 47). If the minimum distance required from an attack helicopter operating at 1,000 m is 300 m, then the perceived area of a UAS operating at altitude is exceeded for 14–16 ft wingspan variants. Furthermore, if the minimum distance required from an attack helicopter moves closer to the objective value of 400 m, variants with 13 ft wingspans quickly fall out of consideration due to perceived area; there is very little tradespace between these two compromising requirements for higher wingspans above 12 ft.

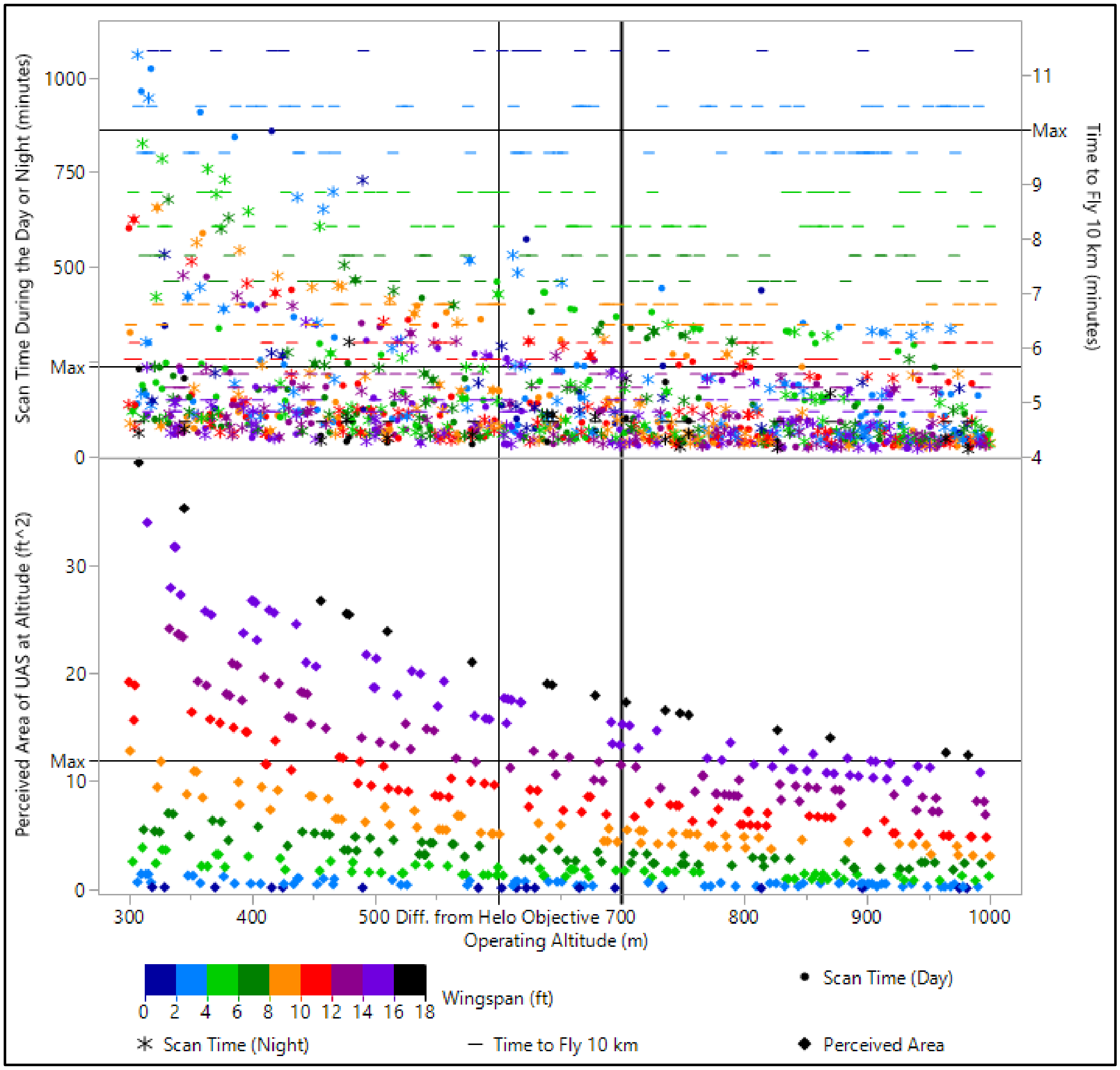


Figure 47. Integrated Set Space 1: Mission Perspective on Operating Altitude (Piston Team)

2. Step 9: Communicate the Specialty Set Space Preferences

Each specialty communicates its preferences and important findings to the design integration manager and a KAR is created (see Table 21).

Table 21. Knowledge and Action Record 1 (Piston Team)

Specialty Information Received	Implication
Structures	
All wingspans from 1 ft to 16 ft are less than 50 lbs All wingspans meet dwell time and day and night scan times Endurance greater than 15 hrs for 15 ft and 16 ft wingspans Airspeed greater than 60 knots for 13-16 ft wingspans Perceived area of UAS at Altitude too large for 16 ft wingspans Time to fly 10 km exceeded for 1 ft and 2 ft wingspans Max Payload = 1.526+0.8307*Wingspan Endurance = 6.55+0.59*Wingspan Airspeed=25.51+2.76*Wingspan Air Vehicle Weight=4.92+2.68*Wingspan Integrated Set Space 1: Structures Perspective on Wingspan plot provided Integrated Set Space 1: Structures Perspective on Perceived Area and Time to Fly plot provided	Over-performance in endurance by 15 ft and 16 ft wingspans Over-performance in airspeed by 13-16 ft wingspans Perceived area proportional to wingspan Time to fly 10 km inversely proportional to wingspan
Weight and Balance	
Several variants exceed the max payload Some variants with appropriate payloads exceed 50 lbs max portage requirement Max Payload = 1.526+0.8307*Wingspan Air Vehicle Weight=4.92+2.68*Wingspan Integrated Set Space 1: Weight and Balance Perspective on Payload plot provided	Sensor/comms weight cannot exceed max payload Sensor/comms weight plus air vehicle weight cannot exceed 50 lbs Max payload proportional to wingspan Air vehicle weight proportional to wingspan
Sensors	
Total (EO+IR) resolution cannot exceed 1,600 pixels in general, varies by wingspan Maximum (EO or IR) resolution is 1,400 pixels in general, varies by wingspan Max Sensor Resolution for Piston Variants table provided	Max total (EO+IR) resolution for 12-16 ft wingspans is 1,600 pixels Max total (EO+IR) resolution for 8-11 ft wingspans is 1,4000 Max total (EO+IR) resolution for 5-7 ft wingspans is 1,200 pixels Max total (EO+IR) resolution for 2-4 ft wingspans is 1,000 pixels Max total (EO+IR) resolution for 1 ft wingspans is 800 pixels
Mission	
Design variants meet day and night scan time and time to fly 10 km across all operating altitudes Perceived area of UAS exceeded for 14-16 ft wingspans Integrated Set Space 1: Mission Perspective on Operating Altitude plot provided	Difference from attack helicopter and perceived area require compromise Operating altitude cannot be above 300 m in order to meet difference from attack helicopter MOE
Action	**Justification**
Reduce the EO resolution range from 200-1,800 pixels to 200-1,400 pixels	Several variants exceeded max payload due to sensor/comms weight Max Sensor Resolution for Piston Variants plot Integrated Set Space 1: Weight and Balance Perspective on Payload plot
Reduce the IR resolution range from 200-1,800 pixels to 200-1,400 pixels	Several variants exceeded max payload due to sensor/comms weight Max Sensor Resolution for Piston Variants plot Integrated Set Space 1: Weight and Balance Perspective on Payload plot
Reduce the wingspan range from 1-16 ft to 3-13 ft	Endurance and airspeed over-performance Perceived area and time to fly 10 km exceeded Integrated Set Space 1: Structures Perspective on Perceived Area plot Time to Fly and Integrated Set Space 1: Mission Perspective on Operating Altitude plot
Reduce operating altitude range from 300-1,000 m to 300-700 m	Difference from attack helicopter operating at 1,000 m altitude must be at least 300 m
Cumulative Design Space Description	
Integrated Set Space 1 Wingspan (1-16 ft); EO and IR Resolution (200-1,800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-1,000 m) 512 Design Choices -- 6 are Viable -- Value (OMOE) Range: 51.82-60.46 -- Cost Range: $140.67K-$143.99K	
Reduction 1 Eliminate 1-2 ft and 14-16 ft wingspans; Eliminate 1,600-1,800 pixel EO resolutions; Eliminate 1,600-1,800 pixel IR resolutions; Eliminate Operating Altitudes > 700 m 53 Design Choices -- 6 are Viable -- Value (OMOE) Range: 51.82-60.46 -- Cost Range: $140.67K-$143.99K	
Integrated Set Space 2 Wingspan (3-13 ft); EO and IR Resolution (200-1,400 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-700 m) 565 Design Choices -- 108 are Viable -- Value (OMOE) Range: 43.95-60.91 -- Cost Range: $138.81K-$145.22K	
Information Communicated to the Specialties	
Integrated Set Space 1: Structures Perspective on Wingspan plot Max Sensor Resolution for Piston Variants table New Integrated Set Space 2 New Wingspan Range: 3 ft to 13 ft	New EO Resolution Range: 200 pixels to 1,400 pixels New IR Resolution Range: 200 pixels to 1,400 pixels New Operating Altitude Range: 300 m to 700 m

3. Step 10a: Reduce the Set Space by Elimination

The design integration manager reduces the current set space (Integrated Set Space 1) by carrying out the actions from Table 21, including: eliminating 1 ft, 2 ft, and 14–16 ft wingspans; eliminating 1,600 pixel and 1,800 pixel EO resolutions; eliminating 1,600 pixel and 1,800 pixel IR resolutions; and eliminating operating altitudes above 700 m (see Table 22).

Table 22. Reduction Table 1 (Piston Team)

Reduction 1 (Piston Team)					
Structures	Sensors				Mission
Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)
1	200	200	15	15	300
2	400	400	30	30	350
3	600	600	45	45	400
4	800	800	60	60	450
5	1000	1000	75	75	500
6	1200	1200	90	90	550
7	1400	1400			600
8	1600	1600			650
9	1800	1800			700
10					750
11					800
12					850
13					900
14					950
15					1000
16					

4. Step 11a: Refine the Reduced Set Space in Greater Detail

The reduced set space is refined by performing NOB LHC sampling with the new MOP ranges and running the UAS simulation tool again with these new LHC samples to generate the new integrated set space (Integrated Set Space 2).

5. Step 12a: Explore the Refined Set Space

With the refined set space created, each specialty now takes the opportunity to explore it from its own perspective by:

- Looking at the input factors (MOPs) it is responsible for and the operational requirements (MOEs) that are a function of these input factors;

- Considering what part of the input factor ranges can be eliminated; and

- Identifying important information and new discoveries.

a. Structures

Structures confirms the new MOP ranges still offer design solutions across all wingspans, and they meet the day and night scan times, time to fly 10 km, and dwell time MOEs it is responsible for. There is also nothing blaringly dominant about any of the wingspans when they are plotted against the MOEs, except that 13 ft wingspans barely meet the perceived area of UAS at altitude MOE (see Figure 48). This was not evident before in the first integrated design space because the knowledge about sensor weight limitations was not applied yet. Structures starts to explore the dwell time more since it exceeds stakeholder requirements by so much. For awareness and knowledge that might be convenient for future characterization, down-selection, or other decision purposes, Structures learns the relationship between dwell time and wingspan is linear and the dwell times are on the order of 335–600 minutes (5.5-10 hours) for the 3–13 ft wingspans in the current design space.

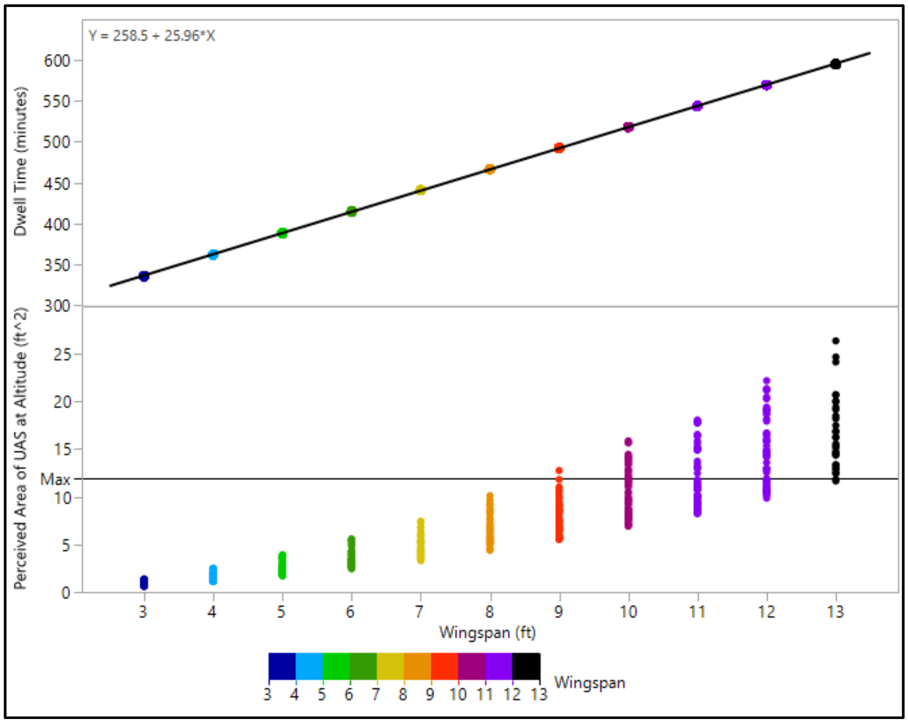


Figure 48. Integrated Set Space 2: Structures Perspective on Wingspan (Piston Team)

b. Weight and Balance

Weight and Balance checks that all design solutions are under the portage load threshold of 50 lbs now that sensor resolutions are linked to wingspan (see Figure 49). Variants with 13 ft wingspans slightly exceed the limit due to comms weight, unless they have less than 1,600 pixels maximum; all other variants and wingspans are acceptable.

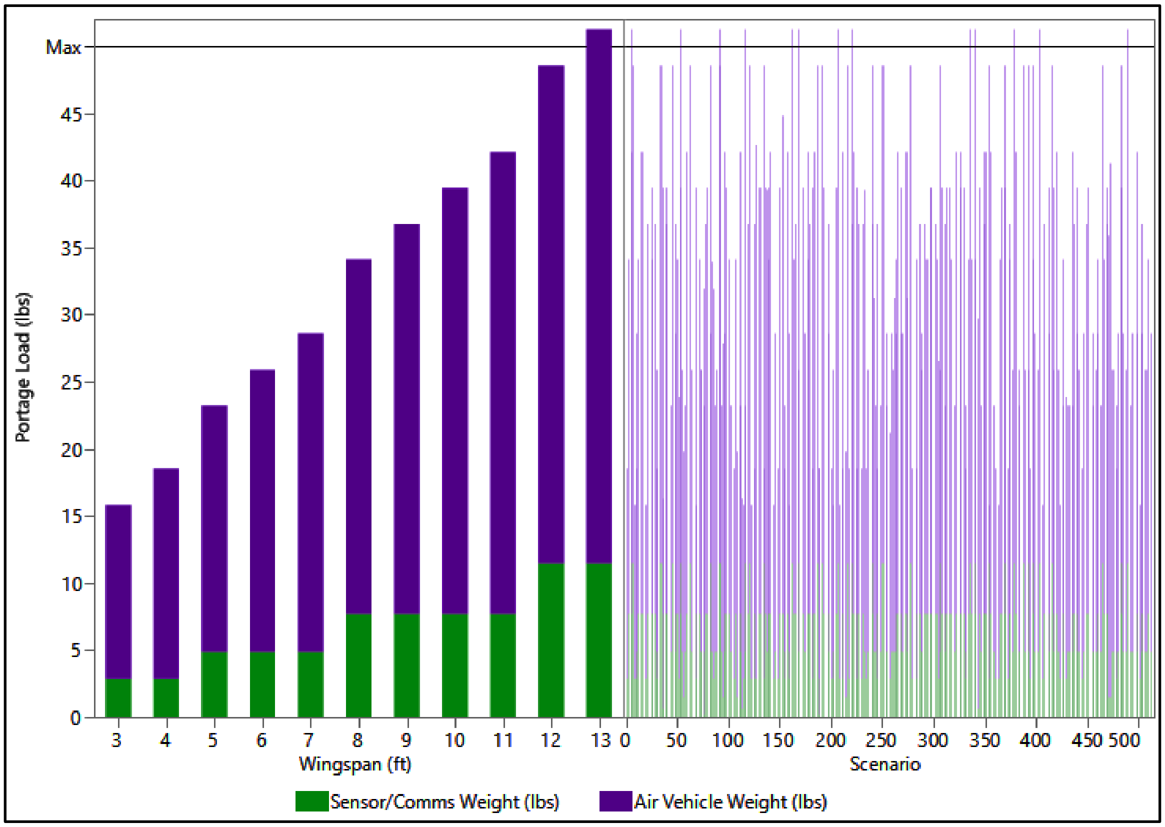


Figure 49. Integrated Set Space 2: Weight and Balance Perspective on Portage Load (Piston Team)

c. Sensors

Sensors explores the resolution and FOV MOPs it is responsible for and the scan time and probability of detection MOEs that are a function of those MOPs. Plotting the day and night scan times and probabilities of detection against the respective EO or IR resolutions and FOVs shows no design variants are capable of meeting the day or night scan time requirement at 15 degree EO or IR FOVs (see Figure 50). Additionally, the bulk of the variants at 200 pixel EO or IR resolutions fail to meet the probability of detection requirements for both day and night and are dominated by all other resolutions.

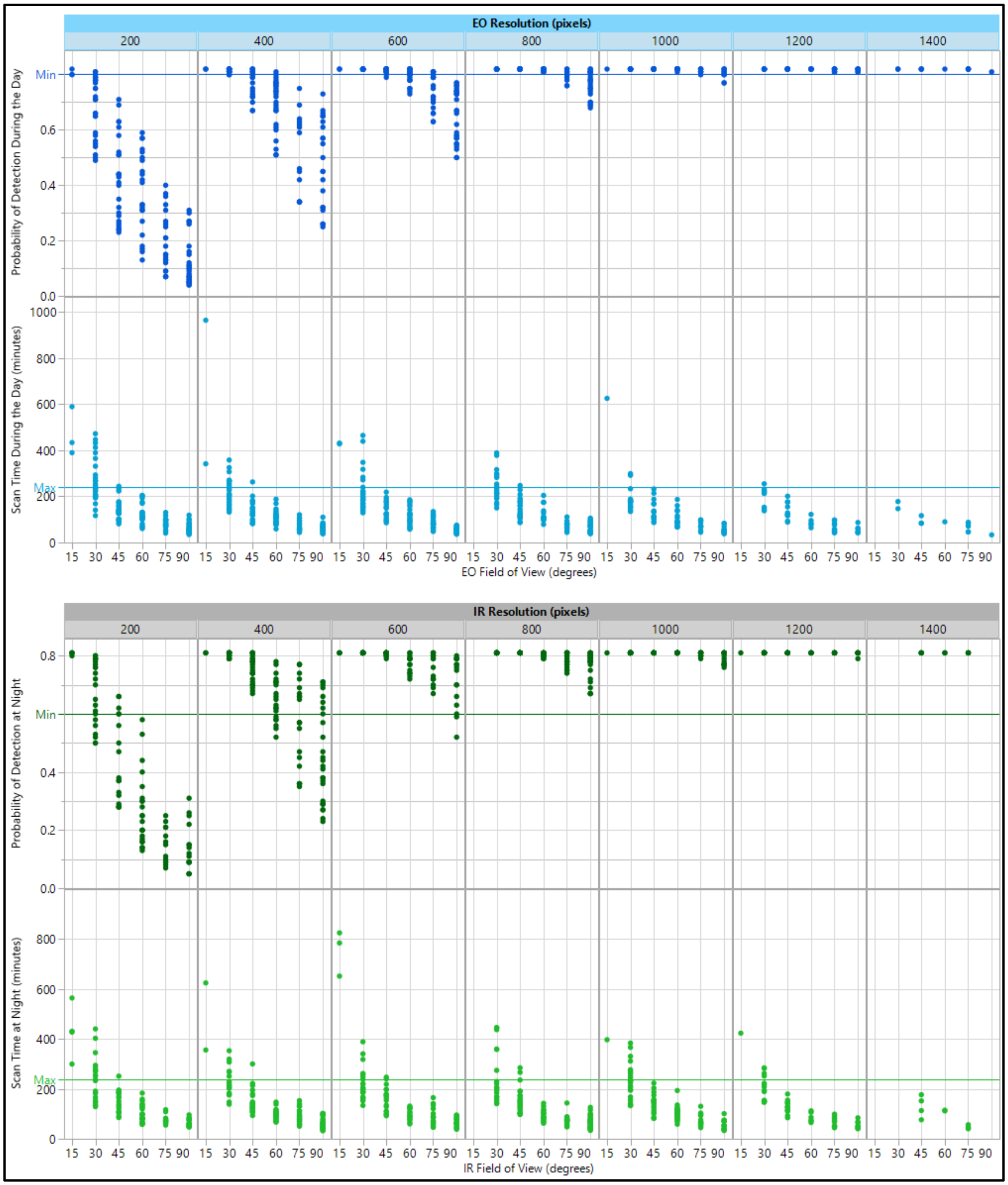


Figure 50. Integrated Design Space 2: Sensors Perspective on Scan Time and Detection (Piston Team)

If 200 pixel EO and IR resolutions are eliminated, the maximum individual EO or IR resolution possible becomes 1,200 pixels (instead of 1,400); the difference between the maximum total (EO+IR) resolution of 1,600 pixels, and the EO or IR resolution must be at least 400 pixels since the 200 pixel option has been eliminated (see Table 23).

Table 23. Valid Sensor Resolution Combinations for Piston Variants

Piston Engine				
Wingspan (ft)	EO Resolution (pixels)	Max Resolution, EO+IR (pixels)	IR Resolution, Max-EO (pixels)	Valid
12	400	1600	1200	✓
	600	1600	1000	✓
	800	1600	800	✓
	1000	1600	600	✓
	1200	1600	400	✓
	1400	1600	200	✕
8 - 11	400	1400	1000	✓
	600	1400	800	✓
	800	1400	600	✓
	1000	1400	400	✓
	1200	1400	200	✕
	1400	1400	0	✕
5 - 7	400	1200	800	✓
	600	1200	600	✓
	800	1200	400	✓
	1000	1200	200	✕
	1200	1200	0	✕
	1400	1200	-200	✕
2 - 4	400	1000	600	✓
	600	1000	400	✓
	800	1000	200	✕
	1000	1000	0	✕
	1200	1000	-200	✕
	1400	1000	-400	✕

d. *Mission*

Mission revisits the perceived UAS area versus operating altitude, while keeping in mind the tradeoff required between it and the difference from an attack helicopter MOE, and also the tidbit of information passed on from Structures that variants with 13 ft wingspans over-perform in airspeed. If there is already a possibility of eliminating 13 ft wingspans, then Mission can offer a further distance from attack helicopter by reducing the upper end of the operating altitude MOP and moving closer to the objective value (see Figure 51).

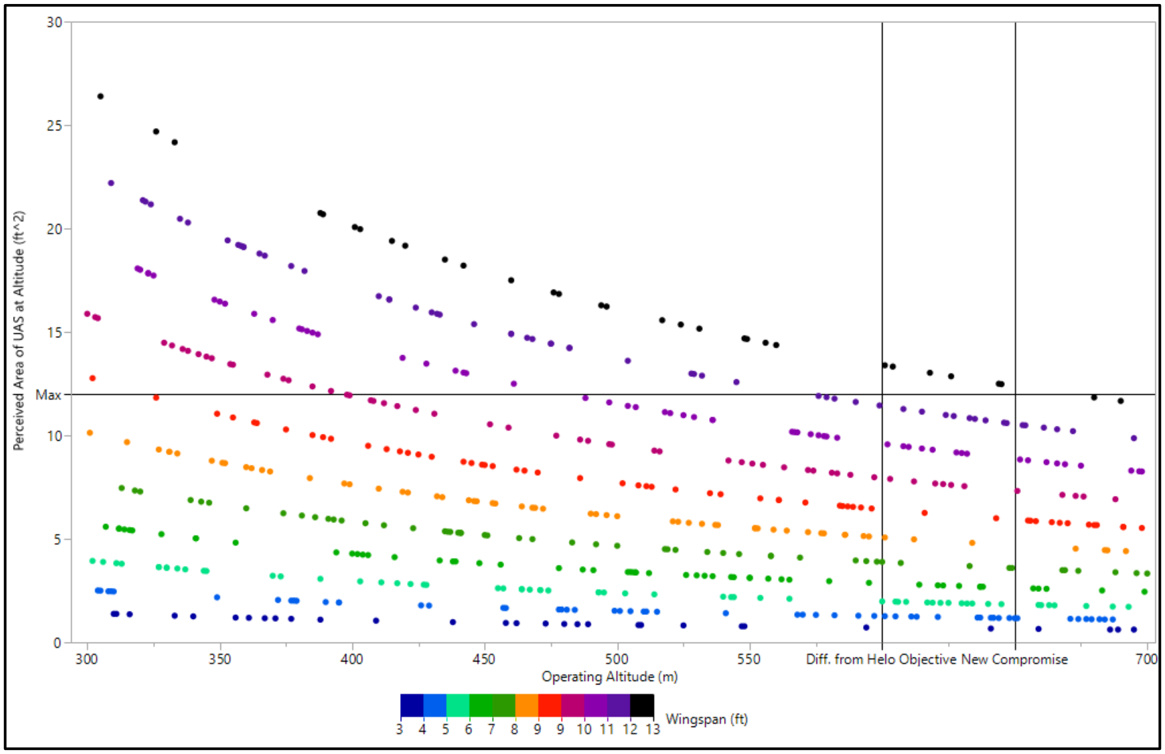


Figure 51. Integrated Set Space 2: Mission Perspective on Operating Altitude (Piston Team)

6. Step 13a: Communicate Specialty Set Space Preferences

Each specialty communicates its preferences and important findings to the design integration manager and a KAR is created (see Table 24).

Table 24. Knowledge and Action Record 2 (Piston Team)

Specialty Information Received	Implication
Structures	
Design variants with 13 ft wingspan barely meet the perceived UAS area MOE Dwell Time=258.50+25.95*Wingspan; range from 335-600 minutes (5.5-10 hrs) Design variants meet day and night scan times, time to fly 10 km, and dwell time MOEs across all wingspans; Airspeed still greater than 60 knots for 13 ft wingspans Integrated Set Space 2: Structures Perspective on Wingspan plot provided	Over-performance in airspeed by 13 ft wingspans Perceived area proportional to wingspan Dwell time could be used to compare design solutions in the future
Weight and Balance	
All new variants adhere to maximum payloads for each wingspan Max portage load requirement exceeded by 13 ft wingspans with more than 1,600 pixels combined EO+IR resolution Integrated Set Space 2: Weight and Balance Perspective on Portage Load plot provided	Comms weight put design solutions with 13 ft wingspans and 1,600 pixels combined EO+IR resolution over the portage load limit Design variants with 13 ft wingspans and less than 1,600 pixels combined EO+IR resolutions are below portage load limit All other design variants across all wingspans have their total (EO+IR) resolutions maximized without exceeding portage load threshold
Sensors	
No design variants meet day or night scan times at 15 degree EO or IR FOVs Design variants with 200 pixel EO or IR resolutions are highly dominated Valid Sensor Resolution Combinations for Piston Variants table provided Integrated Set Space 2: Sensors Perspective on Scan Time and Detection plot provided	Ground swath is a tangential function of FOV Smaller FOV, longer scan time Higher resolutions are needed to meet a higher probability of detection FOV and probability of detection are inversely proportional Max individual EO or IR resolution is 1,200 pixels
Mission	
Design variants meet day and night scan time and time to fly 10 km across all operating altitudes Very limited sliver of the design space where design variants with 13 ft wingspans meet perceived UAS area threshold requirement Integrated Set Space 2: Mission Perspective on Operating Altitude plot provided	Difference from attack helicopter and perceived area require compromise Operating altitude cannot be above 300 m in order to meet difference from attack helicopter MOE; objective value is 400 m
Action	**Justification**
Reduce the EO and IR resolution range from 200-1,400 pixels to 400-1,200 pixels	Design variants with 200 pixel EO or IR solutions are highly dominated when meeting day or night probability of detection Max combined (EO+IR) resolution is 1,600 pixels and no option for 200 pixels Integrated Set Space 2: Sensors Perspective on Scan Times and Detection plot Valid Sensor Resolution Combinations for Piston Variants table
Reduce the EO and IR FOV range from 15-90 degrees to 30-90 degrees	Minimum scan time during the day or night unachievable for all 15 degree FOV variants Integrated Set Space 2: Sensors Perspective on Scan Times and Detection plot
Reduce the wingspan range from 3-13 ft to 3-12 ft	Airspeed over-performance; perceived UAS area exceeded Integrated Set Space 2: Structures Perspective on Wingspan plot
Reduce operating altitude range from 300-700 m to 300-650 m	Difference from attack helicopter operating at 1,000 m altitude objective 400 m

Cumulative Design Space Description
Integrated Set Space 1 Wingspan (1-16 ft); EO and IR Resolution (200-1,800 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-1,000 m) 512 Design Choices -- 6 are Viable -- Value (OMOE) Range: 51.82-60.46 -- Cost Range: $140.67K-$143.99K
Reduction 1 Eliminate 1-2 ft and 14-16 ft wingspans; Eliminate 1,600-1,800 pixel EO resolutions; Eliminate 1,600-1,800 pixel IR resolutions; Eliminate Operating Altitudes > 700 m 53 Design Choices -- 6 are Viable -- Value (OMOE) Range: 51.82-60.46 -- Cost Range: $140.67K-$143.99K
Integrated Set Space 2 Wingspan (3-13 ft); EO and IR Resolution (200-1,400 pixels); EO and IR FOV (15-90 degrees); Operating Altitude (300-700 m) 565 Design Choices -- 108 are Viable -- Value (OMOE) Range: 43.95-60.91 -- Cost Range: $138.81K-$145.22K
Reduction 2 Eliminate 13 ft wingspans; Eliminate 200 pixel and 1,400 pixel EO resolutions; Eliminate 200 pixel and 1,400 pixel IR resolutions; Eliminate 15 degree EO FOV; Eliminate 15 degree IR FOV; Eliminate Operating Altitudes > 650 m 278 Design Choices -- 91 are Viable -- Value (OMOE) Range: 47.78-60.91 -- Cost Range: $138.81K-$145.22K
Integrated Set Space 3 Wingspan (3-12 ft); EO and IR Resolution (400-1,200 pixels); EO and IR FOV (30-90 degrees); Operating Altitude (300-650 m) 790 Design Choices -- 262 are Viable -- Value (OMOE) Range: 47.78-60.91 -- Cost Range: $138.81K-$145.22K

Information Communicated to the Specialties	
Valid Sensor Resolution Combinations for Piston Variants plot New Integrated Set Space 3 New Wingspan Range: 3 ft to 12 ft New Operating Altitude Range: 300 m to 650 m	New EO Resolution Range: 400 pixels to 1,200 pixels New IR Resolution Range: 400 pixels to 1,200 pixels New EO FOV Range: 30 degrees to 90 degrees New IR FOV Range: 30 degrees to 90 degrees

7. Step 10b: Reduce the Set Space by Elimination

The design integration manager reduces the current set space (Integrated Set Space 2) by carrying out the actions from Table 24, including: eliminating 13 ft wingspans; eliminating 200 pixel and 1,400 pixel EO resolutions; eliminating 400 pixel and 1,400 pixel IR resolutions; eliminating 15 degree EO FOVs; eliminating 15 degree IR FOVs; and eliminating operating altitudes above 650 m (see Table 25).

Table 25. Reduction Table 2 (Piston Team)

Reduction 2 (Piston Team)					
Structures	Sensors				Mission
Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)
1	200	200	15	15	300
2	400	400	30	30	350
3	600	600	45	45	400
4	800	800	60	60	450
5	1000	1000	75	75	500
6	1200	1200	90	90	550
7	1400	1400			600
8	1600	1600			650
9	1800	1800			700
10					750
11					800
12					850
13					900
14					950
15					1000
16					

8. Step 11b: Refine the Reduced Set Space in Greater Detail

The reduced set space is refined by performing NOB LHC sampling with the new MOP ranges and running the UAS simulation tool again with these new LHC samples to generate the new integrated set space (Integrated Set Space 3).

9. Step 12b: Explore the Refined Set Space

Each specialty explores the refined set space from its own perspective.

10. Step 13b: Communicate the Specialty Set Space Preferences

Each specialty communicates to the design integration manager that it is satisfied with the current set space and has no further reduction recommendations.

11. Step 14: Create the Viable Set Space

The viable set space is created by:

- Verifying all the set reduction criteria have been applied and all input factors (MOPs) are within the reduced and agreed upon ranges;
- Eliminating design variants that do not meet every MOE; and
- Checking that the viable design variants do indeed spread across the whole

Using the most current integrated set space (Integrated Set Space 3), the design integration manager ensures all of the MOPs for each variant lie within the agreed upon ranges (see Figure 52). Vertical lines illustrate the bounds of each color-coordinated MOP with corresponding arrows to cover the range of permissible values. Every colored marker must fit within the vertical bands of matching color. For example, the black wingspan markers must lie between the two vertical black lines coinciding with 3 ft and 12 ft wingspans. The ends of the normalized x-axis relate to the initial ranges of preferred MOP values put forth by the specialties (e.g., from 1 ft to 16 ft for wingspan, 200 pixels to 1,800 pixels for resolution, etc.). All of the variants fit within the colored bounds appropriately.

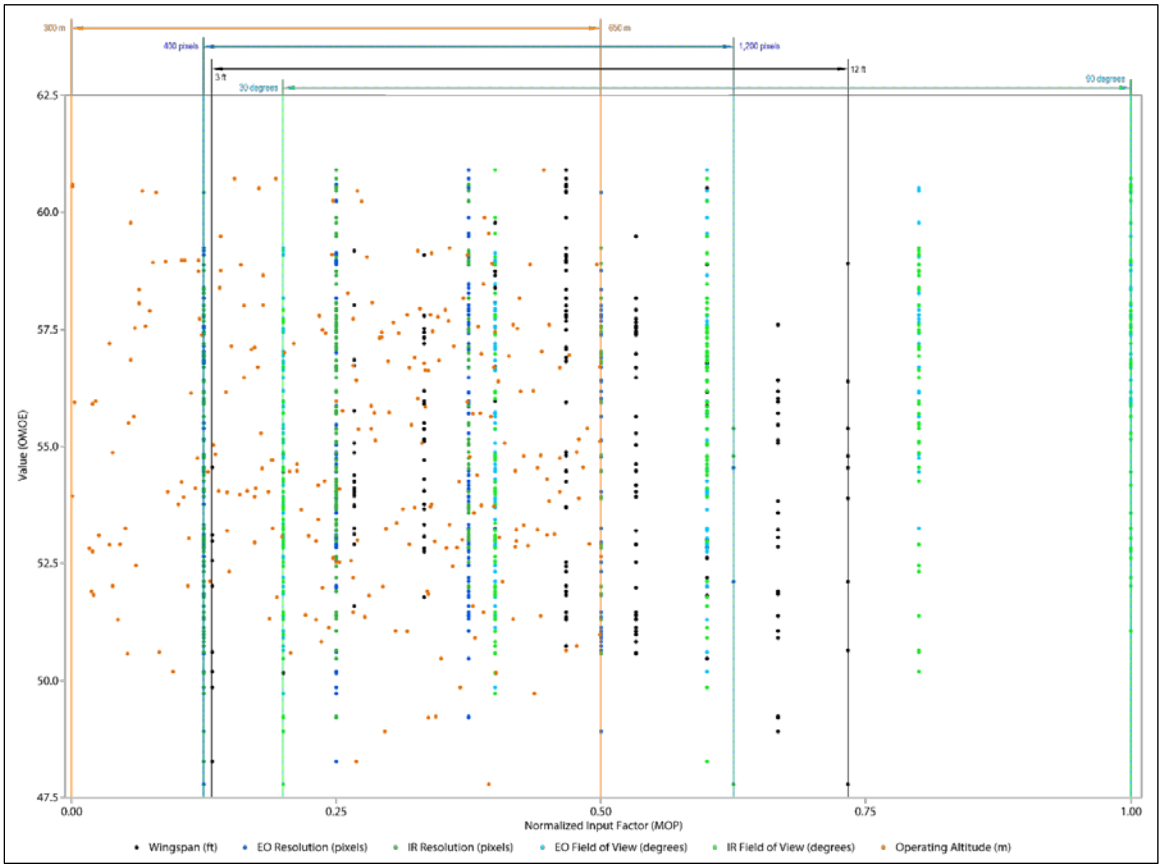


Figure 52. Viable Set Space: Value vs. Input Factor (Piston Team)

Figure 53 demonstrates the concept of viability by ensuring all design variants meet (or exceed) stakeholder MOEs. The ends of the normalized x-axis relate to the range of responses produced for each MOE across all of the simulation runs (e.g., scan times are from 20 minutes to 1,050 minutes, difference from attack helicopter is from 0 m to 700 m, etc.). All of the variants are viable, since none escape their respective colored bounds.

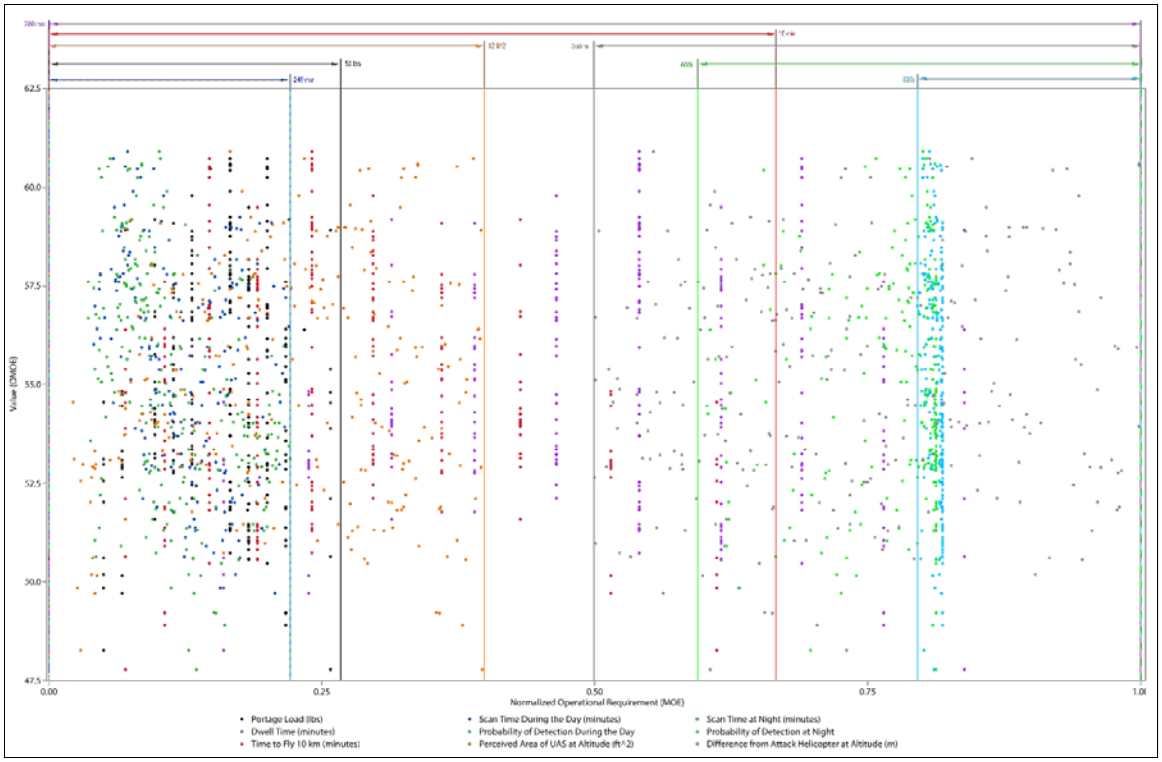


Figure 53. Viable Set Space: Value vs. Operational Requirement (Piston Team)

J. SUMMARIZED FINDINGS FOR BOTH TEAMS

The compiled information for both teams helps illuminate the differences between the two engine types. Some of the important findings are described in terms of the MOPs and MOEs for both the electric team and the piston team.

1. Findings Related to MOPs

Table 26 summarizes some of the design and performance characteristics as a function of wingspan. It is evident electric engine variants can have a larger wingspan and longer length for less weight (nearly half to one-third the weight) compared to piston engine variants. Piston engine variants are about twice as fast and can achieve higher overall airspeeds, in addition to having nearly fifteen times the endurance as electric engine variants.

Table 26. Wingspan Relationships (Combined Teams)

	Electric Engine	Piston Engine
Characteristic	Equation	Equation
Length	0.52*Wingspan	0.62*Wingspan
Endurance	0.04*Wingspan + 1.31	0.59*Wingspan + 6.55
Airspeed	1.42*Wingspan + 24.59	2.76*Wingspan + 25.51
Air Vehicle Weight	1.30*Wingspan + 0.91	2.68*Wingspan + 4.92
Max Payload	0.23*Wingspan + 0.16	0.83*Wingspan + 1.53

The air vehicle weight and max payload equations are used to calculate the maximum total (EO+IR) sensor resolution for each wingspan to ensure the sensor weight does not exceed the max payload allowance (without accounting for the max portage load requirement) (see Table 27). Piston variants can accommodate higher total (EO+IR) resolutions simply because they can carry more payload.

Table 27. Max Sensor Resolutions (Combined Teams)

	Electric Engine				Piston Engine				Totals	
Wing-span (ft)	Air Vehicle Weight (lbs)	Max Payload (lbs)	Sensor Ball Diameter (in)	Res-olution (pixels)	Air Vehicle Weight (lbs)	Max Payload (lbs)	Sensor Ball Diameter (in)	Res-olution (pixels)	Electric Engine Max Resolution, EO+IR (pixels)	Piston Engine Max Resolution, EO+IR (pixels)
16	21.71	3.91	5.42	1112.88	47.80	14.82	8.45	1744.03	1000	1600
15	20.41	3.67	5.31	1089.90	45.12	13.99	8.28	1710.52	1000	1600
14	19.11	3.44	5.19	1065.91	42.44	13.16	8.12	1675.64	1000	1600
13	17.81	3.21	5.07	1040.82	39.76	12.33	7.94	1639.27	1000	1600
12	16.51	2.97	4.94	1014.46	37.08	11.49	7.76	1601.23	1000	1600
11	15.21	2.74	4.81	986.69	34.40	10.66	7.57	1561.30	800	1400
10	13.91	2.50	4.67	957.29	31.72	9.83	7.37	1519.24	800	1400
9	12.61	2.27	4.52	925.99	29.04	9.00	7.15	1474.74	800	1400
8	11.31	2.04	4.36	892.46	26.36	8.17	6.93	1427.42	800	1400
7	10.01	1.80	4.18	856.25	23.68	7.34	6.68	1376.76	800	1200
6	8.71	1.57	3.99	816.75	21.00	6.51	6.42	1322.13	800	1200
5	7.41	1.33	3.78	773.10	18.32	5.68	6.13	1262.62	600	1200
4	6.11	1.10	3.55	723.99	15.64	4.85	5.82	1196.99	600	1000
3	4.81	0.87	3.28	667.32	12.96	4.02	5.47	1123.35	600	1000
2	3.51	0.63	2.95	599.25	10.28	3.19	5.06	1038.72	400	1000
1	2.21	0.40	2.53	511.41	7.60	2.36	4.58	937.76	400	800

Both teams discover 15 degree FOVs are not viable, and the piston team justifies operating altitudes as high as 650 m, while the electric team prefers 700 m.

2. Findings Related to MOEs

The 50 lb max portage load requirement allows the electric variants to carry their full payloads, while the piston variants have to sacrifice some payload capacity due to higher air vehicle weights.

The day and night scan times are faster in general for piston variants because they have faster airspeeds, and ground coverage rate is a function of airspeed. The probabilities of detection during the day and night are the same for electric and piston variants, except there are more FOV and resolution combinations available for piston variants, since they have additional sensor resolutions to choose from.

Piston variants require less time to fly 10 km due to faster airspeeds. Dwell times are almost ten times longer for piston variants because they fly to the dwell location faster and have much higher endurance. Perceived UAS areas tend to be smaller for electric variants because they are shorter in length. The difference from an attack helicopter operating at 1,000 m altitude is the same for both engine types, since it is directly based on operating altitude.

K. SBD CONTINUED FOR BOTH TEAMS COMBINED

1. Step 7: Create the Integrated Set Space by Intersection

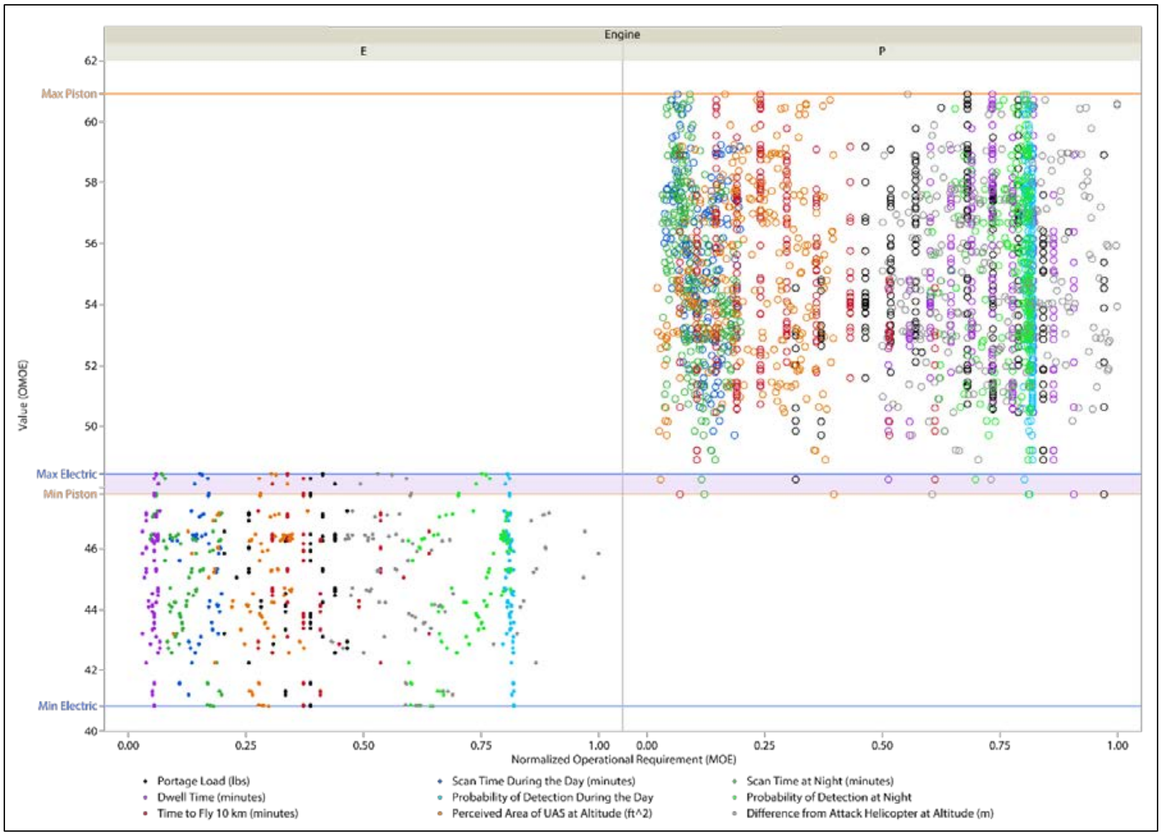
After reviewing the summarized information for each of the two teams, the company wants to explore the consolidated information to graphically see how they compare against each other. The integrated set space for the electric and piston teams combined is created by joining the 86 viable points from the electric team with the 262 viable points from the piston team. Further investigation into the overlying regions will conclude whether electric variants are currently capable of competing in the same value space as piston variants, might insinuate the possibility of offering both product lines to certain stakeholders with requirements identified in the overlap, and will still highlight future investment areas in the regions that do not overlap.

2. Step 8: Explore the Integrated Set Space

Figures 54 and 55 depict system-level views of the integrated set space for the combined teams in terms of value versus input factor and MOE. The primary observation is electric UAS designs inherently exhibit less value than piston UAS designs for the stakeholder requirements in this example and may be ill-suited for the mission at their current level of technology. The horizontal purple band of overlap shows some electric designs are capable of meeting the requirements, but are still at the lower end of the value offered by piston designs.

Note: 68.4 is the maximum value obtainable with every MOE at full performance.

Figure 54. Integrated Set Space 1: Value vs. Input Factor (Combined Teams)



Note: 68.4 is the maximum value obtainable with every MOE at full performance.

Figure 55. Integrated Set Space 1: Value vs. Operational Requirement (Combined Teams)

Figure 56 introduces an example of a new sun plot representing a specialty-level view of the integrated set space for the combined teams in terms of wingspan; similar plots can be generated appropriately for each of the specialties.

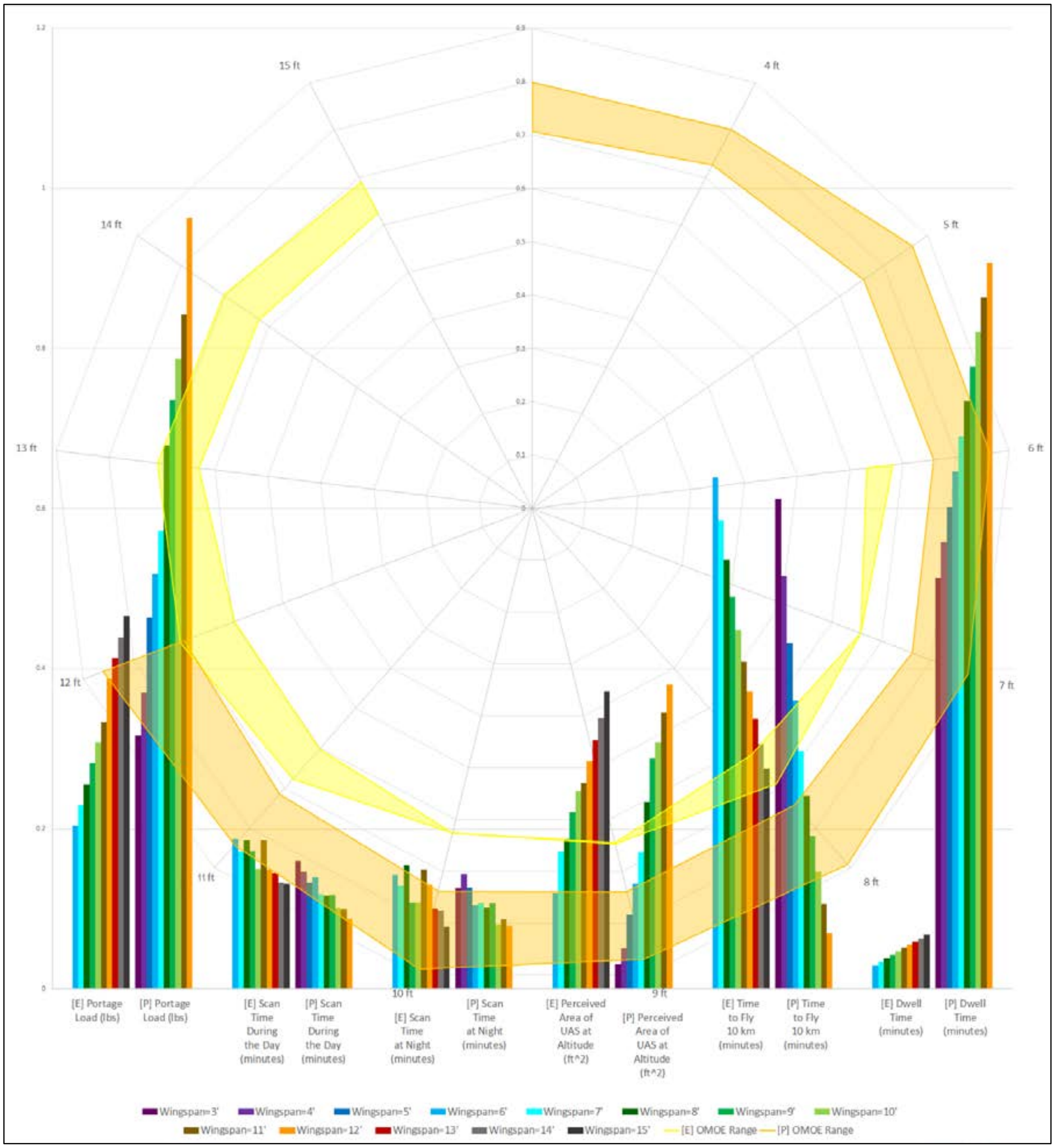


Figure 56. Sun Plot for Wingspan (Combined Teams)

Each MOE that is a function of wingspan is shown in bar chart form for both engine types juxtaposed. Plotting the MOEs in this way emphasizes the performance differences more definitively than in the system-level plots where they are all normalized and plotted together.

The two circular suns drawn as radar plots show the value range as a function of wingspan for both engine types. This plot is consistent with the system-level plot in that the electric variants inherently exhibit less value, and there is very little overlap in value between the two engine types. The electric variants make up the inner circle, which rarely expands to the higher value ring for piston variants. Overlap in value is shown at 12 ft wingspans by both engine types, but it is also more obscurely detected for electric variants at 13 ft and 14 ft wingspans in the fact the values there exceed the minimum value shown by the piston variants, which is around 0.7 at a 3 ft wingspan.

The incomplete circles give important information as well in that they show what wingspans remain in the viable design space for each engine type. One note to be aware of is nothing is being held constant in these plots, so the trend within an MOE does not always shift the value as expected. For example, the time to fly 10 km (a LIB MOE) for the electric variants starting at 6 ft wingspans decreases as wingspan increases; although an increase in value is expected with increasing wingspan, it does not necessarily do so because of the influence of the opposing MOEs (such as dwell time, a MIB MOE). The general proportionality of the MOE behavior to wingspan is discernible through the ascending or descending direction of the bars, however.

Incorporating value into its considerations, each specialty looks at the MOPs and MOEs within its cog and explores the integrated set space in terms of a maximum L1-norm distance to ideal equal to 0.4 (note that this is subjective, but the general intent is to cut the design space in half).

a. Structures

Looking at the integrated set space in terms of L1-norm distance to ideal, the electric Structures team discovers the larger variants with 14 ft and 15 ft wingspans exceed 0.4, while the smaller variants with 3 ft and 4 ft wingspans do for the piston team.

b. Weight and Balance

The Weight and Balance teams observe design variants tend to be the heaviest for those that exceed a 0.4 L1-norm distance to ideal.

c. Sensors

The electric Sensors team sees that every design solution with 200 pixel IR resolution exceeds a 0.4 L1-norm distance to ideal. The piston Sensors team does not observe any real consistency between the sensor FOVs and resolutions and determines they are diversified across all variants.

d. Mission

The Mission specialty for the electric team considers dropping the operating altitude to 650 m as the piston team did, and does discover that higher altitudes coincide with L1-norm distances to ideal greater than 0.4. The piston Mission team observes the operating altitudes of piston design solutions are spread across the entire range, even at maximum L1-norm distances to ideal.

3. Step 9: Communicate the Specialty Set Space Preferences

Each specialty communicates its preferences and important findings to the design integration manager and a KAR is created. Note, the KAR is combined for simplicity of example, but, more realistically, each team would communicate its KAR through its own design integration manager, and then either the two design integration managers would communicate, or they would go through headquarters (see Table 28).

Table 28. Knowledge and Action Record 1 (Combined Teams)

Specialty Information Received	Implication
Structures	
Electric variants with 14 ft and 15 ft wingspans exceed 0.4 L1-norm Piston variants with 3 ft and 4 ft wingspans exceed 0.4 L1-norm	L1-norm distance to ideal based on value (OMOE) and cost at an ideal point of coordinates (1,1); these wingspans offer too little value or have too high cost
Weight and Balance	
The heaviest variants tend to be the ones that exceed 0.4 L1-norm	Consistent with the fact larger wingspans are heavier based on relationship between air vehicle weight and wingspan
Sensors	
Every electric variant with 200 pixel IR resolution is above 0.4 L1-norm No real trend in FOVs and resolutions for piston variants at L1-norms above 0.4	200 pixel resolutions offer lower value Max total (EO+IR) resolution for electric variants is 600 pixels if there is no 200 pixel option
Mission	
Higher operating altitudes above 650 m coincide with higher L1-norm values for electric variants No real trend in operating altitudes for piston variants at L1-norms above 0.4	L1-norm distance to ideal based on value (OMOE) and cost at an ideal point of coordinates (1,1); higher operating altitudes offer less value or cost more

Action	Justification
Reduce the EO resolution range from 400-800 pixels to 400-600 pixels for electric	Max combined (EO+IR) resolution is 1,000 pixels and no option for 200 pixels
Reduce the IR resolution range from 200-800 pixels to 400-600 pixels for electric	Electric design variants with 200 pixel IR resolution exhibit the worst L1-norm Max combined (EO+IR) resolution is 1,000 pixels and no option for 200 pixels
Reduce the wingspan range from 3-12 ft to 5-12 ft for piston variants	The lowest wingspans offer the worst L1-norm values
Reduce the wingspan range from 6-15 ft to 6-13 ft for electric variants	The highest wingspans offer the worst L1-norm values
Reduce operating altitude range from 300-700 m to 300-650 m for electric	Difference from attack helicopter operating at 1,000 m altitude objective 400 m Consistent with findings from piston team; above 650 m has the worst L1-norms

Cumulative Design Space Description
<u>Integrated Set Space 1</u> ELECTRIC Wingspan (6-15 ft); EO Resolution (400-800 pixels); IR Resolution (200-800 pixels); EO FOV (30-75 degrees); IR FOV (30-90 degrees); Operating Altitude (300-1,000 m) 86 Design Choices -- 86 are Viable -- Value (OMOE) Range: 40.82-48.45 -- Cost Range: $137.23K-$139.91K PISTON Wingspan (3-12 ft); EO and IR Resolution (400-1,200 pixels); EO and IR FOV (30-90 degrees); Operating Altitude (300-650 m) 262 Design Choices -- 262 are Viable -- Value (OMOE) Range: 47.78-60.91 -- Cost Range: $138.81K-$145.22K
<u>Reduction 1</u> ELECTRIC Eliminate 14 ft and 15 ft wingspans; Eliminate 800 pixel EO resolutions; Eliminate 200 pixel and 800 pixel IR resolutions; Eliminate operating altitudes > 650 m 52 Design Choices -- 52 are Viable -- Value (OMOE) Range: 42.86-48.45 -- Cost Range: $137.23K-$139.53K PISTON Eliminate 3 ft and 4 ft wingspans 241 Design Choices -- 241 are Viable -- Value (OMOE) Range: 47.78-60.91 -- Cost Range: $140.27K-$145.22K
<u>Integrated Set Space 2</u> ELECTRIC Wingspan (6-13 ft); EO Resolution (400-600 pixels); IR Resolution (400-600 pixels); EO FOV (30-75 degrees); IR FOV (30-90 degrees); Operating Altitude (300-650 m) 281 Design Choices -- 70 are Viable -- Value (OMOE) Range: 41.85-48.45 -- Cost Range: $137.23K-$139.53K PISTON Wingspan (3-12 ft); EO and IR Resolution (400-1,200 pixels); EO and IR FOV (30-90 degrees); Operating Altitude (300-650 m) 519 Design Choices -- 308 are Viable -- Value (OMOE) Range: 47.78-62.65 -- Cost Range: $140.67K-$145.22K

Information Communicated to the Specialties	
New Integrated Set Space 2 New Electric Wingspan Range: 6 ft to 13 ft New Piston Wingspan Range: 5 ft to 12 ft	New Electric EO Resolution Range: 400 pixels to 600 pixels New Electric IR Resolution Range: 400 pixels to 600 pixels New Electric Operating Altitude Range: 300 m to 650 m

4. Step 10: Reduce the Set Space by Elimination

The design integration manager reduces the current set space (Integrated Set Space 1) by carrying out the actions from Table 28, including: eliminating piston variants with 3 ft and 4 ft wingspans; and, for the electric variants, eliminating 14 ft and 15 ft wingspans; eliminating 200 pixel EO resolutions; eliminating 200 pixel and 800 pixel IR resolutions; and eliminating operating altitudes above 650 m (see Table 29).

Table 29. Reduction Table 1 (Combined Teams)

Total Reductions (Combined Teams)											
Piston Team						Electric Team					
Structures	Sensors				Mission	Structures	Sensors				Mission
Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)	Wingspan (ft)	EO Resolution (pixels)	IR Resolution (pixels)	EO FOV (degrees)	IR FOV (degrees)	Operating Altitude (m)
1	200	200	15	15	300	1	200	200	15	15	300
2	400	400	30	30	350	2	400	400	30	30	350
3	600	600	45	45	400	3	600	600	45	45	400
4	800	800	60	60	450	4	800	800	60	60	450
5	1000	1000	75	75	500	5	1000	1000	75	75	500
6	1200	1200	90	90	550	6	1200	1200	90	90	550
7	1400	1400			600	7	1400	1400			600
8	1600	1600			650	8	1600	1600			650
9	1800	1800			700	9	1800	1800			700
10					750	10					750
11					800	11					800
12					850	12					850
13					900	13					900
14					950	14					950
15					1000	15					1000
16						16					

5. Step 11: Refine the Reduced Set Space in Greater Detail

The reduced set space is refined by performing NOB LHC sampling with the new MOP ranges and running the UAS simulation tool again with these new LHC samples to generate the new integrated set space (Integrated Set Space 2).

6. Step 12: Explore the Refined Set Space

Each specialty from both teams explores the refined set space from its own perspective.

7. Step 13: Communicate the Specialty Set Space Preferences

Each specialty from both teams communicates to the design integration manager that it is satisfied with the current set space and has no further reduction recommendations.

8. Step 14: Create the Viable Set Space

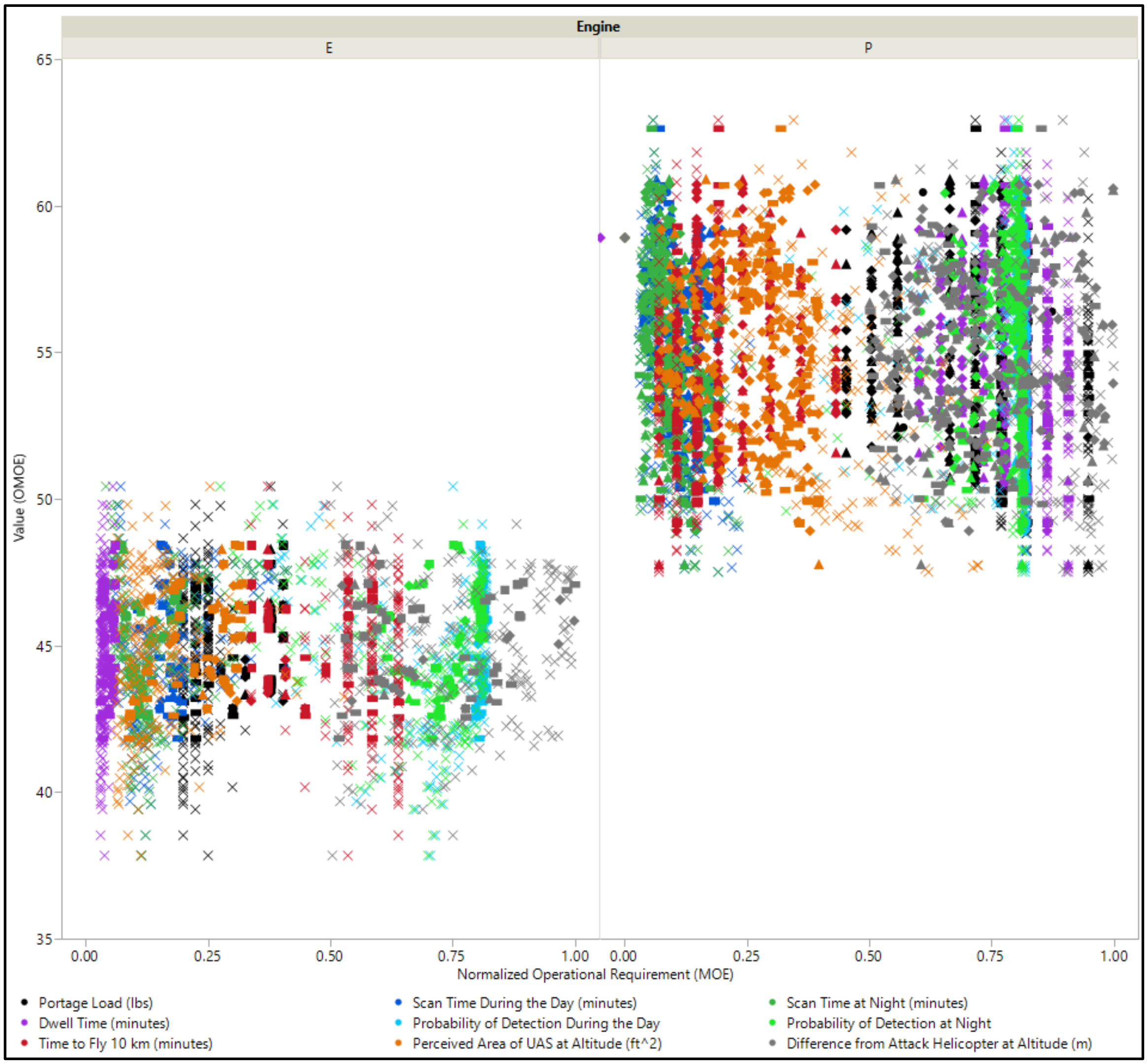
The viable set space is created for the electric and piston teams combined by:

- Verifying all the set reduction criteria have been applied and all input factors (MOPs) are within the reduced and agreed upon ranges;

- Eliminating design variants that do not meet every MOE; and
- Checking that the viable design variants do indeed spread across the whole

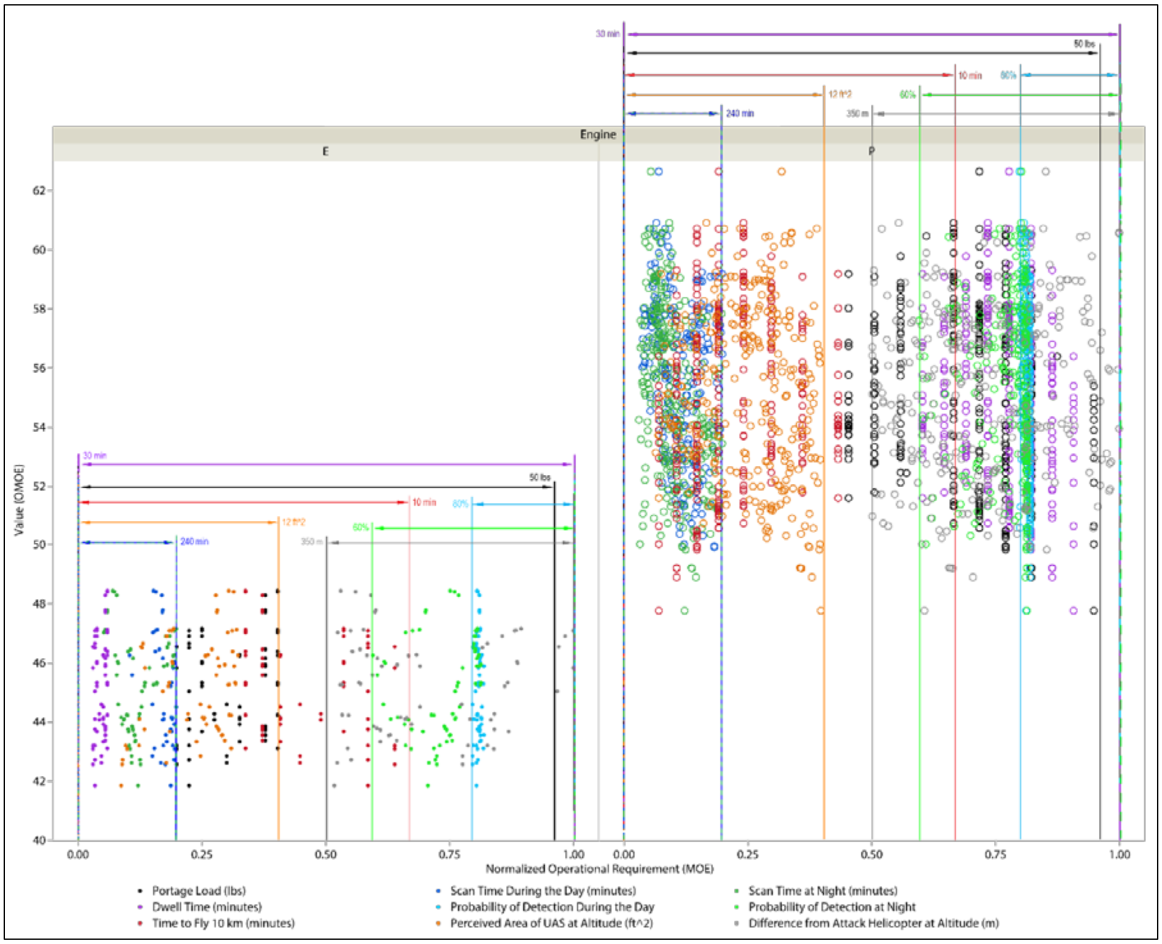
Using the most current integrated set space for the combined teams (Integrated Set Space 2), design points are eliminated in accordance with the cumulative, agreed upon set reduction criteria captured in Table 29. The majority of the points fall within the acceptable ranges, especially those generated in the most recent simulation run, since the new reduction criteria are applied to the NOB LHC tool when generating the refined inputs. Points carried in from previous reductions and integrations do not necessarily adhere to the most current criteria and need to be eliminated. When only points within acceptable ranges remain, those that do not meet every MOE are eliminated. As a clean-up or finish pass, each MOP range is checked to ensure design solutions cover the whole range, otherwise it may be worth considering another reduction.

Figure 57 shows the value versus MOE for all points in the integrated set space that fall within the acceptable and agreed upon MOP ranges. The x's indicate points that are eliminated because of failure to meet every MOE. Although there are around 800 points to consider, it is noticeable the majority of them are not viable. Dot, triangle, and diamond markers are variants generated in iterations of electric or piston integrated set spaces that made it through the first, second, or third reduction, respectively; i.e., they exhibit MOP values that are still within the current, agreed upon ranges. Square markers are those leftover from the first reduction in the integrated set space for the two teams combined, and rectangular markers are newly refined variants. A cleaned up view with the non-viable solutions eliminated and vertical lines for the acceptable bounds of each MOE is shown in Figure 58.



Note: 68.4 is the maximum value obtainable with every MOE at full performance.

Figure 57. Creating the Viable Set Space (Combined Teams)



Note: 68.4 is the maximum value obtainable with every MOE at full performance.

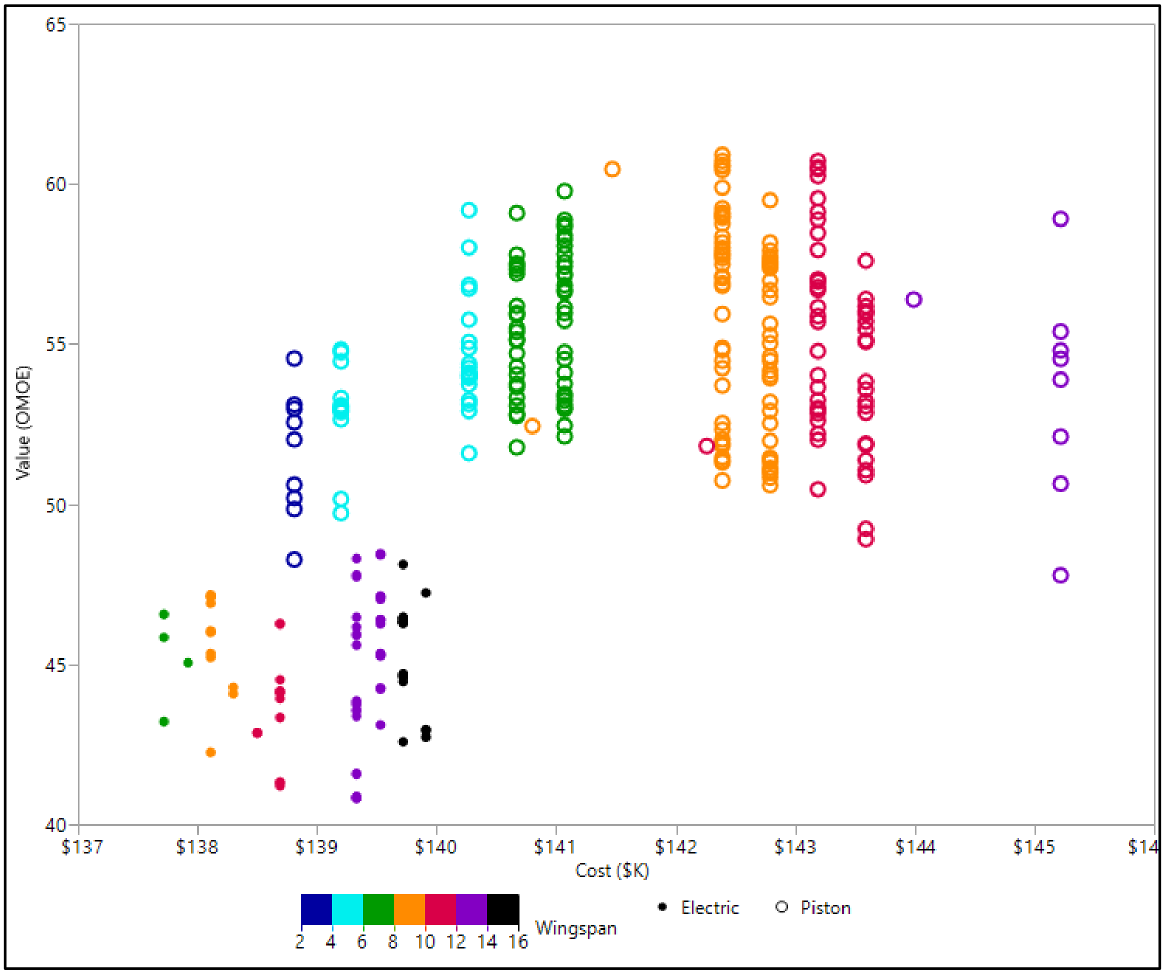
Figure 58. Viable Set Space: Value vs. Operational Requirement (Combined Teams)

9. Step 15: Explore the Viable Set Space

The viable solutions are explored and considered in a manner that leads to selection (e.g., Pareto (or other optimality) frontier, value versus cost, etc.).

10. Step 16: Select a Design Solution

Graphically displaying the value versus cost for all viable design solutions leads to the probable down-selection to piston variants (see Figure 59).



Note: 68.4 is the maximum value obtainable with every MOE at full performance.

Figure 59. Value vs. Cost of Viable Solutions (Combined Teams)

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RESULTS AND DISCUSSION OF APPLYING THE IMPROVED SBD APPROACH TO UAS DESIGN

A. RESULTS

1. SBD Results

Through the application of SBD and based on the current technology and the desired stakeholder value, a piston variant is recommended, unless there is a threshold where the two engine types overlap. For the mission associated with this example, it makes sense to pursue the next part of the design effort with a piston variant.

As a separate effort, which is also intended in this example, the company can identify opportune areas of technology to invest in and decide where to develop to improve electric variants and make them a more attractive option in the future. For example, the limitations on payload capacity that lead to a disadvantage in sensor packages and the drawbacks in airspeed and endurance serve as good starting points for a path forward; efforts to advance engine, sensor, and battery technologies in order to reduce sensor weight, improve engine performance, and increase battery output, storage, and charging capacity are beneficial.

2. PBD Results

A SBD approach is demonstrated in this example, but the problem could also have been solved right up front using a PBD approach and selecting the design solution that looks most promising. Table 30 summarizes the number of design solutions available to compare in the design space, how many are viable, and the value (OMOE) and cost ranges for these viable points as the design progresses from initial set creation (Integrated Set Space 1) through the various iterations of set reduction and refinement (all subsequent integrated set spaces).

Table 30. Set Space Progression and Summary Statistics

		Number of Design Points	Number of Viable Design Points	Value Range (OMOE)	Cost Range ($K)
Electric Team	Integrated Set Space 1	512	0	---	---
	Integrated Set Space 2	566	10	40.87 - 48.30	$137.72 - $139.91
	Integrated Set Space 3	779	36	40.82 - 48.30	$137.23 - $139.91
	Integrated Set Space 4	922	86	40.82 - 48.45	$137.23 - $139.91
Piston Team	Integrated Set Space 1	512	6	51.82 - 60.46	$140.67 - $143.99
	Integrated Set Space 2	565	108	43.95 - 60.91	$138.81 - $145.22
	Integrated Set Space 3	790	262	47.78 - 60.91	$138.81 - $145.22
Combined Teams	Integrated Set Space 1	Electric			
		86	86	40.82 - 48.45	$137.23 - $139.91
		Piston			
		262	262	47.78 - 60.91	$138.81 - $145.22
	Integrated Set Space 2	Electric			
		281	70	41.85 - 48.45	$137.23 - $139.53
		Piston			
		519	308	47.78 - 62.65	$140.67 - $145.22

Using a PBD approach would mean choosing between six viable points in the piston regime and having nothing to select from in the electric regime (see the number of viable design points in Integrated Set Space 1). Of the six viable piston solutions, the maximum achievable value is 60.46 (out of 68.4), but when SBD is applied, 308 viable solutions are available to consider with the maximum value increased to 62.65. The case is similar for electric variants and perhaps even more compelling; what starts off as zero viable options turns into seventy with the maximum value increasing from 48.30 to 48.45 over the iterations. These are terrific examples of how the implementation of SBD allows for further exploration of the design space in increased fidelity and opens up more options to consider with higher overall system value.

Figure 60 shows the six original viable piston points—one of which is eliminated based on difference from attack helicopter—and emphasizes some of the new points generated through SBD (in the [EO FOV, IR FOV, EO resolution, IR resolution, operating altitude] format).

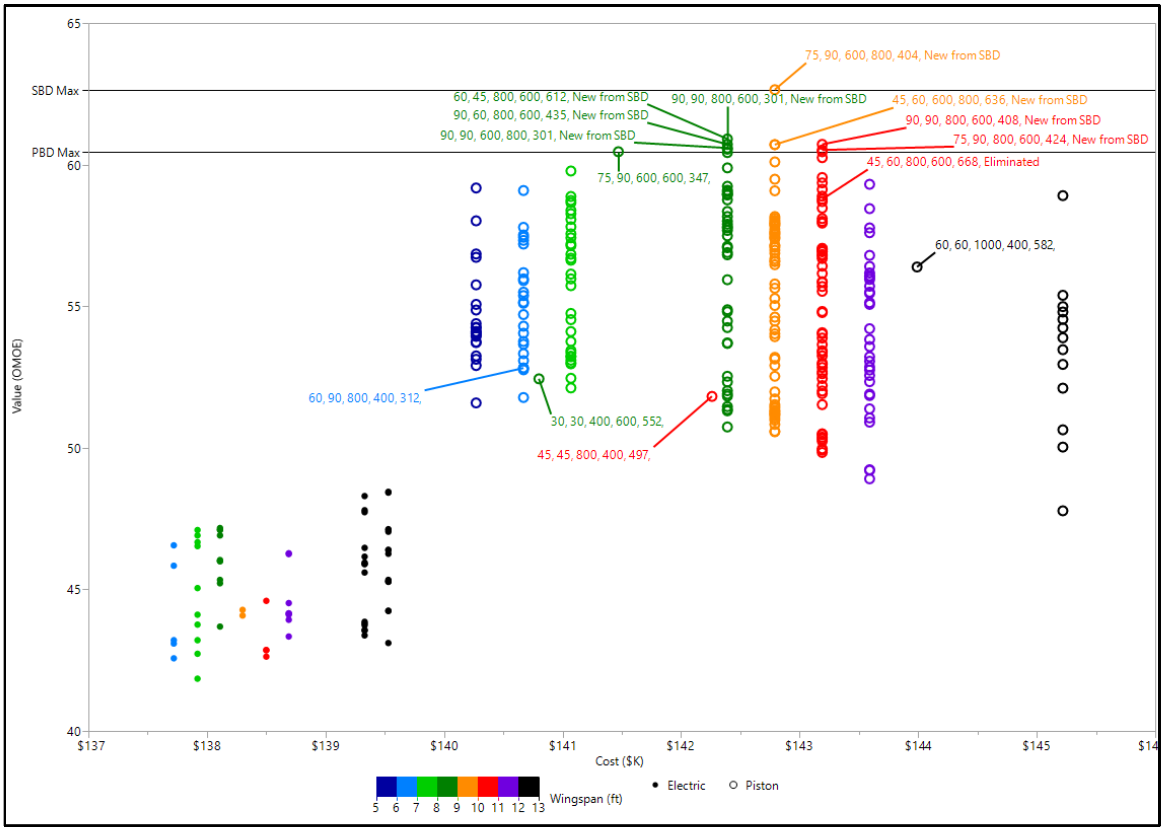


Figure 60. PBD and SBD Comparison: Value vs. Cost

3. Benefits of SBD Compared to PBD for the UAS Example

A notable benefit of solving this problem from an SBD approach is the amount and type of knowledge and understanding gained throughout the process to end up with the best selection based on learning and history. Each individual specialty (e.g., Structures, Propulsion, Weight and Balance, Sensors, and Mission) discusses within itself and explores the design space from its own perspective, giving it the opportunity to truly explore the tradespace, MOPs, and MOEs under its purview. The specialties then communicate their inputs, preferences, and expertise as a whole, which instills a sense of belonging, importance, and respect (e.g., negotiations, KARs, etc.). The design space is looked at as an integrated whole, giving everyone an idea of how their individual parts contribute to the overall system behavior and impact other specialties, while new information causes the specialties to have dialogue (e.g., new and refined integrated design spaces).

B. DISCUSSION

The implementation of SBD exemplified in this example is consistent with the seven characteristics of SBPD and two principles of SBT required to identify a design approach as set-based (Ghosh and Seering 2014).

1. Seven Characteristics of SBPD Exhibited in the UAS Example

a. Emphasis on Frequent, Low-Fidelity Prototyping

The information, knowledge, and understanding gained and learned in the computational study carried out in this example lead to where the company can focus on prototyping. The individual specialties may have engaged in their own prototyping efforts along the way that are not discussed, such as wind tunnel experiments by Structures to explore wingspan. The results of this example narrow down what starting specifications to use for building prototypes going into the next stage of design.

b. Tolerance for Under-defined System Specifications

Each of the specialties presents an initial range of input factors as opposed to a few very select and specific values (e.g., wingspans from 1 ft to 16 ft). The preferred input factors and reductions are communicated as sets of possible values or eliminations instead of singular quantities (e.g., preferred operating altitudes between 300 m and 1,000 m, or eliminate operating altitudes from 651 m to 700 m). The specialties continue to work with sets of values throughout the entire design process and over every iteration making "piecewise commitments" (Ghosh and Seering 2014, 4) until they are sufficiently narrowed down or ready for selection.

c. More Efficient Communication among Subsystems

The entire design team is made up of several individual specialties (Structures, Propulsion, Weight and Balance, Sensors, Mission) and a lead or chief engineer role (design integration manager). Each of the specialties is eager to work together with the other specialties from the beginning to obtain their preferences and realize the opportunities and constraints each one has on the other (such as how the sensor options are limited based on wingspan).

As the design progresses, the specialties engage in dialogue appropriately as new information is received, sets are reduced, and impacts arise. All preferences and important information are concurrently communicated to the design integration manager (KARs, etc.), who also helps prompt the interface between certain specialties requiring more elaborate discussion or negotiation. Rich sets of information are shared by each specialty, which produces a much more efficient communication scheme and bypasses inefficiencies that would result from missing information or poor understanding.

d. Emphasis on Documenting Lessons Learned and New Knowledge

All of the lessons learned and new information are documented by the individual specialties in the form of analysis findings, plots, and communicated preferences. The design integration manager documents all the information received, implications, resulting actions, justification, cumulative design space description, and knowledge passed on to the specialties in a KAR; no decisions are made without justification and buy-in from the specialties, and they are all traceable and documented.

The set reductions are captured in reduction tables in order to follow how the initial sets of input factors change chronologically over subsequent reductions from those that were originally proposed.

Both the KARs and reduction tables enable the convenience of going backwards and re-visiting previous phases of the design space when changes are introduced and reduce re-work. There is an emphasis on learning new information and gaining new knowledge by refining the design space (new simulation runs); adding to the number of points gives more information, while lessening the number of points leads to a selection.

e. Support for Decentralized Leadership Structure and Distributed, Non-collocated Teams

The electric team and piston team are not geographically collocated, work for different divisions of the same company, and are afforded considerable autonomy. Both teams are tasked by company leadership, but are encouraged to execute the design process in a manner of their choosing (for the sake of example, it is SBD) with minimal oversight and their own organization, structure, and hierarchy. A design integration manager acts as a hub for each team to facilitate communications and ensure documentation, but is not located at company headquarters. Common software, simulation tools, and remote technologies empower the teams to work at a distance from each other and still collaborate on the combined space to solve the problem and accomplish the intent.

f. Supplier/Subsystem Exploration of Optimality

Part of this problem focused on comparing the value of two different product lines (electric and piston) to illuminate any shortcomings and identify future areas of research or investment for improvement. Often times a third party vendor will supply some of the more specialized parts and subsystems. The findings from a design effort such as this can be used to make suppliers aware of their strengths and deficiencies in value proposition. A supplier might be motivated to explore how it can re-design or improve its products and maintain its stance at the forefront of technology if told a specific change would increase performance (e.g., suggesting to the sensor supplier that it could reduce the sensor size and increase performance).

Another way of interpreting this characteristic in the context of this example (and similar to how Toyota interacts with its suppliers) is to work directly with the suppliers in parallel: start by telling the sensor supplier there is a requirement for EO and IR sensors that can scan for and detect humans and they will be used on a small UAS with 1–16 ft wingspan; the sensor supplier puts together a list of options and available products for the UAS team to consider and choose from; the design space is narrowed, so the supplier is notified the sensors cannot weight more than 15 lbs; the sensor supplier re-evaluates its inventory and responds with new options and possibly some negotiation to offer the best solution for the UAS team, while still maintaining profit (and without being tasked to develop a product that meets specifications initially determined by the UAS team).

In terms of subsystem exploration of optimality, the individual specialties conduct tradeoff analysis at their level and study design spaces across interfaces by integration. Communicating in sets of design solutions fosters an extended exploration of the design space, which leads to more optimal solutions at the system level.

g. Support for Flow-Up Knowledge Creation

All of the individual specialties are involved in the design decision-making process, which is in contrast to a head boss making all the decisions up front and flowing them top-down to the specialties. The specialties explore their set space and the integrated design spaces from their own perspectives and use their subject matter expertise to develop preferences for communicating bottom-up. Design decisions are not made up front and passed down, but rather evolve through learning and understanding and then elevated.

2. Two Principles of SBT Exhibited in the UAS Example

a. Considering Sets of Distinct Alternatives Concurrently

The first principle encompasses three unique aspects: sets, distinct alternatives, and concurrent. The above explanations supporting the seven characteristics substantiate the design process for both teams involves sets – synthesizing sets, or the creation of sets (NOB LHC), sets of input factors (ranges of values), sets of design solutions (integrated set spaces through simulation), and set elimination, or the reduction of sets.

Ideally, as many options as possible are considered to accomplish a prime advantage of SBD, which is better realized with truly distinct concepts. The UAS example is based on the development of distinct concepts as opposed to the development of sets of variants of a single concept in that it considers electric and piston engine types. If it was only focused on one engine type, then it would restrict the solution space of the UAS and subsist as a limited case of SBD with the risk of reverting to traditional PBD.

"Concurrent development describes the simultaneous development of distinct concepts" (Ghosh and Seering 2014, 7). All of the possible design alternatives for the electric and piston concepts synthesized across multiple sets of input factors are explored and considered simultaneously, which signifies the UAS example meets the first principle of considering sets of distinct alternatives concurrently.

b. Delaying Convergent Decision Making

The second principle supports delaying convergence to a single solution and opens up the opportunity to consider additional options for as long as reasonably possible, while new information becomes available. The solution space is gradually narrowed in designs that follow this principle, as opposed to rapidly converging on single, promising solution like is normally seen in PBD.

The UAS example starts with a set space that covers wingspans of 1–16 ft, EO and IR resolutions of 200–1,800 pixels, EO and IR FOVs of 15–90 degrees, and operating altitudes of 300–1,000 m. The set space is narrowed with each set reduction until it is appropriate to converge on a design selection. By the time selection is desired, the set space, including both engine types, has been narrowed to contain 5–13 ft wingspans, EO and IR resolutions of 400–1,200 pixels, EO and IR FOVs of 30–90 degrees, and operating altitudes of 300–650 m. The decisions to narrow the sets of input factors are also delayed from one reduction to the next. Large chunks of the design space are not removed too hastily to leave more of it open to explore and to ensure there is solid justification for elimination.

The decision for where to focus future investments is also delayed and converged in the UAS example. The areas of technology that need further development or improvement to make electric variants more competitive and viable in an emerging market are unknown at the start of design, or if there is even any significant difference between the two engine types. Iterating through the design space, learning more about the relationships of MOPs to MOEs and MOEs to OMOEs, and observing the distinguishing characteristics and performance of each engine type, make it apparent what is needed and what direction to pursue. Working with sets of electric solutions and sets of piston solutions while accomplishing design, ultimately converges on piston solutions (based on higher value space), but illuminates the strengths and shortcomings in the process, leading to more informed decisions.

VII. CONCLUSION AND FUTURE WORK

A. CONCLUSION

The UAS example demonstrates a complete implementation of SBD, which is proven through its ability to meet the seven characteristics of SBPD and two principles of SBT required to identify a design approach as set-based. The concept of SBD is better defined and executed in a systems context through the sixteen process steps introduced in this dissertation than is currently found in the literature. Methods for creating and eliminating sets are introduced, including a space-filling (NOB LHC) and response-driven (simulation) method for creating and refining sets and several unique ways of reducing the sets and eliminating infeasible or highly dominated design solutions (system-level viability and OMOE investigations, specialty-level MOP and MOE investigations, distance to the ideal point, and visual inspection).

The improved definition of SBD steps introduced in this dissertation are applied to illustrate the: identification of engineering specialties, design factor sets, and engineering modeling tools (steps 1-3); creation of sets (steps 4, 7, and 14); exploration and evaluation of the specialty set space and integrated set spaces (steps 5, 8, 12, and 15); learning and understanding of new knowledge and communication of preferences and important information (steps 6, 9, and 13); reduction and refinement of sets (steps 10 and 11); convergence and selection of a design solution (step 16); and thorough documentation of actions, decisions, and justifications (overarching).

The emphasis on distinguishing the various set spaces (specialty, integrated, reduced, refined, and viable set spaces) and the objective and repeatable steps to ultimately converge on a design solution facilitate an enhanced understanding of SBD and clarify its execution. The consistent terminology and iterative nature of the steps also instill a better sense of which specialties are involved when, what the demand for resources is, where the design is at relative to the overall design space, whether the designs under consideration are feasible and viable, and what the impact and level of confidence is for decisions.

Approaching the UAS example problem in a set-based way results in more viable options to consider with higher system-level performance for comparable cost. Major benefits of SBD stand out in the UAS example (such as rich and efficient communications, greater exploration of the design space and opportunity for specialties to consider it from their own perspectives, globally optimal solutions, in-depth knowledge about the technical problem and potential solution set, delayed decisions and ensured feasibility (and viability) before commitment), while remediating some of the challenges (including poor definition and lack of instruction on how to execute SBD, how to assess design solutions for viability, how to assess mission effects on a large number of alternatives, difficulty determining when and where to make set reductions, and set communication and negotiation among disparate design teams). Incorporating the SBD approach offered in this dissertation results in a viable design that is comparable to other design methodologies.

The improved definition and demonstration of how to create and eliminate sets in a systems context for the implementation of SBD as an alternative design approach to PBD is encouraging. SBD yields recognizable improvements over PBD in complex engineering problems, and the contributions made herein stimulate the momentum towards this new approach and way of thinking, reasoning, and communicating about design problems.

B. FUTURE WORK

1. Incorporating SBD into SET

The U.S. Navy has recently begun a Systems Engineering Transformation (SET) initiative to accelerate the delivery of fully integrated capabilities, which are designed, developed, and sustained in a model-based digital environment (see Figure 61).

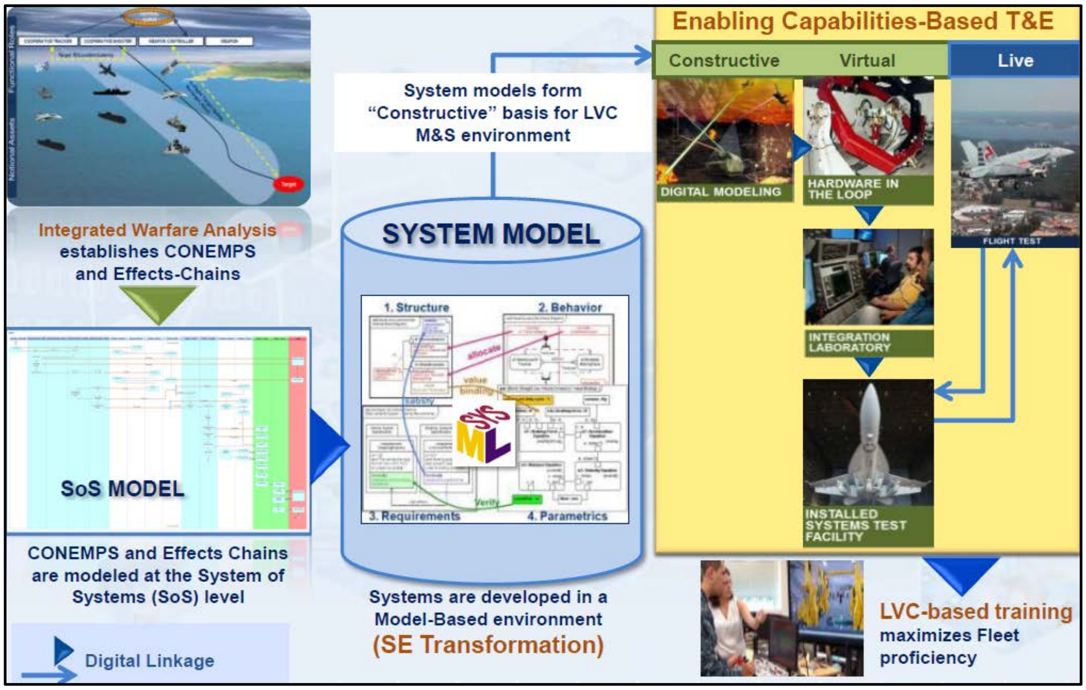


Figure 61. SET Environment. Source: Grosklags (2018).

In an effort to reduce development cycle times and re-gain warfighting advantage, SET aims to better manage requirements traceability, eliminate or reduce SETR events and contract data requirements lists (CDRLs), and leverage multidisciplinary design, analysis, and optimization (MDAO) and high performance computing (HPC) by developing and specifying systems as models. The system specification (performance spec) is developed in a collaborative workspace by a cross-functional team, instantiated in a model using systems modeling language (SysML), and placed on contract as a model instead of a paper spec. There are four elements to the SET framework (see Figure 62):

1. Effectiveness analysis and capability development document (CDD) optimization;

2. Instantiate specification in a system model;

3. Develop initial balanced design via MDAO/SBD and instantiate/verify with models; and

4. Collaborate design and manufacture release decisions via integrated development environment (IDE).

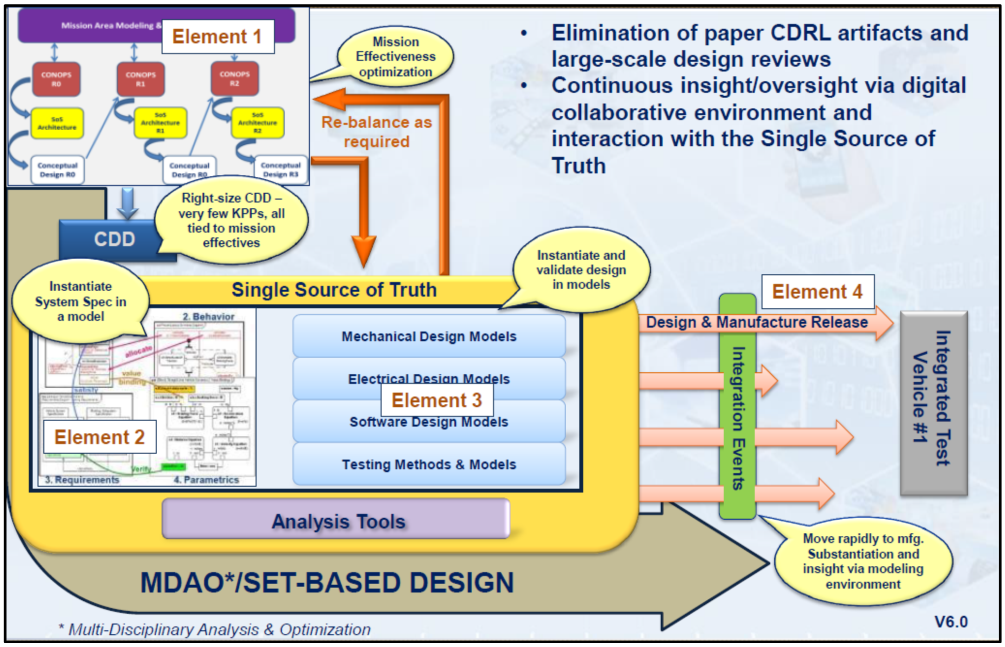


Figure 62. Elements of SET. Source: Grosklags (2018).

In order for SBD to be applicable in the current DoD capability-based acquisition drive for MBSE, it must fit within the SET environment. Elements 1 and 2 pertain to pre-contract award, while Elements 3 and 4 pertain to post-contract award. It makes sense for SBD to fit within Elements 1 and 2, but further exploration and guidance is needed for how to formally incorporate it. The mission effectiveness models and concepts of employment (CONEMPs) stream down from Element 1 and into the decomposed and allocated subsystem requirements, emerging designs, and verification tools of Element 2. SBD might also be applicable to Element 3, since it deals directly with the individual technical domains and their processes and methods for applying systems engineering models to reduce system design, development, and verification costs, while increasing system effectiveness; but it depends on whether the timing of the Elements is hardened, or if they are just guidelines. SBD is most effective in the phases before detailed design, which typically occur during pre-contract award activities.

2. Incorporating SBD into MBSE

SysML tools for MBSE are a critical enabler for the SET initiative to move from a document-centric to model-centric environment. SysML diagrams describe a system in a model-based manner based on structure, behavior, requirements, and parametric pillars. It enables the system model to be viewed from multiple perspectives and identifies and organizes the fundamental system components, relationships, interfaces, processes, boundaries, constraints, and behaviors. SysML is supported by the International Council on Systems Engineering (INCOSE) and stems from a RFP pertaining to "UML [unified modeling language] for Systems Engineering."

The INCOSE repository holds a variety of MBSE methodologies, and several major companies and organizations have defined SysML products to enable MBSE methods and processes (such as No Magic Cameo – formerly MagicDraw with SysML plugin – Innoslate, Vitech CORE, and integrated MBSE environment tools like IBM Rhapsody and Papyrus 4 SysML). Most SysML tools establish a hierarchy that flows from capabilities to components, which is conducive to traditional PBD methodologies. Additional efforts are needed to develop SysML software packages for SBD, which instead map to the conceptual level with banded tolerances after performing several iterations of set creation, refinement, and elimination.

Perhaps the majority of the SBD tasks are accomplished during the analytical modeling portion of MBSE and then feed into the system architectural model as shown in in Figure 63, or maybe the sets of values resulting from SBD execution are inputted into SysML products as constraints within the parametric pillar; regardless, more investigation is needed to determine how to meld SBD efforts into a MBSE and SysML environment.

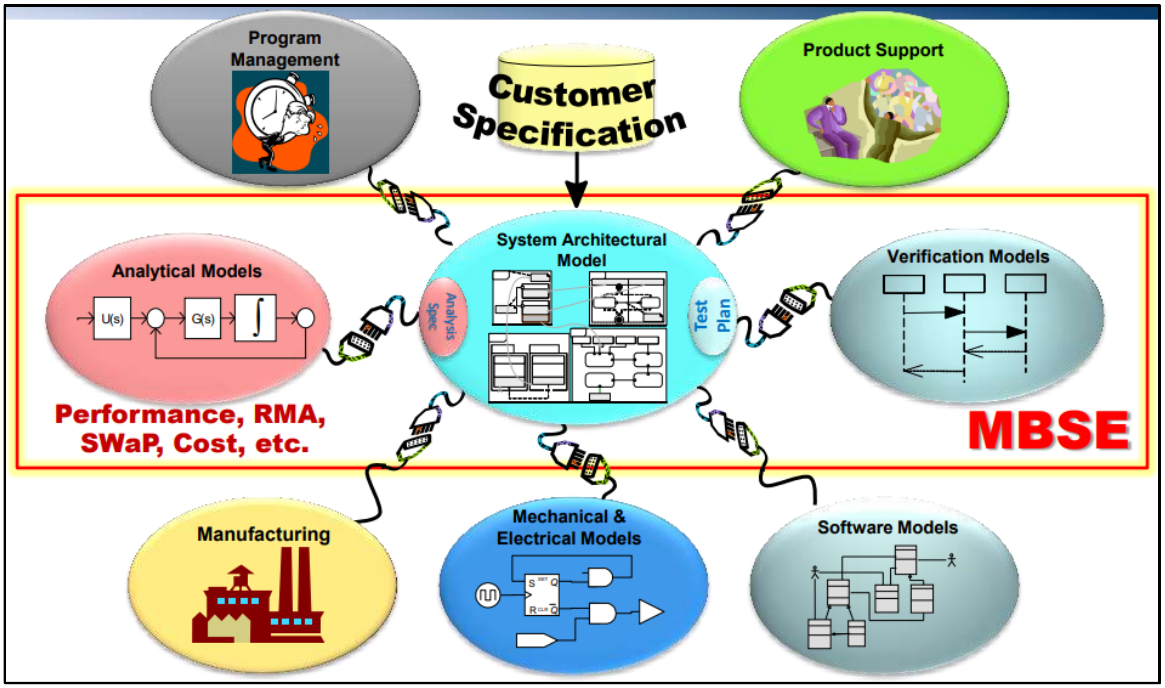


Figure 63. MBSE Integration across Domains. Source: Hart (2015).

There must also be a way to perform sensitivity analysis within the model. Current SysML tools have the ability to simulate the model, but doing this for sets (ranges) of values has not been specifically accommodated. Beery's (2016) work related to a MBSE Methodology for Employing Architecture in a Systems Analysis (MEASA) might be a good reference as it provides a comprehensive framework for the creation of SysML products to build dynamic architectures that can be combined with designed experiments of external models and simulations for use in assessing how the various system parameters impact operational performance and feasibility (see Figure 64).

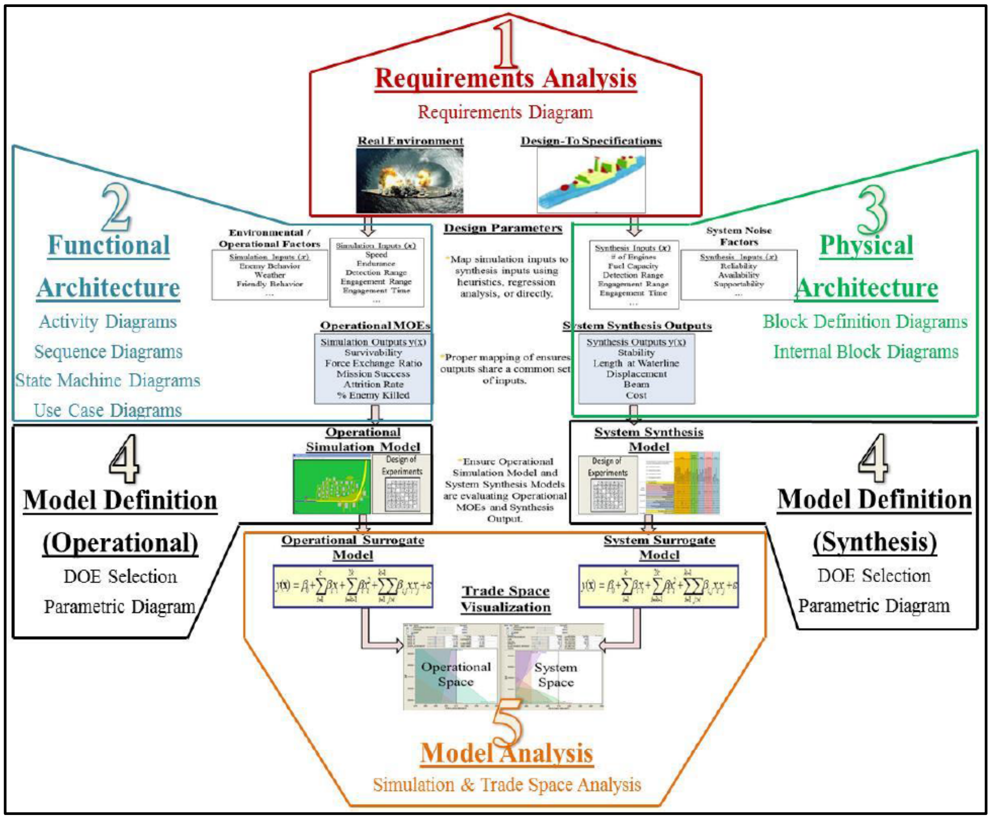


Figure 64. MBSE MEASA. Source: Beery (2016).

While the MEASA establishes a link between the SysML architecture models and analysis models, simulating a SysML architecture has yet to be achieved in practice. Building behavior models from system architecture products is the best approach to ensure traceability and aid in model validation. Research "has largely ignored the need to link system architecture products to detailed external models and simulations” to narrow down the feasible tradespace (Beery 2018, 3), but the exploitation of models capable of representing subsystem interactions, demonstrating emergent behavior, and realizing the impact of system operation and implementation modifications is promising for agent based models (ABM) especially (e.g., Orchestrated Simulation through Modeling [OSM]; Cummings 2015), and other executable architectures, including equation based models (EBM), petri nets, Markov chains, and network models.

3. Validating the Improved SBD Process

The instances of SBD application in the DoD (i.e., SSC, MCM, ACV, Small Surface Combatant, and LDUUV) vary in implementation, but the general trends are extracted and combined with explanations in the literature to create the sixteen SBD process steps. The UAS example demonstrates support for the new steps in a theoretical sense, but formal attempts to apply and observe them are needed to determine and confirm their effectiveness in a practical sense. New and upcoming programs (such as rotary wing future vertical lift, next generation fixed wing, extended range 2x.2.75” rocket motor, and other complex systems) serve as good opportunities to try the SBD steps.

4. User-Defined LHCs in Ship Design Tools

The U.S. Navy has a vision for moving to a high-end toolset that integrates design definition and physics-based analysis tools for exploring ship design alternatives as is supported by Sullivan (2008), Eccles (2010), and Doerry (2012). Currently, Advanced Ship and Submarine Evaluation Tool (ASSET) software is used for total ship synthesis by combining numerous design disciplines, while the Leading Edge Architecture for Prototyping Systems (LEAPS) tool serves as a central hub for integrating the tools of each discipline in a common data environment. The Computational Research and Engineering Acquisition Tools and Environment (CREATE) program focuses on leveraging HPC power to develop high end toolsets and enable a process for rapidly designing and analyzing large numbers of ship designs (Kassel, Cooper, and Mackenna 2010).

Part of the CREATE program for ships includes the Rapid Ship Design Environment (RSDE) modeling tool, which is used for the rapid development, optimization, assessment, and integration of ship designs (Department of Defense 2019). RSDE can use the synthesized ships from ASSET to explore the design space and determine feasibility (through the ability to converge) and has a built-in LHC algorithm that is capable of varying inputs over a user-specified number of runs. A major contribution in this dissertation is using LHCs to create the sets for SBD, and it is important which LHC is selected because the design points are addressed vice curve fits with RSM. A future area of research is in modifying the RSDE source code to accommodate other types of LHCs, so the user can make an appropriate choice for each problem-specific design.

5. Role of Engineering Reasoning in SBD

Whitcomb and Hernandez (2017) describe the engineering reasoning approach of non-SBD processes as linear, starting with deduction to formulate the system objectives and then flowing through inductive and abductive until a singular solution emerges. SBD uses a non-linear, circular approach to converge on a design solution and operates on a different timeline for decisions using retroductive thinking (see Figure 65). Although the basic concepts have been identified, further studies are beneficial for understanding the role of engineering reasoning in SBD and will lead to an improved definition based on logic, phased processes, and directed applications of engineering methods.

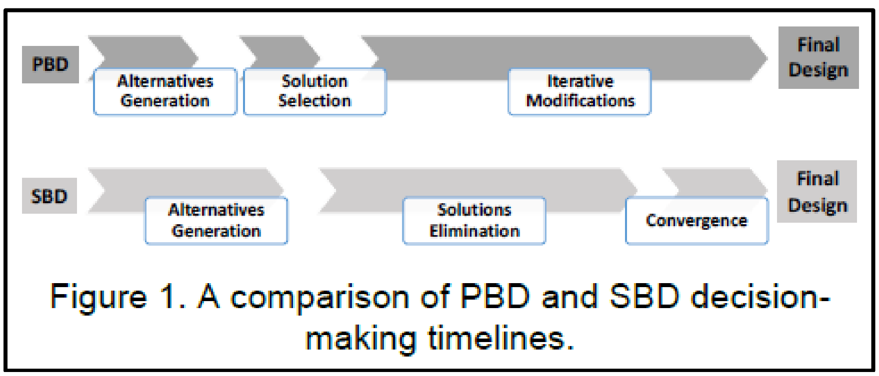


Figure 65. Decision-Making Timelines for PBD and SBD. Source: Whitcomb and Hernandez (2017).

6. SBD and Agile

Current DoD acquisition strategies tend to favor the design spiral for ships and vee model for other systems, however there is a recent push towards agile. Generally speaking, agile refers to the system development life cycle (SDLC) for the creation and management of software applications, but it is gaining in popularity as an approach that extends beyond software. For SBD to be competitive in future DoD system design and program acquisition efforts, it needs to be compatible with agile methodologies.

Several of the twelve agile principles ring true to SBD, but require a shift in mentality towards SBT (see Figure 66). For example, Principle 1 prioritizes early and continuous delivery of a product, Principle 2 welcomes changing requirements, Principle 5 offers motivated teams autonomy and trust, and Principle 9 is attentive to technical excellence and good design. From a set-based mindsight, each refined set could be the product being delivered continuously; changes are easier to accommodate in SBD; each individual specialty is encouraged to approach and explore the problem from its own perspective; and SBD yields designs optimized at the system-level. Even though a large part of agile is reducing documentation and SBD favors it, executing SBD in SET and MBSE environments will inherently minimize documentation. It is not to say that SBD is against reducing documentation, it is that the documentation serves a direct purpose and is beneficial for requirements changes, communication, and justification during design reviews.

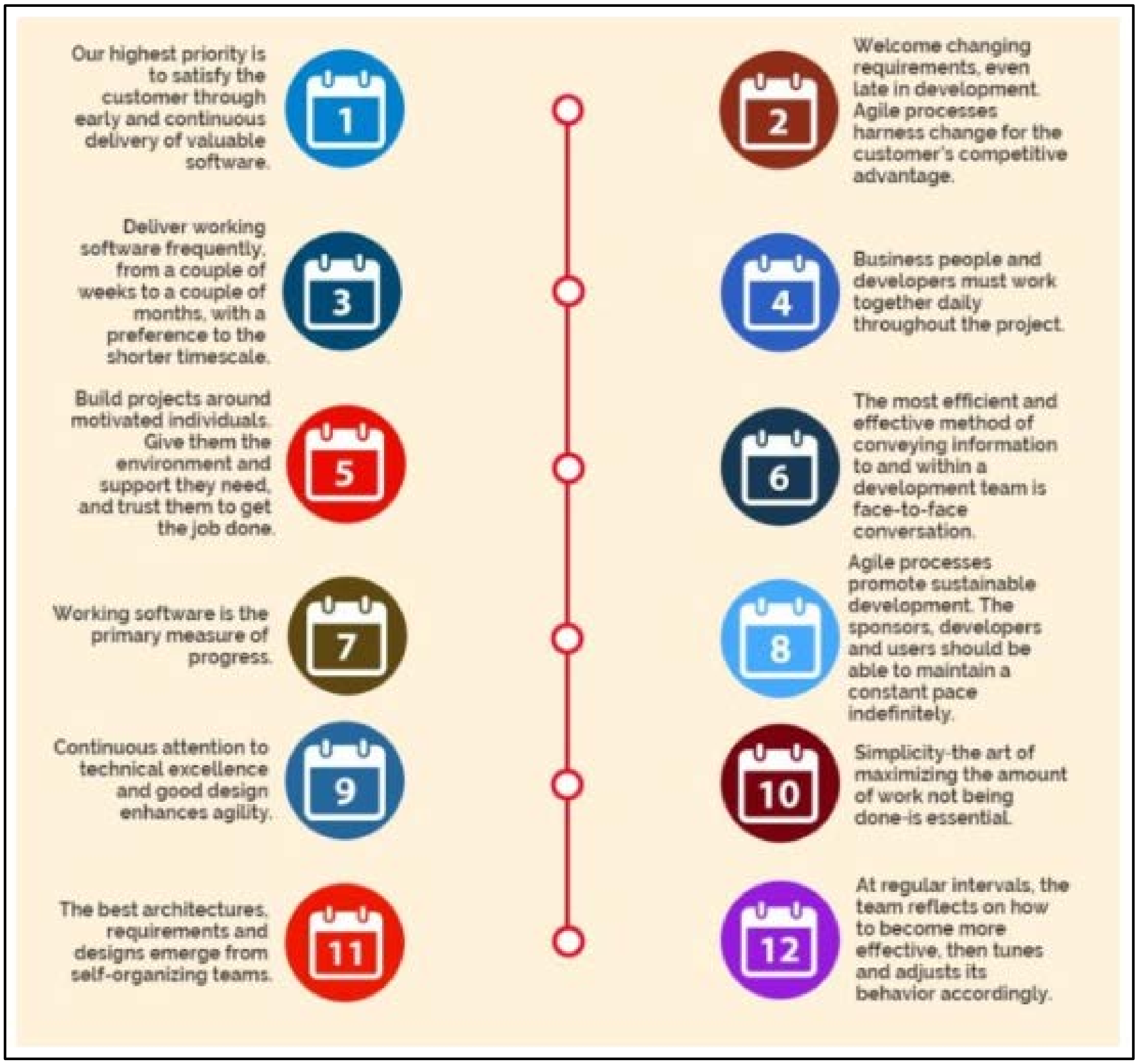


Figure 66. 12 Principles of Agile. Source: Landau (2017).

An alternative layout for the agile process (as opposed to the sprint view in Figure 17) illuminates some areas where a paradigm shift can support agile methods in a set-based manner and actually starts to resemble the circular logic of set-based reasoning (recall Figure 5). For example, if the sequence is more reflective of a *refine-explore-communicate-reduce* path instead of a *develop-release-accept-adjust* focus, it would be more conducive to SBD and SBT (see Figures 67 and 68). The sprint view could also be modified in a similar way to show *refine-explore-communicate-reduce* sprints as opposed to *discover-design-develop-test* sprints (see Figure 69). Further investigation into how (or if) SBD methodologies can be used with agile process models is needed as the modified images of Figures 68 and 69 are simply brainstorming.

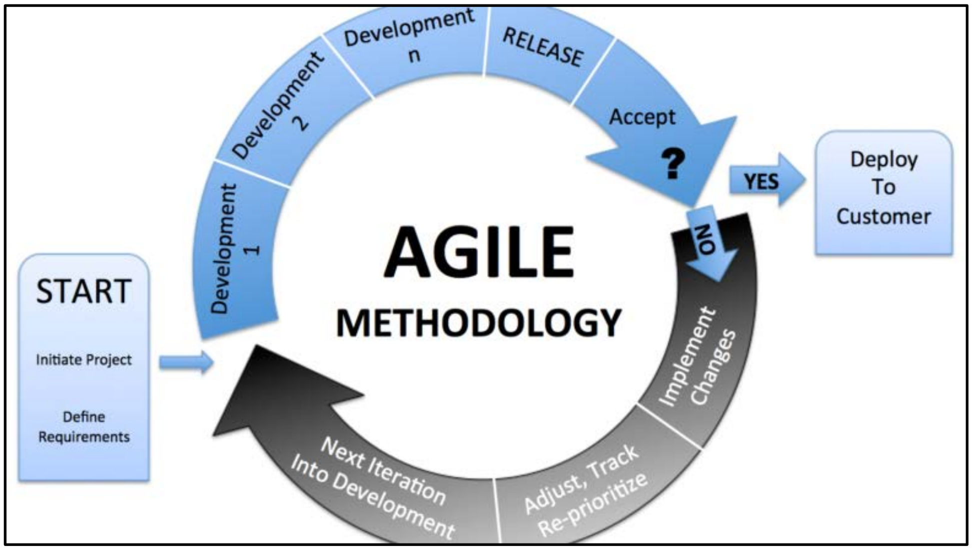


Figure 67. Alternate View of Agile. Source: Landau (2017).

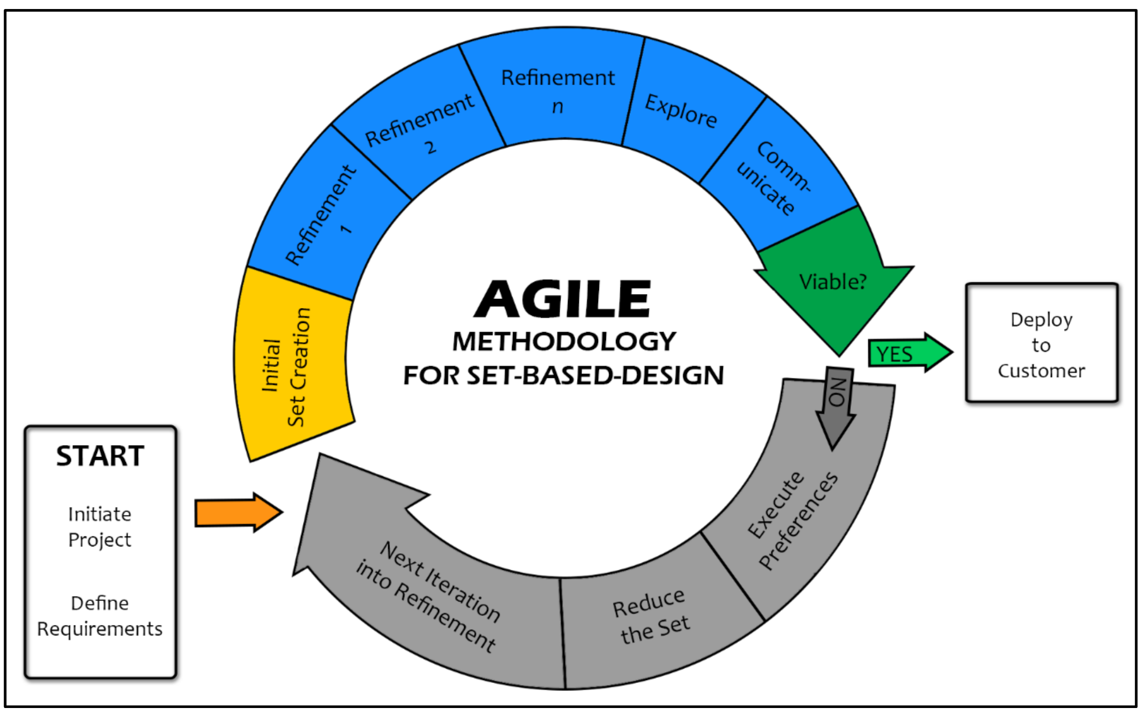


Figure 68. Modifying Agile to Accommodate SBD

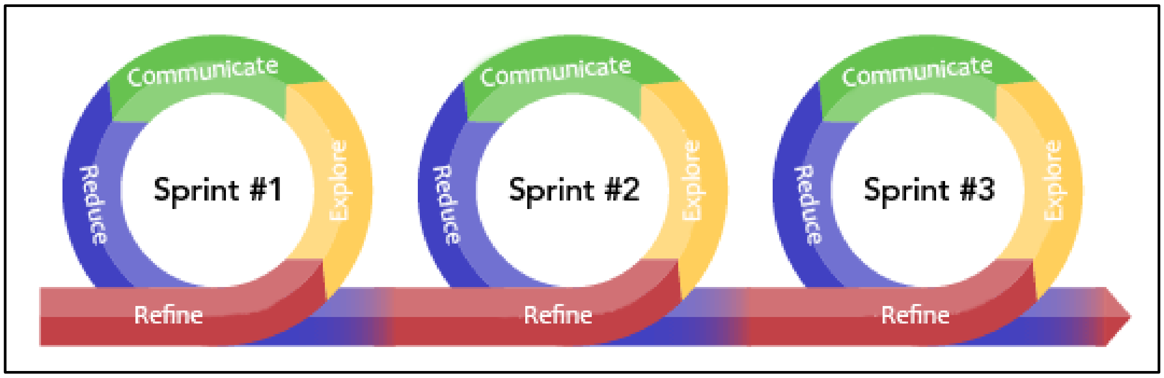


Figure 69. Agile Sprints Modified for SBD. Adapted from Hallman (2013).

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Agile Manifesto. 2001. "Manifesto for Agile Software Development." http://agilemanifesto.org/.

Akao, Yoji. 2004. *Quality Function Deployment: Integrating Customer Requirements into Product Design*. New York: Taylor and Francis.

ASQ. 2018. "House of Quality Template and Benefits." Accessed December 1, 2018. http://asq.org/learn-about-quality/qfd-quality-function-deployment/overview/overview.html/.

Audze, P., and V. Eglajs. 1977. "New Approach for Planning Out of Experiments." *Problems of Dynamics and Strengths*, 35: 104–107.

Bates, SJ, J. Sienz, and VV Toropov. 2004. "Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm." Presented at the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (April 19-22). Palm Springs, CA.

Beery, Paul T. 2016. "A Model Based Systems Engineering Methodology for Employing Architecture in System Analysis: Developing Simulation Models Using Systems Modeling Language Products to Link Architecture and Analysis." PhD Dissertation, Naval Postgraduate School.

Bernstein, Joshua I. 1998. "Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices." Master's thesis, Massachusetts Institute of Technology.

Bhushan, Navneet, and Kanwal Rai. 2004. *Strategic Decision Making: Applying the Analytic Hierarchy Process*. London: Springer-Verlag. http://www.springer.com/978-1-85233-756-8.

Black, Sue, Pail P. Boca, Jonathan P. Bowen, Jason Gorman, and Mike Hinchey. 2009. "Formal Versus Agile: Survival of the Fittest?" *Computer* 42, no. 9 (Sept): 37–45.

Blanchard, Benjamin S., and Wolter J. Fabrycky. 2011. *Systems Engineering and Analysis.* Upper Saddle River, NJ: Prentice Hall.

Boehm, Barry W. 1988. "A Spiral Model of Software Development and Enhancement." *Computer* 21(5): 61–72. https://doi.org/10.1109/2.59.

Boehm, Barry. 2000. *Spiral Development: Experience, Principles, and Refinements*. CMU/SEI-2000-SR-008. Pittsburg, PA: Carnegie Mellon University Software Engineering Institute.

Boehm, Barry. 2006. “A View of 20th- and 21st- Century Software Engineering.” In *Proceedings of the 28th International Conference on Software Engineering*: 12–29. Shanghai, China.

Box, George E. P., and Norman Richard Draper. 1987. *Empirical Model-Building and Response Surfaces.* New York: Wiley.

Burrow, John, Norbert Doerry, Mark Earnesty, Joe Was, Jim Myers, Jeff Banko, Jeff McConnell, Joshua Pepper, and Tracy Tafolla. 2014. “Concept Exploration of the Amphibious Combat Vehicle.” Paper presented at the Society of Naval Architects and Marine Engineers (SNAME) Conference, Houston, TX.

Chan, Jonathan, Amy Hays, Lucas Romas, and Jason Weaver. 2016. “Implementing Set Based Design into Department of Defense Acquisition.” Master’s capstone project, Naval Postgraduate School.

Chen, Daijun, and Shifeng Xiong. 2017. “Flexible Nested Latin Hypercube Designs for Computer Experiments.” *Journal of Quality and Technology* 49, no. 4 (Oct): 337–353.

Chen, Ray-Bing, Dai-Ni Hsieh, Ying Hung, and Weichung Wang. 2013. “Optimizing Latin Hypercube Designs by Particle Swarm.” *Statistics and Computing* 23, no. 5 (Sept): 663-676.

Cioppa, T.M., and T.W. Lucas. 2007. “Efficient Nearly Orthogonal and Space-filling Latin Hypercubes.” *Technometrics* 49(1): 45–55.

Clowney, Patrick, Jason Dever, and Steven Stuban. 2016. “Department of Defense Acquisition Program Terminations: Analysis of 11 Program Management Factors.” *Defense Acquisition Research Journal* 23, no. 3 (July): 298–328.

Cummings, Mary Ann. 2015. “Identifying and Quantifying Emergent Behavior through System of Systems Modeling and Simulation.” PhD Dissertation, Naval Postgraduate School.

Dalbey, Keith R., and George N. Karystinos. 2010. “Fast Generation of Space-Filling Latin Hypercube Sample Designs.” Paper presented at the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX.

Dam, Evan, B. Husslage, D den Hertog, and H. Melissen. 2006. “Maximin Latin Hypercube Designs in Two Dimensions.” *Operations Research* 55(1): 158–169.

De Geus, Arie P. 1988. “Planning as Learning.” *Harvard Business Review* 66(2): 70–74.

Department of Defense. 2017. “Operation of the Defense Acquisition System.” DoD Instruction DODI 5000.02. Washington, DC: Department of Defense.

Department of Defense. 2019. "High Performance Computing Modernization Program." Accessed February 6, 2019. https://www.hpc.mil/index.php/2013-08-29-16-03-23/computational-research-and-engineering-acquisition-tools-and-environments/create-ships/.

Department of Navy. 2011. *Department of the Navy Implementation and Operation of the Defense Acquisition System and the Joint Capabilities Integration and Development System*. SECNAV Instruction 5000.2E. Washington, DC: Department of Navy.

Doerry, N. 2012. "A Vision for Ship Design and Analysis Tools." *Marine Technology, SNAME* 3, no 1 (July): 8–9.

Doerry, Norbert. 2015. "Measuring Diversity in Set-Based Design." Presented at ASNE Day 2015 (March 4, 2015). Arlington VA.

Duan, Weitao, Bruce Ankenman, Susan M. Sanchez, and Paul J. Sanchez. 2017. "Sliced Full Factorial-Based Latin Hypercube Designs as a Framework for Batch Sequential Design Algorithm." *Technometrics* 59, no 1 (Feb): 11–21.

Eccles, T.J. 2010. "Ship Design and Analysis Tool Goals." COMNAVSEA Memorandum 9000 Ser 05T/015 (September 29, 2010). Washington, DC.

Evans, J. Harvey. 1959. "Basic Design Concepts." *Naval Engineers Journal* 21 (Nov): 671–678. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1559-3584.1959.tb01836.x

Fang, K.T., and D.K.J. Lin. 2003. "Uniform Experimental Designs and their Applications in Industry." *Handbook of Statistics* 22: 131–170.

Fisher, Ronald A. 1971[1935]. *The Design of Experiments*. London: Macmillan Publishing Company.

Forsberg, Kevin, and Harold Mooz. 1991. "The Relationship of System Engineering to the Project Cycle." Paper presented at the INCOSE/ASEM symposium, Chattanooga, TN. https://doi.org/10.1002/j.2334-5837.1991.tb01484.x.

Forsberg, K., H. Mooz, and H. Cotterman. 2005. *Visualizing Project Management, 3rd ed*. New York: John Wiley and Sons.

Frye, Matthew C. 2010. "Applying Set Based Methodology in Submarine Concept Design." Master's thesis, Massachusetts Institute of Technology.

Garner, Matt, Norbert Doerry, Adrian MacKenna, Frank Pearce, Chris Bassler, Shari Hannapel, and Peter McCauley. 2015. "Concept Exploration Methods for the Small Surface Combatant." Paper presented at the World Maritime Technology Conference, Providence, RI. https://www.researchgate.net/publication/305222624_Concept_Exploration_Methods_for_the_Small_Surface_Combatant.

Genta, John. 2016. "Using the Principles of Set-Based Design to Realize Ship Design Process Improvement." Master's thesis, Massachusetts Institute of Technology.

Ghosh, Sourobh, and Warren Seering. 2014. "Set-Based Thinking in the Engineering Design Community and Beyond." DETC2014-35597. In Proceedings of the IDETC/CIE conference, Buffalo, NY.

Grosso, A., A. Jamali, and M. Locatelli. 2009. "Finding Maximin Latin Hypercube Designs by Iterated Local Search Hueristics." *European Journal of Operations Research* 197(2): 541–547.

Grosklags, Paul. (2018). "Systems Engineering Transformation." Presentation at Industry Day, Southern Maryland Higher Education Center, California, MD, March 8, 2018. http://www.paxpartnership.org/docs/SE%20Transformation%20Overview%20%28Industry%20Day%29%20V39%20David%20Cohen.pdf

Hall, Arthur. 1969. "Three-Dimensional Morphology of Systems Engineering." *IEEE Transactions on Systems Science and Cybernetics* 5(2): 156–160.

Hallman, James. 2013. "Squiggles and Diamonds and Sprints - How Interdisciplinary Teams Visualize Design Process." *THINK News*, July 22, 2013. http://www.thinkinc.com/blog/squiggles-and-diamonds-and-sprints-how-interdisciplinary-teams-visualize-design-process/.

Hannapel, Shari E., Nickolas Vlahopoulos, and David J. Singer. 2012. "Including Principles of Set-Based Design in Multidisciplinary Design Optimization." AIAA 2012–5444. In Proceedings of the 12th AIAA ATIO Conference, Indianapolis, IN.

Hart, Laura E. (2015). "Introduction to Model-Based System Engineering (MBSE) and SysML." Presented at the Delaware Valley INCOSE Chapter Meeting, July 30, 2015. https://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf

He, Xu, and Peter Z.G. Qian. 2011. "Nested Orthogonal Array-Based Latin Hypercube Designs." *Biometrika* 98(3): 721–731.

He, Yuanzhen, and Boxin Tang. 2013. "Strong Orthogonal Arrays and Associated Latin Hypercubes for Computer Experiments." *Biometrika* 100(1): 254–260.

Hernandez, A.S. 2008. “Breaking Barriers to Design Dimensions in Nearly Orthogonal Latin Hypercubes.” PhD diss., Naval Postgraduate School.

Hernandez, Alejandro S., Thomas W. Lucas, and Paul J. Sanchez. 2012. “Selecting Random Latin Hypercube Dimensions and Designs through Estimation of Maximum Absolute Pairwise Correlation.” In Proceedings of the 2012 Winter Simulation Conference, Berlin, Germany.

Hernandez, Alejandro S., Thomas W. Lucas, and Matthew Carlyle. 2012. “Constructing Nearly Orthogonal Latin Hypercubes for any Non-saturated Run-Variable Combination.” *ACM Transactions on Modeling and Computer Simulation* 22(4): 1–17.

Holt, Ryamond, and Catherine Barnes. 2010. “Towards an Integrated Approach to “Design for X”: An Agenda for Decision-Based DFX Research.” *Research in Engineering Design* 21: 123–136.

Hung, Y., Y. Amemiya, and C.F.J. Wu. 2010. “Probability-Based Latin Hypercube Designs for Slid-Rectangular Regions.” *Biometrika* 97: 961–968.

Hung, Ying. 2011. “Adaptive Probability-Based Latin Hypercube Designs.” *Journal of the American Statistical Association* 106(493): 213–219.

INCOSE. 2015. *Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities*. INCOSE-TP-2003-002-04-2015. Hoboken: Wiley and Sons, Inc.

ISO/IEC. 2010. “Systems and Software Engineering, Part 1: Guide for Life Cycle Management.” ISO/IEC TR 24748-1:2010, IEEE std 24728-1-2011. International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Geneva, Switzerland.

ISO/IEC/IEEE. 2015. “Systems and Software Engineering - System Life Cycle Processes.” ISO/IEC 15288:2015. International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers, Geneva, Switzerland.

Jin, R.C., W. Chen, and A. Sudjianto. 2005. “An Efficient Algorithm for Constructing Optimal Design of Computer Experiments.” *Journal of Statistical Planning and Inference* 134: 539–548.

Johnson, M.E., L.M. Moore, and D. Ylvisaker. 1990. “Minimax and Maximin Distance Designs.” *Journal of Statistical Planning and Inference* 26: 131–148.

Joseph, V. Roshan, and Ying Hung. 2008. “Orthogonal-Maximin Latin Hypercube Designs.” *Statistica Sinica* 18: 171–186.

Kassel, Ben, Seth Cooper, and Adrian Mackenna. (2010). “Rebuilding the NAVSEA Early Stage Ship Design Environment.” In *Naval Surface Warfare Center Carderock Division*. https://apps.dtic.mil/dtic/tr/fulltext/u2/a522824.pdf

Kennedy, James, and Russel Eberhart. 1995. “Particle Swarm Optimization.” In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia.

Kennedy, Brian M., Durward K. Sobek II, and Michael N. Kennedy. 2014. “Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process.” *Systems Engineering* 17, no. 3 (Nov): 278–296.

Kumar, D. Uday. 2015. “Remote Sensing Platforms and Sensors.” August 5, 2015. https://www.slideshare.net/udaykumardevalla/unit2-51304046/.

Kuo, Tsai-C., Samuel H. Huang, and Hong-C. Zhang. 2001. “Design for Manufacture and Design for ‘X’: Concepts, Applications, and Perspectives.” *Computers and Industrial Engineering* 41: 241–260.

Landau, Peter. (2017). “Top 12 Agile Principles.” ProjectManager. Last modified September 5, 2017. https://www.projectmanager.com/blog/agile-principles

Leary, Stephen, Atul Bhaskar, and Andy Keane. 2003. “Optimal Orthogonal-Array-Based Latin Hypercubes.” *Journal of Applied Statistics* 30(5): 585–598.

Li, W., and C.F.J. Wu. 1997. “Columnwise-Pairwise Algorithms with Application to Supersaturated Designs.” *Technometrics* 39: 171–179.

Liefvendahl, M., and R. Stocki. 2006. “A Study on Algorithms for Optimization of Latin Hypercubes.” *Journal of Statistical Planning and Inference* 136(9): 3231–3247.

Liker, Jeffrey K., Durward K. Sobek II, Allen C. Ward, and John J. Cristiano. 1996. “Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering.” *IEEE Transactions on Engineering Management* 43(2): 165–178.

Loeppky, J.L., L.M. Moore, and B.J. Williams. 2010. “Batch Sequential Designs for Computer Experiments.” *Journal of Statistical Planning and Inference* 140: 1452–1464.

MacCalman, A.D., H. Vieira, and T. Lucas. 2017. “Second-Order Nearly Orthogonal Latin Hypercubes for Exploring Stochastic Simulations.” *Journal of Simulation* 11: 137–150.

McKay, MD, RJ Beckman, and WJ Conover. 1979. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code.” *Technometrics* 21: 239–245.

McKenney, Thomas A., Lauren F. Kemink, and David J. Singer. 2011. "Adapting to Changes in Design Reqirements Using Set-Based Design." *American Society of Naval Engineers* 3: 67–77.

McKenney, Thomas Abbott. 2013. "An Early-Stage Set-Based Design Reduction Decision Support Framework Utilizing Design Space Mapping and a Graph Theoretic Markov Decision Process Formulation." PhD diss., University of Michigan.

McKenney, Thomas, and David Singer. 2014. "Set-Based Design: A Concurrent Engineering Approach with Particular Application to Complex Marine Products." *Marine Technology, SNAME* 3, no. 1 (July): 51–55.

Mebane, Walter L., Craig M. Carlson, Chris Dowd, David J. Singer, and Michael E. Buckley. 2011. "Set-Based Design and the Ship to Shore Connector." *American Society of Naval Engineers* 3: 79–92.

Montgomery, Douglas C. 2001. *Design and Analysis of Experiments*. New York: John Wiley.

Morris, M., and T. Mitchell. 1995. "Exploratory Design for Computer Experiments." *Journal of Statistical Planning and Inference* 43: 381–402.

NAVSEASYSCOM. 2012. "Ship Design Manager (SDM) and Systems Integration Manager (SIM) Manual." Technical Report S9800-AC-MAN-010. Washington, DC.

Norman, D.O., and M.L. Kuras. 2006. "Engineering Complex Systems." In *Complex Engineered Systems – Science Meets Technology*, edited by D. Braha, A. Minai, and Y. Bar-Yam, 206–245. New York: Springer.

Owen, A.B. 1992. "Orthogonal Arrays for Computer Experiments, Integration and Visualization." *Statistica Sinica* 2: 439–452.

Park, Jeong-Soo. 1994. "Optimal Latin Hypercube Designs for Computer Experiments." *Journal of Statistical Planning and Inference* 39: 95–111.

Pozmantir, Serge. 2018. "Calculations." Accessed December 1, 2018. Simple Motors, LLC. https://simplemotor.com/calculations/.

Pugh, S. 1981. "Concept Selection: A Method That Works." In Proceedings for the ICED, M3/16 (March): 497–506. Rome, Italy.

Qian, P. Z. G. 2009. "Nested Latin Hypercube Design." *Biometrika* 96: 957–970.

Qian, Peter Z. G., and C.F. Jeff Wu. 2009. "Sliced Space-Filling Designs." *Biometrika* 96(4): 945–956.

Qian, Peter Z. G. 2012. “Sliced Latin Hypercube Designs.” *Journal of the American Statistical Association* 107(497): 393–399.

Ranjan, Pritam, Derek Bingham, and George Michailidis. 2008. “Sequential Experimental Design for Contour Estimation from Complex Computer Codes.” *Technometrics* 50(4): 527–541.

Ranjan, Pritam, and Neil Spencer. 2014. “Space-filling Latin Hypercube Designs based on Randomization Restrictions in Factorial Experiments.” *Statistics and Probability Letters* 94 (Nov): 239–247.

Royce, Winston W. 1970. “Managing the Development of Large Software Systems.” In Proceeding of the IEEE WESCON (Aug): 1–9. Los Angeles, CA.

Sacks, J., S.B. Schiller, and W.J. Welch. 1989. “Designs for Computer Experiments.” *Technometrics* 31(1): 41–47.

Sacks, J, WJ Welch, TJ Mitchell, and HP Wynn. 1989. “Design and Analysis of Computer Experiments.” *Statistical Science* 4(4): 409–423.

Sage, Andrew P. 1977. “A Case for a Standard SE Methodology.” *IEEE Transactions on Systems, Man, and Cybernetics* 7, no. 7 (July): 499–504.

Sanchez, Susan M., and Wan Hong. 2015. “Work Smarter, Not Harder: A Tutorial on Designing and Conducting Simulation Experiments.” In Proceedings of the 2015 Winter Simulation Conference, Huntington Beach, CA.

Santner, TJ, BJ Williams, and WI Notz. 2003. *The Design and Analysis of Computer Experiments*. New York: Springer-Verlag. http://dx.doi.org/10.1007/978-1-4757-3799-8/.

Sarvaiya, Dushyant. 2016. “What Is the Relation between Engine RPM and Engine Torque?” May 31, 2016. Quora. https://www.quora.com/What-is-the-relation-between-engine-RPM-and-engine-torque/.

Singer, David J., Norbert Doerry, and Michael E. Buckley. 2009. “What is Set-Based Design?” *ASNE Naval Engineers Journal* 121(4): 31–43.

Singer, David, Jason Strickland, Norbert Doerry, Thomas KcKenney, and Cliff Whitcomb. 2017. “Set-Based Design.” T&R Bulletin 7–12. Society of Naval Architects and Marine Engineers (SNAME).

Small, Colin. 2018. “Demonstrating Set-Based Design Techniques - A UAV Case Study.” Master’s thesis, University of Arkansas.

Smith, Robert P. 1997. “The Historical Roots of Concurrent Engineering Fundamentals.” *IEEE Transactions on Engineering Management* 44(1): 67–78.

Sobek II, Durward K., Allen C. Ward, and Jeffrey K. Liker. 1999. “Toyota’s Principles of Set-Based Concurrent Engineering.” *Sloan Management Review* 40, no. 2 (Winter): 67–83.

Specking, Eric A., Cliff Whitcomb, Gregory S. Parnell, Simon R. Goerger, Ed Pohl, and Naga Sai Achyuth Kundeti. 2017. “Trade-off Analytics for Set-Based Design.” Paper presented at the American Society of Naval Engineers (ASNE) Intelligent Ships Symposium, Philadelphia, PA.

Sullivan, P.E. 2008. “Ship Design and Analysis Tool Goals.” COMNAVSEA Memorandum 9000 Ser 05D/047 (Feb. 4, 2008). Washington, DC.

Taguchi, Genichi. 1993. *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*. Translated by Shih-Chung Tsai. New York: ASME Press.

Tang, Boxin. 1993. “Orthogonal Array-Based Latin Hypercubes.” *Journal of American Statistical Association* 88(424): 1392–1397.

Tang, Boxin. 1998. “Selecting Latin Hypercubes Using Correlation Criteria.” *Statistica Sinica* 8, no. 3 (July): 965–977.

Telford, Jacqueline K. 2007. “A Brief Introduction to Design of Experiments.” *John Hopkins APL Technical Digest* 27(3): 224–232.

Velasquez, Mark, and Patrick T. Hester. 2013. “An Analysis of Multi-Criteria Decision Making Methods.” *International Journal of Operations Research* 10(2): 56–66.

Viana, FAC, G. Venter, and V. Balabanov. 2010. “An Algorithm for Fast Generation of Optimal Latin Hypercube Designs.” *International Journal for Numerical Methods in Engineering* 82: 135–156.

Viana, Felipe A. C. 2013. “Things you wanted to know about the Latin hypercube design and were afraid to ask.” Paper presented at the 10th World Congress on Structural and Multidisciplinary Optimization (May 19–24). Orlando, FL.

Viana, Felipe A. C. 2016. “A Tutorial on Latin Hypercube Design of Experiments.” *Quality and Reliability Engineering International* 32: 1975–1985.

Vieira, H. Jr. 2012. “NOB_Mixed_512DP_v2.xls design spreadsheet.” Naval Postgraduate School. Accessed September 14, 2018. http://harvest.nps.edu/.

Vieira, Helcio Jr., Susan M. Sanchez, and Karl Heinz Kienitz. 2011. “Improved Efficient, Nearly Orthogonal, Nearly Balanced Mixed Designs.” In Proceedings of the 2011 Winter Simulation Conference (Dec. 11–14). Phoenix, AZ.

Vieira, H. Jr, S. M. Sanchez, K. H. Kienitz, and M. C. N. Belderrain. 2013. “Efficient, Nearly Orthogonal-and-Balanced, Mixed Designs: An Effective Way to Conduct Trade-off Analyses using Simulation.” *Journal of Simulation* 7: 265–275.

Wade, Z., S. Georger, G.S. Parnell, E. Pohl, and E. 2018 (in review). “Incorporating Resilience in an Integrated Analysis of Alternatives.” *Military Operations Research*, submitted March 26, 2018.

Wang, Xiao-Lei, Yu-Na Zhao, Jian-Feng Yang, and Min-Qian Liu. 2017. “Construction of (Nearly) Orthogonal Sliced Latin Hypercube Designs.” *Statistics and Probability Letters* 125: 174–180.

Ward, A.C. 1989. “A Theory of Quantitative Inference for Artificial Sets, Applied to a Mechanical Design Compiler.” PhD diss., Massachusetts Institute of Technology.

Ward, A.C., and W. Seering. 1989. “Quantitative Inference in a Mechanical Design ‘Compiler.’” *Journal of Mechanical Design* 115, no. 1 (March 1993): 29–35. https://doi.org/10.1115/1.2919320.

Ward, Allen, Durward K. Sobek II, John J. Cristiano, and Jeffrey K. Liker. 1995a. “Toyota, Concurrent Engineering, and Set-Based Design.” In *Engineered in Japan: Japanese Technology Management Practices*, edited by Jeffrey K. Liker, John E. Ettlie, and John C. Campbell, 192–216. New York: Oxford University Press.

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II. 1995b. “The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster.” *Sloan Management Review* 36(3): 43–61.

Watts, J., R. Mackay, D. Horton, A. Hall, B. Douthwaite, R. Chambers, and A. Acosta. 2007. “Institutional Learning and Change: An Introduction.” Paper presented at Institutional Learning and Change Initiative, ILAC working paper 3. Rome, Italy.

Welch, WJ, RJ Buck, J Sacks, HP Wynn, TJ Morris, and MD Mitchell. (1992). “Screening, Predicting, and Computer Experiments.” *Technometrics* 34: 15–25.

Wheelwright, Steven C., and Kim B. Clark. 1994. “Accelerating the Design-Build-Test Cycle for Effective Product Development.” *International Marketing Review* 11(1): 32–46. https://doi.org/10.1108/02651339410057509/.

Whitcomb, Clifford, and Alejandro Hernandez. 2017. “Engineering Reasoning in Set-Based Design.” Working paper, Naval Postgraduate School.

White, B.E. 2016. “A Complex Adaptive Systems Engineering (CASE) Methodology – The Ten-Year Update.” Paper presented at the 9th IEEE Systems Conference (April). Orlando, FL. https://doi.org/10.1109/SYSCON.2016.7490547/.

Ye, Kenny Q. 1998. “Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments.” *Journal of the American Statistical Association* 93(444): 1430–1439.

Ye, Kenny Q., William Li, and Agus Sudjianto. 2000. “Algorithmic Construction of Optimal Symmetric Latin Hypercube Designs.” *Journal of Statistical Planning and Inference* 90: 145–159.

Zhu, H., L. Liu, T. Long, and L. Peng. 2012. “A Novel Algorithm of Maximin Latin Hypercube Design Using Successive Local Enumeration.” *Engineering Optimization* 44(5): 551–564.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California