Structural Complexity and its Implications for Design of Cyber-Physical Systems

by

Kaushik Sinha

B.E., Construction Engineering, Jadavpur University (1997)

M.E., Aerospace Engineering, Indian Institute of Science (2000)

Submitted to the Engineering Systems Division in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

Author. . . . . .
Engineering Systems Division
January 28, 2014

Certified by. . . . . .
Olivier L. de Weck
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Certified by. . . . . .
Franz S. Hover
Associate Professor of Mechanical Engineering
Thesis Committee Member

Certified by. . . . . .
Daniel E. Whitney
Senior Research Scientist, Emeritus, Engineering Systems Division
Thesis Committee Member

Certified by. . . . . .
Raissa M. D'Souza
Associate Professor, University of California, Davis
Thesis Committee Member

Accepted by. . . . . .
Richard C. Larson
Mitsui Professor of Engineering Systems
Chair, Engineering Systems Division Education Committee

This page is intentionally left blank

Structural Complexity and its Implications for Design of Cyber-Physical Systems

by

Kaushik Sinha

Submitted to the Engineering Systems Division on January 3, 2014, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

Most modern era software-enabled, electro-mechanical systems are becoming more complex as we demand more performance and better lifecycle properties (e.g. robustness) from them. As a consequence system development projects are becoming increasingly challenging and are falling behind in terms of schedule and cost performance.

The complexity of technical systems depends on the quantity of different elements and their connectivity, i.e., complexity means a measurable system characteristic. There are three main dimensions of complexity that emerged in the context of system design and development: (1) *Structural Complexity*; (2) *Dynamic Complexity* and (3) *Organizational Complexity*. Structural complexity pertains to the underlying system architecture or more generally, the enabling *infrastructure*. Dynamic complexity refers to the complexity of the system behavior or process running on the underlying infrastructure. Organizational Complexity relates to the system development process and the organizational structure of the development team.

This dissertation primarily focuses on developing a theoretical framework for structural complexity quantification of engineered systems and subsequently a complexity-based design paradigm of interconnected, complex engineered system.

There are four distinct thematic parts in this dissertation: (i) theoretical development of the proposed structural complexity metric, including the metric's qualification as a valid complexity measure based on its mathematical properties; (ii) empirical validation of the proposed complexity metric based on simple experiments and application of the methodology to compute structural complexity of complex engineered systems like jet engines and advanced printing systems; (iii) systemic implications from a complexity management standpoint, including introduction of *complexity budgeting* for system development and linking actual complexity to human perception of complexity through the notion of *complicatedness*, and (iv) extension of the proposed metric to *system-of-systems* and a computational framework for measuring *dynamic complexity.*

The topological complexity metric, C_3 is shown to clearly distinguish between system architectural regimes (e.g., centralized, hierarchical, transitional and distributed). The ball and stick experiment empirically establishes the super-linear relationship between structural complexity (X) and development effort (Y) with exponent, b=1.48.

Thesis Supervisor: Olivier L. de Weck

Professor of Aeronautics and Astronautics and Engineering Systems

Acknowledgements

First, I would like to thank my wife Swatilekha for all her support and endurance under stressful circumstances that we went through as a family and our son Arkajyoti for being the necessary distraction. Without their love, trust and patience, this thesis would have been difficult to complete. My parents, Mr. Rajendra Nath Sinha and Mrs. Dolly Sinha have supported me throughout my life and particularly through the challenging times over the last few years. I owe my thesis to them, and am extremely grateful for the inspiration and encouragement they have always provided. I would like to thank my mother-in-law, Mrs Alaka Maiti and my father-in-law, Mr. Bidyut Kumar Maiti (1942-2012) for their support and patience. I am truly blessed to have them all in my life. Thanks to all our friends and family for their love and best wishes.

I would like to thank Prof. Olivier de Weck, my thesis advisor, for his guidance and support during the course of my graduate studies, without whose help and advice this thesis would not exist. Prof. de Weck inspired me to come back to academia after a number of years in corporate research at Mercedes-Benz Research and Technology Center, India and provided me with a number of opportunities to shape and steer my research and also gain invaluable experience in academic research. I would like to express my gratitude to my thesis committee members, Prof. Franz Hover, Dr. Daniel Whitney and Prof. Raissa D'Souza from UC Davis for their guidance and shaping my research in and outside of the usual committee meetings.

I would like to acknowledge many people who, in one way or another, influenced my research and provided invaluable support through my graduate studies: Dr. Gergana Bounova for introducing me to the area of Network Science; Prof. Eun Suk Suh (while at Xerox Research Center, Webster) for focusing my exploratory studies to specific application areas in complex engineered systems; all my friends and colleagues in Strategic Engineering Research Group (SERG), including Paul Grogan, Denman James (at MIT and later at Pratt and Whitney) and Narek Shougarian for wide-ranging technical and also non-technical discussions.

This research was initiated with the generous financial support from the Xerox-MIT Fellowship Program, later supported by Pratt & Whitney (MIT project number 6921437), the Defense Advanced Research and Projects Agency (DARPA) and Vanderbilt University (VU-DSR #21807-S8).

Table of Contents

List of Figures

List of Symbols

a	Model parameter
b	Model Parameter
α	Component Complexity
β	Interface Complexity
γ	Scaling parameter in the Structural Complexity Metric
δ	Complexity gap
α_i	Complexity of i^{th} component
β_{ij}	Complexity of an interface connecting i^{th} and j^{th} components
$\sigma_i(.)$	i^{th} singular value of a matrix
Δ	Integrative Structural Complexity Component
Ω	Sum of component complexities of multiple, connected systems/subsystems
A	Binary Adjacency matrix
B	Weighted adjacency matrix
A_{ij}	$(i , j)^{th}$ entry of the binary adjacency matrix
$E(A)$	Sum of singular values of binary adjacency matrix, in short, graph energy
C	Structural Complexity Metric
C_1	Sum of component complexities
C_2	Sum of interface complexities
C_3	Topological Complexity Metric
n	Number of system components or the number of nodes in a graph; rate of performance gain
m	Number of system interfaces or the number of edges in a graph; rate of complexity penalty
x	Fractional parameters
y	Fractional parameters
P	Performance measure
R^2	Coefficient of multiple determination
X	Independent variable

Y	Dependent Variable
p	Fractional parameters
r	Fractional parameters
u	Fractional parameters
v	Fractional parameters
w	Fractional parameters
p(.)	Probability density function
H(.)	Information entropy
h(.)	Differential entropy
I(. , .)	Mutual Information

This page is intentionally left blank

Chapter 1

Introduction

Today's large-scale engineered systems are becoming increasingly complex due to numerous reasons including increasing demands on performance, and improved lifecycle properties. As a consequence, large product development projects are becoming increasingly challenging and are falling behind in terms of schedule and cost performance. For example, in 13 aerospace projects reviewed by the US Government Accountability Office (GAO) since 2008, large development cost growth of about 55% was observed. The fundamental tenet of this thesis is that such large development cost overruns can largely be attributed to our current inability to characterize, quantify and manage system complexity [DARPA report, 2011]. With increasing complexity of engineered systems, typically the associated Life Cycle Cost (LCC) also increases [Sheard and Mostashari, 2009].

The complexity of today's highly engineered products and systems is rooted in the interwoven web defined by its components and their interactions. Today's engineered systems are hard to design and maintain since they are *complicated* as well as *complex*. Modern systems are *complicated* in the sense that they typically exceed the bounds of human understanding in a sense that they are so large or detailed that no single individual can understand how they work [Sturtevant 2013]. Only collectively as teams and organizations of individuals are we able to bring these systems into being. By *complex,* we mean that interactions between parts can result in unexpected behavior that is difficult to anticipate and which can threaten safe and reliable operations. One of the defining features of complex systems is that they are often interconnected in ways that enable unanticipated behavior to emerge as a result of unanticipated interactions between system components. Because of

this *emergent behavior,* the whole often does not behave in a manner that would follow directly from the independent functioning of its constituent parts. Keeping complexity under control is paramount as overly complex systems carry a variety of costs and risks. They are more expensive to design, harder to maintain, and can be more prone to unanticipated failure.

System complexity can add value as well in terms of enhanced performance and robustness. Over the last century, increasingly complex machines and infrastructures have provided new capabilities that were previously unimaginable. Examples are the human landings on the Moon between 1962-1972 (Project Apollo) as well as the recent complete decoding of human DNA using sequencing machines. While complexity can be costly, a higher complexity system may very well be worth the price of this additional complexity if the added functionality and improved performance gains outweigh the negatives. A natural tradeoff therefore exists between enabling valuable functionality or performance characteristics while keeping complexity under control.

Complexity across different regions of the same system can vary widely. In the continuous battle to constrain and channel the behavior of a large system so that complexity is appropriately managed, development of appropriate complexity quantification techniques is of primary interest in the context of large, complex system development efforts. In the context of system/product design, the challenge of quantification and efficient management of system complexity is a central theme.

The difficulty is that we require a clear definition for complexity, particularly one that is measurable and objective. By objective, we mean that two observers tasked with quantifying complexity using identical inputs will arrive at an identical – or at least similar conclusion. An appropriate summary of the issue is reflected in the assertion “I can't define it for you, but I know it when I see it.” [Corning 1998]. A root cause in this lack of unified definitions is that there are in fact several types of complexity.

There are different facets of complexity and it is therefore important to specify which one of these is the focus of this thesis. In the context of a complex engineered system design and development effort, we can categorize the associated

complexities into (i) ***internal*** and (ii) ***external***. There are three main dimensions of *internal* complexity that emerge from existing literature in the **context of system design and development** - (1) **Structural Complexity**; (2) **Functional or Dynamic Complexity** and (3) **Organizational Complexity** [Malik F. 2003, Weber C. 2005, Riedl 2000, Lindemann 2009]. These three aspects of complexity are described in fig. 1.1 below with arrows showing their observed correlations [Lindemann 2009].

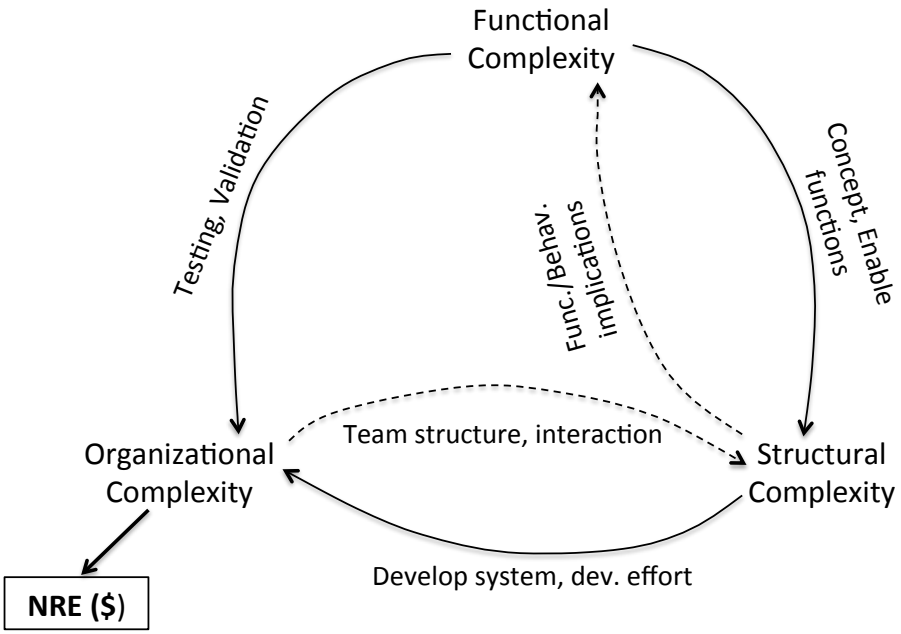


Fig. 1.1: Interplay among three main dimensions of complexity in system/product development context. NRE stands for the Non-Recurrent Engineering Cost/Effort.

Functional or dynamic complexity is driven by the requirements that the system must satisfy and how they are satisfied. The functional or dynamic complexity is driven by what the system does (i.e. its functions as well as the chosen concept). A system is deemed dynamically complex if its external behavior/dynamics is difficult to describe and predict effectively. Since the system behavior is described over an operational envelope of the system, dynamic complexity is an implicit function of this envelope. If changes in the operational envelope impacts system behavior significantly, the dynamic complexity is likely to change significantly as manifestation of changes in system behavior over the current operational envelope. Please note that the system behavior is often bounded by the

underlying system architecture and therefore, dynamic complexity possesses a strong positive correlation with structural complexity. There are two primary sources of dynamic complexity: (i) interactivity among functional attributes and (ii) uncertainties in their interactions.

There is strong interdependence between functional/dynamic and structural complexity (they relate to the interplay between form and function). The functions and behavior of the system are enabled by components. Structural complexity characterizes the system architecture (i.e., the pattern of interactions amongst the functionality enabling components).

A system is structurally complex if it has numerous components whose interaction is difficult to describe or understand. Structural complexity relates to the notion of the architecture of a system, which is a skeleton that connects the components of the system. Therefore, in a sense, structural complexity represents complexity determined by the *form*. Given a set of basic functions, there are multiple forms that can perform that set of functions. An example is the function of slowing food spoilage. Over millennia humans have developed numerous concepts for achieving this particular function – cooling, freezing, irradiating, salting, canning and vacuum-packing the food. Achieving the desired functionality and system behavior with minimal structural complexity becomes an important criterion for down-selection of design/architectural concept.

Designs that require more than $7^3 =\sim 300$ atomic components require teams of designers which can be organized in different ways [de Weck *et al*., 2011]. Organizational Complexity therefore relates to the system development process and the organizational structure of the development team. Organizational structure often mirrors the system's architecture and is thereby closely related to structural complexity. The fig. 1.2 below describes a complexity typology for engineered complex systems from a system design and development perspective.

The longer-term dynamic complexity includes external complexities that stem from market dynamic, stakeholder mechanism, etc. and is not considered hereafter in this thesis. In general, it has been observed that structural complexity

strongly correlates with organizational complexity [Conway 1968; MacCormack et. al, 2011] and also dynamic complexity [Riedl 2000, Lindemann et al., 2009].

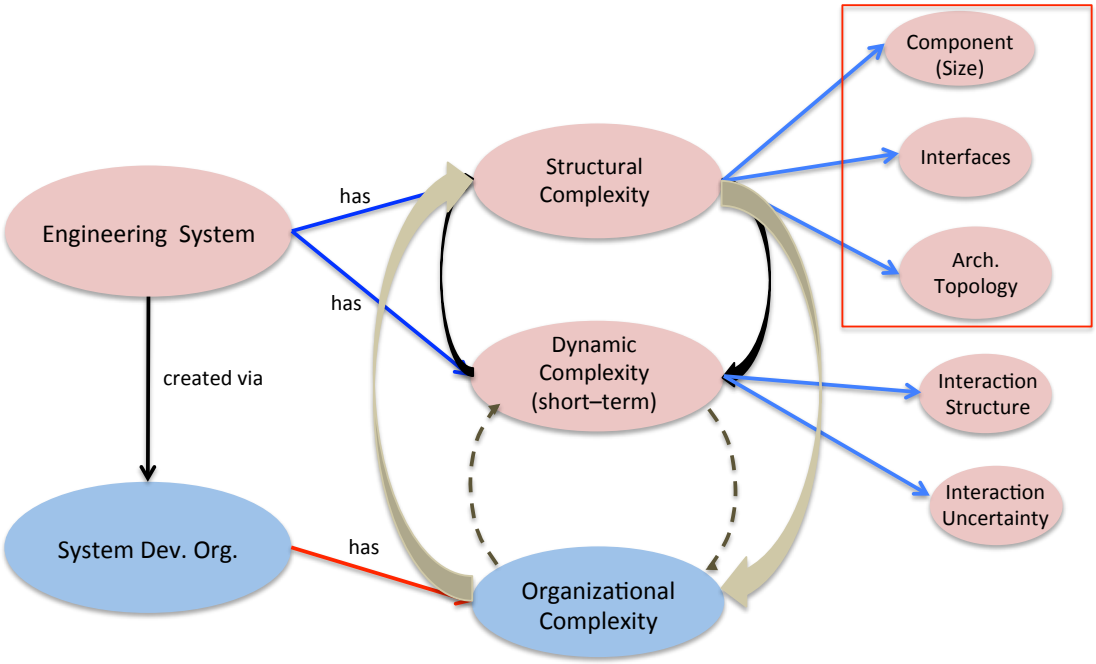


Fig. 1.2: Complexity Typology for Engineered Systems [adapted from Sheard and Mostashari , 2009].

The system functionality is enabled by the underlying system architecture. The system architecture has significant impact on both traditional properties like performance measures that can be observed in the short-term as well as longer-term properties like the "ilities" [de Weck *et al.*, 2012]. The system architecture impacts the complexity of the system during its initial design phase, during implementation and during the changes that will occur in its lifetime.

As we stretch the limits of efficiency and attempt to design more robust systems, it tends to make the architecture more complex [Carlson and Doyle 2002]. While targeting to reduce complexity, ideally towards an unknown level of essential complexity [Crawley, 2007], is important, the addition of new functionality/performance requirements generally drives complexity up [Doyle 2002; Dan Frey et al., 2007]. Increased performance and robustness usually are the upsides of increased complexity in engineered complex systems [Doyle 2002].

The *external* complexities relates to facets that are usually not under control of the system development organization. They typically include complexities associated with funding mechanisms, market dynamics, political and institutional complexities, etc. They are much larger in scope and usually tend to encompass very large-scale projects [Sussman 2000]. In this thesis, we are primarily concerned with the internal complexities that are largely under control of the development team or organization entrusted with development of a particular engineered system.

1.1 Thesis Objectives and Contributions

The primary objective of this research is to develop a *rigorous*, *quantifiable* and *repeatable measure* for *structural complexity* of an engineered complex system. This work strives to bridge the current gap between the theoretical formulation of complexity metrics and their practical applicability to real-world system development. Such a complexity measure shall be computed and traced during the course of system design and development. It can also become a key element of any *complexity-inclusive* system optimization methodology/framework where different system architectures can be traded for system complexity, in addition to performance and lifecycle-driven measures. The proposed complexity metric is expected to apply to a large class of cyber-physical systems.

The primary contributions of this thesis are summarized below and will be discussed in greater detail in Chapter 8.

1. To develop a rigorous and quantitative structural complexity metric for architecture evaluation and optimization, incorporating the fundamental underlying characteristics of system architectures. This lends objectivity to the process of system architecture selection and design.
2. To establish a relationship between the development cost/effort of a system and the underlying structural complexity. Empirical validation of the hypothesis that the cost/effort scales super-linearly with structural complexity by performing a set of simple experiments involving assembly of

molecular structures. Empirical evidence of such behavior is presented using case studies across a realm of real-world engineered systems.

3. Introduce the notion of structural complexity distribution across the system architecture and how this impacts strategic decisions in system development efforts. Extend the structural complexity quantification to *system of systems*, and describe the source of *integrative complexity* that cannot be attributed to the individual systems.
4. Demonstrate extension of structural complexity estimation framework to characterize short-term (governed by physics of the system) dynamic complexity of physical systems.

1.2 Thesis Outline

A compact roadmap of the thesis is provided in fig. 1.3 below. The next chapter introduces the notion of complexity as applied to engineering systems and provides an analysis of the existing literature on this topic. Chapter 3 presents the formulation of the structural complexity metric, its properties and associated methodological framework used throughout the thesis. Chapter 4 illustrates the empirical validation of the proposed structural complexity metric relating system level observables like development cost and defects to structural complexity respectively. Chapter 5 illustrates application of the metric to case studies across a set of real-world engineered systems. Chapter 6 introduces the notion of structural complexity distribution across the system architecture, including its relationship to modularity and applications to System of Systems (SoS) and how this impacts strategic decisions in system development efforts. Chapter 7 presents extensions of the methodology to develop a related dynamic complexity metric, which looks only at the functional and behavioral aspects of the system. Finally, Chapter 8 concludes the thesis with a summary, contributions, and suggestions for future work.

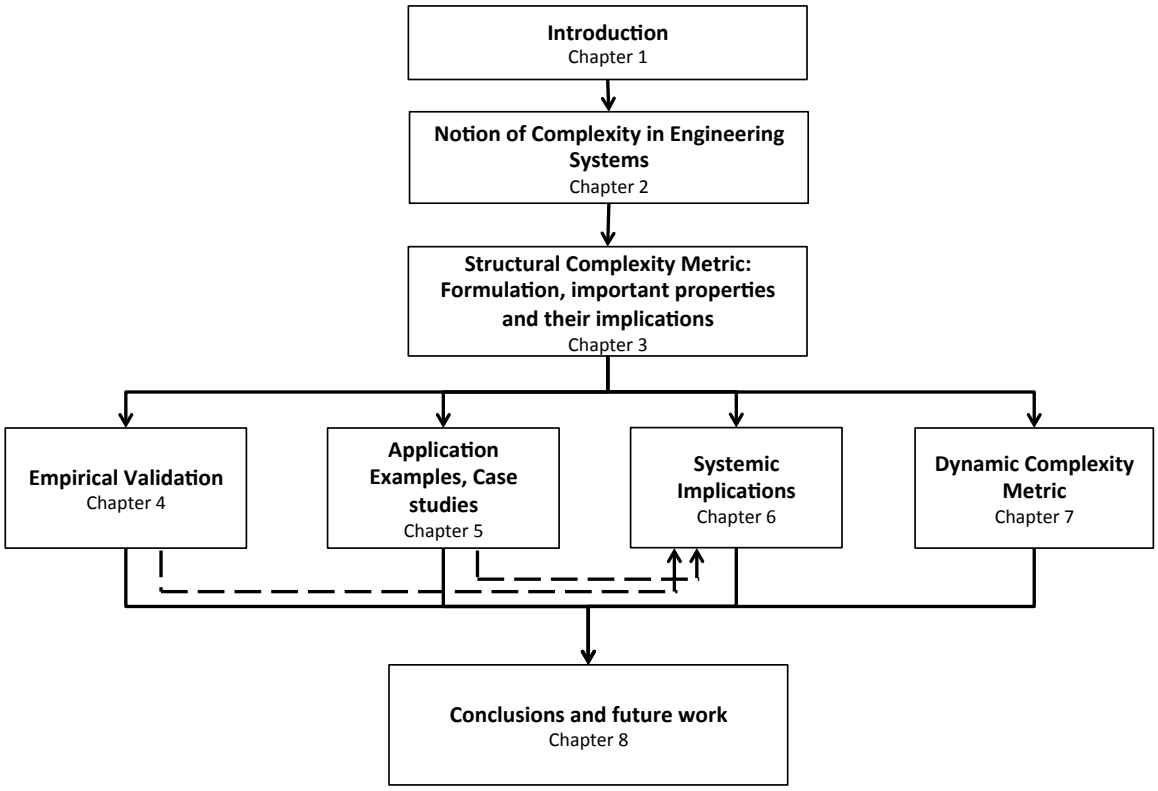


Fig. 1.3: Thesis Roadmap.

References:

Carlson J. M. and Doyle J. C., "Complexity and Robustness," PNAS, 2002.

Sussman, J. M., "Ideas on Complexity in Systems--Twenty Views", M.I.T., 2000.

Corning, P.A., "Complexity is Just a Word!" *Technological Forecasting and Social Change*, 58:1-4, 1998.

Conway, M. E., "How do Committees Invent?", *Datamation* **14** (5): 28–31, 1968.

Dan Sturtevant, "System Design and the Cost of Architectural Complexity", PhD Thesis, MIT, 2013.

http://www.darpa.mil/Our_Work/TTO/Programs/AVM/AVM_Design_Tools_(META).aspx, 2011.

Frey D.D., Palladino J., Sullivan J.P., and Atherton M., "Part Count and Design of Robust Systems," *Systems Engineering (INCOSE)* **10**(3), pp. 203- 221, 2007.

Lindemann, U.; Maurer, M.; Kreimeyer, M. (2005): Intelligent Strategies for Structuring Products. In: Clarkson, J.; Huhtala, M. (Eds.): Engineering Design – Theory and Practice. Cambridge, UK: Engineering Design Centre 2005, pp 106-115.

Lindemann, U, Maurer, M. and Braun, T. "Structural Complexity Management - An Approach for the Field of Product Design" – Springer, 2008.

MacCormack, A. D., Rusnak J, and Baldwin C.Y. "Exploring the Duality between Product and Organizational Architectures: A Test of the Mirroring Hypothesis." Harvard Business School Working Paper, No. 08–039, March 2008. (Revised October 2008, January 2011).

Malik, F., "Strategie des Managements komplexer Systeme". Bern: Haupt 2003.

Riedl, R., "Strukturen der Komplexität – Eine Morphologie des Erkennens und Erklärens", Berlin: Springer 2000.

Sheard S.A. and Mostashari A., "A Complexity Typology for Systems Engineering", Systems Engineering, 2009.

de Weck O., Roos D., Magee C., "Engineering Systems: Meeting Human Needs in a Complex Technological World", MIT Press, 2011.

de Weck O.L., Ross A.M., Rhodes D.H., "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)", 3rd International Engineering Systems Symposium, Delft, 2012.

Crawley E., "Lecture notes for ESD.34 - System Architecture (2007), Massachusetts Institute of Technology," unpublished.

Chapter 2

Complexity in Engineered Systems

Today‘s engineered systems are hard to design and maintain since they are *complicated* and they are *complex*. Modern systems are *complicated* in the sense that they have far exceeded the bounds of a single human’s understanding in a sense that they are so large or detailed that no single individual can understand how they work. The complexity of today’s highly engineered products is rooted in the interwoven web defined by its components and their interactions. Keeping complexity under control is paramount as overly complex systems carry a variety of costs and risks, and are more expensive to design, harder to maintain, and can be prone to unanticipated failure.

The aspect of *cost of complexity* was mentioned by Norman Augustine [Augustine,1997], where he commented – “In the year 2054, the entire defense budget will purchase just one aircraft⋯”. He also predicted that, if the same trend continues, by another 100 years, the entire GNP (i.e., Gross National Product) of the US would buy only one fighter airplane (see fig. 2.1 below). This aspect was dealt in detail in the RAND Corporation report [Arena *et al.*, 2008].

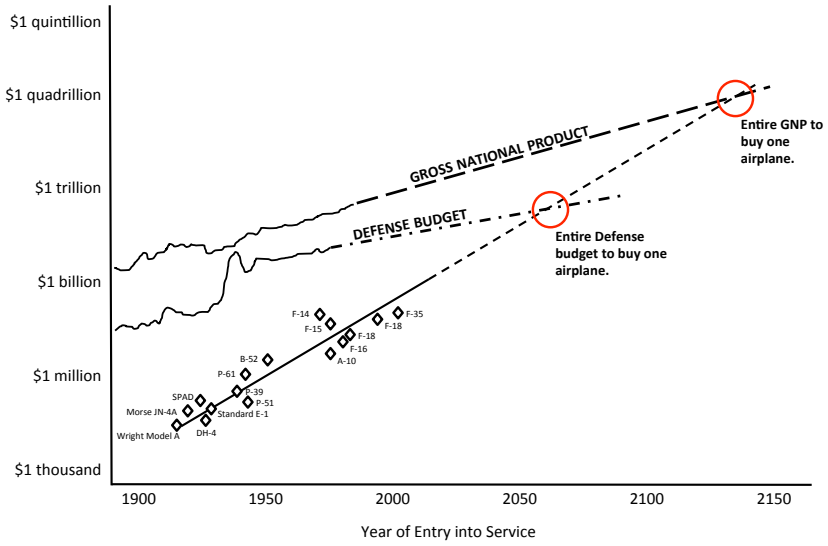


Fig. 2.1: Cost of fighter aircrafts against their year of entry into service. Assuming same cost escalation pattern, the entire defense budget of the US will be spent on buying only one fighter aircraft by 2054 and by 2150, it would require the entire GNP of the US.

This brought forth the impact of *cost of complexity* on the overall price escalation for two fighter aircraft programs and showed the contribution of system complexity to the overall price escalation (see fig. 2.2 below). The *cost of complexity* was estimated by subtracting the sum of costs due to other cost drivers from the total cost.

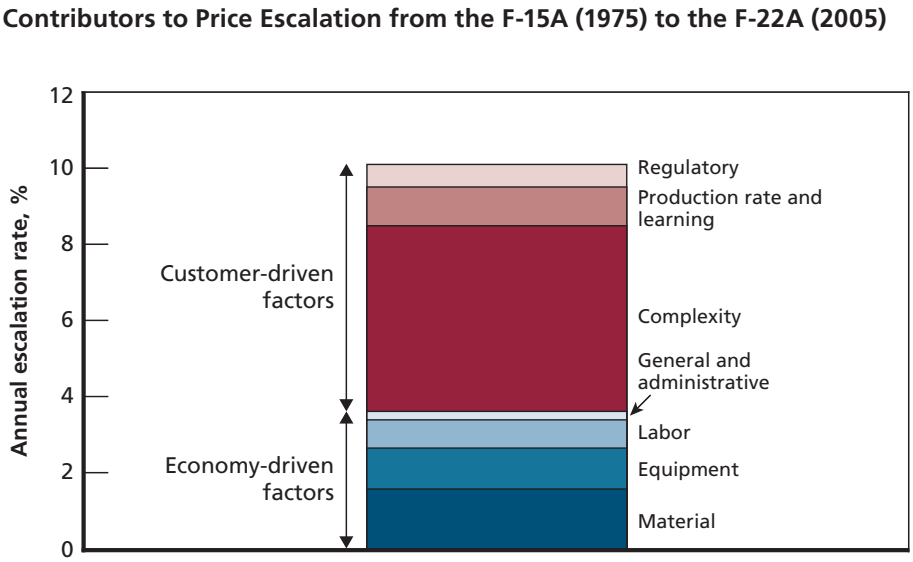


Fig. 2.2: Complexity is a major source for price escalation for fighter aircrafts [RAND Corporation report, Arena *et al.*, 2008].

There are numerous other examples with most chronicled examples coming from NASA and other government-funded programs [Butts and Linton, 2009]. There have been instances of entire programs being cancelled due to cost escalation. Most programs discussed in this report had a substantial enhancement in system's capabilities and also reported to have witnessed significant increase in complexity. There is a hint of strong positive correlation between the functional capabilities and complexity of systems, in line with predictions in the previous chapter. While complexity can be costly, a higher complexity system may very well be worth the price of this additional complexity if the performance and efficiency gains outweigh the negatives. A natural tradeoff therefore exists between enabling valuable functionality or performance characteristics while keeping complexity under control. In order to constrain and actively manage the complexity, development of appropriate complexity quantification technique is of primary interest.

Most modern era software-enabled, electro-mechanical products are becoming more and more complex as we demand more performance and better lifecycle properties (e.g. robustness) from them [Arthur 1993, Frey *et al.*, 2007]. This point was argued for in the explicit case of aircraft engines [Frey *et al.*, 2007] and also in other natural, technological and biological systems [Carlson and Doyle, 2002]. As a consequence system development projects are becoming increasingly challenging and are falling behind in terms of schedule and cost performance [DARPA, 2011]. There is consensus that this is due to our poor understanding and insufficient ability to measure and manage complexity [Arena *et al.*, 2008; DARPA, 2011].

As discussed in the previous chapter, there are the three main dimensions of complexity that emerged in the **context of system design and development** can be grouped as (1) **Structural Complexity**; (2) **Dynamic Complexity** and (3) **Organizational Complexity** [Malik F. 2003, Weber C. 2005, Riedl 2000, Lindemann 2009]. All three dimensions of complexity described above are positively correlated among themselves [Conway 1968; MacCormack et. al, 2011, Riedl 2000, Lindemann

et al., 2009]. This means that as one type of complexity increases, others tend to follow suit.

In this research, we focus on *Structural Complexity*. Structural complexity characterizes the system's architecture. A system is structurally complex if it has numerous components whose interactions are difficult to describe or understand. Structural complexity relates to the notion of the architecture of a system, which is a skeleton connecting the components of the system. Therefore, in a sense, structural complexity represents complexity induced by the *form* or objects that compose the system [Dori, 2002].

2.1 Why measure *Complexity*?

Two things that make today's systems challenging to design and maintain are that they are *complicated* and they are *complex*. By *complicated*, we mean that they are so large or detailed that no single individual can fully understand how they work. By *complex,* we often mean that interactions between parts can result in unexpected behavior that is difficult to anticipate and which can threaten safe and reliable operation [Sturtevant 2013]. This was not always the case. During the time period since the beginning of the Industrial Revolution until the advent of complex systems in the early twentieth century, many individuals running design and manufacturing organizations were capable of fully understanding their processes and products. During this "epoch of great inventions and artifacts" [de Weck *et al.*, 2011], large hierarchically structured organizations grew by taking advantage of differentiated labor and interchangeable parts [Smith, 1985].

The design process, however, remained in the hands of small groups of people. Once a problem was understood, managers coped with the demands of accomplishing a large task by dividing it until each sub-task was small enough for a person or team to handle. Hierarchical control, division of task, and assembly of standard parts led from Adam Smith to Ford's assembly line, and Edison's electrification at the turn of the twentieth century. Then something began to change. Systems such as the telephone network [Fagen et al. 1975] seemed to increasingly

resist reductionist approaches. The process of designing and operating modern machines began to change in fundamental ways [de Weck et al. 2011].

The technical knowledge required to complete a modern system‘s design is much larger than could be learned by a single person over the course of a lifetime. These systems have far exceeded the bounds of human understanding [Crawley 2007, Sussman 1999]. Complicated systems sometimes consist of large numbers of components connected in different ways. Hundreds of engineers make intellectual contributions to the design of these artifacts. As a result, it is no longer only the organization, the product, and the production process that must be decomposed. The *design process* itself must be subdivided and allocated to large groups of people with different skills. Those charged with designing and evolving a complicated system must grope for means of managing the *structure of the design process* (the layout of teams and the communication channels between them) even though everyone involved has only partial visibility [Sturtevant 2013]. It is often now difficult for a group of engineers to really know if a flaw in the decomposition of the design organization will lead them to miss opportunities to create a good *technical structure*, or if the collective "*unknown-unknowns*" will wreak havoc on the end result [Baldwin and Clark 1999].

A defining feature of complex systems is that of *emergent behavior* – the idea that the whole often does not behave in a manner that logically follows from the independent functioning of its parts [Sturtevant 2013]. The need to avoid or control emergent behavior (to prevent defects or disasters) or the desire to find and exploit it requires modern organizations to employ strategies, processes, and structures beyond hierarchical reductionism. Some properties that we require in our complex systems – such as safety – cannot be obtained by assigning responsibility to a single group because they are systemic in nature [Leveson 2004]. Accidents often result from unanticipated interactions *between* parts, not from problems identifiable within individual components [Leveson 2004].

A major goal in complex engineered systems is to manage *structural complexity* so as to keep the *dynamic and emergent* complexity of a system well understood and under control. As we observe, the more complex the system, the

more expensive is the design and implementation effort. Measuring and understanding the complexity of a proposed system's architecture is, however, very important for the whole system development enterprise. There is a multitude of reasons for worrying over and measuring complexity. Many complexity measures in use today are based on anecdotal evidence or intuitive reasoning, due to a lack of detailed, controlled experiments. There are quite natural reasons for this. Large systems development projects are relatively rare and not easily repeatable, making empirical (comparative) studies hard to perform. Consequently, there are no widely agreed upon or standardized complexity measures for engineered systems.

2.2 *Complexity* and *Development Cost*:

According to Rechtin [Rechtin 2002], *the more complex a system, the more difficult it is to design, build and use,* and, intuitively, the more difficult a task, the more expensive it is, if not for any other reason than requiring access to select experts or lots of time to complete all the task involved. Beyond simply requiring a large amount of time for designing and integrating components, complex systems present intricate topologies or patterns that challenge their designers and lead to lower levels or productivity and higher error rates during development as will be shown later in this thesis.

The systems architecting phase usually consumes only a very small amount of the total development budget, but deciding the architecture determines most of the total development cost. According to [Simmons et al. 2005] systems architecting decisions are made after using often less than 1 percent of the total cost and they define up to 80 percent of the total cost. This is because changing a system's architecture later during development is very expensive. Therefore, it is prudent to try to avoid mistakes in systems architecting.

Measuring complexity, and trying to reduce complexity as much as possible while still meeting functional requirements or performance targets, is one way of avoiding mistakes. The same reasoning is found in [Muller 2005], where Muller writes, "*an enabling factor for an optimal result is simplicity of all technical aspects.*

Any unnecessary complexity is a risk for the final result and lowers the overall efficiency". Meyer [Meyer and Lehnerd 1997] argues that: "*Reducing complexity almost always reduces direct and indirect costs. Complexity fuels those costs, which grow geometrically if not exponentially. Every additional part requires that it be made or purchased requiring time, people, and capital. Greater complexity means more purchase orders and more stockroom space"*.

Furthermore, architectural complexity spills over to the organizational design: *"The complexity of architecture will be mirrored in the firm's organizational complexity"*. Liebeck (Boeing) also emphasizes the need to consider the whole product system, not only the product itself - the more complex the product, the more complex the supporting systems and the whole product system.

According to Lankford [Lankford 2003], if a subsystem is significantly more complex that the other parts of the system, it is likely to be more problematic over the entire development and maintenance process. This is especially true if resources and attention are divided irrespectively of the distribution of complexity. Therefore, given a measure of complexity, systems architects and product development managers should strive for a more deliberate or possibly an even distribution of complexity across the system. If this is not possible, they should devote extra resources and attention to the more complex subsystems.

McCabe [McCabe and Butler 1989] state that measuring complexity of a design is essential in being able to predict the cost and time needed to implement the design. They say that before proceeding to implement the design, one must understand the complexity of it. McCabe introduced *cyclomatic complexity*, which is the number of linearly independent paths in a software code, as a measure of software complexity. This metric has been generalized as a network-based complexity metric, which measures the number of linearly independent cycles present in the network. Let us think of a network with *n* nodes and *m* links. The cyclomatic complexity for this simple, connected network is given as (*m-n+1*).

Understanding the complexity of design also gives us a hint, whether the design as such is comprehensible for humans. Empirical studies on software systems referred to in [McCabe and Butler 1989], show a strong positive correlation

between complexity and number of errors found in the implemented system (see fig. 2.3 below).

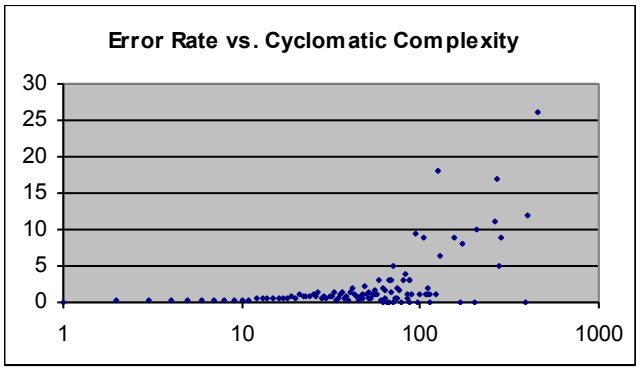


Fig. 2.3: Empirical data on error-rate vs. cyclomatic complexity for software systems [Chapman and Soloman, 2012].

Measuring complexity of problematic subsystems gives an idea whether the problems are inherent in the design or somewhere else (i.e., level of expertise or experience of the development team in charge of a given subsystem). Complexity measurement is also important, when developing new versions of existing products. While trying to minimize complexity for a given level functionality and performance is important, the addition of new functionality usually results in increased complexity. Measuring complexity of an existing product is important and can be done while updating design documentation at the end of product development projects.

If complexity has increased significantly, trying to reduce it during development of the succeeding versions is very important in a continuous development process. A good example in this context is that of the Swiss watchmaker, Swatch. The fig. 2.4 below shows how Swatch reduced the parts count from 91 to 51 and number of assemblies from 55 to 29, eliminating 40 components and 26 assembly operations over a generation of the same product.

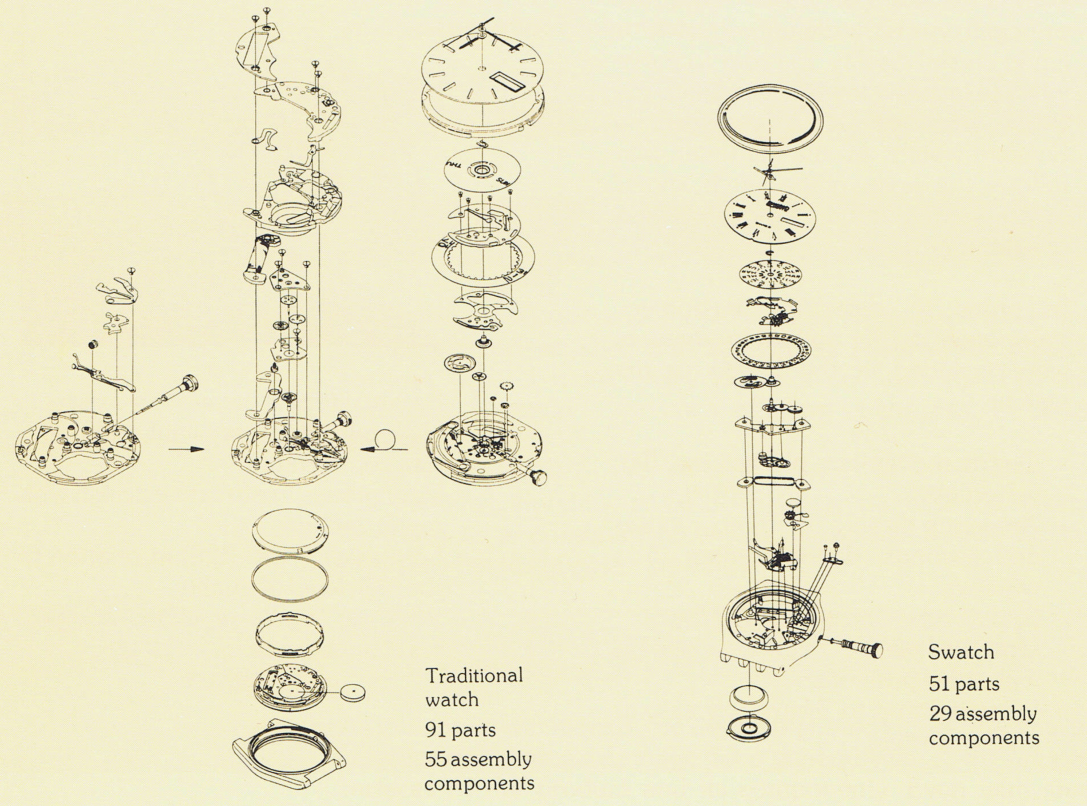


Fig. 2.4: The more traditional design with 91 parts and 55 assemblies vis-à-vis new Swatch design with 51 parts and just 26 assemblies [Andreasen and Ahm, 1988].

The reduction in part count and associated architectural changes led to effective complexity control and reduced production cost. In the new design, components become more multifunctional and the number of interfaces has also reduced. The individual components and the interfaces themselves were more complex on average, but the overall complexity was contained with more centralized architecture. One drawback of the new design was that the watch could no longer be dismantled [Andreasen and Ahm, 1988].

2.3 *Complexity* and *Emergence* in Engineering Systems:

Complexity is blamed for many problems in system development. Measurement of complexity of systems, which could in theory lead to understanding and possibly control of complexity, is problematic in practice. Difficulty arises in

part because “even students and scholars in complexity...use the word ‘complexity’ to describe different ideas and perceptions.” [Suh, 2005].

In other words, complexity is a semantically overloaded term. Sometimes complexity implies a large number of components. More often the problem lies in how those components are interconnected, because interconnections, and the architectures derived from them, can create emergent patterns and/or unintended behaviors. Such patterns and behaviors are collectively termed as *emergent property* of the system and can lead to both, positive and negative consequences.

A good example of *emergent property* can be found in the World Wide Web [Corning, 1998; Fisher, 2006]. World Wide Web is a layered system that provides *services* globally [Fisher, 2006]. Internet Protocol (IP) routing in the Internet is a particularly good example of an emergent global service. No IP router knows the complete topology of interconnections for the Internet or even the configuration of local interconnections in its own neighborhood. Because the configuration of links among routers changes continuously, as does the available bandwidth on a given link, routing tables always correspond to an earlier configuration. And yet, IP routing is a reliable and efficient process that predictably gets messages from their source to their intended destination. Each IP router along the path of a message decides which of its immediate neighbor routers will constitute the next hop without knowledge of routers or likely paths beyond that immediate neighbor. IP routing, like most emergent services, must operate with incomplete, imprecise, and outdated information; yet able to provide efficient and predictable functionality. IP routing implementations do not guarantee optimal paths and in particular, they generate paths whose lengths are strictly less than order n where n is the total number of possible destinations. The Internet must be able to dynamically adapt to accidents, user errors, equipment failures, natural disasters, and attacks by intelligent adversaries. IP routing manages this tradeoff between performance and adaptability in a way that, while adaptable and suboptimal, is always scalable and affordable without risk of local routing errors cascading into system-wide failures.

This contrasts with the electric power grid where issues of local and global performance are often in conflict, leading at times to widespread power outages as chronicled in literature [Buldyrev *et al.*, 2010].

A good example of unintended behavior is often manifested in our transportation systems, on ground or air transportation. Let us consider the national highway system. Under light traffic, minor braking and acceleration among the various vehicles go largely unnoticed. As traffic increases, this braking and acceleration has a traffic wave effect through the entire system. Similar kind of emergent behavior can be seen throughout most other engineered complex systems (or in System-of-Systems), such as air traffic control, aircraft development, automotive system development, and data-centric systems.

Another example is that of autonomous systems operating in the terminal area of an airport. As the dynamic, integrated, and rapidly changing environment of terminal area traffic fluctuates, the systemic interactions can lead to unpredictable results from autonomous systems. This lack of predictability (and non-determinism) causes a lack of trust and difficulty in certifying these systems. A detailed account of emergent characteristics in different engineered complex systems and in system of systems can be found in [Fisher, 2006].

Emergence in the context of system architecture is related to the internal connectivity structure among components and is not just the number of connections between components; it is also about how these connections are organized. It relates to the internal organization or topology of the connectivity structure. Apart from the examples of emergence described before, discontinuities are most dramatically visible among nonlinear emergent properties. A physical example occurs in the stalling of an aircraft. At small angles, the lift and (indirectly) altitude of the aircraft increase with the inclination of the wing. At larger angles, however, turbulence in front of the wing causes abrupt loss of lift [Anderson, 2010].

We can categorize emergence into (i) weak emergence and (ii) strong emergence. Weak emergence describes new properties arising in systems as a result of the interactions at an elemental level. But if, on the other hand, systems can have qualities not directly traceable to the system's individual components, but rather to

how those components interact, and one is willing to accept that a system supervenes on its components, then it is difficult to account for an emergent property's cause. These new qualities are irreducible to the system's constituent parts [Laughlin 2005]. The whole is greater than the sum of its parts. This view of emergence is called strong emergence.

The likelihood of *emergence* is embedded in the system architecture through the connectivity information. It is about the extent of global connectivity information that can be inferred from only the local connectivity information. It is a topological property of the underlying system architecture and in a more general setting, relates to a combination of topology and behavioral aspects of individual components. In case of a centralized architecture or architecture with replicated/symmetric structures we may be able to effectively reduce the system and can still learn a lot about the architecture.

Sometimes the system itself may be well understood, but the problem of developing the system is complex because of the number of teams, contractors or the number of tasks in the development schedule, dependencies among these nodes, or socio-political aspects of the development effort.

Complexity was identified as a major problem in the postwar effort when large aerospace and computer systems required coordination of thousands of engineers [Hughes, 1998]. Whatever the type of complexity (in fact, the type is usually unspecified), the complexity of a system is correlated to: (a) product life cycle costs; (b) difficulty of getting engineering changes implemented (in fact, excessive complexity can generate changes which in turn further increase the complexity of the system); (c) need for sophisticated manufacturing tools and technologies; (d) difficulty in servicing (leading to new failure modes); (e) need for a complex, and therefore costly, design process; etc.

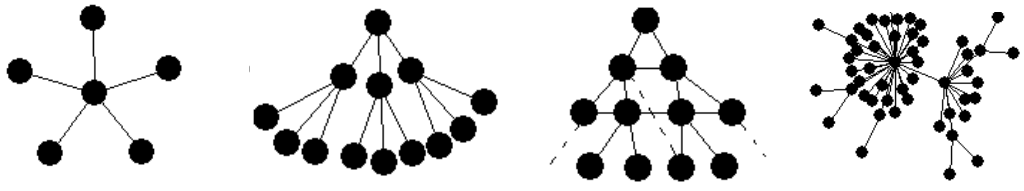
In order to be able to understand or fix these problems, engineers need to understand what complexity is and how to measure it. Thus the need for an objective, quantitative and repeatable measure of structural complexity.

2.4 *Networks* and *Engineering Systems*

System engineers have begun to lean heavily on the language and mathematics of graphs and networks when describing engineering systems and their properties. A monograph about system architecture released by MIT [Whitney *et al*., 2004] employs the term *network* 40 times. This makes sense because many important engineering systems, including the Internet, are really technical networks [Mahadevan et al. 2006].

The monograph on system architecture tells us that, "some architectures can be represented fairly completely as networks. In such cases, a lot can be determined about their behavior from graph theory". After all, if architecture is an "abstract descriptions of entities... and [their] relationships", then a network - defined by its nodes and edges, is a natural corollary. In addition, the authors argue persuasively that many of the properties we care to measure and manage over a system's lifecycle including "robustness, adaptability, flexibility, safety, and scalability... might be measured using network models of a particular architecture."

Methods for complexity quantification should include the *intricateness* (i.e., having many complexly arranged elements that is comprehensible only with painstaking effort) of the structural connectivity patterns observed in modern engineered systems. For example, if the architectures are nearly "tree-like", the impact of internal structure could be nominal but that is certainly not true for modern day, large-scale engineered systems that are becoming architecturally more distributed.



(a) (b) (c) (d)

Fig. 2.5: Structural patterns with increasing topological complexity [Bounova, 2010] – (a) *star* network, (b) tree architecture, (c) hierarchical architecture with cross-links; (d) network with local hubs. Please refer to section 2.8 for details on topological complexity and how it is computed.

We argue that any representative complexity metric should therefore explicitly account for the underlying pattern in the underlying connectivity structure.

Although there is an extensive literature on complexity, so far it is relatively fragmented. Especially the structural complexity measures are often simplistic and only account for the number of components (nodes) and interfaces (edges), but not the complexity contributions due to the pattern of interconnections. As the density of interactions increases there appears to be an additional contribution to structural complexity due to the inherent patterns in the connectivity structure that is only poorly understood.

In particular, the consideration of network structures describing the connectivity information among system components, i.e., their structural complexity, attracts attention in various scientific works because dependency-based system structures affect system characteristics and behavior [Riedl 2000]. Ulrich & Probst define a system as consisting of parts which are linked to each other; the interaction between these parts influences the system's behavior. Ulrich & Probst further mention dynamics as an important system characteristic [Ulrich & Probst 2001].

The term "structure" is directly linked to the definition of a system. The existence of elements and relations between them automatically results in a structure. Typically, the structure is seen as an attribute of the system and in a way describes an inherent order. In the context of product design, a specific focus is product architecture, which describes the dependency structure within components of a product [Ulrich & Eppinger, 1995].

During the conceptualization and design of any large-scale engineered systems, it is very important to adopt an architecture that is as simple as possible, while being robust and efficient. In addition, there are additional constraints like usability, extensibility etc. It is often perceived that as we stretch the limits of efficiency and attempt to design more robust system, we tend to make architectures more complex [Doyle and Carlson, 2002].

2.5 *Complexity* and *Cognition*

The inherent human cognitive ability manifests itself through *perceived complexity*. *Perceived complexity* is an observer-dependent property that characterizes an actor's/observer's ability to unravel, understand and manage the system under consideration.

In contrast, actual complexity is an inherent system property. For example, the complexity of an automobile's automatic transmission or a software application's code may be hidden from a user and it is perceived to be less complex tan it actually is. Conversely a system may be perceived to be more complex than it actually is by a novice observer while an expert observer may perceive that same system to be less complex based on abstractions he or she may have formed in mind over time.

The observer-dependent quantity that was considered in the above case was not really complexity, but perceived complexity. Therefore separating *perceived complexity* from *actual complexity* improves the clarity by which systems can be described, analyzed and certain classes of system observables (e.g., like predicted development cost or the number of rework iterations) be predicted [Tang and Salminen, 2001]. We can think of *perceived complexity* as a conduit through which *complexity* manifests itself at the level of system-level observables like the *system development cost* [Tang and Salminen, 2001]. Therefore, complexity can be a desirable property if it happens to enable functionality, but remains distinct from perceived complexity [Ramasesh and Browning, 2012].

The notion of perceived complexity provides insights to the cognitive aspects of the observer and his/her ability to handle a certain level of complexity. Development of a rigorous measure for complexity helps in studying the interaction of complexity and human cognitive ability in a quantitative fashion.

2.6 Existing Measures of *Structural Complexity*

The system's architecting/engineering communities have developed several approaches for empirically measuring complexity of engineered systems.

The term "structure" is typically understood as the network formed by dependencies between system elements and it represents a basic attribute of each system. Like the terms "system" and "structure", many definitions can be found for the term "structural complexity", and a number of disciplines focus on its different aspects.

Malik defined the quantity of different states a system can assume [Malik 2003], as an important characteristic of complexity. Malik also mentions that complexity originates from interactions between elements and notes the importance of combinatorics for the determination of system complexity. There is a difference between the "complexity of technical systems" and the "complexity of components" [Lindemann 2009].

The complexity of components or objects is characterized by the parameters like quantity of variables, completeness of understanding about the objects that make up the system, etc. The complexity of technical systems depends on the quantity of different elements and their connectivity, i.e., complexity refers to a measurable system characteristic. This internal product architecture can be represented by complex networks, which are graph-theoretic representations of complex systems. The nodes, representing components of the systems, are connected by links if there exists a physical interaction between such components. Physical interactions can be of four primary types including a direct physical connection (typically transmitting forces and/or moments), transfer of energy, transfer of matter and transfer of information. For *complex* engineered systems, the origin of emergence lies in non-local connectivity that a component might have with the rest of the architecture (local neighborhood of a component consists of its direct connections to its immediate neighbors and might extend to second level neighbors). Such understanding vanishes as the underlying architecture becomes more distributed, leading to larger structural complexity.

Before we look at the existing complexity measures in the literature, let us enumerate the general requirements for any sensible complexity metric. Edmonds

[Edmunds 1999] requires that a complexity measure must not depend on the observer. This means that complexity is an objective and measurable quantity that depends only on the system being measured. Primarily aimed at software systems, McCabe [McCabe and Butler 1989] presented an informal set of desired properties of complexity metrics.

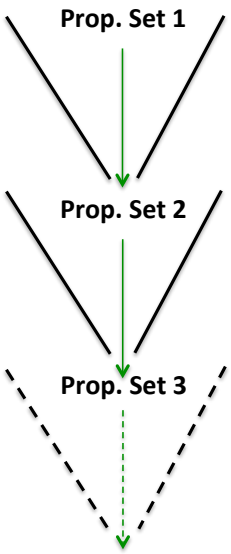
More formally, Weyuker [Weyuker 1988], developed a formal set of nine required properties of complexity metrics that can be used as the set of basic minimal properties that a valid complexity measure must satisfy. We follow Weyuker's criteria and start with additional set of four more informal properties, designated as desirable properties, and then move on to state the set of nine formal properties that are used to establish mathematical validity of complexity measures in a more formal way (see fig. 2.6 below).

In general, a complexity metric should possess the following set of desirable characteristics [Bashir and Thomson, 1999]:

- Objective and mathematically rigorous.
- Correlate intuitively with the difficulty of comprehending, constructing or reconstructing the system.
- Related to the effort required to integrate the system.
- Of operational help in the context of system development.

All these properties are clearly meaningful. A complexity metric should intuitively correlate with the difficulty of comprehending and the ability to construct or reconstruct the design. It is difficult to qualify the extent of what "intuitively" means in this context.

Looking at the diagrammatic representation in fig. 2.6, the property set 1 act as the primary filter that establishes the *construct validity* of a complexity measure. This addresses the first bullet point and the property set 2 addresses the last three bullet points mentioned above. The property set 3 includes any domain specific considerations, if any, and might lead to a modified version of the structural complexity metric.



Property Set 1: theoretical construct (Weyuker's Criteria)

Property Set 2: Practicability perspective – easy to apply; ability to reasonably differentiate between system architectures; applicable to a wide variety of engineered systems; makes intuitive sense

Property Set 3: additional domain-specific filters/ considerations – incorporation of specific models for estimation of different components of complexity

Fig. 2.6: Characteristics of any proposed complexity metric – a filtering diagram.

Weyuker's criteria represent a formal set of necessary properties any valid complexity metric must satisfy. Let *A* and *B* be two different systems and *K(A)* denotes the complexity of system *A*:

1. *There exist A and B such that K(A) differs from K(B),* for a metric, which gives the same value for all systems is useless.
2. *There are only finitely many systems of complexity c,* for the metric should have a range of values. Too narrow range of values is not practical.
3. *There are distinct systems A and B for which K(A) = K(B),* for a metric which gives each system unique value is not useful since such a metric would be a simple bijective mapping of systems.
4. *There are functionally equivalent systems A and B for which K(A) ≠ K(B),* for the structure of the system determines its complexity.
5. *For all A and for all B, K(A) is smaller than K(A∪B), and K(B) is smaller than K(A∪B),* for a system is more complex than the union of its subsystems.

6. *There exists A, B, and O such that, K(A) = K(B) and K(A∪O) ≠ K(B∪O),* for *O* may interact with *A* in different manner than with *B*. Namely, the interfaces between *A* and *O* may be more complex than the interfaces between *B* and *O*.
7. *There are systems A and B such that B is a permutation of components of A and K(A) ≠ K(B),* for changing the way connecting the components to each other, may change the level of complexity.
8. *If A is a renaming of B, then K(A) = K(B),* for complexity does not depend on the naming of the system. This relates to the property of invariance under isomorphic transformation.
9. *There exist A and B such that K(A) + K(B) is smaller than K(A∪B),* for putting systems together creates new interfaces. This pertains to the notion of "a system is greater than the sum of its parts".

In addition to the above *necessary conditions* for any structural complexity metric, there are additional conditions or the *sufficiency conditions* that any proposed complexity metric should satisfy. They include aspects of computability and empirical validation using data from the real-world. The metric should be applicable across a wide swath of systems, for example, say *nano-systems* and even socio technical systems. Please note that the characterization of factors influencing structural complexity could be vastly different across different classes of systems.

In the context of system/product design, there were various indicators used as a complexity measure in practice. The application of complexity metrics was initiated with the use of *cyclomatic complexity* in software system design [McCabe 1976]. This metric can be traced to the first *Betti number* in algebraic topology and graph theory [Munkres, 1993]. This metric measures the number of linearly independent loops/cycles in the system architecture (applied primarily to software systems) and can be measured by counting the number of interfaces and the number of

components. In the context of electro-mechanical systems, the most prevalent indicator has been the number of components in the system [Meyer and Lehnerd, 1997].

Suh [Suh 2005] based his theory of complexity on semantic theory of information. He defines "*complexity as a measure of uncertainty in achieving the specified functional requirements".* The idea is that the greater the information needed to achieve the functional requirement, the greater is the information content (of the functional requirements) and thus the greater the complexity.

Crawley [Crawley, 2007] defines complex systems as "*having many interrelated, interconnected, or interwoven elements and interfaces* and a system which *requires a great deal of information to specify".* He emphasizes that complexity is an absolute and quantifiable system property, which is not dependent on the observer. The observer's cognitive limit of comprehension or ability to deal with complexity defines the upper limit of acceptable/manageable complexity. If system is more complex than the observer can comprehend within a specified period of time, the system will be expensive and error-prone to architect, design and implement. Crawley defines three complexity measures: the number of parts and their types, the number of interconnections and their types and the *sophistication* of the interconnections. Note that any interconnection of higher sophistication means higher interface complexity. Thus, Crawley adds the idea that the quality or nature of interfaces is important. In other word, some interfaces are more complex than others. Simply counting the components and interfaces is not enough. Crawley defines *part* as an element that cannot be taken apart, as the atomic unit of measurement of complexity. A *module* is a collection of elements. Parts are connected with interconnections, which are one of the four types: logical relational, topological, implementation, or operational. Crawley also distinguishes between essential, perceived, and actual complexity.

Essential complexity is the minimum amount of complexity, which is necessary to deliver the required functionality. To best of our knowledge, there is currently no method to a priori determine the essential complexity of a system, given a set of functional requirements or performance targets.

Perceived complexity is the complexity, which the observer perceives when looking at a model of the system at a certain abstraction level.

Actual complexity is the amount of complexity that the system has and it is therefore a system property. The actual complexity is never smaller than the essential complexity and the difference is the *excess* complexity.

Although this framework makes intuitive sense, there exists some vagueness around estimation of the essential complexity and more so with the definition of perceived complexity, which is an observer dependent property.

As shown in fig. 2.7, the perceived complexity, an observer dependent property, is mostly perceived to be less that both the actual and essential complexity. In fact, one of the major reasons for persistent cost overruns on system development programs is that budgets are frequently based on perceived complexity rather than on actual complexity.

This is the sign of over-optimism [Sega *et al.*, 2010] that we seem to be suffering from where we always underestimate the actual complexity in the early/planning stages. The fig. 2.7 below succinctly captures this view of system complexity and shows how the actual complexity increases with the abstraction level during the product development process.

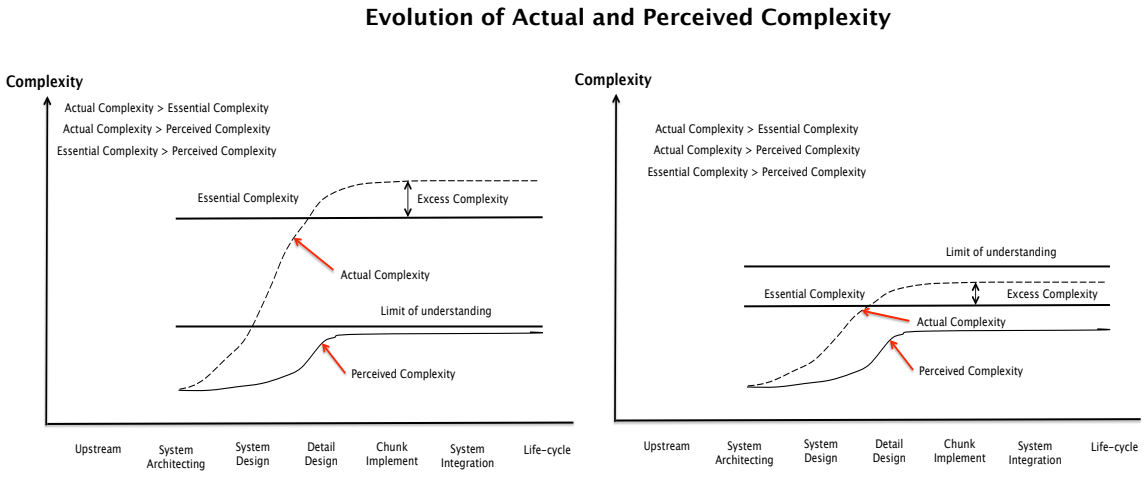


Fig. 2.7: Evolution of complexity during development of a complex system – (a) once actual complexity exceeds the limits of understanding of a single individual, we require a team of individuals whose collective limit of understanding will be larger than the actual complexity of the system; (b) for simpler systems, the actual complexity can still be within the limits of understanding

of an individual. The difference between the actual and perceived complexity is a sign of over-optimism [Sega *et al.*, 2010].

There have been empirically developed measures of complexity based on historical data [Bearden, 2000]. Such measures are usually based on a database of similar systems and are not generalizable. In fact there is an inherent assumption that there are no drastic changes in the class of system architectures across projects considered and that a system whose complexity is to be quantified, falls between the extreme low and high values in the database.

More recently, complexity metrics were developed using number of components and the number of interactions [Jones et al., 2009]. Also there is a relatively recent literature linking different structural aspects of the connectivity structure, mostly pertaining to modularity and importance of individual nodes [Sosa et al., 2003, 2005].

2.7 Critique of Existing *Complexity Metrics*

The complexity metrics developed to date have generated a large amount of debate for a number of reasons. Some have highlighted the limitations of these types of metrics. One limitation of complexity metrics is that we lack a clear explanation for why some complexity metrics correlate with other system observables even when they do. This is especially true for composite metrics. Because many metrics lack an underlying theory, many believe that they are only crude indicators that are often useless due to a lack of robustness or prescriptiveness. Another important limitation of those traditional metrics is that they are local in nature and does not account for the long-range, global properties that system's structure implies. Such metrics fail to capture the complexity and complicatedness created by inter-component coupling patterns. They usually fail to satisfy the set of validation criteria outlined by Weyuker. The table 2.1 shows a shortlist of prominent complexity metrics and their comparison with respect to satisfaction of Weyuker's criteria and their computability (for any medium to large size graphs). The chosen spectrum of complexity metrics covers a variety of structural aspects that they

emphasize as shown in table 2.1 below. Note that the graph energy (introduced later in section 2.8) satisfies Weyuker's criteria while being computable for large graphs (for details on proofs, please refer to chapter 3 of this thesis). This makes it a fundamentally rigorous and practical complexity measure. It can be shown that any affine transformation of this metric also satisfies the Weyuker's criteria. The dependence of computed graph energy on the level of abstraction, used to represent a system, is found to be relatively weak in the print system case study (see chapter 5). The aspect of *complexity gap* due to difference in the level of abstraction for the same system is discussed in chapter 6.

Table 2.1: List of existing complexity metrics showing their computability and whether they satisfy Weyuker's criteria. Graph Energy stands out as both computable and satisfies Weyuker's criteria and establishes itself as a theoretically valid measure (i.e., construct validity) of complexity.

Complexity Measure	Computability	Aspect emphasized	Weyuker's Criteria
Number of components [Bralla, 1986]	✔	Component development (count-based measure)	✗
Number of interactions [Pahl and Beitz, 1996]	✔	Interface development (count-based measure)	✗
Whitney Index [Whitney *et al.*, 1999]	✔	Components and interface developments	✗
Number of loops, and their distribution []	✗	Feedback effects	✗
Nesting depth [Kerimeyer and Lindemann, 2011]	✗	Extent of hierarchy	✗
Graph Planarity [Kortler *et al.*, 2009]	✔	Information transfer efficiency	✗
CoBRA Complexity Index [Bearden, 2000]	✔	Empirical correlation in similar systems	✗
Automorphism-based Entropic Measures [Dehmer *et al.*, 2009]	✗	Heterogeneity of network structure, graph reconfigurability	✔
Matrix Energy / Graph Energy	✔	Graph Reconstructability	✔

A particular concern with the work done in the area of complexity estimation is that less than one-fifth of the studies even attempted to provide some degree of objective quantification of complexity [Tang and Salminen 2001]. The aspect of empirical validation will be handled in chapter 4 and 5 that includes simple

experiments and application of the framework to compute structural complexity of real-world, large engineered complex systems.

The next section provides an overview of the structural complexity metric. An objective and quantifiable measure of structural complexity is imperative for systematic search and optimization of system architecture.

2.8 Overview of the proposed *Structural Complexity Metric*

As previously stated, there are three primary sources of structural complexity (i.e., complexity of the physical embodiment of the system). They arise from (i) the complexities of individual components alone; (ii) the complexities of each pair-wise interaction; and (iii) effect of architecture or the arrangement of the interfaces. Given the same number of interfaces, they can be arranged in a variety of patterns. The number of interfaces alone does not dictate how they should be arranged among themselves, given there are no additional system constraints.

The next step is about defining a functional form for the proposed structural complexity metric, using the three primary sources of complexity mentioned above.

We use a functional form motivated by the relationship expressing the π electron energy of organic molecular system. The energy can be expressed using the Hamiltonian of the molecular system where the sources of energy can be shown to come from (i) *self-energy* of the individual atoms in isolation; (ii) *interaction energy* between neighboring atoms; and (iii) effect of the molecular structure (i.e., how atoms are connected amongst themselves). We argue that any engineered system can be represented by a number of components that are connected amongst themselves in varying ways where each system component can be thought of as an atom and with the interfaces between system components as inter-atomic interactions. The molecular structure is replaced by the connectivity structure amongst the system components.

The time-independent Schrodinger equation is a second-order partial differential equation of the form:

$$H\psi = \varepsilon\psi \qquad (2.1)$$

where ψ is the wave function of the system considered, H is the Hamiltonian operator of the system considered, and ε is the energy of the system considered. When applied to a particular molecule, the Schrodinger equation enables one to establish their energies by solving eq. 2.1, which is an eigen-system problem of the Hamiltonian operator, and also describe the behavior of the electrons.

According to the Huckel's molecular orbital theory (HMO theory), wave functions of a conjugated hydrocarbon with *n* carbon atoms are expanded in an n-dimensional space of orthogonal basis functions, whereas the Hamiltonian matrix is a square matrix of order *n*, defined as:

$$[\mathbf{H}]_{ij} = \begin{cases} \alpha & \text{if } i = j \\ \beta & \text{if the atoms } i \text{ and } j \text{ are chemically bonded} \\ 0 & \text{if there is no chemical bond between the atoms } i \text{ and } j. \end{cases}$$

According to the HMO model, one needs to solve the eigen system problem of an approximate Hamiltonian matrix of the form:

$$H = \alpha I_n + \beta A(G) \qquad (2.2)$$

where α and β are certain constants, $\mathbf{I}_n$ is the identity matrix of order *n*, and **A**(G) is the adjacency matrix of the graph G on *n* vertices corresponding to the carbon–atom skeleton of the underlying molecule. The absolute energy levels ε_i of the π electrons are related to the singular values σ_i of the graph *G* (determined by the singular values of the binary adjacency matrix, A) by the simple relation:

$$\left|\varepsilon_i\right| = \alpha + \beta\sigma_i \qquad (2.3)$$

Using the HMO approximation, the form of total energy of all π electrons can be expressed as:

$$\varepsilon_\pi = \sum_{i=1}^{n} h_i \left|\varepsilon_i\right| \qquad (2.4)$$

where h_i acts as weights associated to each energy level and is constrained by the following relation amongst the weights:

$$h_1 + h_2 + \ldots + h_n = n \qquad (2.5)$$

since the number of π electrons in the molecules is *n*.

The total π electron energy is bounded from above (see appendix A for derivation):

$$\varepsilon_\pi \le n\alpha + \beta \underbrace{\left(\sum_{i=1}^{n} h_i \right)}_{n} \underbrace{\left(\sum_{i=1}^{n} \sigma_i \right)}_{E(A)} \qquad (2.6)$$

where the sum of singular values of the binary adjacency matrix, $E(A) = \sum_{i=1}^{n} \sigma_i$ is defined as the *graph energy* or the *matrix energy* or the *nuclear norm*. We can write inequality 2.6 as,

$$\varepsilon_\pi \le n\alpha + n^2 \beta \left(\frac{E(A)}{n} \right) \qquad (2.7)$$

Looking at the RHS of inequality 2.7, the first term, is the sum of self-energy associated to each atom. The $n^2\beta$ term can be thought of as the upper bound of the sum of interaction energy, where β is a representative interaction energy between atoms. Notice that the number of entries or the non-zero entries in the atomic interaction part of the Hamiltonian matrix scales as n^2 where *n* is the number carbon atoms in the molecule.

The remaining part of expression 2.7 is related to the arrangement of the interactions, that is, the topology of interactions as manifested by the adjacency matrix, *A*. This term differentiates between connectivity patterns even if other parts of expression 2.7 remain identical.

We introduce a notion of *configuration energy*, Ξ expressed as,

$$\Xi := n\hat{\alpha} + m\hat{\beta}\left(\frac{E(A)}{n}\right) \qquad (2.8)$$

where *m* is the number of pair-wise interfaces in the system.

The above *configuration energy* expresses the innate ability of the interacting system to respond to the environment and a higher value indicates increasing difficulty to manage the system. The notional configuration energy described above can be thought of as the complexity associated to the system structure.

We term this quantity as the Structural Complexity of the system where $\alpha's$ stand for component complexity while $\beta's$ stand for interface complexity. Assuming $\hat{\alpha}$ to be the average component complexity and $\hat{\beta}$ to be the average interface complexity, we can express the structural complexity, C as,

$$C = n\hat{\alpha} + m\hat{\beta}\left(\frac{E(A)}{n}\right) \qquad (2.9)$$

where *m* is the number of pair-wise interfaces in the system.

We propose the generic form of the Structural Complexity matric as,

$$\begin{aligned} C &= C_1 + C_2 C_3 \\ &= \sum_{i=1}^{n} \alpha_i + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij} A_{ij}\right)\left(\frac{E(A)}{n}\right) \end{aligned} \qquad (2.10)$$

where the first term C_1 represents the sum of complexities, α_i of individual components alone, the second term has two factors: (i) the sum of all pair-wise interaction complexities, C_2 (local effect) with β_{ij} representing individual pair-wise interaction complexity and (ii) effect of architecture or the arrangement of the interfaces C_3 (global effect).

The above expression can be further generalized as,

$$C = \sum_{i=1}^{n} \alpha_i + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij} A_{ij}\right)\gamma E(A) \qquad (2.11)$$

where we have used $\gamma = \frac{1}{n}$ as the scaling factor in eq. 2.10. It is possible to modify the structural complexity metric by using different values of parameter, γ. Such modification can be used to control the impact of the connectivity structure on the overall structural complexity metric.

In a nutshell, we use the functional form of the total π electron energy expression for the conjugated hydrocarbons, based on the system Hamiltonian. We can think about the *complexity equivalents* of energy in the structural complexity expression. The component complexity, α_i is the complexity equivalent of the self-energy of each atom in isolation. Similarly, complexity equivalent of the interaction energy is the interface complexity, β_{ij}. The effect of the connectivity or the network structure among system components acts as a scaling factor and is captured as the sum of singular values of the binary adjacency matrix (see fig. 2.8 below), defined here as the *graph energy,* $E(A)$.

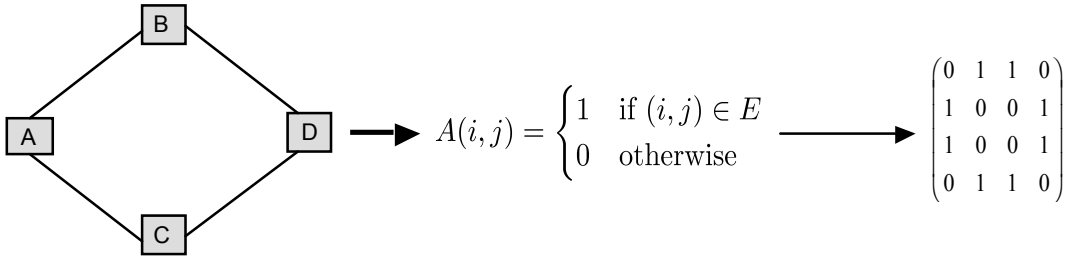


Fig. 2.8: Connectivity structure of a simple system as a simple graph and its associated adjacency matrix.

The *graph energy* captures the impact of topological differences in the connectivity structure and acts as a measure of topological complexity. Theoretical properties of the topological complexity measure and their implications are described in rest of the chapters in this thesis. It will be shown as a valid construct for measuring complexity in chapter 3.

Later in chapter 7, we use the same functional form to compute *dynamic complexity*. Dynamic complexity refers to the complexity of the dynamical behavior of the system and is a form of complexity that stems from the underlying physics

and interrelationships that govern the performance of the system. We can view the system behavior as a physical process and dynamic complexity is the complexity of that physical process. While structural complexity is a measure of complexity of the system architecture, the dynamic complexity develops due to uncertainties in the system during its operation and the direct dependency structure amongst the system responses. In this case, the interaction structure among the associated behavioral responses of the system is represented using a network of interactions among the behavioral responses of a system. In terms of complexity equivalents of energy, the uncertainty in the dynamic behavior of an individual system response in isolation over an *operational envelope* of the system maps to the self-energy of each atom. The interaction uncertainty between a pair of system responses over the operational envelope maps to the interface energy term while the effect of network structure among system responses is captured by the graph energy of the binary adjacency matrix of the system response network, similar to that shown in fig. 2.8, where the nodes are the system responses.

References:

Baldwin, C.Y and Clark, K.B, "Design Rules", MIT Press, 2000.

Browning, T. (2001),"Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions", IEEE Transactions on Engineering Management 48 (2001) 3, pp 292-306.

Jones, R., Hardin, P., and Irvine, A., "Simple Parametric Model for Estimating Development (RDT&E) Cost", ISPA/SCEA Joint Conference, 2009.

Lindemann, U.; Maurer, M.; Kreimeyer, M. (2005): Intelligent Strategies for Structuring Products. In: Clarkson, J.; Huhtala, M. (Eds.): Engineering Design – Theory and Practice. Cambridge, UK: Engineering Design Centre 2005, pp 106-115.

Lindemann, U, Maurer, M. and Braun, T. "Structural Complexity Management - An Approach for the Field of Product Design", Springer, 2008.

McCabe T.J., "A Complexity Measure", IEEE Transactions on Software Engineering Vol. 2, No. 4, pp. 308 - 320, 1976.

Malik, F. (2003): Strategie des Managements komplexer Systeme. Bern: Haupt 2003.

Matti J Kinnunen, "Complexity Measures for System Architecture Models", 2006 (MIT Thesis).

Riedl, R. (2000): Strukturen der Komplexität – Eine Morphologie des Erkennens und Erklärens. Berlin: Springer 2000.

Sosa, M., "A Network Approach to Define Component Modularity", Proceedings of the 7th International Dependency Structure Matrix (DSM) Conference, Seattle. Seattle, USA: Boeing 2005.

Sosa, M. E.; Eppinger, S. D.; Rowles, C. M., "Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions", Journal of Mechanical Design 125 (2003) 2, pp 240-252.

Ulrich, K.; Eppinger, S., "Product Design and Development", New York: McGraw-Hill, 1995.

Ulrich, H.; Probst, G., "Anleitung zum ganzheitlichen Denken und Handeln – Ein Brevier für Führungskräfte", Bern: Paul Haupt, 2001.

Weber, C., "What Is "Complexity"? ", Proceedings of the 15th International Conference on Engineering Design (ICED 05), Melbourne. Melbourne: Institution of Engineers 2005.

Weyuker, E., "Evaluating software complexity measures", IEEE Transactions on Software Engineering, 1988, 14 (9), 1357-1365.

Whitney D. E., "Physical limits to modularity", Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division, 2004.

Yassine, A.; Falkenburg, D.; Chelst, K., "Engineering Design Management: An Information Structure Approach", International Journal of Production Research 37 (1999), pp 2957– 2975.

Rechtin, E and Maier, M.W.,"The art of systems architecting", CRC, 2002.

Simmons, W.; Koo, B.; and Crawley, E.,"Architecture generation for moon-mars exploraton using an executable meta-language", volume AIAA-2005-6726, American Institute of Aeronautics and Astronautics, 2005.

Muller, G., "System Architecting", ESI, 2005.

Meyer, M. and Lehnerd, A., "The Power of Product Platforms", Free Press, 1997.

Lankford, J., "Measuring system and software architecture complexity", Aerospace Conference, Proceedings, March 2003.

Suh, N. P., "Complexity: theory and applications", Oxford University Press, 2005.

Hughes, T. P., "Rescuing Prometheus", Pantheon Books, 1998.

McCabe T.J. and Butler C. W., "Design complexity measurement and testing", Communications of the ACM, 32(12): pp 1415-1425, 1989.

Carlson J. M. and Doyle J. C., "Complexity and Robustness," PNAS, 2002.

Sussman, J. M., "Ideas on Complexity in Systems--Twenty Views", M.I.T., 2000.

de Weck O.L., Roos D., and Magee C.L., "*Engineering systems: Meeting human needs in a complex technological world*", The MIT Press, 2011.

Smith M.R., "*Military enterprise and technological change: Perspectives on the American experience*", The MIT Press, 1985.

Fagen M.D., Joel A.E., and Schindler G., "*A history of engineering and science in the Bell system*", New York: Bell Telephone Laboratories, 1975.

Crawley E., "Lecture notes for ESD.34 - System Architecture (2007), Massachusetts Institute of Technology," unpublished.

Baldwin C.Y. and Clark K.B., "*Design rules: The power of modularity, Volume 1",* The MIT Press, 1999.

Sussman J.M., "The new transportation faculty: The evolution to engineering systems," 1999.

Leveson N., "A new accident model for engineering safer systems," *Safety Science,* vol. 42, pp. 237- 270, 2004.

Whitney D., Crawley E., de Weck O.L., Eppinger S., Magee C.L., Moses J., Seering W., Schindall J., and Wallace D., "The influence of architecture in engineering systems", *Engineering Systems Monograph,* 2004.

Mahadevan P., Krioukov D., Fomenkov M., Dimitropoulos X., and Vahdat A., "The Internet AS-level topology: Three data sources and one definitive metric," *ACM SIGCOMM Computer Communication Review,* vol. 36, pp. 17-26, 2006.

Dan Sturtevant, "System Design and the Cost of Architectural Complexity", PhD Thesis, MIT, 2013.

Corning, P.A., "Complexity is Just a Word!" *Technological Forecasting and Social Change*, 58:1-4, 1998.

Conway, M. E., "How do Committees Invent?", *Datamation* **14** (5): 28–31, 1968.

MacCormack, A. D., Rusnak J, and Baldwin C.Y. "Exploring the Duality between Product and Organizational Architectures: A Test of the Mirroring Hypothesis." Harvard Business School Working Paper, No. 08–039, March 2008. (Revised October 2008, January 2011).

Tang, V. and Salminen, V., "Towards a Theory of Complicatedness: Framework for Complex Systems Analysis and Design", 13th International Conference on Engineering Design, Glasgow, Scotland, August 2001.

Bashir, H.A. and Thomson, V., "Estimating design complexity", Journal of Engineering Design Vol. 10 No 3 (1999), pp 247–257.

Hirschi N.W, and Frey D.D., "Cognition and complexity: An experiment on the effect of coupling in parameter design", Research in Engineering Design 13, 2002, pp 123 – 131.

Frey D., Sullivan J., Palladino J., Atherton M., "Part Count and Design of Robust Systems", Systems Engineering (INCOSE) 10(3):203-221, 2007.

Arthur B. W., "Why Do Things Become More Complex?," *Scientific American*, May edition, 1993.

Ramasesh R.V. and Browning T.R., "Toward a theory of unknown unknowns in project management", preprint, 2012.

Bearden D.A., "A complexity-based risk assessment of low-cost planetary missions: when is a mission too fast and too cheap? ", 4th IAA international conference on low-cost planetary missions, May 2-5, 2000.

Dori D.,"Object-Process Methodology: A Holistic Systems Paradigm", Springer, ISBN-10: 3540654712 2002.

Munkres, J. R.,"Elements of Algebraic Topology", Perseus Books, 1993.

Chapman M. and Solomon D.,"The Relationship of Cyclomatic Complexity, Essential Complexity and Error Rates", http://www.docstoc.com/docs/6802632/The-Relationship-of-Cyclomatic-Complexity-Essential-Complexity, 2012.

Arena M.V., Younossi O., Brancato K., Blickstein I., Grammich C.A., "A Macroscopic Examination of the Trends in U.S. Military Aircraft Costs over the Past Several Decades", RAND Corporation Report, http://www.rand.org/content/dam/rand/pubs/monographs/2008/RAND_MG696.pdf , 2008.

Augustine, N. R., "Augustine's Laws", 6th edition, AIAA Press, 1997.

Butts G., Linton K., "The Joint Confidence Level Paradox: History of Denial", NASA Cost Symposium, April 28, 2009.

Andreasen M.M and Ahm T., "Flexible Assembly Systems", Springer-Verlag, ISBN-10: 0387502467, pp 192, 1988.

Fisher D.A, "An Emergent Perspective on Interoperation in Systems of Systems", Technical Report, CMU/SEI-2006-TR-003 ESC-TR-2006-003, 2006.

Buldyrev S.V, Parshani R., Paul G., Stanley H.E., Havlin S., "Catastrophic cascade of failures in interdependent networks", Nature 464, pp 1025-1028, 2010.

Anderson J.D., "Fundamentals of Aerodynamics", 5th edition, McGraw-Hill, ISBN-10: 0073398101, 2010.

Kreimeyer M., Lindemann, U. "Complexity Metrics in Engineering Design: Managing the Structure of Design Processes", Springer-Verlag, ISBN-10: 3642209629, 2011.

Kortler S., Kreimeyer M. and Lindemann U., "A Planarity-based Complexity Metric", 17th International Conference on Engineering Design, ICED'09, 2009.

Bralla J.G., "*Handbook of Product Design for Manufacturing",* NewYork: McGraw-Hill, 1986.

Pahl G., Beitz W., "Engineering design: a systematic approach", Springer, New York, 1996.

Whitney D. E., Dong Q., Judson J., Mascoli G., "Introducing Knowledge-Based Engineering into an Interconnected Product Development Process", DETC99/DTM-8741, ASME iDETC, 1999.

Dehmer M., Barbarini N., Varmusa K., Graber A., "A Large Scale Analysis of Information-Theoretic Network Complexity Measures Using Chemical Structures", PLoS ONE 4(12), 2009.

Chapter 3

Structural Complexity Quantification

The complexity of today's highly engineered products is rooted in the interwoven architecture defined by their components and their interactions. As discussed in chapter 2, this can be modeled by graphs and associated adjacency matrix. To evaluate a complex system and to compare it to other systems, numerical assessment of its structural complexity is essential.

At the end of chapter 2, we presented an overview of the structural complexity metric and introduced the functional form of the proposed metric.

In this chapter, we dive deeper and explore the proposed measure for structural complexity, discuss its constituents and their important properties, perform mathematical verification for the proposed metric and demonstrate its applications.

As discussed in chapter 2, we adopt the following functional form for estimating the structural complexity of an engineered complex system:

$$\textit{Structural Complexity, } C = C_1 + C_2 C_3 \qquad (3.1)$$

The first term C_1 represents the sum of complexities of individual components alone (local effect) and does not involve architectural information. The second term has two factors: (i) number and complexity of each pair-wise interaction, C_2 (local effect) and (ii) effect of architecture or the arrangement of the interfaces C_3 (global effect). Now, given the same number of interfaces these can be arranged in a variety of patterns and the number of interfaces alone may not dictate how they should be

arranged among themselves, given there are no additional system constraints. Hence in this sense, we assume that C_2 and C_3 are mutually independent, therefore the multiplicative model (see appendix A for details). This can be shown to be a conservative assumption (see appendix B for detailed analysis).

The effect of system architecture captured in C_3 represents a global, *system level effect* that is not visible at the local level (i.e., component engineering or pair-wise interface management) and cannot be obtained from 'local' information alone. This quantity is not amenable to simple addition with its impact realized during system integration. Also, C_3 is likely to have profound impact on various aspects of performance and other life-cycle properties. Similar functional forms are found in quantum mechanical analysis of molecular systems (see appendix A for details) where the system Hamiltonian is the matrix of importance [Gutman *et al.* 1998, Sinha and de Weck 2012, 2013].

3.1 Functional form of the Structural Complexity Metric

As discussed in chapter2, the structural complexity metric is defined using the following analytical form,

$$C = C_1 + C_2 C_3 = \sum_{i=1}^{n} \alpha_i + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{i,j} A_{i,j} \right) \gamma E(A)$$

$$= \underbrace{\sum_{i=1}^{n} \alpha_i}_{C_1} + \underbrace{\left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{i,j} A_{i,j} \right)}_{C_2} \underbrace{\frac{E(A)}{n}}_{C_3}, \quad using\ \gamma = \frac{1}{n} \qquad (3.2)$$

The component complexities, α_i are attached to individual compositional elements of the system and therefore, local to each particular element alone. The second term involves complexities due to pair-wise interfaces, β_{ij} while the third term, C_3 reflects the effects of the underlying connectivity structure. This term is defined as topological complexity, which generally scales with the challenges of system integration. Higher topological complexity will likely lengthen system

integration efforts significantly and it is a global property that is not visible locally (i.e., during component or interface development).

The detailed reasoning for using the above functional form and the scaling factor γ can be found the appendices A and B. The implication of different terms of the structural complexity measure is described in the fig 3.1 below.

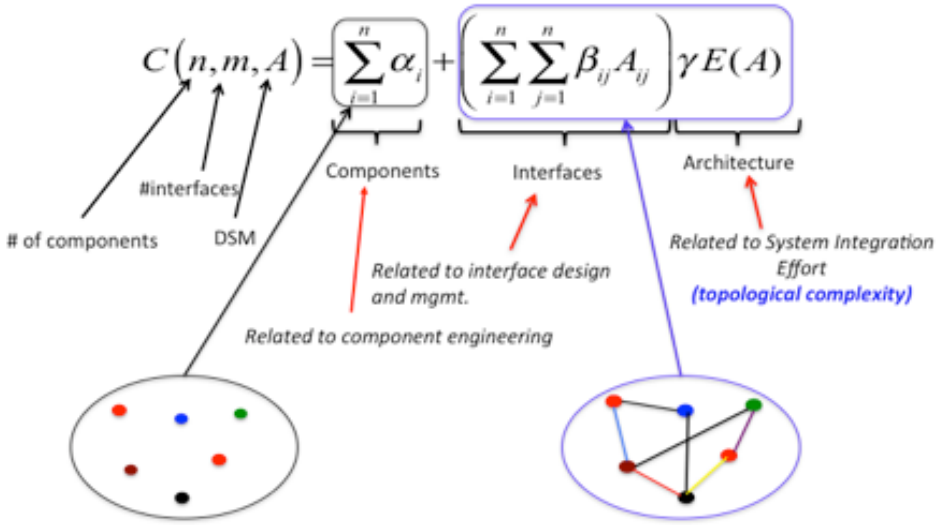
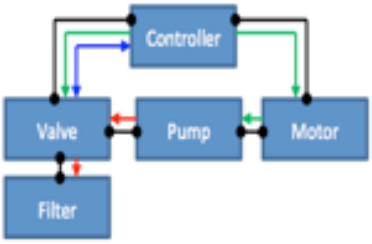


Fig. 3.1: Constituents of different parts of the overall structural complexity metric and their implications in the context of system development. Different components are differentiated using different colors. DSM stands for Design Structure Matrix [Ulrich and Eppinger, 1995].

The structural complexity metric can be treated as an affine transformation of the topological complexity metric, $C_3 = \frac{E(A)}{n}$. Here $E(A)$ stands for *graph energy* (discussed later in this chapter) and *n* stands for the number of system components (i.e., number of nodes).

Illustrative example: We present a small example of a hypothetical system for demonstrating the mechanics of the method, using a hypothetical fluid flow system as shown in Fig. 3.2 below. The associated binary adjacency matrix, A is shown below and the graph energy is computed as, *E(A)* = 5.6. Now let us differentiate among components and let the component complexity vector (i.e., α_i's)

be {(controller=5); (pump=2);(valve=1); (filter=1);(motor=3)}. The sum of component complexities C_1 = 5+3+2+1+1 = 12.



$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 3.2: (a) Sample system: it shows different connection types amongst components - physical/mechanical connection (black); material/fluid flow (red); energy flow (green) and information/control signal flow (blue); (b) binary adjacency matrix.

Let us use the following connection complexities: β^{mech} = 0.5 (black interfaces), $\beta^{flow/energy}$ = 1.0 (red and green interfaces respectively) and β^{info} = 1.0 (blue interface). Note that mechanical and information connections are bidirectional (they are counted twice), while fluid and energy flows are unidirectional (counted once). Therefore, the sum of connection complexities C_2 = 2*5*0.5 + 1*5*1 + 2*1*1 = 12. The binary adjacency matrix is constructed such that any two components (represented by rows of the adjacency matrix) are considered connected, if there is at least one pair-wise connection of any kind between these components. If we use $\gamma = 1/n = 1/5$, the structural complexity is (12+12*(5.6/5)) = 25.44.

The same methodology was applied to two different jet engine architectures, namely a dual spool turbofan and a geared turbofan engine. The specific details can be found in [Sinha and de Weck 2012] and in chapter 5 of this thesis. The detailed sensitivity analysis revealed that primary functionality generators (e.g., those generating thrust) are significant contributors to component complexity while supporting systems (e.g., lubrication systems, accessory gearbox, robust control systems) are the primary contributors to topological complexity and have significant impact on system integration efforts [Denman et al. 2011, Sinha and de Weck 2012]. In practice, assignment of component and connection complexities could be uncertain during the conceptual stage or even after the product

architecture is finalized. In such cases, the resulting structural complexity will not be a single number but a distribution, depending upon the distribution of individual component and connection complexities. The detailed application examples will be discussed later in chapter 5.

Now, we will look into the details of the three individual sources of structural complexity and how to compute/estimate them. We start with the topological complexity quantifier, C_3 since this is a novel contribution of this thesis. Subsequently, we will get into the detailed characterization of component and interface complexities that are required to compute the structural complexity metric.

3.2 Topological Complexity Metric, C_3

Any engineered system can be represented with many components that are linked to each other [Ulrich 1995, Lindemann et al. 2008]. The interaction between these parts influences the behavior of the system. This *system architectural pattern* leads to structural complexity of the system, which is associated with the difficulty of describing the connectivity patterns and their individual complexities [Kinsner 2010]. System architecture is an abstract description of the compositional entities and the interactions between those entities. The structure or configuration of the system is necessitated by constraints and requirements that the system is mandated to satisfy. This internal product architecture can be represented by complex networks, which are a graph-theoretic representation of complex systems. The nodes, representing components of the systems, are connected by links if there exists a direct interaction between any pair of components [Sheard and Mostashari, 2010]. The complexity of technical systems depends on the heterogeneity and quantity of different elements and their connectivity pattern, and is a measurable system characteristic.

It is observed that low topological complexity implies centralized architectures and that higher levels of topological complexity generally indicate highly distributed architectures.

It is often perceived that as we stretch the limits of efficiency and attempt to design more robust systems, we tend to make architectures more complex. One such example is the evolution of jet engine architecture [Frey *et al.*, 2007] where complexity rises to accommodate ever demanding and uncertain environments while maintaining consistent performance levels.

3.2.1 *Graph* or *Matrix Energy*:

Let G be a finite and undirected simple graph, with vertex set V(G) and edge set E(G). The number of vertices of G is n, and are labeled by v_1, v_2, … , v_n . The adjacency matrix A(G) of the graph G is a square matrix of size n, whose (i, j)-entry is 1 if the vertices v_i and v_j are adjacent and is zero otherwise [Li *et al.*, 2012]. The associated adjacency matrix A(G) is symmetric of size n with the diagonal elements being all zeros (see fig. 3.2 below). The total number of edges in a graph is denoted by m. The adjacency matrix described above has many other incarnations like Design Structure Matrix (DSM) or N^2 diagram [Ulrich and Eppinger, 1995] depending on the application area. DSM is the often-used terminology in the engineering design community for the binary adjacency matrix [Ulrich and Eppinger, 1995; Eppinger and Browning, 2012].

The topological complexity is defined as the matrix energy or graph energy of the adjacency matrix. Please note that we follow the definition of matrix or graph energy as defined in [Nikiforov, 2007] for the rest of this thesis.

Topological complexity originates from interaction between elements and depends on the nature of such connectivity structure. The adjacency matrix $A \in M_{nxn}$ of a network is defined as follows:

$$A_{ij} = \begin{cases} 1 \ \forall[(i,j) \mid (i \neq j) \text{ and } (i,j) \in \Lambda] \\ 0 \text{ otherwise} \end{cases}$$

where Λ represents the set of connected nodes and n being the number of components in the system. The diagonal elements of A are zero. The associated *matrix energy* [Nikiforov, 2007; Sinha and de Weck, 2013] or *modified* version of

graph energy (referred to as *graph energy* in rest of this thesis) of the network is defined as the sum of singular values of the adjacency matrix:

$$E(A) = \sum_{i=1}^{n} \sigma_i, \text{ where } \sigma_i \text{ represents i}^{th} \text{ singular value}$$

This definition is slightly modified version of original definition of graph energy used in [Gutman, 1978] where graph energy is defined as the sum of absolute eigenvalues of the adjacency matrix. This definition extends the applicability of the metric to any simple graph, undirected and directed alike. In case of a directed or mixed graph with combination of undirected and directed edges, the original definition of graph energy will not work since the eigenvalues could become complex.

The singular values of any matrix are always positive or zero and therefore, the modified graph energy or the matrix energy works for any simple graph. In case of undirected graphs, both definitions of graph energy converge since the singular values are the absolute eigenvalues in case of symmetric matrices. For the rest of this dissertation, we will refer to this modified version of graph energy as the graph energy. Hence, the graph energy in this dissertation is the matrix energy applied to the graph binary adjacency matrix (see appendices A and B for details).

The *matrix energy* also expresses the minimal effective dimension embedded within the connectivity pattern represented through the binary adjacency matrix. Notionally, this quantity encapsulates the "intricateness" of structural dependency among components. Using singular value decomposition (SVD), we can express matrix A as:

$$A = \sum_{i=1}^{n} \sigma_i \underbrace{u_i v_i^T}_{E_i} = \sum_{i=1}^{n} \sigma_i E_i \qquad (3.3)$$

where E_i represents simple, building block matrices of unit matrix energy and unit norm. Using this view, we observe that *matrix energy* or *graph energy* express the sum of weights associated with the building block matrices required to represent or reconstruct the adjacency matrix *A*.

Other mathematical connections of matrix energy or the modified graph energy relates to linear algebra and functional analysis - if the singular values of matrix *A* are labeled $\sigma_1,\sigma_2,...,\sigma_n$ and arranged in a non-increasing order, then the *Ky Fan* k-*norm* of matrix **A** is $\sum_{i=1}^{k}\sigma_i$. For k = n, the Ky Fan *k*-norm is referred to as the *Ky Fan trace norm* or *nuclear norm*. Evidently, the graph energy is simply the Ky Fan *n*-norm of the adjacency matrix. This latter norm is widely studied in matrix theory and functional analysis [Li *et al.*, 2012]. The *p-norm*, also called *Schatten p-norm*, defined as $\left(\sum_{i=1}^{n}\sigma_i^p\right)^{1/p}$, is another norm that is frequently used in analysis, where *p* is a real number and $p \geq 1$. Its special case is the Ky Fan trace norm (p = 1). Hence, the graph energy can be viewed as the p-norm of the adjacency matrix for p = 1.

The matrix energy or the nuclear norm has also been used in matrix reconstruction problem, where minimization of nuclear norm was shown to yield the optimal matrix [Candes and Tao, 2009] and forms a basis for semi-definite programming [Candes and Recht, 2009]. Recent research is exploring application of compressive sensing [Candes and Tao, 2009] to networks [Jiang et al., 2013].

3.2.2 Properties of *graph energy*

With this basic definition in tow, let us look at some important properties of the matrix or graph energy. The original graph energy [Li *et al.,* 2012] is defined as: $E(A)=\sum_{i=1}^{n}|\lambda_i|$ where λ_i stands for i^{th} eigenvalue of the adjacency matrix. For undirected graphs, the adjacency matrix A is symmetric and for symmetric matrices, we have $|\lambda_i| = \sigma_i$ where σ_i represents the i^{th} singular value of the adjacency matrix. The modified version of graph energy introduced here is the sum of singular values and coincides with the definition of the original graph energy metric defined by Gutman [Li *et al.,* 2012]. Since all diagonal entries are zero in *A* we can write,

$$\sum_{i=1}^{n}\lambda_i = 0 \qquad (3.4)$$

Using the Frobenius norm, $\left\|A\right\|_F$ [Bernstein, 2009] of A, we have:

$$\left\|A\right\|_F^2 = \sum_{i=1}^{n} \lambda^2{}_i = \sum_{i=1}^{n} \sigma^2{}_i$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \left|a_{ij}\right|^2 = 2m \qquad (3.5)$$

where *m* is the number of edges in the simple graph. Let us use the identity:

$$\left(\sum_{i=1}^{n} \lambda_i\right)^2 = \sum_{i=1}^{n} \lambda_i^2 + 2\sum_{1\le i\le j\le n} \lambda_i\lambda_j$$

$$\Rightarrow \sum_{1\le i\le j\le n} \lambda_i\lambda_j = -m \qquad (3.6)$$

In terms of singular values, we can write:

$$\left(\sum_{i=1}^{n} \sigma_i\right)^2 = \sum_{i=1}^{n} \sigma_i^2 + 2\sum_{1\le i\le j\le n} \sigma_i\sigma_j$$

$$= 2m + 2\sum_{1\le i\le j\le n} \sigma_i\sigma_j \qquad (3.7)$$

Linking singular and eigenvalues of the symmetric adjacency matrix A (i.e., for undirected edges), we can write:

$$\sum_{1\le i\le j\le n} \sigma_i\sigma_j = \sum_{1\le i\le j\le n} \left|\lambda_i\right|\left|\lambda_j\right| \ge \left|\sum_{1\le i\le j\le n} \lambda_i\lambda_j\right|$$

$$\Rightarrow \sum_{1\le i\le j\le n} \sigma_i\sigma_j \ge \left|\sum_{1\le i\le j\le n} \lambda_i\lambda_j\right|$$

$$\Rightarrow \sum_{1\le i\le j\le n} \sigma_i\sigma_j \ge m \text{ since } m>0 \qquad (3.8)$$

Now, using eq. 3.7 and 3.8, we can find a lower limit for the sum of singular values of the symmetric adjacency matrix A, representing the undirected edges:

$$\left(\sum_{i=1}^{n}\sigma_i\right)^2 = 2m + 2\sum_{1\le i\le j\le n}\sigma_i\sigma_j \ge 2m + 2m$$

$$\Rightarrow \left(\sum_{i=1}^{n}\sigma_i\right)^2 \ge 4m$$

$$\Rightarrow \left(\sum_{i=1}^{n}\sigma_i\right) \ge 2\sqrt{m} \qquad (3.9)$$

Therefore, we obtain the following limit on the matrix or graph energy, $E(A) \equiv \sum_{i=1}^{n}\sigma_i$

$$\therefore\ E \ge 2\sqrt{m} \qquad (3.10)$$

For connected graph, $m \ge (n-1)$ and therefore we can write: $E \ge 2\sqrt{n-1}$. This minimal graph energy is same as the graph energy of the centralized network where a central node connects to all other nodes and those peripheral nodes are connected only to the central nodes [Li *et al*., 2012].

Now, for any connected graph (valid for both, undirected and directed graphs), the adjacency matrix is shown to be *nonnegative* and *irreducible* [Chung, 1997] and therefore, applying *Perron-Frobenius* theorem [Chung, 1997; Van Mieghem, 2011], we have,

$$\sigma_1 \ge \frac{2m}{n} \equiv \langle k \rangle \quad [since\ \lambda_1 = \sigma_1 = spectral\ radius\ (A)]$$

$$\Rightarrow \sigma_1 \ge \langle k \rangle \qquad (3.11)$$

where $\langle k \rangle$ is the average degree of the network. Hence the dominant eigenvalue/singular value should be greater than the average degree $\langle k \rangle$ with equality holding for complete graph. Therefore as the density of a network increase, the dominant singular value increase (see fig. 3.3 below). This fact is corroborated by simulation and this relationship is found to be linear.

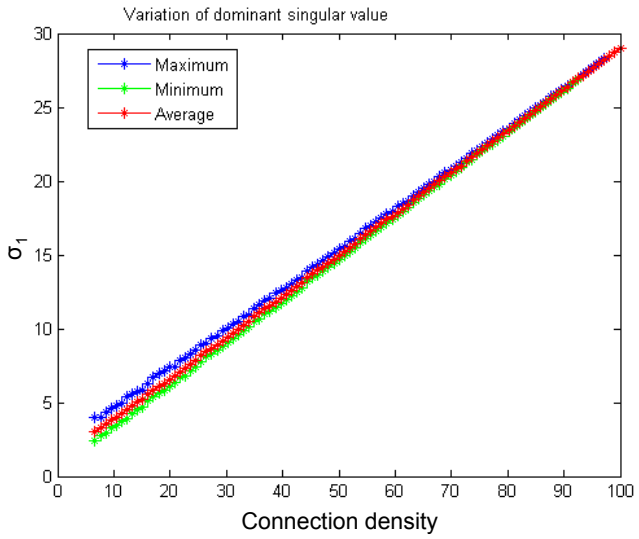


Fig. 3.3: The largest singular value increases linearly with the number of links in the ER random network. The number of nodes were held constant at, n = 30 nodes.

Using the Cauchy-Schwarz inequality,

$$\left(\sum_{i=1}^{N} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{N} a_i^2\right)\left(\sum_{i=1}^{N} b_i^2\right) \qquad (3.12)$$

which holds for arbitrary real-valued numbers a_i, b_i with i=1,2, . . . ,*N*. If we choose N = n, $a_i = \sigma_i$ and $b_i = 1$, we get the famous *McClelland Inequality*:

$$E \leq \sqrt{2mn} \qquad (3.13)$$

Now, for a connected graph, we must have $n \leq 2m$. Using McClelland inequality, $E \leq \sqrt{2mn}$, we get the following bound: $E \leq 2m$.

Hence we have:

$$2\sqrt{m} \leq E \leq 2m \qquad (3.14)$$

The bounds in 3.14 assumed symmetric adjacency matrix and is therefore valid for undirected graph only. In general, for engineered systems, we come across mixed graphs that has both, directed and undirected links. In that case, we can express the total number of entries in the adjacency matrix as *sm* where *m* is the number of links and $s \in [1,2]$ is a parameter, expressing the relative number of undirected

nodes in the mixed graph with $s=1$ for purely directed graph and $s=2$ for purely undirected graph. Using a similar analytical procedure, we can show that (see appendix G for the formal proof),

$$\sqrt{2sm} \le E \le sm \qquad (3.15)$$

Note that, relation 3.15 converges to 3.13 for $s=2$, the purely undirected graph case.

We can derive useful bounds for the matrix or graph energy by invoking the Cauchy-Schwarz inequality given by eq. 3.12 in different ways. Using the Cauchy-Schwarz inequality above with N = n-1, $a_i = \sigma_{i+1}$ and b_i = 1, then we obtain,

$$(E-\sigma_1)^2 \le (n-1)(2m-\sigma_1^2)$$
$$\Rightarrow E \le \sigma_1 + \sqrt{(n-1)(2m-\sigma_1^2)} \qquad (3.16)$$

Using the earlier relations: $\sum_{i=1}^{n}\sigma^2{}_i$ =2m and $\sigma_1 \ge \frac{2m}{n}$, the above relationship implies,

$$E \le \frac{2m}{n} + \sqrt{(n-1)[2m-\left(\frac{2m}{n}\right)^2]} \qquad (3.17)$$

Expressing in terms of average degree, the above inequality becomes:

$$E \le \langle k \rangle + \sqrt{(n-1)[2m-\langle k \rangle^2]} \qquad (3.18)$$

with equality holding for complete graph (e.g., fully integral architecture).

The limiting form of the above relation can be expressed in the following functional form for fixed *n,* (e.g., given matrix size):

$$f(m) = 2m/n + \sqrt{(n-1)[2m-(2m/n)^2]} \qquad (3.19)$$

Let us maximize the function $f(m)$ above where *n* is fixed. This allows us to find the number edges, m^* for a given number of nodes (alternatively, the graph density) at which the graph energy is maximized. Applying the *Kuhn-Tucker* optimality criteria [Pishkunov, 1976], we should have:

$$\frac{df}{dm} = 0$$

$$\Rightarrow \frac{2}{m} - \frac{(n-1)(4m-n^2)}{n\sqrt{2}\sqrt{m(n-1)(n^2-2m)}} = 0$$

On algebraic simplification, we get,

$$8m(n^2-2m) = (n-1)(4m-n^2)^2$$

$$\Rightarrow (4m-n^2)^2 = n^3$$

$$\therefore m = \frac{n^2+n^{3/2}}{4} = \frac{n^2\left(1+\frac{1}{\sqrt{n}}\right)}{4} \qquad (3.20)$$

Using this value of *m* and inserting that in the second derivative, we observe that the sufficiency condition for maximization is satisfied:

$$\frac{d^2f}{dm^2} = -\frac{(n-1)^2n^3}{2\sqrt{2}\left[(n-1)m(n^2-2m)\right]^{3/2}}$$

$$= -\frac{n^3\sqrt{(n-1)}}{2\sqrt{2}}\left\{\frac{1}{[m(n^2-2m)]^{3/2}}\right\}$$

$$= -\frac{n^3\sqrt{(n-1)}}{2\sqrt{2}}\left\{\frac{8}{n^3(2n+\sqrt{n}-1)}\right\}^{3/2} < 0$$

Therefore, for the maximal limiting value of graph energy, we get:

$$m^* = \frac{n^2\left(1+\frac{1}{\sqrt{n}}\right)}{4} \approx O(n^2) \qquad (3.21)$$

$$f_{\max} = \frac{n(1+\sqrt{n})}{2} \approx O(n^{3/2}) \qquad (3.22)$$

For the general case of mixed graphs, the maximal limiting value of graph energy remains the same, but the corresponding number of links becomes a function of parameter $s \in [1,2]$ (see appendix G for formal proof),

$$m^* = \frac{n^2\left(1+\frac{1}{\sqrt{n}}\right)}{2s} \approx O(n^2)$$

$$f_{\max} = \frac{n(1+\sqrt{n})}{2} \approx O(n^{3/2})$$

Again, the number of links in a network is constrained by the number of vertices in the network for a simple network (not allowing for multiple edges between pair of vertices and enforcing symmetry):

$$m \le \frac{n(n-1)}{2}$$

$$\Rightarrow 2m \le n(n-1)$$

Now, from the *McClelland Inequality*, $E \le \sqrt{2mn}$, we arrive at the following graph energy bound:

$$E \le n\sqrt{n-1} \qquad (3.23)$$

In addition, from eq. 3.18, 3.20 and 3.22, we can write:

$$E \le \frac{n(1+\sqrt{n})}{2} \qquad (3.24)$$

Therefore, the maximal graph energy is bounded by $n^{3/2}$:

$$E_{\max} \le n^{3/2} \qquad (3.25)$$

Now the energy of a complete graph, $E_c = 2(n-1)$ and therefore does not represent maximal graph energy for a network with fixed number of vertices. There exists *hyperenergetic* graph G whose graph energy, E(G) > E_c.

Also for any connected graph, E ≥ $2\sqrt{n-1}$ with equality holding for a network with a hub and all other vertices connecting to the hub. In this case, there are only two non-zero singular values and *(n-2)* zero singular values and therefore the adjacency matrix ix highly rank deficient.

In general, we get the following graph energy bounds for a network with *n* vertices:

$$2\sqrt{(n-1)} \le E \le \frac{n(1+\sqrt{n})}{2} \qquad (3.26)$$

If we were to consider the case of mixed star graph, the above (i.e., star graphs with a mix of directed and undirected graphs) bounds can be expressed as (see appendix G for details),

$$s\sqrt{(n-1)} \le E \le \frac{n(1+\sqrt{n})}{2}$$

Let us consider that case that graph energy, E is the same as that of the complete graph where,

$$E = 2(n-1) \qquad (3.27)$$

Using eq. 3.18 and 3.27, we can write:

$$2(n-1) \le \langle k \rangle + \sqrt{(n-1)(n\langle k \rangle - \langle k \rangle^2)}$$

$$\Rightarrow 4(1-\frac{1}{n}) \le \langle k \rangle \le (n-1) \qquad (3.28)$$

Note that the network density is defined as, $\mu = \langle k \rangle / (n-1)$. On substituting $\mu = \langle k \rangle / (n-1)$ in eq. 3.28, we have:

$$\frac{4}{n} \le \mu \le 1 \qquad (3.29)$$

Hence, the average degree has to be greater than 3 (for any graph of size more than 4) for having graph energy equal to that of complete graph for any given number of vertices (e.g., given matrix size). The lower bound in 3.29 refers to a smaller level of network density at which the graph energy reached the same level as the fully connected graph with the same number of vertices. As the number of vertices, n increases, this lower limit of network density decreases. The lowest possible average degree, $\langle k \rangle$ at this density level has to be at least 4.

We define the density level at which graph energy equals that of the fully connected state as the critical network density. Note that this is a generic result and holds for any simple network, undirected or directed. The impact of asymmetric adjacency matrix on structural complexity can be found in appendix F.

3.2.3 Simulation results:

In this sub-section, we verify the bounds for graph energy developed above with a series of numerical experiments. For a given network size, the percent population is varied and for a given density of interdependencies, the entries are filled randomly, while maintaining connectedness (i.e, there are no disconnected component) of the graph. For a given number of interdependencies *m*, there are multiple arrangements possible. Therefore the graph energy of instantiated graphs exhibits a band of values for any fixed density of the network.

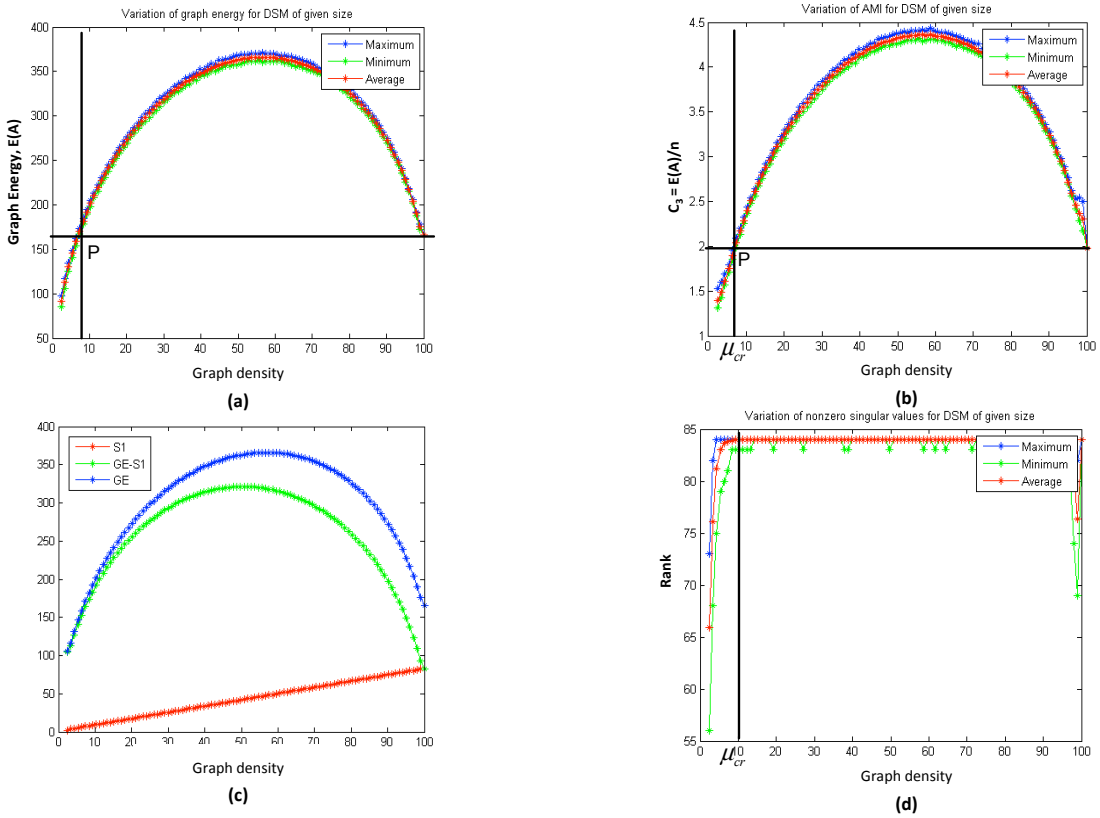


Fig. 3.4: Variation of spectral properties with increasing graph density for Erdos-Renyi random, undirected graphs – (a) the *P* point is where the graph energy becomes equal to that of the fully connected network; (b) at the *P* point, the normalized graph energy, $E(A)/n$ is around 2; (c) variation of largest singular value (red), graph energy less the largest singular value (green) and graph energy (blue) and (d) at the *P* point, on average, the instantiated graphs attains rank-sufficiency.

The average, maximum and minimum values of graph energy are plotted below. In all cases, 500 trials are run and the range of network density was divided into 200 points on the interval [2/n,1], where *n* is the number of nodes in the graph/network. One can observe from fig. 3.3 and fig. 3.4 (c) that the dominant singular value, σ_1 increases linearly with number of connections whiles the sum of remaining singular values, $\left(\sigma_2+\sigma_3+..+\sigma_n\right)$ has a convex relationship with graph density. If we define this as the combinatorial part of graph energy respectively, then we can say that the *combinatorial* part of the energy shows a concave relationship with connection density (e.g., maximum at the middle and diminishes at the two extremes of connection density spectrum). The resulting total graph energy is also concave function of connection density but is asymmetrical.

3.2.4 Notion of *P* point: Hypo and hyper energetic regimes

The matrix or graph energy regime for graphs with a given number of nodes, can be divided into: (i) *hyperenergetic* and (ii) *hypoenergetic*. There does exist an intermediate or transition regime between these two where the energy is higher than that of the *hypoenergetic* regime but is smaller than the *hyperenergetic* one [Li *et al.*, 2012]. The *hyperenergetic* regime is defined by graph energy greater than or equal to that of the fully connected graph:

$$E(A) \geq 2(n-1)$$

The *hypoenergetic* regime is defined as:

$$E(A) \leq n$$

There is also a region termed strongly hypoenergetic where the graph energy satisfies:

$$E(A) < (n-1)$$

Hence, in terms of topological complexity metric, the regimes are defined as:

$$C_3 = \begin{cases} \geq 2\left(1-\frac{1}{n}\right) \approx 2 & \text{- hyperenergetic} \\ < 1 & \text{- hypoenergetic} \end{cases} \qquad (3.30)$$

The intermediate or transitional energetic regime is the interval: $1 \leq C_3 < 2$. Hence, for graphs with topological complexity, C_3 in this intermediate lie in the non-hyperenergetic regime (similarly, they can be termed to lie in the hypo-energetic regime).

In fig. 3.4 and 3.5, the point *P* denotes the transition point to *hyperenergetic* graphs. The network densities that lie on the right of *P* point represents distributed configuration (e.g., architecture). At this point the network becomes rank-efficient on average.

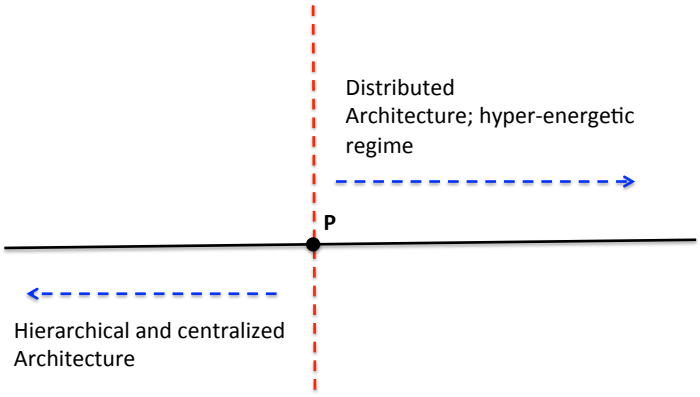


Fig. 3.5: *P* point separates the distributed architectural regime, defined by hyper-energetic graph structures.

Let the critical density corresponding to point *P* be termed as *critical density*, μ_{cr} and critical average degree, $\langle k \rangle_{cr}$ be the corresponding average degree of the graph. They are related as:

$$\mu_{cr} = \frac{\langle k \rangle_{cr}}{n-1} \qquad (3.31)$$

Above this density, the corresponding graph becomes *hyperenergetic*. According to eq. 3.31, as the network size increase, the critical density reduces. This in turn indicates a constant band of values for m_{cr} and this quantity does not vary

much with variation in network size and seem to lie within a relatively compact interval of [5,7].

Based on extensive simulation studies (averaged over 10,000 instances at each network density level) on ER random graphs with given number of vertices, the variation of the critical density and corresponding critical average degree for varying the graph size is shown in fig. 3.6 below. We can observe that as graph size increase, the critical density reduces, but the critical average degree tends to remain almost constant.

The *P* point shows very interesting features relating to interesting characteristics of the graph. It appears that nearness to rank-sufficiency of the network has important bearing on other network metrics as well (see appendix H for details). Simulations indicate saturation in terms of relative improvement in other network metrics like maximum diameter, average path length and other network communicability indices.

Another interesting observation was made in an analytical study by [Valente et al., 2004] regarding the resilience of general random networks against both, targeted and random attack on nodes (see appendix H for details). Using analytical methods, they estimated bounding envelopes of resilience against targeted nodal failure vis-à-vis random nodal failure and found that the outward growth of the envelope saturates beyond the average degree, <k> = 6. This again coincides with *P* point on the graph density plot. It is interesting to note that the *P* point seem to indicate a kind of transition region for network diameter beyond which it settles down to a near constant value with increasing connection density (see appendix H for details). This is an interesting finding and relates to the findings of an earlier empirical study [Whitney *et al.,* 1999] using a large and diverse set of engineered products and systems showed that the average number of connections to any component (i.e., average degree) was about 6 (see appendix G for additional details). This is a very interesting empirical finding and backs up the analytical predictions.

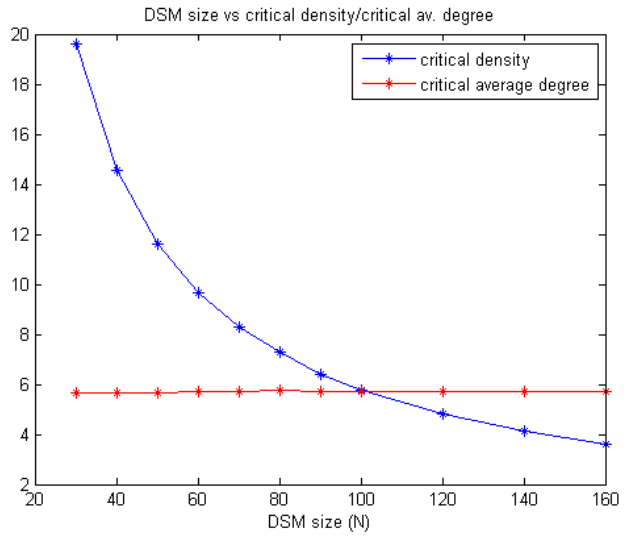


Fig. 3.6: Variation of critical density and corresponding average degree with increasing graph size. DSM stands for Design Structure Matrix [Steward, 1981; Eppinger and Browning, 2012]

It appears that the *P* point might serve as an important system architecting guideline for engineered, networked systems. We define the *P* point as a *critical point* in the graph density space and derive interesting bounds related to the graph density at this point (graph density at *P* point is called the critical density).

The average degree of *non-hyperenergetic* networks is less than this critical value of average degree of network of given number of vertices.

Now, based on eq. 3.28, we can write:

$$\langle k \rangle_{cr} \geq 4\left(1-\frac{1}{n}\right)$$

$$\therefore \quad \mu_{cr} \geq \frac{4}{n} \qquad (3.32)$$

Therefore, at the critical density, the corresponding number of links, m_{cr} is given as:

$$\begin{aligned} m_{cr} &= \mu_{cr}\frac{n(n-1)}{2} \\ &\geq \frac{4}{n}\frac{n(n-1)}{2} \\ &\geq 2(n-1) \qquad (3.33) \end{aligned}$$

The variation of graph energy with density is quadratic in nature and therefore, neglecting statistical variations, we can model graph energy as:

$$E = a\mu^2 + b\mu + c \qquad (3.34)$$

At $\mu = 0$; we have E=0 => c = 0. Hence, we get:

$$E = a\mu^2 + b\mu \qquad (3.35)$$

Now let us use other boundary conditions. At $\mu = 1$ (fully connected graph), E = 2(n-1) and at $\mu = \frac{2}{n}$, $E = E_{ch}$ where E_{ch} is the graph energy for a linear chain.

Using these conditions, we can express graph energy as:

$$E = \frac{n}{(n-2)}[4(n-1) - nE_{ch}]\mu^2 + \frac{n}{(n-2)}[nE_{ch} - \frac{8(n-1)}{n}]\mu \qquad (3.36)$$

For maximal graph energy, E, we must have,

$$\frac{\partial E}{\partial \mu} = 0$$

$$\mu_{\max} = \frac{nE_{ch} - 8(1 - \frac{1}{n})}{2[nE_{ch} - 4(n-1)]} \qquad (3.37)$$

As $n \to \infty$, we have,

$$\mu_{\max} \approx \frac{1}{2\left[1 - \frac{4}{E_{ch}}\right]} \qquad (3.38)$$

Now, since $E_{ch} < 2(n-1)$, we have

$$\left(1 - \frac{4}{E_{ch}}\right) < \left(\frac{n-3}{n-1}\right)$$

$$=> \frac{1}{2\left(1 - \frac{4}{E_{ch}}\right)} > \frac{(n-1)}{2(n-3)} \qquad (3.39)$$

Combining eq. 3.38 and 3.39, we get:

$$\mu_{\max} > \frac{(n-1)}{2(n-3)} \qquad (3.40)$$

From eq. 3.40, for any *n*, we can write:

$$\mu_{\max} > \frac{1}{2} \qquad (3.41)$$

This is a hard limit and can only be approached asymptotically. Hence the maximum value of matrix or graph energy is attained at a density greater than 0.5. As the network size increase, the density corresponding to maximal energy becomes closer to 0.5.

3.2.5 Notion of *Equi-energetic* Graphs

Two non-isomorphic graphs [Cvetkovic *et al.,* 1980] are said to be *equienergetic* if they have the same matrix /graph energy. There exist finite pairs of graphs with identical spectra, called *cospectral graphs* [Cvetkovic *et al.,* 1980]. All such cospectral graphs are off course trivially equienergetic. There exist finite non-cospectral graphs whose matrix or graph energy are equal and are therefore *equienergetic*. It has been observed that there are only finite numbers of cospectral graphs and this number tends to zero as the number of nodes, *n* increases [Li *et al.*, 2012]. Also, there exist finite non-cospectral, equienergetic graphs and that number goes to zero with increasing number of nodes, *n* in the graph. Hence there could distinct graph structures with the same graph energy and therefore, topological complexity. The topological complexity for a graph structure is not unique and is therefore, not a *bijective* mapping of systems.

3.2.6 Other important properties of graph energy

In this sub-section, we state a set of 12 important properties of graph energy that are crucial to perform mathematical verification of the proposed complexity metric using the Weyuker's criteria. Here are the set of important properties of matrix or graph energy:

P.1 For hyperenergetic graphs, we have matrix or graph energy, $C_3 \geq 2$. Also any graph with the number of links, $m \geq \frac{n^2}{4}$ is non-hypoenergetic [Li *et al.*, 2012].

P.2 For hyperenergetic graphs, we must have, $m \geq 2(n-1)$. This is greater than twice the necessary number of links to maintain connectivity (e.g., no disconnected vertices).

P.3 Graphs with $m \leq (2n-1)$ cannot be hyperenergetic [Li *et al.*, 2012].

P.4 Trees or hierarchical structures are non-hyperenergetic.

P.5 Any graph with nonsingular adjacency matrix is non-hypoenergetic.

P.6 For any graph on *n* nodes with *cyclomatic number*, $c = m - n + 1$, there exists a constant γ_c such that the graph energy is bounded [Wagner, 2012] as:

$$E(A) \leq \frac{4n}{\pi} + \gamma_c$$

$$\Rightarrow C_3 \leq \frac{4}{\pi} + \hat{\gamma}_c$$

P.7 For almost all graphs with *n* nodes, where *n* is sufficiently large, the graph energy is bounded [Li *et al.*, 2012] by:

$$\left[\frac{1}{4} + o(1)\right] n^{3/2} < E(A) < \left[\frac{1}{2} + o(1)\right] n^{3/2}$$

$$\Rightarrow \left[\frac{1}{4} + o(1)\right] n^{1/2} < C_3 < \left[\frac{1}{2} + o(1)\right] n^{3/2}$$

Hence, for graphs with *n* nodes, where *n* is sufficiently large, the spread in graph energy and the topological complexity metric are bounded by:

$$\Delta E \leq \left[\frac{1}{4} + o(1)\right] n^{3/2} \text{ and } \Delta C_3 \leq \left[\frac{1}{4} + o(1)\right] \sqrt{n}$$

As we can observe, the spread could be quite large, given the number of nodes or components in the system and varies as the square root of the number of components (e.g., nodes).

P.8 Translating the resulting graph structures to system architectural patterns, we associate typical topological complexity metric C_3 values to those forms:

$$Centralized\ Architecture \rightarrow hypoenergetic,\ C_3 < 1$$

$$Hierarchical\ /\ layered\ Architecture \rightarrow transitional,\ 1 \leq C_3 < 2$$

$$Distributed\ Architecture \rightarrow hyperenergetic,\ C_3 \geq 2$$

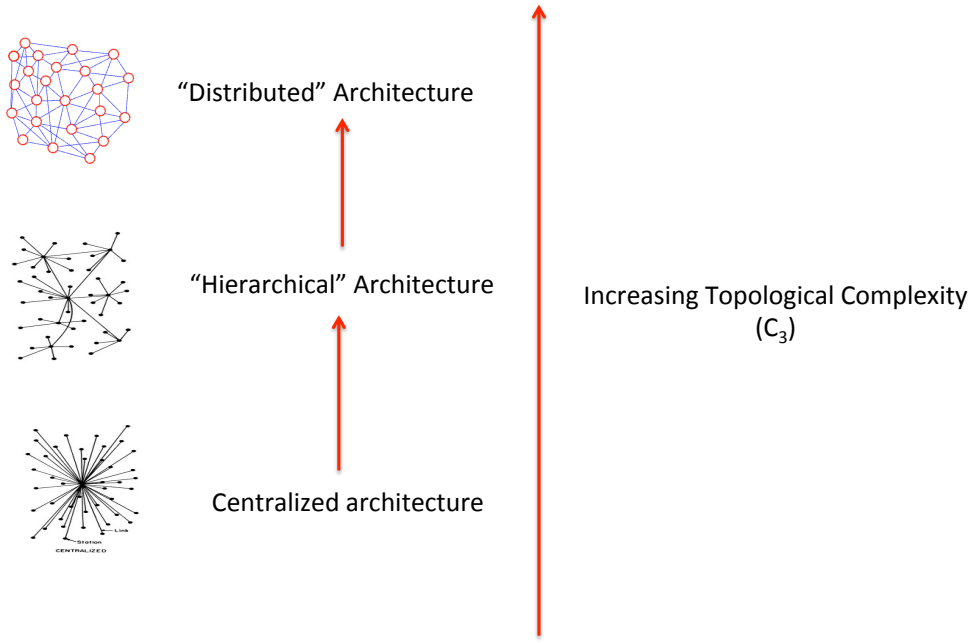


Fig. 3.7: Spectrum of architectural patterns based on topological complexity metric.

P.9 There exists a finite number of connected *n*-vertex *noncospectral, equienergetic* graphs for every $n \geq 20$ [Li *et al.*, 2012].

P.10 Let A, B and C be square matrices of size *n*, such that A + B = C. Then

$$E(A) + E(B) \geq E(C)$$

This is termed the *coalescence* of two graphs - merging of two graphs without introduction of an additional edge. Let A and B be two graphs with disjoint vertex

sets, and $u \in$ A and $v \in$ B. Thus, merging the two graphs at a node, we have the following relation: $|V(A \circ B)| = |V(A)| + |V(B)| - 1$. The graph $A \circ B$ is known as the *coalescence* of A and B with respect to *u* and *v*. Hence, we can write $A + B = A \circ B$ and we have the following inequality:

$$E(A \circ B) \leq E(A) + E(B)$$

This inequality is not very relevant to most engineered complex systems, but may be of importance for reconfigurable systems where a component adapts its functionality based on the environment. Since reconfigurable systems can attain different configurations over time and are suitable for classes of applications in which their ability to undergo changes easily can be exploited to fulfill new demands, allow for evolution, and improve survivability [Siddiqi and de Weck, 2008]. Examples include convertible car, reconfigurable modular robots, morphing UAV's (Unmanned Aerial Vehicle) [Siddiqi and de Weck, 2008],

P.11 For a partitioned matrix $\Lambda = \begin{bmatrix} A & X \\ Y & B \end{bmatrix}$, where both A and B are square matrices, we have:

$$E(\Lambda) \geq E(A) + E(B)$$

Equality holds if and only if X, and Y are all zero matrices. This shows that introduction of edges in the process of connecting two disparate graphs results in an increase in graph energy and therefore increases the topological complexity. This is a restricted version of the general *pinching inequality* [Horn and Johnson, 2000]. This inequality mathematically expresses that the system is more complex that its constituent sub-systems – that is, the system is more complex than the sum of parts.

P.12 Let A be an *n* x n matrix partitioned into blocks such as the diagonal blocks A_{ii} are square:

$$A = \begin{bmatrix} A_{11} & A_{12} & . & . & A_{1N} \\ A_{21} & A_{22} & . & . & A_{2N} \\ . & . & . & . & . \\ . & . & . & . & . \\ A_{N1} & A_{N2} & . & . & A_{NN} \end{bmatrix}$$

The following block diagonal matrix is called the *pinching* of A:

$$\Theta(A) = \begin{bmatrix} A_{11} & 0 & . & . & 0 \\ 0 & A_{22} & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & A_{NN} \end{bmatrix}$$

Pinching reduces the matrix energy of A for every partitioned matrix and the pinching inequality is stated as:

$$E(\Theta(A)) \leq E(A) \Rightarrow \sum_{i=1}^{N} E(A_{ii}) \leq E(A)$$

Combining the previous result with P.11, we obtain the fundamental pinching inequality:

$$\sum_{i=1}^{N} E(A_{ii}) \leq E(A) \leq \sum_{i=1}^{N} E(A_{ii}) + \sum_{i \neq j=1}^{N(N-1)} E(A_{ij})$$

This inequality shows that the graph energy of the total aggregated system is higher than the sum of graph energies of the constituent sub-systems.

3.2.7 Necessary properties of a valid complexity metric: Weyuker's Criteria

Weyuker's criteria represent a formal set of required properties any valid complexity metric must satisfy. To check the construct validity of graph energy as the complexity metric, we benchmarked this against the set of minimal required properties prescribed as Weyuker's criteria [Weyuker, 1988] and also compared against some other complexity metrics that have been proposed in the existing literature (see table 3.1) [Kinnunen, 2006; Lindemann et al., 2008].

As we can observe, the proposed topological complexity metric is fully compliant with Weyuker's criteria [Weyuker, 1988].

Table 3.1: Benchmarking of matrix energy or graph energy against Weyuker's criteria (i.e., a set of nine criterion [Weyuker 1988]) and comparison with other proposed metrics of complexity.

W.C / Metric	1	2	3	4	5	6	7	8	9
Number of components	✓	✓	✓	✓	✓	✗	✗	✓	✗
Number of links	✓	✓	✓	✓	✓	✗	✗	✓	✗
# independent loops	✓	✗	✓	✓	✓	✗	✗	✓	✗
Degree of non-planarity	✓	✓	✓	✓	✗	✗	✓	✓	✗
Nesting depth	✓	✓	✓	✓	✓	✗	✗	✓	✗
Topological Complexity	✓	✓	✓	✓	✓	✓	✓	✓	✓

Now the structural complexity metric, as defined in eq. 3.1, is an *affine transformation* of the proposed topological complexity metric. Therefore, it satisfies the *necessary conditions* for any proposed structural complexity metric. We can think of these *necessary conditions*, as exemplified by the Weyuker's criteria, as conditions for mathematical verification of the proposed complexity metric.

Below we prove that the proposed topological complexity metric fulfills the Weyuker's criteria and thereby, qualifies as a valid complexity metric.

Let *A* and *B* are two different systems and *K(A)* denotes complexity of system *A*:

1. ***There exist A and B such that K(A) differs from K(B),* for a metric, which gives the same value for all systems is useless.**

Proof: Let us consider B is obtained by removing some edges of A. Hence system B is the *induced subgraph* of system A and B = A-e. Depending on the level and pattern of connectivity we have, E(A-e) < E(A) in general or E(A-e) > E(A) if A is a complete graph. There could be non-isomorphic, *equienergetic* graphs, but their number is finite and tends to zero as we increase the graph sixe, *n* [Li *et al*., 2012].

2. ***There exist only finitely many systems of complexity c,* for the metric should have a range of values.**

Proof: As we have observed from the properties of matrix or graph energy, there

are finitely many *equienergetic* graphs of size *n* and their number tends to zero as the graph size increases [Li *et al.*, 2012]. Hence, the proposed graph energy based topological complexity metric satisfies this criterion.

3.*There exist distinct systems A and B for which K(A) = K(B),* for a metric which gives each system unique value is not useful since such a metric would be a simple bijective mapping of systems.

Proof: There exist finite non-cospectral, equienergetic graphs and that number goes to zero with increasing number of nodes, n in the graph. Hence there could distinct graph structures with the same graph energy and therefore, equal topological complexity. Hence, it is possible that there exist two distinct system structures with identical topological complexity.

4. *There exist functionally equivalent systems A and B for which K(A) ≠ K(B),* for the structure of the system determines its complexity.

Proof: The *function to form* mapping is not unique as the same (or nearly identical) functionality can be achieved using different system architectures. They may use different *concepts* to achieve the same functionality. The differences in their system structure yields distinct graph energy values and therefore, have distinct topological complexities.

5. *For all A and for all B, K(A) is smaller than K(A∪B), and K(B) is smaller than K(A∪B),* for a system is more complex than its subsystems.

Proof: This is a restatement of the graph energy property P.11. In P.11, designating $\Lambda = A \cup B$, we observe $E(\Lambda) \geq E(A) + E(B)$. Hence, we have, $E(\Lambda) \geq E(A)$ and $E(\Lambda) \geq E(B)$. In case of system structures formed by *coalescence* of two graphs, we conclude that $E(A \circ B) \geq E(A)$ and $E(A \circ B) \geq E(B)$ since with addition of nodes while keeping the basic system structure constant leads to an increases of graph energy [Li *et al*., 2012].

6. *There exists A, B, and M such that, K(A) = K(B) and K(A∪M) ≠ K(B∪M),* for *M* may interact with *A* in different manner than with *B.* Namely, the

interface structure between *A* and *M* may be more complex than interfaces between *B* and *M*.

Proof: Let us consider the following system structure Λ_1 where *A* and *M* represent subsystems and *X* represents the interfaces between the two:

$$\Lambda_1 = \begin{bmatrix} A & X^T \\ X & M \end{bmatrix}$$

The above system structure can be represented in the following block matrix form:

$$\Lambda_1 = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ X & 0 \end{bmatrix} + \begin{bmatrix} 0 & X^T \\ 0 & 0 \end{bmatrix}$$

$$\Rightarrow E(A) + E(M) \le E(\Lambda_1) \le E(A) + E(M) + 2E(X)$$

$$\Rightarrow E(\Lambda_1) = E(A) + E(M) + \Delta(X)$$

Hence, the resultant graph energy of the integrated system has an *integrative topological complexity* component given by $\Delta(X)$.

Similarly, the system structure Λ_2 where *B* and *M* represent subsystems and *Y* represents the interfaces between the two:

$$\Lambda_2 = \begin{bmatrix} B & Y^T \\ Y & M \end{bmatrix}$$

The above system structure can be represented in the following block matrix form:

$$\Lambda_2 = \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ Y & 0 \end{bmatrix} + \begin{bmatrix} 0 & Y^T \\ 0 & 0 \end{bmatrix}$$

$$\Rightarrow E(B) + E(M) \le E(\Lambda_2) \le E(B) + E(M) + 2E(Y)$$

$$\Rightarrow E(\Lambda_2) = E(B) + E(M) + \Delta(Y)$$

In this case, the resultant graph energy of the integrated system has an *integrative graph energy/topological complexity* component given by $\Delta(Y)$.

Assuming $E(A) = E(B)$, the difference in graph energy between Λ_1 and Λ_2 is:

$$E(\Lambda_1) - E(\Lambda_2) = \Delta(X) - \Delta(Y)$$

Hence, the difference in their topological complexity depends on how the individual subsystems interface with each other. Their topological complexities can only be equal if the subsystem interfaces are identical.

7. *There are systems A and B such that B is a permutation of components of A and K(A)≠ K(B),* for changing the way connecting the components to each other, may change the level of complexity.

Proof: This criterion is satisfied due to property P.7 as the largest difference in graph energy with same number of system components is bounded by $\Delta E \leq \left[\frac{1}{4} + o(1)\right] n^{3/2}$. The difference stems from the way nodes of the system are connected to each other.

8. *If A is a renaming of B, then K(A) = K(B),* for complexity does not depend on the naming of the system. This relates to the property of invariance under isomorphic transformation.

Proof: By definition, the singular values of a matrix are independent of any rearrangement of its rows and columns. The singular values are invariant to isomorphic transformation. Therefore the graph energy, are invariant under isomorphic transformation of the graph [Horn and Johnson, 1997].

9. *There exist A and B such that K(A)+K(B) is smaller than K(A∪B),* for putting systems together creates new interfaces. This pertains to the notion of "system is greater than the sum of parts".

Proof: This criterion is satisfied due to the *pinching inequality* described in P.11. For a partitioned matrix $\Lambda = \begin{bmatrix} A & X \\ Y & B \end{bmatrix}$, where both A and B are square matrices, we have: $E(\Lambda) \geq E(A) + E(B)$. Equality holds if and only if X, and Y are all zero matrices. Hence, introduction of edges in the process of connecting two disparate graphs results in an increase in total graph energy of the aggregated

system and therefore increases the topological complexity. This inequality says that the system is more complex that its constituent sub-systems – that is, the system is larger than the sum of parts.

In addition, the complexity metric should satisfy *sufficient conditions* that might include domain-specific considerations. It should be easily computable across a wide swath of engineered complex systems and more importantly, empirically validated using data from real-world, complex engineered systems.

The empirical validation is accomplished by establishing correlation between the proposed measure of complexity and *measurable* programmatic quantities like system development cost/effort. This validation is the focus of the next chapter (i.e., Chapter 4).

3.2.8 Graph Energy and its relationships to other metrics

The graph energy and by extension, the topological complexity metric, C_3 are instances of spectral metrics since they stem from the spectral properties of the adjacency matrix, A of the underlying graph. Please note that *matrix* or *graph energy* expresses the sum of weights associated with the building block matrices required to represent or reconstruct the adjacency matrix *A*.

Another spectral metric connected to this viewpoint is the reconstructability index, θ of a graph that expresses how many dimensions of the *n*-dimensional space are needed to *exactly* represent or reconstruct the graph [Liu *et al.*, 2010, Van Mieghem, 2011]. A parallel to this viewpoint is the *graph reconstructability* viewpoint [Liu *et al.*, 2010, Van Mieghem, 2011] and the ability to easily reconstruct system structure can be viewed as the dual of topological complexity. Notice that, for exact reconstruction, we have to exactly replicate each element of the adjacency matrix. This requirement of *exact* reconstruction is in contrast to, for example, image compression, where some image information might be lost.

A larger reconstructability index reflects a "simpler" graph that only needs a few dimensions to describe. In other words, the higher the reconstructability index, the easier it is to reconstruct the graph accurately from reduced information.

Simulation studies indicated such dualism between graph energy and graph *reconstructability* (see appendix E for detailed analysis). Minimum topological complexity implies maximum reconstructability and vice-versa. The behavior of graph reconstructability with increasing link density is dual of the graph energy (refer to appendix E for details). The graph energy and the topological complexity metric can be used to detect the change in system connectivity structure in the Fabrikant model [Fabrikant *et al.,* 2002] as it transitions from star networks to tree with multiple hubs with identical number of nodes and edges (see appendix E).

The relationship between graph energy and structural resilience of the underlying graph is explored in appendix H and relates it to the notion of *P* point introduced earlier.

From extensive simulations, it was observed that before reaching the topological complexity level of 2 (i.e., $E(A)/n < 2$), it takes failure of only 5% of nodes on the average to bring down or disconnect the graph. For a specified probability of nodal failure and permissible level of structural resilience, there is a target level of topological complexity that is essential. It appears that networked systems require a minimum complexity level to guard against network disintegration and it appears that topological complexity at or beyond the *P* point may provide a reasonable level of structural robustness against graph disintegration (see appendix H for detailed description).

3.3 Estimation of Component Complexity

Having looked at the topological complexity, let us turn our attention to *local* aspects of structural complexity, namely the *component complexity*, C_1. While the topological complexity component is domain-independent and only requires consideration of a binary connectivity matrix of the system, estimation of component complexity requires domain knowledge or domain-specific information. Characterization of component complexity is based on the technical design/development difficulty associated with the individual component alone, not accounting for the complexity of its interfaces. The method of component

complexity estimation varies depending on the level of fidelity and amount of data, either historical or contextual. Here are some strategies for estimating component complexity and their choice might be dictated by the availability of data.

(a) Estimation based on technological maturity: The implicit assumption here is that the component complexity is proportional to the technological maturity or readiness level (i.e., TRL). If a component and its underlying operating principles are matured and mastered over time, than the component is termed less complex than otherwise. This measure could be based on the widely used notion of component TRL (i.e., Technology Readiness Level) or other similarly motivated measures as a surrogate for component complexity.

We propose a component complexity scale on the interval [0, 5] and computed from the component TRL [Sadin *et al.*, 1988] level as:

$$\alpha_i = 5\left(\frac{TRL_{\max} - TRL_i}{TRL_{\max} - TRL_{\min}}\right) \qquad (3.42)$$

where TRL_i stands for the TRL of the i^{th} component and the technological readiness level is defined in $[TRL_{\min}, TRL_{\max}]$. Hence, eq. 3.42 converts a TRL scale to a component complexity scale, with the component complexity scale being continuous in general. In practice, this approach works well if the organization developing the system is well versed with and rigorously assigns technological maturity levels for components, following a consistent procedure. If we have a distribution for TRL_i, then α_i is also a derived distribution. This is a relatively crude, macro-level measure and its reliability has been questioned in the literature [Babuscia, 2012].

(b) Estimation of component complexity when no reliable data are available: In this scenario, we do not have available data and not enough information to meaningfully assign a maturity index. The only available option in such cases is *expert opinion*. This is particularly applicable for any novel component that is used for the first time. This method usually consists of (a) elicitation of expert opinion;

and (ii) pooling/aggregation of expert opinions (in case of multiple experts, which is usually the case).

As observed from fig. 3.8 below, expert opinion plays its role in providing subjective probabilities when minimal information/knowledge is available. It is also possible to devise a hybrid approach of combining expert opinion and available data using Bayesian approaches [Babuscia, 2012].

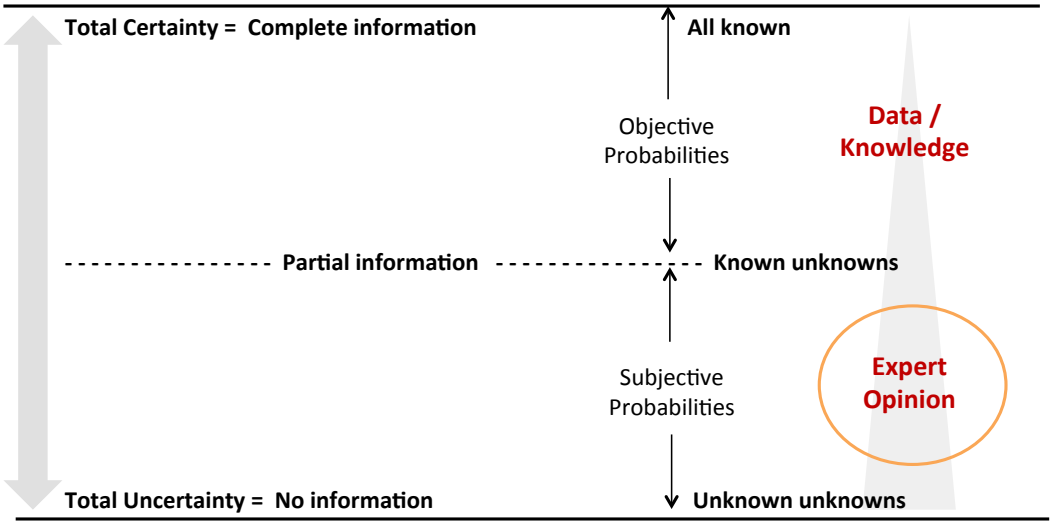


Fig. 3.8: Expert opinion is the only option when minimal information is available. One can use a hybrid approach with availability of additional data [Greenberg, 2013].

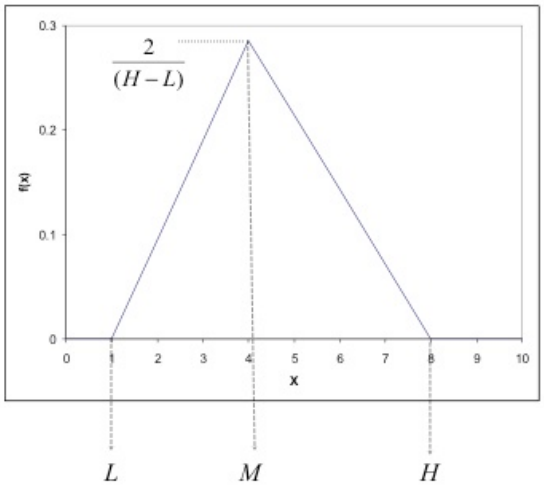
The final output component complexity is usually a probability distribution and triangular distributions are generally used for this purpose due to their inherent advantages as described in [Garvey 2000].

(i) ***Expert elicitation*****:** Expert opinion elicitation is a structured approach to capture subject matter expert's (SME) knowledge base and convert that knowledge into quantitative estimates. This is accomplished using a questionnaire-based survey or similar procedures to converge on an estimated component complexity distribution. The elicitation process consists of multiple cycles.

The triangular distribution is used as a subjective description of a population for which there is only limited sample data. Triangular distribution is a bounded distribution and is useful if the expert has a certain level of confidence in an

"informed guess", and in cases where the relationship between variables is known but data is scarce, possibly because of the high cost of collection [Kotz and van Dorp, 2004].

It is based on knowledge of the minimum and maximum and an "informed guess" as to the modal value [Garvey, 2000] (see fig. 3.9 below).



$$f(x) = \frac{2(x-L)}{(M-L)(H-L)}, \quad L \le x < M$$
$$= \frac{2(H-x)}{(H-M)(H-L)}, \quad M \le x < H$$
$$= 0, \quad otherwise$$

Fig. 3.9: A triangular pdf with most likely estimate M, minimum estimate L and maximum estimate H.

The triangular distribution can characterize skewedness in the distribution and the mean and variance are given by,

$$\left.\begin{aligned} \mu &= \frac{(L+M+H)}{3} \\ \sigma^2 &= \frac{(L^2+M^2+H^2-LM-LH-MH)}{18} \end{aligned}\right\} \qquad (3.43)$$

These three values are specified by the subject matter expert (SME) to describe the subjective probabilities. For these reasons, the triangle distribution has often been called a "lack of knowledge" distribution [Garvey, 2000; Kotz and van Dorp, 2004].

In practice, it is often the case that SME's can only prescribe some upper and lower limits H^* and L^* with adequate confidence, but there exists extreme values H and L, outside of those limits and with finite probabilities. Such a scenario is

described in fig. 3.10 below. This is the usual scenario for expert elicitation in practice.

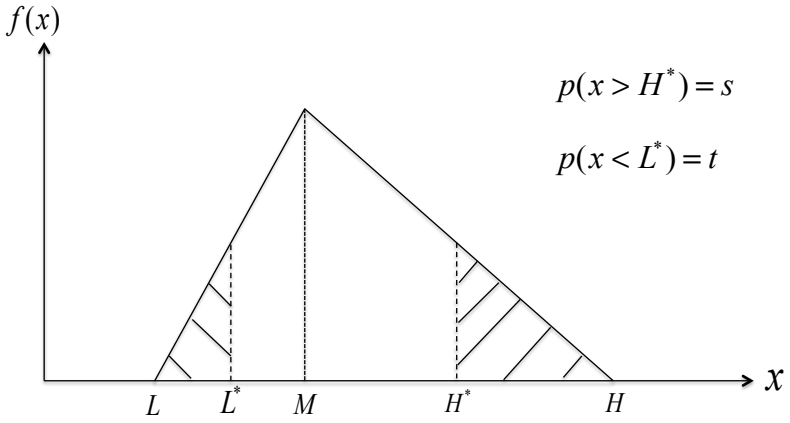


Fig. 3.10: A triangular pdf with limit values (H^*, L^*) specified by SME with probabilities of actual extreme values lying beyond those limits.

There is extensive literature on expert elicitation techniques [Young and Young, 1995; O'Hagan, 1998; Greenberg, 2013]. There is a large swath of literature in social science research dealing with expert opinion elicitation that also points to the lack of consensus on the superiority of any particular elicitation technique [O'Hagan *et al.*, 2006].

(ii) ***Aggregation/pooling of expert opinions*****:** There are multiple techniques for aggregating expert opinion [Cooke, 1991]. Any one of them could be employed for this purpose, but we advocate using the simple aggregation technique based on weights associated with each expert [Cooke 1991]. Goulet [Gaulet *et al.*, 2009] developed an R software package implementing Cooke's convex combination of expert opinions.

Regarding aggregation methods, mathematical compositions have been used intensively [Babuscia, 2012]. Cooke developed the Cooke Classical Model, while O'Hagan's work [O'Hagan *et al.*, 2006] describes different ways to exploit at maximum expert opinion using Bayesian analysis, elicitation processes, subjective probabilities, and expert's cooperation. Clemen [Clemen and Winkler, 1999] reviewed the principal techniques for expert aggregation: linear weighted

combination, logarithmic combination, Cooke generalization, Bayesian approach, and Mendel-Sheridan composition. Since there is not a clear winner, we advocate using the simplest aggregation rule that is applicable under the scenario.

Another alternative approach for expert opinion aggregation is the *Delphi method*. This method resolves the aggregation using an anonymous off-line discussion-based approach, including iterations. This requires generation of a relevant questionnaire and needs to be facilitated by a moderator [Ahn *et al.*, 2013].

(c) Estimation using data analytics for system design: This is a bottom-up, first principle approach to estimation of component complexity based on component characteristics and leveraging existing data. Hence, there is an inherent assumption about availability of historical and contextual data for similar components (or at least some reasonable estimates). This approach is based on data analytics applied to system design and development. Such approaches are becoming increasingly prominent with introduction of concepts like *additive manufacturing* and design by crowdsourcing [DARPA AVM, 2011]. In this approach, the complexity of a component generally scales with the amount of information required to define/replicate the component.

Application of this method involves developing a statistical model of the form, $\alpha = f(X)$ relating a component characteristics vector, X to component complexity, α. Complexity cannot be measured physically using a measuring device, but it is manifested through other observables like component development cost or effort, y.

In this approach, the first and probably the most important step is defining the component characteristics vector, X. Below, we suggest a list eight characteristics that typically impact the complexity of any given component in an engineering system development context.

1. **Measure of performance tolerance, x_1:** Components with extremely tight performance tolerance requirements tends to have increased complexity. As performance tolerance increases, component complexity tends to decrease.

2. **Measure of performance level, x_2:** A higher level of component performance introduces higher levels of complexity in components.

3. **Component *size* indicator, x_3:** Component with larger '*size*' tends to increase its complexity. Here, *size* indicators would mean different things for software components vis-à-vis hardware. Using a *size* indicator that is representative in context is important. There may be cases where a smaller size might reflect higher complexity (i.e., integrated circuits or chips).

4. **Number of coupled disciplines involved, x_4:** If engineering a component involves a number of disciplines (scientific or otherwise), its complexity tends to increase. For example, components that concurrently fulfill structural, fluidic and thermal functions tend to be more complex that components that have only structural requirements to satisfy.

5. **Number of variables and physical processes involved, x_5**: An increased number of different physical processes and associated variables tend to move component complexity upwards.

6. **Component reliability measure, x_6:** Components with high reliability often mandates higher complexity.

7. **Existing knowledge of operating principle, x_7:** This is an indicator in [0,1] where 1 indicates complete knowledge of the operating principle of the component. Existing knowledge about the operating principle reduces the component's complexity.

8. **Extent of reuse/heritage indicator, x_8:** This is an also an indicator in [0,1] where 1 indicates complete reuse of an existing component. Any extent of reuse of an existing component reduces its design/developmental complexity.

Please note that the above list of component characteristics impacting component complexity is only a suggested one and can be modified based on the application and context to a certain degree. We believe this is a promising list in specific case of engineered complex system design and development, but will require further refinement.

The methodology developed here leverages the method developed by Bearden [Bearden, 2000, 2004]. The component complexity measure uses a weighted, rank-based measure. For a given component, each of the component characteristics listed above are ranked based on a historical database for similar components. The procedure for computing the rank for ith component is described below:

$$Let,\ r_j^{(i)} = percent\ rank\ wrt\ variable\ x_j^{(i)}\ in\ the\ historical\ database$$

$$R^{(i)} = \left\{ \begin{array}{c} perf.\ tolerance\ rank \\ perf.\ level\ rank \\ component\ 'size'\ rank \\ \#\,coupled\ disciplines\ rank \\ \#variable\ involved\ rank \\ component\ reliability\ rank \\ existing\ knowledge\ rank \\ extent\ of\ reuse/heritage\ rank \end{array} \right\} = \left\{ \begin{array}{l} r_1^{(i)} \\ r_2^{(i)} \\ r_3^{(i)} \\ r_4^{(i)} \\ r_5^{(i)} \\ r_6^{(i)} \\ r_7^{(i)} \\ r_8^{(i)} \end{array} \right\}$$

In order to compute the individual ranks, let us first divide the factors into two groups: (i) LIH = Larger Is Higher – where larger value leads to higher component complexity (i.e., positive correlation with complexity), and (ii) SIH = Smaller Is Higher – where a smaller value maps to higher component complexity (i.e., negative correlation with complexity). For the given factors, here is how their ranks are computed,

$$\left.\begin{aligned} r_j^{(i)} &= \frac{size\{p \mid x_j(p) \le x_j^{(i)}\}}{n} \;\; if \;\; x_j \in LIH \\ &= \frac{size\{p \mid x_j(p) \ge x_j^{(i)}\}}{n} \;\; if \;\; x_j \in SIH \end{aligned}\right\} \qquad (3.44)$$

$$with \; LIH = \{2,3,..,6\}; \; SIH = \{1,7,8\}$$

where n = #instances of i^{th} component in the database. Please note that the size characteristic can belong to either category, depending on the application area.

Then we assign weights to each component characteristic and compute the component complexity value. To estimate the weights we have to build a statistical model linking component development cost/effort to the same set of component characteristics vector, X as shown below,

$$y^{(i)} = h(X^{(i)}) + \varepsilon \qquad (3.45)$$

The model described in eq. 3.45 is usually developed using *stepwise regression* technique [Mosteller and Tukey, 1977; Martinez and Martinez, 2007]. The exact functional form $h(.)$ is obtained using statistical/econometric modeling techniques and therefore dependent on available data. The polynomial model is usually adopted for its simplicity and ease of interpretation. The weights used for computing the component complexity are derived using the associated coefficients from the above model linking development effort and complexity charateristics vector:

$$w_j^{(i)} = \frac{representative \; coeff. \; of \; x_j^{(i)}}{\min[representive \; coeff. \; of \; x^{(i)}]} \qquad (3.46)$$

$$= weight \; associated \; to \; j^{th} \; factor \; for \; i^{th} \; component$$

The weighted, rank-based component complexity is computed by,

$$\alpha_i = \frac{\sum_{j=1}^{m} w_j^{(i)} r_j^{(i)}}{m} \qquad (3.47)$$

$$where\ m = size(R^{(i)}) = size\ of\ the\ X\ vector$$

where m = 8 since we have considered 8 component characteristics here.

Please note that the associated weights are always larger that one by definition, and therefore, the component complexity metric could also be higher than unity. This weighted measure assigns relative importance or contribution of each component characteristic towards system development cost/effort and help link component complexity to its development cost/effort in an implicit fashion.

There is also an alternative model that can be used to link the component characteristic vector, X to its complexity, α. This involves relating the component development cost/effort, y to component complexity, α, based on the historical data. Let us assume that they are related as,

$$y = g(\alpha) + \varsigma \qquad (3.48)$$

Using eq. 3.45 and 3.48, we can arrive at the functional form, assuming the functional inverse $g^{-1}(.)$ exists,

$$\alpha = \underbrace{g^{-1}h}_{(f)}(X) + \delta$$
$$= f(X) + \delta \qquad (3.49)$$

The above procedure is depicted pictorially in fig. 3.11 below.

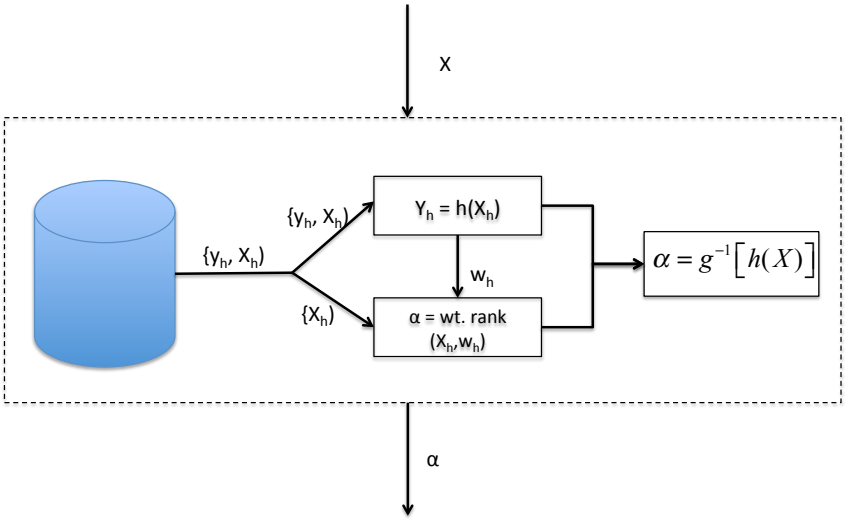


Fig. 3.11: Data analytics based, first principle approach for computing component complexity, α from component characteristics vector, X.

Assuming the functional form between y and α, as obtained by Bearden [Bearden 2004], $y = ae^{b\alpha}$, we can arrive at the following functional form linking component complexity, α to component characteristics vector, X:

$$\alpha = \underbrace{\frac{1}{b}\ln\left(\frac{1}{a}\right)}_{constant} + \frac{1}{b}\ln[h(X)] + \delta$$

$$= K + \frac{1}{b}\ln[h(X)] + \delta \qquad (3.50)$$

This is an alternative form for computing component complexity based on component characteristics using historical/contextual data. The suggested data driven estimation method needs to be matured over time by application to multitude of systems across different domains before it can achieve a level of fidelity and maturity for large-scale industrial applications.

Please note that the data analytics driven estimation of component complexity, α is still an estimate. We can assign an uncertainty band around this estimate to derive a probabilistic distribution for component complexity. The likely candidates are normal (e.g., Gaussian) or other symmetric distributions.

3.4 Estimation of Interface Complexity

In order to compute the term C_2 in the structural complexity metric, we need to define the interface complexity for each connection in the system connectivity graph, G. Interface complexity is estimated based on (i) complexities of the interfacing components, and (ii) characteristic properties related to the interface type. Therefore, for an interface of type *k*, interfacing the $(i,j)^{th}$ components of a system, the interface complexity can be represented as:

$$\beta_{i,j}^{(k)} = g(c^{(k)}, \alpha_i, \alpha_j) \qquad (3.51)$$

where α_i and α_j are the complexities of components *i* and *j*, respectively and $c^{(k)}$ is a constant based on the interface type.

In the context of engineered complex system, table 3.2 lists the primary interface types, including their sub-types [Tilstra, 2010; Tilstra *et al.,* 2013]. In order to be applicable across a large variety of systems, the interface types are extended from the primary interfaces to its more detailed sub-types [Tilstra, 2010]. This list of fine-grained interface sub-types is an extension of the general interactions presented in the contemporary DSM literature [Eppinger and Browning, 2012; Suh et al., 2010]. More details on application of this methodology can be found in chapter 5.

Table 3.2: Various primary interfaces and their sub-types.

Primary Interface types	Interface sub-types
Physical	{load transfer; translational; rotational; spatial; alignment; positional proximity}
Flow	{fluid flow; flow of solids; mixture flow; plasma flow}
Energy	{mechanical; thermal; hydraulic; elastic; pneumatic; electrical; magnetic; electromagnetic; acoustic; chemical; biological; human}
Information	{control signal; status signal; information processing}

Please note that all the sub-types listed above in table 3.2 may not be applicable in a specific scenario. The underlying idea is to use a consistent description of interfaces across systems that are being compared.

The relative complexity of the interfaces with respect to that of the components can be estimated looking at the relative effort associated with component vis-à-vis interface design. One of the essential function in systems engineering and project management for product development is to predict and track the relative effort for component design, interface control document definition and systems integration. This requires consistent book keeping of development efforts differentiating between component development and interface design.

It is crucial to express the interface complexities relative to the component complexities such that they are not *dimensionally mismatched*. Dimensionally, complexity of interface type *k* can be expressed in terms of component complexity as,

$$\beta^{(k)} = \frac{O(\alpha)}{c^{(k)}} \qquad (3.52)$$

Since it is also easier to express the interface development effort as a function of the component development efforts, a simple and probably the most practical functional form to express interface complexity appears to be the following:

$$\beta_{i,j}^{(k)} = \frac{\max(\alpha_i, \alpha_j)}{c^{(k)}} \qquad (3.53)$$

where $c^{(k)}$ characterizes the interface type *k* and is independent of component complexity. A large value of $c^{(k)}$ reduces the interface complexity. The obvious next question is how to estimate the interface type factor, $c^{(k)}$. Here, we utilize the available data on development cost/effort of interfacing components and that for the interface. Let us consider an interface type *k* with multiple instances of such interface types in a database.

Let $z_{(i,j)}^{k}$ being the interface development cost for interface type *k*, connecting components (i, j). If y_i and y_j represent the interfacing component development costs, we express the estimated interface factor for interface type *k* as,

$$c^{(k)} = \underset{\cup k}{mean}\left[\frac{\max(y_i, y_j)}{z_{(i,j)}^{k}}\right] \qquad (3.54)$$

Now, we can also utilize available data to develop a bottom-up, first principle approach to interface factor characterization. We again emphasize the importance of the bottom-up model for interface factor and we propose a data analytics driven approach to characterization of interface factor, similar to that of component complexity.

As the first step, we define interface characteristic vector, $X_{(k)}$ as,

$$X_{(k)}^{(i,j)} = \left\{\begin{array}{c} magnitude\ of\ 'entity'\ transfer \\ tolerance\ requirement\ indicator \\ knowledge\ of\ interface\ mechanism \\ \#\ disciplines\ involved \\ reliability\ requirement\ indicator \\ extent\ of\ reuse/standardization \end{array}\right\} = \left\{\begin{array}{c} x_1^{(i,j)} \\ x_2^{(i,j)} \\ x_3^{(i,j)} \\ x_4^{(i,j)} \\ x_5^{(i,j)} \\ x_6^{(i,j)} \end{array}\right\}$$

The impact of six interface characteristics on the interface complexity is as follows:

1. **Magnitude of 'entity' transfer, x_1:** Interfaces that transfer a large 'entity' of interest (e.g., force, energy, flow, signal) usually are more complex.

2. **Interface tolerance requirement, x_2:** Interfaces with low tolerance requirement often mandates higher interface complexity.

3. **Existing knowledge of interface mechanism, x_3:** This is an indicator in [0,1] where 1 indicates complete knowledge of the operating principle of the interface. Existing knowledge reduces the associated interface complexity.

4. **Number of disciplines involved, x_4:** If designing an interface involves a number of disciplines (scientific or otherwise), its complexity tends to increase. Interfaces that transfer both heat and electrical signals are more complex to develop that purely electrical interfaces.

5. **Interface reliability requirement, x_5:** Interfaces with high reliability requirement often mandate higher complexity.

6. **Extent of reuse/standardization indicator:** This is an also an indicator in [0,1] where 1 indicates complete reuse/standardization of an interface. Any extent of reuse of an existing interface reduces its complexity. Interface standardization is a major issue in industry and ISO, ANSI and IEEE standards are often extensively used to better manage interface complexity over time.

Please bear in mind that the above list of interface characteristics is only a suggested one and can be modified based on the context. For each instance of the interface type *k* in the historical database, we can substitute the ratio $\left[\frac{\max(y_i, y_j)}{z^k_{(i,j)}}\right]$ for $c_{(k)}$ and build a statistical model relating the interface characteristics vector, $X_{(k)}$ and $c_{(k)}$:

$$c_{(k)} = h(X_{(k)}) \qquad (3.55)$$

The model described in eq. 3.55 is developed using the *stepwise regression* technique [Mosteller and Tukey, 1977]. The exact functional form $h(.)$ is dependent on available data. The polynomial model is usually adopted for its simplicity and ease of interpretation. In case of component complexity being available as a

distribution and/or interface factor being defined as a distribution, the resulting interface complexity, β becomes a distribution.

The specification of interface complexity distribution may become rather involved and required application of algebra of random variable. Such an approach requires use of integral transforms, like the Mellin transform to derive the resulting probability distribution for interface complexity [Springer, 1979].

In cyber-physical systems, the interface complexities are usually much smaller than the component complexities with $c_{(k)}$ being much larger than unity.

This may not be the case in other systems though. For example, it is quite possible that interface complexity in novel biological systems might dominate the component complexities. It must be mentioned that other functional forms, like the multiplicative model, can be used to characterize the interface complexity. While maintaining the basic essence of eq. 3.50, we can write the multiplicative model for interface complexity as,

$$\beta_{i,j}^{(k)} = p_{(k)}\alpha_i\alpha_j \qquad (3.56)$$

where $p_{(k)}$ is a coefficient that has to be estimated. The multiplicative model has been used earlier for characterizing interfaces [Eppinger and Browning, 2012]. A similar data-driven methodology described above can be used to estimate the interface coefficient.

3.5 Analysis of the Structural Complexity Metric

We have described the details of individual terms of the structural complexity metric. Now, let us look at the dimensional analysis of the structural complexity measure and then a look at the interplay between the different components and the resulting *isocomplexity* surface.

3.5.1 Order analysis for the proposed Structural Complexity metric

Let us define the averaged component and interface complexities as follows:

$$\left.\begin{aligned}\frac{C_1}{n} &\equiv \bar{\alpha}\\ \frac{C_2}{m} &\equiv \bar{\beta}\end{aligned}\right\} \qquad (3.57)$$

where *n* and *m* represents the number of component and the number of interfaces respectively.

As described earlier, we can group system architectures broadly into three regimes (a) centralized architectures (e.g., *hypoenergetic* $C_3 < 1$), (b) tree and hierarchical architectures (e.g., *non-hypoenergetic* $1 < C_3 \leq 2$), and (c) distributed architectures (e.g., *hyperenergetic* $C_3 > 2$).

(a) For *centralized architectures*: The component complexity dominates in this regime and associated order can be described as follows:

$$C_1 = \bar{\alpha} O(n)$$
$$C_2 = \bar{\beta} O(n)$$
$$C_2 C_3 = \bar{\beta} O\left(\sqrt{n}\right)$$

The order of structural complexity metric in this regime is:

$$C = \bar{\alpha} O(n) + \bar{\beta} O(n^{1/2}) = O(n) \qquad (3.58)$$

Hence, the structural complexity is of the same order as the number of components in the system. A weighted sum of components in the system may yield a good approximation of structural complexity.

(b) For *hierarchical architectures*: This is a *balanced* architectural form where component and interaction structure contributions are of the same order.

$$C_1 = \bar{\alpha} O(n)$$
$$C_2 = \bar{\beta} O(n)$$
$$C_3 = O(1)$$
$$C_2 C_3 = \bar{\beta} O\left(n\right)$$

The order of structural complexity metric in this regime is:

$$C = \bar{\alpha} O(n) + \bar{\beta} O(n) = O(n) \qquad (3.59)$$

Even in this case, the structural complexity is of the same order as the number of components in the system, the difference being the relative contributions from different terms in the structural complexity measure. A weighted sum of components and interfaces in the system may yield a good approximation of structural complexity.

(c) For *distributed architectures*: In this regime, the effect from system interaction structure dominates over the component complexities as described below.

$$C_1 = \overline{\alpha} O(n)$$
$$C_2 = \overline{\beta} O(n^2)$$
$$C_3 = O(n^{0.5})$$
$$C_2 C_3 = \overline{\beta} O\left(n^{2.5}\right)$$

The order of structural complexity metric in this regime is:

$$C = \overline{\alpha} O(n) + \overline{\beta} O(n^{2.5}) = O(n^{2.5}) \qquad (3.60)$$

The structural complexity varies non-linearly with the number of compositional elements of the system in this regime. This system architectural regime is often associated with use of simpler component while the interconnectivity accounting for much of the improved performance and life-cycle properties.

A weighted sum of components and interfaces in the system is usually insufficient to yield a good approximation of structural complexity. The relative balance between component complexity and that arising from the interaction structure is an interesting decision problem and directly relates to the relationship between component engineering and system integration respectively.

Based on the above, one would predict that a new, component innovation-centric system will start off with a more centralized architecture initially, but additional requirements might mandate a departure towards a more hierarchical or distributed architecture. It would be interesting to investigate the evolution of different classes of systems (as part of a system archeology exercise) and see if the above assertion holds true in general.

3.5.2 Isocomplexity surface

Highly distributed architectures feature a high C_3 value and thereby imply high system integration effort. Note that C_3 is only a part of the total structural complexity as the total structural complexity is also dependent on component and interface complexities.

For example, assume a total *structural complexity budget* of C = 100. We can distribute this total complexity among its different components C_1, C_2 and C_3. A finite amount of overall structural complexity can be distributed across different components of the complexity metric. Given a constant level of overall complexity, topological complexity is contained to lower levels for a large swath (i.e., moderate to high) of component and interface complexities (see fig. 3.12 below).

An *isocomplexity* surface is one in which different instantiations of a system have the same total structural complexity but distributed differently within the system.

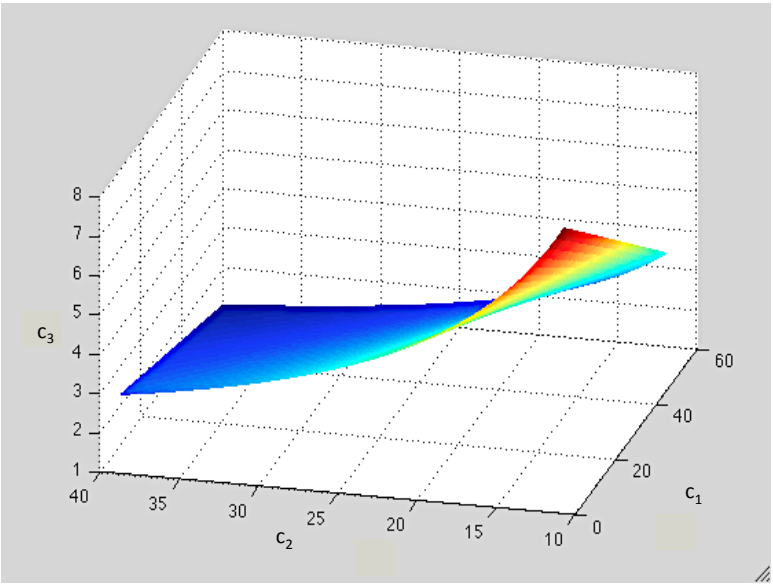


Fig. 3.12: Profile of isocomplexity surface for *n* = 20 components, assuming, c_1 in [10,60]; c_2 in [12,40] and Structural Complexity held constant at C = 100. Different value of C only shifts this profile.

Lower interface complexities allow for larger topological complexity in the system. There is an interesting tradeoff between (i) complex components and simple architecture, or (ii) simpler components and more complex architecture. Assuming we have both options open after considering other life-cycle considerations like *robustness* etc., the first option calls for excellence in component development and very high component reliability while the second option requires expertise in system architecting and integration.

This may often be a strategic decision to be made by the development organization, depending on its own capabilities and strengths. One of the main outcomes of this thesis is a call for more deliberate complexity management within large system/product development efforts. Part of the complexity management is the establishment of an overall complexity budget, C_b and the allocation of sub-budgets to component, C_{1b}, interface, C_{2b} and topological complexity, C_{3b}.

We will dive into this topic in more details in Chapter 6 of this thesis and link complexity budget to *value* of the system, which suggests a way to estimate/allocate a complexity budget based on the system value.

3.6 Sensitivity Analysis of the Structural Complexity Metric

Another aspect of complexity management is to perform sensitivity analysis for factors driving the structural complexity and we discuss this in this section.

Let us look at the sensitivity analysis for structural complexity, C. Consider *architecture 1* as the base architecture and *architecture 2* as the new architecture.

$$\textit{Architecture 1}, \quad C = C_1 + C_2 C_3$$
$$\textit{Architecture 2}, \quad C_* = C_1^* + C_2^* C_3^*$$

Assuming $C > C_*$, the sensitivity is expressed as,

$$Sensitivity = \frac{\Delta C}{C}$$
$$= \frac{(C_1 - C_1^*) + (C_2 C_3 - C_2^* C_3^*)}{C_1 + C_2 C_3}$$
$$\approx \frac{1 - \left(\frac{C_2^* C_3^*}{C_2 C_3} \right)}{1 + \left(\frac{C_1}{C_2 C_3} \right)} \quad assuming \left(\frac{C_1 - C_1^*}{C_2 C_3} \right) \to 0 \; in \; general \qquad (3.61)$$

For the base architecture, if the interface effects dominate over component complexity, then we have, $\left(\frac{C_1}{C_2 C_3} \right) \ll 1$ and we have,

$$Sensitivity \approx 1 - \left(\frac{C_2^*}{C_2} \right) \left(\frac{C_3^*}{C_3} \right) \qquad (3.62)$$

We observe from eq. 3.62 that component complexity does not play any role in sensitivity calculation in this scenario. Now, if interface effects and component complexities are comparable for the base architecture, then we have,

$$Sensitivity \approx R(1) \left[1 - \frac{C_2^* C_3^*}{C_2 C_3} \right] \; where, \; R(1) < \frac{1}{2} \qquad (3.63)$$

Comparing relations 3.62 and 3.63, we observe that as component complexity increases and becomes comparable or dominant over the interface effects, the resulting sensitivity to structural complexity decreases. Therefore, dominance of component complexity tends to reduce the sensitivity of structural complexity to architectural changes.

Looking at the above expressions for sensitivity of structural complexity, it appears that architectures dominated by component complexities (i.e., $C_1 > C_2 C_3$) with minimal changes in interface complexity (i.e., $C_2 \approx C_2^*$) helps dampen the sensitivity of structural complexity. Now let us look at the sensitivity calculations under specific circumstances.

3.6.1 Sensitivity to compositional elements

Sensitivity analysis with respect to individual component complexities is relatively straightforward as it follows the parametric sensitivity analysis procedure since the underlying system architecture remains unchanged. In this case, the underlying connectivity pattern remains the same and therefore, topological complexity, C_3 remains constant. Using the multiplicative model $\beta_{ij}^{(k)} = c^{(k)}\alpha_i\alpha_j$ for the interface complexity, the analytical expression is derived below.

$$C = C_1 + C_2C_3 = \sum_i \alpha_i + C_3\sum_i\sum_j \beta_{ij} = \sum_i \alpha_i + C_3\sum_i\sum_j c^{(k)}\alpha_i\alpha_j$$

$$\therefore \frac{\partial C}{\partial \alpha_i} = \frac{\partial C_1}{\partial \alpha_i} + C_2\frac{\partial C_3}{\partial \alpha_i} + C_3\frac{\partial C_2}{\partial \alpha_i} = \frac{\partial C_1}{\partial \alpha_i} + C_3\frac{\partial C_2}{\partial \alpha_i} \;;\; since\; \frac{\partial C_3}{\partial \alpha_i} = 0 \qquad (3.64)$$

Assuming longer range (beyond immediate neighbor) interaction sensitivities are negligible, we obtain

$$\frac{\partial C_1}{\partial \alpha_i} = 1 + \sum_{i\neq j}\frac{\partial \alpha_j}{\partial \alpha_i} = 1 + \sum_{i\sim j}\frac{\partial \alpha_j}{\partial \alpha_i} \qquad (3.65)$$

where $(i \sim j)$ indicates that they are connected to each other.

Assuming interface factor, $c^{(k)}$ being independent of changes in complexity of compositional elements, we obtain the following expression for sensitivity of interface complexity to changes in compositional elements:

$$\frac{\partial C_2}{\partial \alpha_i} = \sum_{i\sim j}\left[\frac{\partial c^{(k)}}{\partial \alpha_i}\left(\alpha_i\alpha_j\right) + c^{(k)}\frac{\partial}{\partial \alpha_i}\left(\alpha_i\alpha_j\right)\right]$$

$$= \sum_{i\sim j}\left[c^{(k)}\frac{\partial}{\partial \alpha_i}\left(\alpha_i\alpha_j\right)\right] = \sum_{i\sim j}\left[c^{(k)}\alpha_j\left(1 + \underbrace{\frac{\alpha_i}{\alpha_j}\left(\frac{\partial \alpha_j}{\partial \alpha_i}\right)}\right)\right] \qquad (3.66)$$

Notice the underlined normalized sensitivity term of immediate neighbor element complexity. In other words, the sensitivity of total interface complexity to component complexity also depends on all immediate neighboring components. Putting everything together, we obtain the final expression for sensitivity of structural complexity to compositional elements in the system:

$$\frac{\partial C}{\partial \alpha_i} = \left[1 + \sum_{i \sim j} \frac{\partial \alpha_j}{\partial \alpha_i} \right] + \gamma E(A) \left\{ \sum_{i \sim j} \left[c^{(k)} \alpha_j \left(1 + \frac{\alpha_i}{\alpha_j} \left(\frac{\partial \alpha_j}{\partial \alpha_i} \right) \right) \right] \right\}$$

$$= \left[1 + \sum_{i \sim j} \frac{\partial \alpha_j}{\partial \alpha_i} \right] + \underbrace{\frac{E(A)}{n}} \left\{ \sum_{i \sim j} \left[c^{(k)} \alpha_j \left(1 + \underbrace{\frac{\alpha_i}{\alpha_j} \left(\frac{\partial \alpha_j}{\partial \alpha_i} \right)} \right) \right] \right\} \qquad (3.67)$$

Notice that there is a term describing the architectural pattern involved, in addition to a normalized sensitivity term involving immediate neighbor element complexity (indicated by underlined terms in eq. 3.67).

3.6.2 Sensitivity to system architectural changes

Let us now look at the sensitivity of component deletion on the structural complexity metric. This has a combinatorial effect as the underlying system architecture is changed in addition to deletion of a single system component. Let us consider a system architecture be represented as a simple graph G with *n* components and *m* interactions and whose binary adjacency matrix is A.

Now the k^{th} component (i.e., k^{th} node of graph G) is removed from the system and this results in deletion of all interactions associated to this component. However, it is assumed that this component deletion still maintains a connected system (e.g., deletion does not lead to fragmentation of the overall system).

For simplicity, assume also that such deletions do not render the overall system totally dysfunctional (but performance might partially degrade) [Agte *et al.*, 2012]. In case of component removal leading to structural disintegration of the system (i.e., system fragments into multiple disconnected fragments), the sensitivity

for that component is set to a very large number, indicative of this behavior. We can also employ the same strategy for absolutely necessary components of the system (i.e., components that cannot be removed else the system becomes *dysfunctional*) from the system functionality stand point. We also assume that component deletion does not result in any re-distribution of component and interface complexities while maintaining at least limited functionality.

Please note that we are not imposing the multiplicative model for estimating interface complexity here. Imposing the multiplicative model would result in a slightly different mathematical expression, but essential characteristics remain the same.

Under these assumptions, we can express the difference in structural complexity due to removal of the k^{th} system component as below:

$$\Delta C = (C_1 - C_1^{(k)}) + C_2 C_3 \left[1 - \frac{C_2^{(k)} C_3^{(k)}}{C_2 C_3} \right]$$

$$= \underbrace{\left(\sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n-1} \alpha_i \right)}_{\substack{\text{effect of individual} \\ \text{component}}}$$

$$+ \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} \right) \frac{E(A)}{n} \left[1 - \underbrace{\left(1 - \frac{\sum_{i=1}^{n} \beta_{i,k} + \sum_{i=1}^{n} \beta_{k,i}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}} \right)}_{\text{effect of deleted interfaces}} \left(\frac{n}{n-1} \right) \underbrace{\left(\frac{E(A^{(k)})}{E(A)} \right)}_{\text{topological effect}} \right]$$

$$= \alpha_k + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} \right) \frac{E(A)}{n} \left[1 - \underbrace{\left(1 - \frac{\sum_{i=1}^{n} \beta_{i,k} + \sum_{i=1}^{n} \beta_{k,i}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}} \right)}_{\text{effect of deleted interfaces}} \left(\frac{n}{n-1} \right) \underbrace{\left(\frac{E(A^{(k)})}{E(A)} \right)}_{\text{topological effect}} \right] \qquad (3.68)$$

Looking at the different terms of the above expression, we observe that for any component, sensitivity to its deletion on structural complexity consists of three

sources: (i) complexity of the deleted component itself (α_k), (ii) complexities of the deleted direct interactions that were associated with the removed component, and (iii) re-structuring of the underlying system architecture due to removal of the k^{th} component.

The only impact that organization of system elements has on the sensitivity expression is through the changes in the topological complexity term after removal of any system component. If removal of any component makes the system structure less distributed than before, then the ratio of topological complexities in eq. 3.68 becomes smaller than unity.

3.7 Extension to System-of-Systems (SoS)

The system-of-systems (SoS) concept describes the large-scale integration of many independent, self-contained systems in order to satisfy a requirement by forming a large, integrated system (see fig. 3.13). System of systems problems are a collection of trans-domain networks of heterogeneous systems that are likely to exhibit operational and managerial independence, geographical distribution, and emergent and evolutionary behaviors that would not be apparent if the systems and their interactions are modeled separately [Maier, 1998].

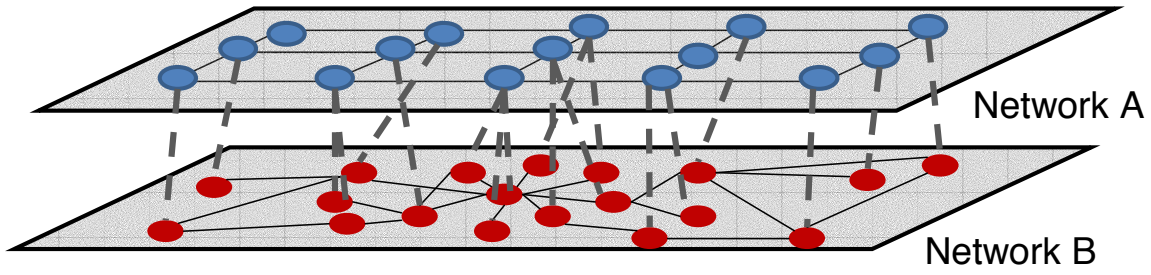


Fig. 3.13: System-of-Systems (SoS) represented as layered, multi-domain networks [Maier, 1998].

The proposed structural complexity metric can be extended to system-of-systems (SoS). The mathematical form is quite similar to that observed in computing the structural complexity of modular systems with individual systems acting as

'*modules*'. Here, the system of systems could be represented as a layered one with each layer representing one individual system.

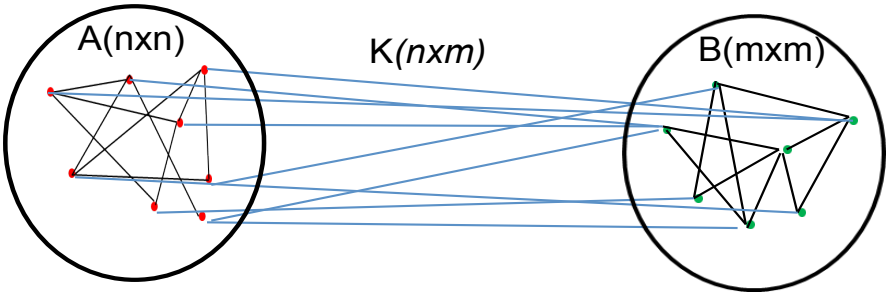


Fig. 3.13: 2D network representation of the SoS with *A* and *B* being their adjacency matrices.

Here, network *A* has *n* components or nodes, while network *B* is characterized by *m* components. The inter-system links are captured in the rectangular, domain-mapping matrix *K* (see fig. 3.13). The adjacency matrix of the resultant system-of-systems can be written as,

$$\Lambda = \begin{bmatrix} A & K \\ K^T & B \end{bmatrix}_{(n+m)*(n+m)} = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} + \begin{bmatrix} 0 & K \\ K^T & 0 \end{bmatrix}$$

From the pinching inequality, we have

$$E(A) + E(B) \leq E(\Lambda)$$

Let us write,

$$E(\Lambda) = E(A) + E(B) + \Delta$$

The additional graph or matrix energy component, Δ originates from the inter-system connectivity structure. The resultant structural complexity of the system-of-systems can be expressed as,

$$
\begin{aligned}
C &= \left(\sum_{i=1}^{n} \alpha_i^A + \sum_{i=1}^{m} \alpha_i^B \right) + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{i,j}^A + \sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{i,j}^B + \sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{i,j}^{K_1} + \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{i,j}^{K_2} \right) \gamma \left(E(A) + E(B) + \Delta \right) \\
&= \left(\sum_{i=1}^{n} \alpha_i^A + \sum_{i=1}^{m} \alpha_i^B \right) + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{i,j}^A + \sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{i,j}^B + \sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{i,j}^{K_1} + \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{i,j}^{K_2} \right) \gamma [E(A) + E(B)] \\
&\quad + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{i,j}^A + \sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{i,j}^B + \sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{i,j}^{K_1} + \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{i,j}^{K_2} \right) \gamma \Delta \\
&= \left(\sum_{i=1}^{n} \alpha_i^A + \sum_{i=1}^{m} \alpha_i^B \right) + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{i,j}^A + \sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{i,j}^B \right) \left[\frac{E(A) + E(B)}{N} \right], \ \textit{since } \gamma = \frac{1}{n+m} = \frac{1}{N} \\
&\quad + \left(\sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{i,j}^{K_1} + \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{i,j}^{K_2} \right) \left[\frac{E(A) + E(B)}{N} \right] \\
&\quad + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{i,j}^A + \sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{i,j}^B + \sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{i,j}^{K_1} + \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{i,j}^{K_2} \right) \frac{\Delta}{N} \Bigg\} \quad \textit{(integrative complexity)} \qquad (3.69)
\end{aligned}
$$

where α represents component complexities of comprising systems and β stands for corresponding interface complexities.

Notice that there are inter-system interfaces, in addition to the usual intra-system interfaces. From the above expression, we observe that the structural complexity of the System-of-Systems (SoS) has an additional component, in addition to contributions from the individual systems. This is termed as *integrative complexity* and this drives the structural complexity of the SoS upwards (see eq. 3.69).

The *integrative complexity* arises due to two factors: (i) inter-system interfaces; and (ii) connectivity structure of the inter-system interfaces, characterized by Δ in eq. 3.69. This aspect can be shown clearly by defining the following quantities, and recasting the above expression for structural complexity of the SoS,

$$\left.\begin{aligned} C_2^{add} &\equiv \sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{i,j}^{A} + \sum_{i=1}^{m}\sum_{j=1}^{m}\beta_{i,j}^{B} \\ C_2^{\text{int}} &\equiv \sum_{i=1}^{n}\sum_{j=1}^{m}\beta_{i,j}^{K_1} + \sum_{i=1}^{m}\sum_{j=1}^{n}\beta_{i,j}^{K_2} \\ C_2^{eff} &\equiv c_2^{add} + c_2^{\text{int}} \end{aligned}\right\} \qquad (3.70)$$

Using the definitions in eq. 3.70 above, we can express the structural complexity eq. 3.69 as,

$$\begin{aligned} C &= \left(\sum_{i=1}^{n}\alpha_i^A + \sum_{i=1}^{m}\alpha_i^B\right) + \left(C_2^{add} + C_2^{\text{int}}\right)\left[\frac{E(A)+E(B)+\Delta}{N}\right] \\ &= \left(\sum_{i=1}^{n}\alpha_i^A + \sum_{i=1}^{m}\alpha_i^B\right) + \left(C_2^{add} + C_2^{\text{int}}\right)\left[\frac{E(A)+E(B)}{N}\right] + \underbrace{\left(C_2^{add} + C_2^{\text{int}}\right)}_{c_2^{eff}}\left[\frac{\Delta}{N}\right] \\ &= \left(\sum_{i=1}^{n}\alpha_i^A + \sum_{i=1}^{m}\alpha_i^B\right) + \left(C_2^{add}\right)\left[\frac{E(A)+E(B)}{N}\right] + \underbrace{\left(C_2^{\text{int}}\right)\left[\frac{E(A)+E(B)}{N}\right] + \left(C_2^{eff}\right)\left[\frac{\Delta}{N}\right]}_{\textit{integrative complexity}} \end{aligned} \qquad (3.71)$$

In the above expression, notice the factors C_2^{int} and Δ that leads to the *integrative complexity*. Also, more simplistically, we can express eq. 3.71 as,

$$\begin{aligned} C &= \left(\sum_{i=1}^{n}\alpha_i^A + \sum_{i=1}^{m}\alpha_i^B\right) + \left(C_2^{eff}\right)\left[\frac{E(A)+E(B)+\Delta}{N}\right] \\ &= \left(\sum_{i=1}^{n}\alpha_i^A + \sum_{i=1}^{m}\alpha_i^B\right) + \left(C_2^{eff}\right)\left[\frac{E(A)+E(B)}{N}\right]\left\{1 + \frac{\Delta}{E(A)+E(B)}\right\} \end{aligned} \qquad (3.72)$$

Notice that the interface complexity term C_2^{eff} is bolstered by the additional complexity of inter-system interfaces C_2^{int} and their topological counterpart Δ. A practical challenge here is the characterization of the interface between individual systems, C_2^{int}. This poses significant challenge and uncertainty in estimation of associated complexity and remains an open research question that needs to be addressed in future.

The above approach can be applied to different scenarios where one layer could be the technical architecture while the other network layer refers to the system development organization [Pasqual and de Weck, 2012; Sinha *et al.*, 2012]. A similar multi-layer approach can be applied to other systems like healthcare, transportation system architecture, etc. Please note that in each application scenario, the implication and characterization of components and interfaces could be vastly different.

Chapter Summary

This chapter lays the foundation for the complexity quantification framework and the resulting metric developed in this thesis. The proposed structural complexity metric involved component and interface complexities, in addition to topological complexity and can be written as, $C = C_1 + C_2C_3$ following the Hamiltonian form (see appendix A for details).

We have developed a measure for capturing the topological complexity of a graph/network structure representing the physical architecture of the system. It is a global measure that does not require anything more than just the existence of connections between the components of the system. Important properties of the proposed topological complexity measure were discussed with classification of connectivity structures according to the proposed measure.

Lower and upper bounds were developed for the matrix or graph energy of the system's adjacency matrix. It was shown that the topological complexity measure satisfies the Weyuker's criteria. We also discussed about the dual nature of graph energy and matrix reconstructability problem as well as the relationship between graph energy and probabilistic resilience of networks, the details of which can be found in appendices E and H respectively. The transition from *non-hyperenergetic* to *hyperenergetic* regimes is characterized by the *P* point that heralds the regime of *distributed* networks. Extensive simulations indicated that the state of network at the *P* point provides an acceptable level of network resilience against nodal failure.

Different approaches for estimating component complexity, depending on the availability of prior knowledge and data, were discussed. An estimation model for interface complexity was proposed as a function of the complexities of interfacing components and the type of interface.

We also performed an order analysis for structural complexity across different regimes of the underlying system architecture and demonstrated that structural complexity could be of the order of $n^{2.5}$ for distributed architecture. Since the structural complexity metric captured three constituent elements of complexity emerging from different aspects of a system, understanding of their relative contribution is important.

The iso-complexity surface involving these three dimensions of structural complexity indicated aspects and implications of different trade-offs in the context of engineered complex system design and development. Finally we investigated the sensitivity of structural complexity measure under few different scenarios that indicated the importance of topological complexity.

In addition to providing a foundational basis for this thesis, this chapter introduces interesting questions and future research directions. As we open our eyes to the increasing importance of networks and their contribution towards the design of engineered complex systems, an interesting question is about leveraging the *strength of networks* to provide improved functionality and better life-cycle properties. Do distributed systems based on simpler components and higher topological complexity meet the goals of future systems? What does this imply for the typical system development expertise that has largely been based in effective use of reductionism (e.g., designing systems with complex components arranged in a relatively simple architectural pattern)?

The complexity *vis-a-vis* performance trade-off provides another interesting area of study. It appears that increasing system complexity usually enhances its performance but the curve tapers down beyond a certain level of complexity. Assuming that there is no precipitous complexity, what increased complexity does buy is enhanced performance, but the upside diminishes beyond a certain level of complexity and any further complexity does not provide additional performance.

This research suggests a relationship of diminishing return after topological complexity level increases beyond the *P* point. Any guideline on characterizing the level of complexity beyond which performance gains saturate, if any, would have enormous practical implications for system design and development.

Subsequently, we developed the analytical expression for structural complexity metric for system-of-systems (SoS). They have an additional contribution to complexity, defined as *integrative complexity*, in addition to contributions from the individual intra-system complexities. The integrative complexity terms was shown to arise from the inter-system interfaces and the inter-system connectivity structure.

References:

Bernstein D. S., "Matrix Mathematics: Theory, Facts, and Formulas", Princeton University Press, Second Edition (2009), ISBN: 9781400833344.

Chung, F. and Lu, L., "complex graphs and networks", CBMS volume 7 2004.

Chung, F. "Spectral Graph Theory", American Mathematical Society, 1st edition, 1997.

Cvetkovic´ D., Doob M., Sachs H., "*Spectra of Graphs – Theory and Application",* Academic, New York, 1980.

Fabrikant A., Koutsoupias E., Papadimitriou C., "Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet", *Automata, Languages and Programming*, Vol. 2380, January 2002.

Holtta, K., Suh, E.S. and de Weck, O. (2005). "Tradeoff Between Modularity and Performance for Engineered Systems and Products", ICED05, Melbourne, Australia.

Gutman, I., "The energy of a graph", Ber. Math.-Stat. Sekt. Forschungszent. Graz 103 (1978), 1–22.

Gutman, I., Soldatovic, T. and Vidovic, D., "The energy of a graph and its size dependence. A Monte Carlo approach", Chemical Physics Letters 297 (1998), 428–432.

Koolen, J. and Moulton, V., "Maximal energy graphs" Advances in Appld. Math., 26 (2001), 47–52.

Koolen, J., Moulton, V. and Gutman, I., "Improving the McClelland inequality for total p-electron energy" Chemical Physics Letters 320 (2000) 213–216.

Kharaghani H., Tayfeh-Rezaie B., "On the energy of (0; 1)-matrices" Lin. Algebra Appl. 429 (2008) 2046-2051.

Horn and Johnson, "Topics in Matrix Analysis", Cambridge Press, 1994.

Klau G.W. and Weiskircher R., “Robustness and Resilience", chapter in "Network Analysis: Methodological Foundations", Ulrik Brandes and Thomas Erlebach (eds.), Lecture Notes in Computer Science, Vol. 3418, Springer, Berlin, 2005, pp.417-437.

Li X., Shi Y., Gutman I., “Graph Energy”, Springer (2012), ISBN: 978-1-4614-4219-6.

Liu, D., Wang, H., and Van Mieghem, P.,”Spectral perturbation and reconstructability of complex networks”, Physical Review E 81, 016101, 2010.

Najjar W. and Gaudiot J.L., “Network Resilience: **A** Measure of Network Fault Tolerance”, IEEE transactions on computers, vol. 39. no. 2., pp 174-181, 1990.

Van Mieghem P., “Graph Spectra for Complex Networks”, Cambridge University Press, ISBN 978-0-521-19458-7, 2011.

Nikiforov V., “The energy of graphs and matrices”, J. Math. Anal. Appl. 326 (2007), 1472–1475.

Newman, M. E. J., Barabasi, A. L., and Watts, D. J., “The Structure and Dynamics of Networks” Princeton University Press, Princeton (2003).

Pishkunov N., “Differential and Integral Calculus”, CBS Publishers & Distributors, 1996, ISBN: 8123904924.

Wagner S., “Energy bounds for graphs with fixed cyclomatic number”, MATCH Commun. Math. Comput. Chem. **68**, 661–674 (2012).

Fabrikant A., Koutsoupias E., Papadimitriou C., “Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet”, *Automata, Languages and Programming*, Vol. 2380, January 2002.

Liu, D., Wang, H., and Van Mieghem, P.,”Spectral perturbation and reconstructability of complex networks”, Physical Review E 81, 016101, 2010.

Van Mieghem P., “Graph Spectra for Complex Networks”, Cambridge University Press, ISBN 978-0-521-19458-7, 2011.

Kinsner W.,”System Complexity and Its Measures: How Complex Is Complex”, Advances in Cognitive Informatics and Cognitive Computing Studies in Computational Intelligence, Vol 323, pp 265-295, 2010.

Candès E.J. and T. Tao. “The power of convex relaxation: Near-optimal matrix completion”, *IEEE Trans. Inform. Theory*, 56(5), 2053-2080, 2009.

Candes E.J. and Recht B., “Exact Matrix Completion via Convex Optimization”, Found. of Comput. Math., 2717-772, 2009.

Garvey P.R., “Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective”, CRC Press (2000), ISBN-10: 0824789660.

Kotz S., van Dorp J.R., “Beyond Beta: Other Continuous Families Of Distributions With Bounded Support And Applications”, World Scientific, 2004, ISBN-10: 9812561153.

Babuscia A., “Statistical Risk Estimation for Communication System Design”, PhD thesis, MIT, 2012.

Cooke R.M., “Experts in Uncertainty: Opinion and Subjective Probability in Science”, Oxford University Press, 1991.

Goulet V., Jacques M. and Pigeon M., "Modeling without data using expert opinion", The R Journal, 1:31–36, 2009.

O'Hagan A., "Eliciting expert beliefs in substantial practical applications", Journal of the Royal Statistics Society, 47:21–35, 1998.

O'Hagan A., Buck C., Daneshkhah A., Eiser R. and Garthwaite P., "Uncertain Judgements: Eliciting Experts Probabilities", Lavoisier Libraire, 2006.

Bearden D.A., "Evolution of complexity and cost for Planetary Missions throughout the development lifecycle", IEEE Aerospace Conference, 2012.

Bearden D.A., "A complexity-based risk assessment of low-cost planetary missions: when is a mission too fast and too cheap? ", 4th IAA international conference on low-cost planetary missions, May 2-5, 2000.

Bearden D.A., "Complexity based cost estimating relationships for space systems", IEEE Aerospace Conference, Volume 6, 2004.

Greenberg M., "A Step-Wise Approach to Elicit Triangular Distributions", ICEAA Professional Development & Training Workshop June 18-21, 2013.

Young, D. C., and Young P.H., "A Generalized Probability Distribution for Cost/Schedule Uncertainty in Risk Assessment", Proceedings of the Society for Computer Simulation, 1995.

Clemen R.T and Winkler R.L., "Combining probability distributions from experts in risk analysis", Society for Risk Analysis, 19:187–203, 1999.

Martinez W.L. and Martinez A.R., "Computational Statistics Handbook with MATLAB", 2nd edition, CRC Press (2007), ISBN-10: 1584885661.

Mosteller F. and Tukey J.W., "Data Analysis and Regression", Addison-Wesley Publishing Company, Don Mills (1977).

DARPA Adaptive Vehicle Make (AVM),"http://www.darpa.mil/Our_Work/TTO/ Programs/Adaptive _Vehicle_Make__(AVM).aspx", 2011.

Ahn J., de Weck O.L., Steele M., "Credibility Assessment of Models and Simulations Based on NASA's Models and Simulation Standard Using the Delphi Method", Systems Engineering, 2013.

Eppinger S.D., Browning T.R., "Design Structure Matrix Methods and Applications", MIT Ptress, 2012, ISBN-10: 0262017520.

Steward D., "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management 28 (3), pp 71-74, 1981.

Ulrich, K.; Eppinger, S., "Product Design and Development", New York: McGraw-Hill, 1995.

Sinha K., de Weck O.L., "Structural Complexity Metric for Engineered Complex Systems and its Application", *14th International DSM Conference*, 2012.

Denman J., Sinha K., de Weck O.L., "Technology Insertion in Turbofan Engine and assessment of Architectural Complexity", *13th International DSM Conference*, Cambridge, MA, September 14-15, 2011.

Sinha K., de Weck O., "A network-based structural complexity metric for engineered complex systems", IEEE Systems Conference (SysCon), pp 426-430, 2013.

Sinha K., de Weck O., "Structural Complexity quantification for engineered complex systems and implications on system architecture and design", ASME International Design Engineering Technical Conference, 2013.

Jiang X., Yao Y., Liu H., Guibas L., "Compressive Network Analysis", IEEE Transactions on Automatic Control. To appear. 2013.

Tilstra A.H., "Representing Product Architecture and Analyzing Evolvable Design Characteristics", PhD thesis, University of Texas, 2010.

Tilstra A.H., Backlund P., Seepersad C.C and Wood K., "Principles for Designing Products with Flexibility for Future Evolution", DETC2006-99583, 2013.

Suh E.S., Furst M.R., Mihalyov K.J. de Weck O.L., "Technology infusion for complex systems: A framework and case study", Systems Engineering, 13: 186–203, 2010.

Springer M.D., "The Algebra of Random Variables", John Wiley & Sons, 1979, ISBN-10: 0471014060.

Giesa T., Spivak D., Buehler M.J., "Category theory based solution for the building block replacement problem in materials design," *Advanced Engineering Materials*, 2012.

de Weck O.L., "Simulated Annealing: A Basic Introduction", Course Notes 16.888/ ESD 77, Massachusetts Institute of Technology, 2010.

Jilla C.D., Miller D.W., "A Multi-Objective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems", Journal of Spacecraft and Rockets, Vol. 41, No. 1, 2004.

Pinto A., Becz S. and Reeve H.M., "Correct-by-Construction Design of Aircraft Electric Power Systems", AIAA ATIO/ISSMO Conference, 2010.

Siddiqi A., de Weck O.L.,"Modeling Methods and Conceptual Design Principles for Reconfigurable Systems", *Journal of Mechanical Design*, 139, 101102, 2008.

Frey D., Palladino J., Sullivan J., Atherton M.,"Part count and design of robust systems", Systems Engineering, Volume 10, Issue 3, pages 203–221, 2007.

Whitney D.E., Dong Q., Judson J., Mascoli G., "Introducing knowledge-based engineering into an interconnected product development process", Proceedings of the 1999 ASME Design Engineering Technical Conferences, 1999.

Maier, M.W., "Architecting Principles for System of Systems". *Systems Engineering* **1** (4): 267–284, 1998.

Pasqual M.C., de Weck O.L.,"Multilayer Network Model for Analysis and Management of Change Propagation", *Research in Engineering Design*, Special Issue on Engineering Change, June 2012.

Sinha K., James, D. and de Weck O.L, "Interplay between Product Architecture and Organizational Structure", 14th International Dependency and Structure Modelling Conference, 2012.

Matti J Kinnunen, "Complexity Measures for System Architecture Models", 2006 (MIT Thesis).

Chapter 4

Empirical Validation

Although defining measures of complexity has been a recurring theme in variety of academic literature, a vast majority has been of theoretical in nature and less than one-fifth of the studies even attempted to provide some degree of objective quantification of complexity [Tang and Salminen, 2001**]**. This is understandable as large system development projects are rare and definitely not repeatable, making empirical/experimental studies hard to perform. This has posed a significant roadblock to widespread adaptation of generic complexity quantification methods exist for engineered systems.

While formulating the structural complexity metric in chapter 3, we performed formal mathematical consistency check and verification using Weyuker's criteria. That only looked at the mathematical validity or the *construct validity* of the proposed metric as a complexity measure. But in order to establish the proposed metric as a valid and useful measure of structural complexity, both mathematical verification and empirical validation based on real-world applications are necessary.

The first obstacle for empirical validation is the inability to directly *measure* complexity. Therefore we have to depend on the indirect measures or well-accepted manifestation of complexity in terms of other system observables. The most visible of these system level observables is the system development cost/effort.

In this thesis, existence of a strong correlation between structural complexity and system development cost/effort, after accounting for confounding effects, is construed as empirical validation of the proposed complexity metric. The underlying theme is that the structural complexity manifests itself through the system development cost/effort. Below, we state the primary hypothesis relating

structural complexity to system development cost/effort and the rest of this chapter is aligned with this overarching hypothesis.

Hypothesis: *System development cost/effort correlates super-linearly with Structural Complexity* (see figure 4.1).

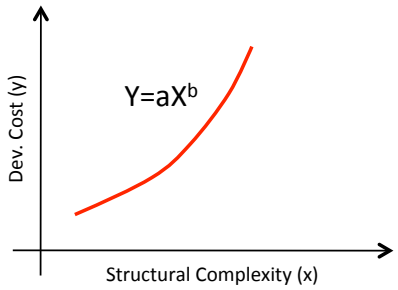


Fig. 4.1: Predicted super-linear growth in development cost with increasing structural complexity.

We propose a single variable model using structural complexity, as a predictor of cost/effort. The following functional form is chosen for simplicity, and the empirical evidence supporting this form [Garvey, 2000; Bashir and Thomson, 2001]:

$$Y = aX^b \qquad (4.1)$$

where, Y = estimated design cost/effort; X = structural complexity. The parameters $\{a,b\}$ are estimated by the least squares method. The parameters vary from one data set to another.

We start with illustrating supporting evidence from the literature, followed by an experimental validation using natural experiments involving experimental subjects building simple ball and stick models. Using the experimental data, we perform rigorous statistical analysis and build a model relating structural complexity to development effort. The methodology based on pseudo-jackknifing technique [Mosteller and Tukey, 1977] applies to situations where the number of observations is limited and increasing sample size is a *costly* option. We also demonstrate the hierarchical nature of the process of model building where fundamental observations are made at the level of individuals while the final model is for the group of individuals under study.

We subsequently discuss how and why observations made in experiments involving simple ball and stick models can help us understand real-world engineered complex systems development and discuss the generalizability aspect of this empirical validation.

Finally, we comment on the parameters {a,b} in the above functional form and discuss how they relate structural complexity to *perceptive complexity* or *complicatedness*. *Complicatedness* is an observer-dependent property that characterizes an actor's/observer's ability to unravel, understand and manage the system under consideration. In contrast, *complexity* is an inherent system property and a complex system may display different degrees of *complicatedness* depending on the observer. We can think of *complicatedness* as a conduit through which *complexity* manifests itself at the level of system-level observables like the *system development cost or effort* [Tang and Salminen, 2001].

4.1 Empirical Evidence

We begin with a couple of instances of empirical evidence from literature in support of the stated hypothesis. They represent simpler systems (e.g., family of electrical drills) at one end [Wood et al. 2001] and highly complex satellite systems at the other end [Wertz and Larson, 1996, Larson and Wertz, 1999, DARPA Report 2011]. The development costs and system architectural information were taken from the existing literature [Wood et al. 2001, DARPA Report 2011].

The development costs and structural complexities are normalized with respect to the respective minimum values within that product category. For example, the development costs and structural complexities of each satellite are each divided by the respective minimum values among the set of satellite programs considered. In this case, Orsted satellite had the minimum structural complexity and minimum development cost. Hence Orsted is mapped to point (1,1) on the normalized complexity vs. development cost plot (see fig. 4.3(a)). This was done to homogenize the representation since the reported development costs were masked using normalization techniques. The system architectural details of three satellites

can be found in the literature [Wertz and Larson, 1996, Larson and Wertz, 1999, DARPA Report 2011]. The versatility (i.e., different mission profiles and functional capabilities) and availability of data were the primary reason for choosing this set of satellite systems (see appendix J for further details).

In all cases, we assumed component complexities to have triangular distribution with most likely estimated being the point estimates (see fig. 4.2 below).

In all satellite examples, the inequality $(b-\alpha_m)>(\alpha_m-a)$ or $(a+b)>2\alpha_m$ was ensured while choosing the most optimistic and pessimistic values randomly. We assumed heavy-tailed or right-skewed distributions in satellite examples. The most optimistic values *a* and the most pessimistic values *b* were set based on the ranking of point estimates of component complexities α_m. We also assumed b/α_m as 2 and a/α_m as 0.85 for the most complex component and the values were varied based on the relative ranking of the component in terms of its complexity.

For all electric drill examples, we assumed symmetric triangular distribution with most optimistic and pessimistic estimates being ± 5 percent from the most probable value. Hence, we have b/α_m as 1.05, a/α_m as 0.95 and $(a+b)=2\alpha_m$.

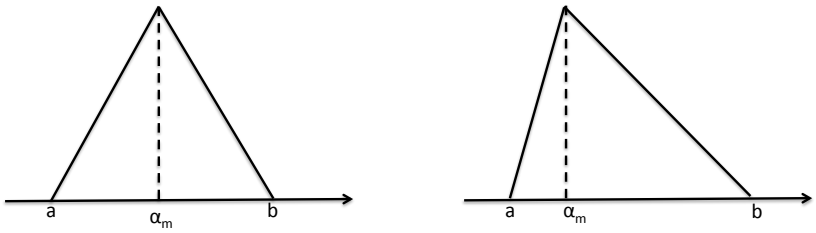


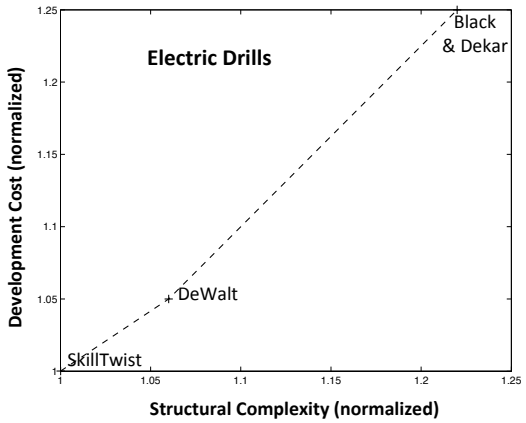
Fig. 4.2: (a) Symmetric triangular pdf of component complexities used in electric drill example; (b) right-skewed triangular pdf of component complexities [Garvey 2000] in case satellites.

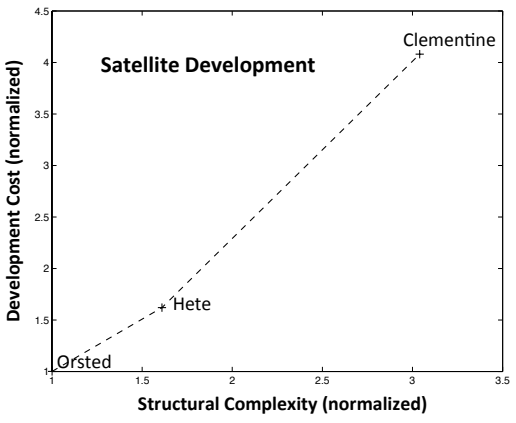
For the satellite examples, we assume the most likely component complexity, α_m to be one for all components with b=2 and a=0.85. Similarly for the electric drill examples, we use α_m =1 for all components with b=1.05 and a=0.95. We sample the component complexities from the triangular distribution defined by the three parameters and computed the structural complexity for each sample. The procedure outlined in chapter 3 was followed for computing the distribution of structural

complexities in all cases. Within each group of products, the structural complexities are normalized as described before. Additional details about the procedure can be found in appendix J.

We plot the mean structural complexity (normalized) as the representative number (see fig. 4.3(a)) for the group of electric drills and satellites.

Subsequently, we present another evidence from a summer class [2012] at MIT (ESD 39, 2012) that involved building toy vehicles using *LEGO™*. In this case, subjects organized in teams, were made to build three different kinds of cars as shown below using LEGO™ blocks. After the building process, we looked at the average number of defects for each car type and found that the number of defects increases with structural complexity in a super-linear fashion (see fig. 4.3(b)). The structural complexities for the LEGO™ car variants were computed assuming all component and interface to be of unit complexity. In this scenario, the development effort would consist of fixing any defect that might exist and therefore the number of defects (that needs to be fixed) serves as a proxy for the development effort. Please note that no normalization was carried out in this example. In this case, 3 teams had to build different variants of *LEGO™* vehicles as shown in fig. 4.3(b):





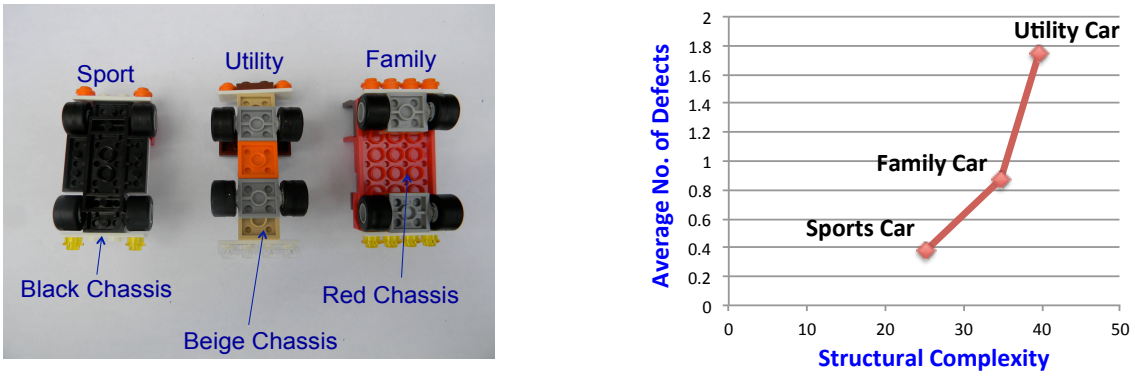


Fig. 4.3: (a) Empirical evidences of super-linear growth in development cost with increasing structural complexity from the literature (after normalization) for a class of electric drills and satellites, (b) Empirical evidence from building of toy vehicles using LEGO™ blocks showing super-linear relationship between average number of defects in building categories of LEGO™ cars (averaged over 3 teams for each car type) and structural complexity.

In all cases shown above, if we fit a power law as per our hypothesis, we obtain very strong correlation (with R^2 as high as 0.98). The parameters of the model {a,b} depend on the nature and category of systems.

4.2 Empirical Validation using Natural Experiments

Apart from such empirical evidence, we concentrated on conducting experiments with human subjects to see if we observe a similar behavior. The apparent behavior with three data points for each group of systems (i.e., screwdrivers, LEGO™ cars and Satellites) is consistent with our hypothesis, but to obtain statistically significant results, we conducted simple experiments with a group of nearly homogeneous subjects, using simple ball and stick models as described in the following section. Although the experiments were clearly defined in terms of the final goal, there were no additional controls in working details and participants could choose any method or strategies in order to achieve the final goal of building the skeleton molecular structure correctly. In this sense, these experiments were closer to *natural* experiments.

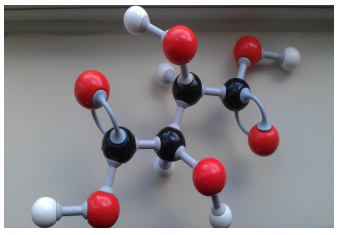
In this section and throughout the chapter, we focus on explicit empirical validation of the proposed structural complexity metric by exploring its correlation

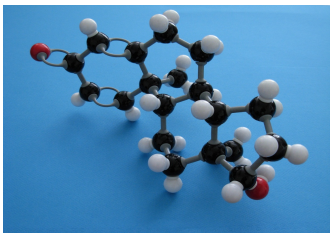
with system development time or effort. As has been highlighted elsewhere [Tang and Salminen, 2001; Bashir and Thomson, 1999], there is not much publicly available data to validate our hypothesis for large, real-world complex engineered systems. Given the lack of available data, we choose to perform simple natural experiments related to assembly of simpler structural models by human subjects. We perform an experiment with a molecular modeling kit from Prentice Hall [Prentice Hall Molecular toolkit, 1997], used in chemistry for constructing structure of organic molecules. The atoms are the components, and the bonds between them are the interfaces. Test subjects were required to correctly assemble structures given this molecular kit and a 2D picture of the structure to be built. The order of molecules was randomized for each test subject and for each molecule one would start with the entire, fully dissembled kit.

4.2.1 Structural Complexity of ball and stick models

We picked a set of 12 ball and stick structures to be built by the subjects. They were chosen such that they spanned a reasonable spectrum of structural complexities while the expected build time is not too high. This was done keeping in view of the availability of subjects for successfully conducting the experiments. All the ball and stick models were based on molecular structures that could be built from the molecular toolkit.

A sample of molecular structures used is shown in fig. 4.4 below.





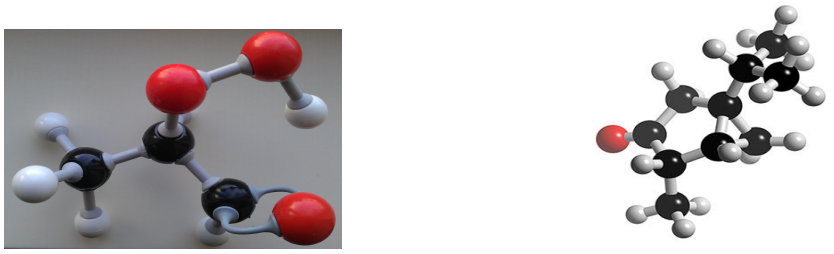


Fig. 4.4: Representative molecular structures build using a molecular kit – (counter-clockwise from top left corner): ID 8 with 16 atoms (see table 4.1 below); ID 3 with 12 atoms; ID 10 with 27 atoms and ID 12 with 46 atoms, respectively.

In all cases, we assumed $\alpha = 0.1$ for all atoms, $\beta = 0.1$ for all links and $\gamma = 1/n$ where n is the number of atoms in a given molecule (see the table 4.1 below). Each molecular bond is treated as a bi-directional edge and the number of interfaces, m is twice the number of molecular bonds. Also notice that the curved double bonds are treated as a single interface between the atoms. This is because all atoms are used as is and there is no perceptible difference observed in using different bond types (i.e., curved vs. straight bonds). Please note that the number of edges is computed assuming each physical link is bi-directional. In table 4.1, please note that for the most of the models, the ratio of the number of edges to the number of atoms (i.e., average degree), m/n can be expressed as $2(1-1/n)$ for most of the models and has a value close to 2. If you look at just the average degree, the molecular structures are closer to simple chains and binary trees, but their internal topological structure could be more complicated (i.e., *intricate*) in cases, leading to higher topological complexity (for example, see molecule no. 9). Any incorrect construction involved rework and led to increasing total build time. Additional details about the molecules and the experimental procedure can be found in appendix K.

Table 4.1: Details of the set of 12 ball and stick structures of varying structural complexities used for the experiments.

Molecule No.	n	m	C1	C2	C3= E(A)/n	C2*C3	SC = C1 + C2*C3
1	3	4	0.3	0.4	0.94	0.38	0.68
2	7	12	0.7	1.2	1.13	1.35	2.05
3	12	22	1.2	2.2	1.13	2.48	3.68
4	12	22	1.2	2.2	1.00	2.20	3.40
5	12	22	1.2	2.2	1.27	2.80	4.00
6	14	26	1.4	2.6	0.96	2.50	3.90
7	15	28	1.5	2.8	0.97	2.70	4.20
8	16	30	1.6	3	1.40	4.21	5.81
9	19	38	1.9	3.8	1.58	6.00	7.90
10	27	56	2.7	5.6	1.08	6.05	8.75
11	39	80	3.9	8	1.12	8.96	12.86
12	46	100	4.6	10	1.19	11.92	16.52

We also looked at the sensitivities of component and interface complexity values and found no significant impact on the fundamental nature of the complexity vs. development cost relationship. Notice that this is a natural experimental setting and the idea is to mimic the real-world assembly process with the sequence in which different subjects were given the molecular structures was randomized to contain any significant learning effects.

4.2.2 Structural Complexity-Development Cost: A parametric model

With simple ball and stick model building experimental setup, it is easier to contain and isolate other exogenous, *confounding* factors [Mosteller and Tukey, 1977].

We track the total build time for each structure as the observable representing system development effort. Any incorrect assembly involves rework and leads to increasing total assembly time. The overall experimental setup and steps followed are listed below:

- For the experiment we choose 12 different structures to assemble. They were all based on real molecules but that had no importance for our experiment. Photographs were taken of the assembled structures from angles such that the topology of the molecules was visible.

- The experimental subject received an initial briefing including the explanation of what will be expected from him. Before before started, they were shown the molecular kit to familiarize themselves with the components and the two different kinds of bonds, rigid and flexible.

- The subjects were given the completely unassembled kit and were shown a picture of a molecule. Our test subjects consisted of 17 student volunteers with largely similar backgrounds to keep the sample as homogeneous as possible. To contain any influence of the *learning effect*, the order of molecules was randomized.

- The volunteers were asked to assemble these structures as quickly as possible and without error. The total build time, $T_{total} = T_{cognition} + T_{construction} + T_{rework}$ was recorded. We focused only on the total build time T_{total} and not its individual constituents. When completed, the structure will be unassembled and then the next picture was shown to the subject under study.

This experimental setup helps isolate the effect of structural complexity on the system development cost/effort since components of dynamic and organizational complexities are not present here. This helps us capture the effect of structural complexity on development cost/effort by using a simple, single variable parametric model. In order to smooth out the individual differences, we consider the averaged build time for the group of 17 subjects for each ball and stick structure.

The parametric model is developed using regression analysis. The dependent variable Y is the averaged build time for a ball and stick model and the sole independent variable X is the structural complexity of the structure (pre-computed as in table 4.1). The functional form is based on our primary hypothesis and given in eq. 4.1, where the model parameters {a,b} are computed using the traditional least square methodology. The degree of super-linearity or sub-linearity depends on the value of exponent *b*. The growth of development cost is super-linear if $b > 1$ and sub-linear if $b<1$. For the specific case of a linear relationship (i.e., b=1), we use a

parametric relationship of the form, Y = a + bX, where X stands for structural complexity and Y represents mean model build time.

Please note that the model parameters are dependent on the data set used to build the parametric model and are themselves random variables. Their estimates can vary depending on the data and this puts a practical restriction on their use. They should not be used too far out of the data set range used to construct them at the first place or when fundamentally novel systems are being developed for which we have no significant historical data.

To obtain estimates of the equation parameters from the data and also statistically validate the model, the pseudo-value jackknife technique [Mosteller and Tukey, 1977] was used because of the small size of the data samples. The jackknife technique is a statistical method used to ameliorate the problem of biased estimates due to the small size of a sample. This is a form of cross-validation technique where the sample set is randomly divided into two sets, one for model building and the other to test the developed model's redictive capability. This division of sample sets are are performed exhaustively till all combinations are covered and the model parameters are *averaged* quantities over all the instances.

In this technique, the desired calculation for all the data is made where the data are divided into subsamples, and then, the calculation is made for each group of data obtained by leaving out one subsample [Martinez and Martinez, 2007]. An outline of the pseudo-value jackknife procedure is given below:

Pseudo-value jackknifing procedure: Let us represent the original samples as {x_1, x_2,..., x_n} where sample size is *n*. The i^{th} jackknife subsample is defined as the set of original samples with the i^{th} sample or data point removed. Hence the i^{th} jackknife subsample is given by the reduced sample set {x_1, x_2,..., x_{i-1}, x_{i+1},..., x_n} of size (*n-1*).
Let ***T*** be our statistic of interest and is represented as:

$$T = g(x_1, x_2, \ldots, x_n) \qquad (4.2)$$

For the i^{th} jackknife subsample, the same statistic is represented as:

$$T_{(-i)} = g(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \qquad (4.3)$$

The jackknife pseudo-values are defined as:

$$\overline{T}_i = nT - (n-1)T_{(-i)} \qquad (4.4)$$

We perform this operation, leaving one sample observation at a time and generate a sequence of *n* jackknife subsamples.

The jackknife estimate of the statistic of interest is given by:

$$\overline{T} = \frac{\sum_{i=1}^{n} \overline{T}_i}{n} \qquad (4.5)$$

The standard error of this pseudo-value jackknife estimate is defined as:

$$SE_{\overline{T}} = \left[\frac{1}{n(n-1)} \sum_{i=1}^{n} (\overline{T}_i - \overline{T})^2 \right]^{\frac{1}{2}} \qquad (4.6)$$

In our case, we estimated the model parameters {a,b} using the pseudo-valu jackknifing procedure. All computations were performed in MATLAB™ environment. The final single variable parametric model relating development effort (Y) to structural complexity (X) for this set of ball and stick model building activity is given by:

$$Y = 14.68 X^{1.4775} \qquad (4.7)$$

All statistical significance analysis assumed 95% confidence level. The most interesting result perhaps is the exponent of the power law relation, b = 1.4775. This suggests that effort increases super-linearly but is not quite quadratic with increasing structural complexity. Please note here that the above model was built at an aggregated level for the group of individuals and not for a given individual and the model parameters for a specific individual will vary. Hence, the model parameters {a = 14.68; b = 1.4775} are at the aggregated level and not at the level of individuals.

Also note that the above model is a best fit and the model parameters {a,b} are random variables themselves with their expected values given in eq. 4.7. A set of statistical measures is used benchmark the model quality. To test the model quality, the following criteria were used:

- **Coefficient of multiple determination (R^2)** – shows the extent of variance accounted for by the independent variable with a high value (closer to 1) indicates additional independent variables are not likely to improve the model.

- **The mean magnitude of relative error (*MMRE*)** – measures the relative estimation accuracy of the model and is defined as:

$$MMRE = \frac{\sum_{i=1}^{n} |(y_i - \widehat{y_i}) / y_i|}{n} \qquad (4.8)$$

  where y_i is the actual building time of i^{th} molecule and $\widehat{y_i}$ is the estimated building time of i^{th} molecule and *n* is the number of ball and stick structures build.

- **Prediction at a given level (*PRED*)** – measures the quality of prediction by counting how many predictions that lie within a given range of their actual values.

The methodological details can be found in [Conte *et al.*, 1986]. A small *MMRE* indicates that on the average, the model is a good predictor. According to the Purdue Software Metrics Group [Conte *et al.*, 1986], the model is considered to be acceptable, if its *MMRE* is equal to 0.25 or less and [*PRED* (0.25)] > 0.75 [Conte *et al.*, 1986] in general. In other words, the model is said to be acceptable if mean relative error is less that 0.25 and 75% of the predicted values are within 25% of their actual values. We adopt the suggested values in this study for model quality checks. The detailed results are listed below in table 4.2 and table 4.3:

Table 4.2: Results based on the single variable parametric model linking structural complexities of the set of 12 ball and stick structures with development effort, averaged over 17 subjects.

Serial No.	Structural Complexity	Mean build time (sec)	Estimated mean build time	Relative error (%)
1	0.68	12	8.25	31.22
2	2.05	54	42.40	21.47
3	3.68	113	100.72	10.87
4	3.40	96	89.63	6.63
5	4.0	126	113.85	9.64
6	3.9	100	109.50	9.50
7	4.2	117	122.45	4.66
8	5.81	181	197.64	9.19
9	7.90	300	311.43	3.81
10	8.748	427	361.80	15.27
11	12.86	605	639.30	5.67
12	16.52	932	925.58	00.69

Table 4.3: Model parameters and model quality measures.

Model functional form	$Y = aX^b$
Model parameters {a, b}	{ 14.68, 1.4775}
Coefficient of multiple determination (R^2)	0.992
Mean magnitude of relative error (MMRE)	0.107
PRED (0.25)	0.9167
Significance test (parameters)	$t_a = 28.2$, $t_b = 30.67$ ($>t_0 = 2.131$)
Significance of regression model (F test)	$f = 124 > f_{0.05,1,15} = 4.54$

For additional details at individual level, including the standard deviation for each individual, please refer to appendix K.

We observe that the developed model demonstrates a vary high R^2 value with MMRE of 0.1 (<0.25) and PRED(0.25) of 0.92 (>0.75), all indicating excellent model quality as per the suggested guidelines [Conte *et al.*, 1986]. The structural complexity versus actual averaged build time recorded for ball and stick is plotted in fig. 4.5 along with the predicted parametric model in eq. 4.7. Here, we used the mean

build time of a group of 17 subjects for each ball and stick structure as the dependent variable. We also carried out the significance test of model parameters {a,b} and found them to be statistically significant following t-tests (see table 4.3 above). The functional form of the non-linear model adopted here was also found to be highly significant (see table 4.3). A linear fit of the form Y = a + bX was found to be inferior and failed the model quality test, performed using the metrics described above.

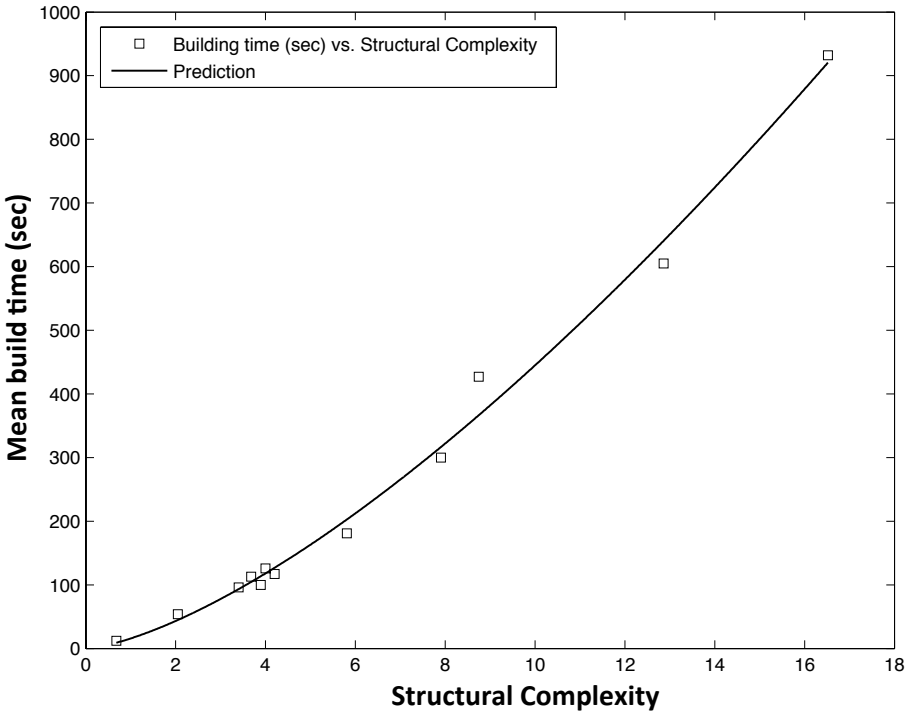


Fig. 4.5: Structural Complexity versus averaged build time in seconds for the set of 12 ball and stick structures (square dots) with the fitted parametric model linking structural complexity to mean build time (solid line).

We can also use a different measure of central tendency like median build time. Using the median build time the individual level data and following the same procedure, we get the model parameters {a=12.28; b=1.54} with R^2 =0.98; MMRE =0.13; PRED(0.25)=0.83, which is not very different from those obtained for the mean build time.

All statistical significance analysis assumed 95% confidence level. The most interesting result is the exponent of the power law relation, b = 1.4775. This

suggests that the system development effort increases super-linearly but is not quite quadratic with increasing structural complexity [Sinha, Omer and de Weck, 2013].

We have focused primarily on a parametric model linking structural complexity and average build time (as a measure of development effort) and choose not to focus on the contribution of individual terms in the structural complexity metric. The ball and stick models in this experiment were constrained by structure of organic molecules, which have inherent architectural features. This restricts us from spanning a similar range of values for each individual component in the structural complexity metric. During the experimentation process, it was observed that for very high C_3 values (i.e., as C_3 value approached 2), the molecules became very complex and lot of subjects gave-up building them due to the larger number of errors that started creeping in. We used molecules for which we had complete data (i.e., correctly constructed by all subjects or nearly all subjects).

Nevertheless, we can look at some observations from the experimental setup to convince ourselves of the importance of individual terms in the structural complexity metric. Since the dependence of the building time on component and interface complexities is more clear and intuitive, we choose to look at the effect of the topological complexity term, C_3. Let us look at the 3 different ball and stick models with identical component and interface complexities (see table 4.4 below), but different molecular structure. We can observe here that the differences in their build time can be account for by the differences in the topological complexity term, C_3. Similar observation can be made if we look at molecular structures 7 and 8 in table 4.1 and 4.2. Their relative difference in component and interface complexities are minimal while one has a much higher topological complexity, which explains the much higher build time was observed.

Table 4.4: Components of structural complexity for a set of ball and stick models where the contributions from component and interface complexities are the same while their molecular arrangement (i.e., connection topology) varies.

ID	C_1	C_2	C_3	$C = C_1 + C_2C_3$	Mean building time (sec)
3	1.2	2.2	1.13	3.68	113
4	1.2	2.2	0.96	3.40	96
5	1.2	2.2	1.27	4.0	126

Please note that the model, linking the build time to structural complexity is nonlinear and therefore a formal, term-wise contribution analysis is not possible. Instead, we used the experimental data and looked a how a parametric model of the functional form $Y = aX^b$ will fit in case we use different sets of independent variable, X . We followed the same jackknifing strategy and the results are consolidated in table 4.5 below for different sets of independent variable as described in the table.

Table 4.5: Model parameters and model quality measures for different predictive models.

Case	Y	X	{a, b}	Model Quality
1	Mean build time (sec)	$C_1 + C_2C_3$	{14.68, 1.4775	[R^2=0.992; MMRE=0.107; PRED(0.25)=0.9167]
2	Mean build time (sec)	C_2C_3	{26.9, 1.43}	[R^2=0.985; MMRE=0.13; PRED(0.25)=0.9167]
3	Mean build time (sec)	$E(A)=nC_3$	{1.781, 1.558}	[R^2=0.983; MMRE=0.14; PRED(0.25)=0.8333]
4	Mean build time (sec)	$C_1 + C_2$	{16.6, 1.49}	[R^2=0.972; MMRE=0.16; PRED(0.25)=0.8333]
5	Mean build time (sec)	C_1	{79.6, 1.607}	[R^2=0.947; MMRE=0.174; PRED(0.25)=0.75]

As we can observe, our proposed functional form for structural complexity metric yields the best overall model. Even if we assume $C_1 = 0$ in all case, we have $X = C_2C_3$ and the model quality is very close for this experimental dataset. If we assume $C_3 = 1$ in all cases, we have $X = C_1 + C_2$ and although a good fit, the model starts to lose some of its predictive capability. Also, detailed t-tests show reduced significance levels for the model parameters for the experimental data considered here. Note that the variation in model quality with reduced form of the structural complexity metric is dependent on the data (i.e., type of systems being considered) and given the availability of complete data, we should use the complete expression for the structural complexity metric, developed in chapter 3 to get the best model quality.

Results from the ball and stick experiments indicate that the super-linearity observed in the build time – structural complexity relationship not due to inherent bias in the structural complexity metric (see appendix K for details). The reason for this super-linear behavior is likely to be embedded in the human cognitive capability and this aspect remains to be explored in future by combining with cognitive scientists.

On the statistical modeling aspect, we have used the average build times at an aggregated level. Please note that it is rather difficult to use this aggregated model at an individual level and such multi-level, hierarchical modeling is complicated since the functional model linking the build time/development effort and structural complexity is nonlinear. In the following section, we discuss a formal linkage between individual level model parameters and that at the aggregated level is discussed in the following section in the context of multi-level or hierarchical modeling.

4.2.3 Discussions on statistical modeling

In the previous section, we have worked towards and developed an aggregate level relationship between Structural Complexity and system development cost/effort. This section takes a deeper analytical look at treating the overall statistical modeling process as an hierarchical or multi-level one, where the fundamental or lowest level is the individual build time that feeds into the higher or aggregated level of group level mean build time for each ball and stick structure (see fig. 4.6 below). There are likely to be significant differences in individual build times for ball and stick models. What might seem simpler to one individual can be difficult or even intractable for someone else. Therefore, the build time curves at individual level may vary significantly.

At complexity level x_j of the jth ball and stick model, the corresponding build time of ith subject is represented as $y_j^{(i)}$. For each individual, the build time is modeled as:

$$y_j^{(i)} = \alpha_i x_j^{\beta_i} + \varepsilon_i \qquad (4.9)$$

where ε_i stands for the error term in the regression model for the i[th] individual. Let there be *n* individuals building the same ball and stick structure with structural complexity x_j. Therefore, the mean build time for the j[th] ball and stick structure with structural complexity x_j is represented as:

$$\overline{y_j} = \frac{\sum_{i=1}^{n} y_j^{(i)}}{n} = \frac{1}{n}\left[\sum_{i=1}^{n} \alpha_i x_j^{\beta_i} + \sum_{i=1}^{n} \varepsilon_i\right]$$
$$= \frac{1}{n}\left(\sum_{i=1}^{n} \alpha_i x_j^{\beta_i}\right)$$

Generalizing from the above expression, we can write the mean build time for any ball and stick structure with structural complexity x as:

$$\overline{y} = \frac{1}{n}\left(\sum_{i=1}^{n} \alpha_i x^{\beta_i}\right) \qquad (4.10)$$

At the aggregated level, we used the same mean build time $\overline{y}$ as the dependent variable in eq. (4.1). Hence we can write $\overline{y}$ as in the aggregated model:

$$\overline{y} = ax^b + \delta \qquad (4.11)$$

where δ is the normally distributed error term. Combining eq. 4.10 and eq. 4.11, we can link the aggregate or group level model to the individual level model:

$$\frac{1}{n}\left(\sum_{i=1}^{n} \alpha_i x^{\beta_i}\right) = ax^b + \delta \qquad (4.12)$$

The above eq. 4.12 relates aggregate level parameters {a,b} to individual level parameters $\{\alpha_i,\beta_i\}$. This relationship between the layers of this hierarchical or multi level representation is illustrated in fig. 4.6 below.

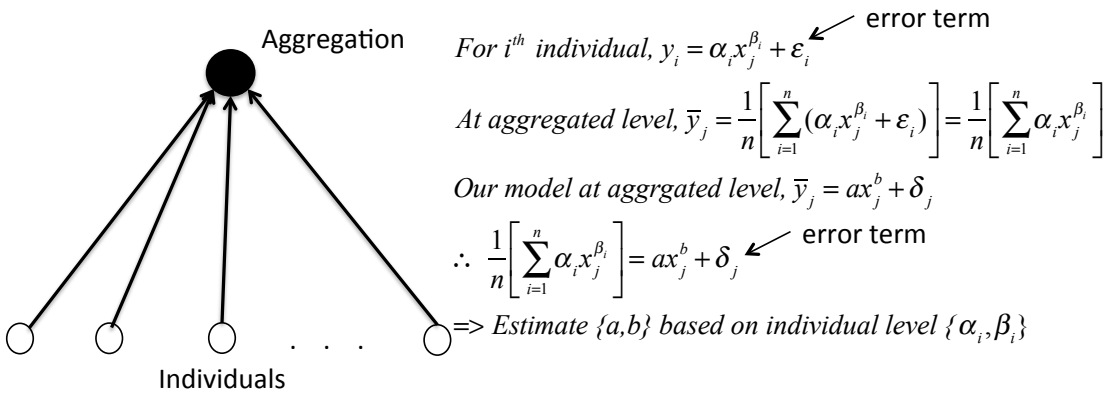


Fig. 4.6: Aggregated level model building using individual level data.

Given the individual level data, we can estimate $\{\alpha_i,\beta_i\}$ for each of the subject who participated in the experiment. For each ball and stick structure, we know the computed structural complexity (see table 4.1). Using the information, one can compute the best fit aggregate level parameters {a,b} using regression analysis. Using this approach, we obtained the aggregate level parameters as {a=14.82; b=1.484} with R^2 = 0.92 and as expected, observe that the values are not significantly different from those obtained using a direct application of averaged build time for each ball and stick structure.

This approach shows the depiction of individual level and aggregate level data as a multi-level structured data and demonstrates a multi-level modeling approach that explicitly brings about the differences in aggregated build time (y) and individual level build time (y_i) in a quantifiable form. This also brings in an important reversed question. Can we estimate the individual level parameters, given the aggregate level parameters? What is the process and a corresponding model that links aggregated, group level parameters {a,b} to individual level parameters $\{\alpha_i,\beta_i\}$?

We can express the individual level parameters using the aggregate level parameters and may be an individual level predictor vector u_i as $\alpha_i = h_\alpha(a,b,u_i)$ and $\beta_i = h_\beta(a,b,u_i)$. Finding a representative individual level predictor vector u_i is a very different and challenging problem that should be investigated in league with social and cognitive scientists.

4.2.4 Comment on generalizability to engineered complex systems

An important question to raise here is about the generalizability of the observed results to real-world, complex engineering system design and development. Are the fundamental observations from this simple ball and stick model building experiment translatable to complex engineering systems? If yes, then to what extent does it translate and why?

We believe that the basic underlying observations about the functional form of the structural complexity vs. system development effort do translate to the much larger, real-world complex engineered systems [Bashir and Thomson, 1999].

We claim generalizability of the functional form found above and that the super-linear form dependence with exponent b>1 will hold for large, real-world engineered systems. We expect that the degree of super-linearity, as measured by the exponent b, will only tend to increase for large, engineered systems. A glimpse of this might be observed from the data shown earlier in the chapter (see section 4.1), although we do not have a sample that is statistically significant for different categories of engineered systems. This is an area for future research in systems archeology.

Building ball and stick models does not have any dynamic aspect and the individual-based building process takes away any organizational complexity. It is a constructive activity, primarily involving visual and/or geometric information processing and ability to decompose and recompose a given structure. This helps in isolating from other confounding effects and understanding the effect of structural complexity on the development cost/effort. A comparison of simple ball and stick model building vis-à-vis complex engineered system development is shown in the table below (see table 4.6):

Table 4.6: Compare and contrast Ball and Stick model building activity with real-world, engineered complex system development.

	Ball and Stick models	**Complex Engineered Systems**
Constructional	Yes	Yes, but at a much larger scale and scope

activity		
Decision making	Not much decision making involved	Involves significant decision making activities at different levels
Extent of team activity	None. Individual activity implies no organizational or social complexity.	Requires multiple people to collaborate with significant social complexity.
Cognitive aspects involved	Primarily involves visual understanding and ability to decompose and recompose.	Involves detailed understanding of physical behavior, spatial and logical understanding and interplay between system architecture and its behavior.
Confounding effects	Almost none. Focused entirely on structural complexity.	Different aspects of complexity [Sheard and Mostashari, 2010] are manifested, including other exogenous factors.

Building ball and stick models is a highly distilled encapsulation of real-world engineering system development activity and underlying effects observed here are likely to magnify in case of engineering system development scenario, which contain various exogenous factors influencing the development cost/effort.

4.2.5 Note on the Spread in build times and Structural Complexity

In the process of building simple ball and stick models, we observe significant differences in individual model building times. The build time profiles vary at the level of individual subjects (see appendix K for details). What is simpler for one subject might turn out to be difficult or even intractable for another. In the current ball and stick model building, the individual level variances in build times are primarily dependent on: (a) cognitive capability of the individual; (b) ability to efficiently accomplish the mechanical work.

We would expect the individual variance to be small for lower structural complexity level, since for less complex systems it is easier to see the best way of assembling it, errors and rework are unlikely and the time to understand and process information is small. Conversely for a more complex structure, time for cognitive processing and likely rework becomes larger and can lead to a larger variance between individuals. To sum up, one would expect significant difference in build times at individual level and that such variance will increase with increasing structural complexity levels.

To investigate this assertion, we build a response model relating the maximum build time and minimum build time for each ball and stick model, to structural complexity. Subsequently, we build a response model relating the standard deviation of the build time for each model and structural complexity. It is shown that the standard deviation of the build times increase super-linearly with structural complexity. The collected data on the respective response variables and structural complexity of ball and stick models are shown in table 4.7 below.

Table 4.7: Structural Complexity and corresponding maximum, minimum build times and the standard deviation of build times for the experiments conducted where each model was build by 17 subjects.

Molecule No.	Structural Complexity	Max. build time (sec.)	Min. build time (sec.)	Build time Standard Deviation (sec.)
1	0.68	18	8	2.97
2	2.05	98	34	17.93
3	3.68	216	77	41.55
4	3.4	205	47	40.83
5	4.0	276	63	53.16
6	3.9	135	66	36.41
7	4.2	187	78	42.61
8	5.81	387	86	76.47
9	7.9	543	159	121.17
10	8.75	775	215	178.74
11	12.86	1140	324	279.28
12	16.52	1496	350	356.88

Let us term the maximum development effort as Y_{max} and minimum development effort as Y_{min} at a given level of structural complexity X. Based on the experimental data, the following relationships linking Y_{max} and Y_{min} to structural complexity, X were obtained:

$$Y_{\max} = 23.31X^{1.5642} \qquad (4.13)$$

$$Y_{\min} = 18.22X^{1.062} \qquad (4.14)$$

Using eq. 4.13 and eq. 4.14, we define the ratio, τ and find the following relationship:

$$\tau = \frac{Y_{\max}}{Y_{\min}} \approx 1.28\sqrt{X} \qquad (4.15)$$

From the experimental data, we observe that the ratio maximum to minimum build for a given ball and stick model increases with increasing level of structural complexity, X.

Now, we turn our attention to the standard deviation of build times for each model and relate that to the structural complexity of that model. For a given ball and stick model, the structural complexity is X and let the build times for all the subjects be given by the vector $\{Y\}$. Let $\hat{Y}_{sd}$ be the standard deviation of the build times (i.e., standard deviation of the data vector $\{Y\}$) for the given ball and stick model. Based on the experimental data on ball and stick model building exercise, we obtain the following relationship linking $\hat{Y}_{sd}$ to structural complexity, X

$$\hat{Y}_{sd} = 8.48X^{1.3} \qquad (4.16)$$

with the following model quality statistics using the jackknifing technique: R^2 =0.96; MMRE =0.19; PRED(0.25)=0.75.

Hence, based on the experimental data, we observe that the standard deviation of the build time for a given ball and stick model increases super-linearly with increasing level of structural complexity, X. This is an otherwise intuitive conclusion and is corroborated by the experimental data.

4.3 Complicatedness function

Let us take a relook at the relationship between development effort, Y and structural complexity, X:

$$Y = aX^b$$

Taking logarithm of both sides, we arrive at a *linear* model where variables are expressed in the logarithmic scale:

$$\ln(Y) = \ln(a) + b\ln(X) \qquad (4.17)$$

Here $\ln(a)$ is the *intercept* term and *b* represents the slope of the curve in logarithmic scale. The parameters of the functional model emphasize very different aspects associated to system development efforts. The parameter *a* characterize work efficiency of the actor (i.e. ability to perform known/specified work efficiently). The parameter *b*, on the other hand, relates to individual/group's innate cognitive ability to synthesize solutions and cognitive capability plays a big role. This parameter becomes more significant at higher regimes of complexity and this aspect is demonstrated segmenting the ball and stick model structural complexity levels into low and high regimes as shown in table 4.8 below.

Let us define the *cognitive demand ratio*, s = b/ln(a). We can see that cognitive demand ratio *s* increased significantly from 0.34 in lower complexity regime to 0.572 in higher complexity regime. This points at increased emphasis of cognitive capability as we move from lower to higher complexity levels. At this point, there is no algorithmic way of segmenting the complexity space into lower and higher complexity regions and we adopt a more visual approach here.

Table 4.8: Model characteristics after segmenting the structural complexity levels into two levels and changes in the cognitive demand ratio, *s* across the structural complexity regimes.

Model Characteristics	Low SC (0,4)	Higher SC (≥ 4)
Model parameters {a, b}	{ 24.2, 1.09}	{14.08,1.497}
Coefficient of multiple determination (R^2)	0.96	0.99
Mean magnitude of relative error (MMRE)	0.12	0.06
PRED (0.25)	0.8	1.0
s = b/ln(a)	0.34	0.572

Notice that at the lower level of structural complexity (i.e., SC<4), the parameter *a*, is larger and the work efficiency, rather than the cognitive ability, of the subject gains importance with a lower value of the cognitive demand ratio, *s*. At

higher complexity regime, higher cognitive demand (i.e., indicated by a higher cognitive demand ratio, *s*) might reduce the work efficiency of the same individuals.

When *s* is smaller, then focusing on efficiency gains might pay off while high value of *s* indicates growing relative importance of cognitive capability on problem solving. This leads us to relate the parameter *b*, the slope of the straight line in log-log plot, to the cognitive capability of an individual or at a group level. We define *b* as *complicatedness function* of the actor (an individual or a group of individuals).

Please note that $b < 1$ implies sub-linear increase in development effort with increasing complexity, which, in turn, implies smaller *complicatedness function* of the development entity (i.e., system development team). It is possible to influence/reduce the complicatedness function by different means, which we will discuss later. This sub-linear behavior, if observed, means that the improvement in cognitive ablity of the development team outstrips the growth in structural complexity. Such behavior is highly unlikely in practice and may be due to a fundamental limitation associated to human cognition. This is where complexity and human cognitive ability meets.

Complicatedness is an observer-dependent property that characterizes an actor's / observer's ability to unravel, understand and manage the system under consideration. In contrast, *complexity* is an inherent system property and a complex system may represent different degrees of *complicatedness* depending on the observer. For example, the complexity of an automobile's automatic transmission may be hidden from a user and is perceived to be less complex. Therefore separating *complicatedness* from *complexity* improves the clarity by which systems can be described, analyzed and certain class of system observables (e.g., development cost, extent of reworks) be predicted.

We can think of *complicatedness* as a conduit through which *complexity* manifests itself at the level of system-level observables like the *system development cost* [Tang and Salminen, 2001].

Complicatedness provides insights to the cognitive aspects of the observer and his/her ability to handle a certain level of complexity. This also calls for future

research in systems archeology and linking system development with cognition science.

Chapter Summary

We looked at the empirical validation aspect of the proposed structural complexity metric in this chapter using an experimental study in which subjects were asked to built molecular models of varying structural complexity and their build times recorded. The proposed complexity metric is shown to have a strong correlation with the development effort/cost and satisfied one of the desirable properties of a complexity metric listed in chapter 3. The primary observations from this empirical study are as follows:

- The exponent of the power law relation between development cost and structural complexity for the ball and stick model building exercise is, b = 1.4775. This suggests that effort increases super-linearly but is not quite quadratic with increasing structural complexity. This parameter is related to the complexity handling capability of the system development team.

- The ratio of maximum to minimum build time varies as the square root of structural complexity for a given ball and stick structure. This result demonstrates that the variance in build time increases with increasing complexity levels.

- Demonstrated a multi-level modeling approach to build an aggregate level from the individual level data model. Such a hierarchical methodology may throw light in the hierarchical relation linking lower level parameters to those at the aggregate level using an individual level variable.

- The notion of complicatedness or perceived complexity is an observer-dependent property that manifests inherent structural complexity in terms of the development effort. This is essentially related to the complexity handling capability of the group/individual.

Further empirical studies using historical system development project databases are required to validate the proposed model for complicatedness function and examine the human characteristics of adaptation and innovation for solving problems of complex system design and development.

References:

Bashir, H.A. and Thomson, V., "Estimating design complexity", Journal of Engineering Design Vol. 10 No 3 (1999), pp 247–257.

Bashir, H.A. and Thomson, V., "Models for estimating design effort and time", Design Studies 22 (2001), pp 141–155.

Conte S.D., Dunsmore H.E. and Shen V.Y., "Software Engineering Metrics and Models", The Benjamin Cummings Publishing Company, Menlo Park (1986).

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Hirschi N.W, and Frey D.D., "Cognition and complexity: An experiment on the effect of coupling in parameter design", Research in Engineering Design 13, 2002, pp 123 – 131.

Maier M.W. and Rechtin E., "The Art of Systems Architecting", CRC Press; 3rd edition (2009), ISBN-10: 1420079131.

Martinez W.L. and Martinez A.R., "Computational Statistics Handbook with MATLAB", 2nd edition, CRC Press (2007), ISBN-10: 1584885661.

Mosteller F. and Tukey J.W., "Data Analysis and Regression", Addison-Wesley Publishing Company, Don Mills (1977).

Prentice Hall Molecular toolkit, "Prentice Hall Molecular Model Set for General and Organic Chemistry". 1997.

Sinha K., de Weck O.L, "Structural Complexity Metric for Engineered Complex Systems and its Application", 14th International DSM Conference, 2012.

Sinha K., Omer H., de Weck O., "Structural Complexity: Quantification, Validation and its systemic implications for engineered complex systems", International Conference on Engineering Design, 2013.

Tang, V. and Salminen, V., "Towards a Theory of Complicatedness: Framework for Complex Systems Analysis and Design", 13th International Conference on Engineering Design, Glasgow, Scotland, August 2001

Wertz J.R. and Larson W.J., "Reducing Space Mission Cost", ISBN-10: 1881883051, Springer, 1996.

Larson W.J and Wertz J.R., "Space Mission Analysis and Design", ISBN-10: 9781881883104, 1999.

Wood et al, "Interfaces and Product Architecture", ASME DETC, 2001.

Chapter 5

Applications and Case Studies

In this chapter, we explore application of the proposed methodology of structural complexity quantification introduced in chapter 3. We demonstrate application of structural complexity quantification to various engineered systems, starting from simple ones and moving towards large, real-world systems. Impact of uncertainty propagation on structural complexity is illustrated for each example.

5.1 Simple Examples

We illustrate calculation of structural complexity with uncertain component complexity estimates for simpler systems. The propagation of uncertainty is demonstrated in the context of structural complexity calculation and shows the effect of right-skewedness on the resulting distribution. Before we get into the details, the first step is to describe the system modeling approach followed in this thesis for representing complex engineered systems.

5.1.1 System representation methodology

The first step in applying the developed methodology to complex systems is to develop the product DSM of the system. The product DSM's presented in this chapter were based on the process developed by Suh and co-authors [Suh *et al.*, 2010] and in Tilstra [Tilstra *et al.*, 2013]. The connection types were generalized to four primary categories (i.e., physical connection, mass flow, energy flow and information/signal flow) as shown in table 5.1 below for most examples. Please note that all four categories may not be present in a single system. Also note that the four primary connection types can be further broken down (see table 3.2 in chapter 3)

and the level of detail depends on system and associated variations in modeling of interface complexities.

Table 5.1: Different Types of Interactions in a DSM

Connection Type	Description	Examples
Physical	Physical adjacency/connectedness between two components	Weld, Bolt, Socket, Wiring
Mass	Material/mass flow between two components	Toner, Paper, NOHAD flow
Energy	Energy/power transfer between two components	Electrical, Mechanical, Chemical energy
Information	Data/signal exchange between two components	Diagnostics information to UI, Sensor signals

Each cell in the matrix at the intersection of a component-row and component-column is divided into four sub-cells. The upper left sub-cell represents a direct physical interaction between two components, typically involving a mechanical connection, the transfer of motion or at a minimum indicating immediate physical proximity. If the sub-cell is 1 such an interaction is present, if it is 0 then there is no such interaction between the two specific components indicated by the row and column, respectively. Likewise the other three sub-cells indicate mass transfers (upper right sub-cell), energy transfers (lower left sub-cell) or information transfers (lower right sub-cell), respectively. There are particular specific aspects about different types of connections. For example, mass flow can happen typically only across physical connections. Hence mass flow has an underlying dependency on physical/ spatial connectivity between components and mass flows are usually asymmetric. But the same is not true about the information flow. The mass flow and energy flow usually happen across physical connections but such interactions can occur in a “wireless” fashion as well (especially in case of energy flow). In this fashion a “complete” DSM model of the system can be developed. Once the basic DSM is built, one can go about reordering or partitioning

the DSM to reveal the logical structure or architecture of the product. An example system Design Structure Matrix (DSM) shows the main elements or sub-systems as the rows and columns of a matrix. The connections between the elements are shown as the off-diagonal elements.

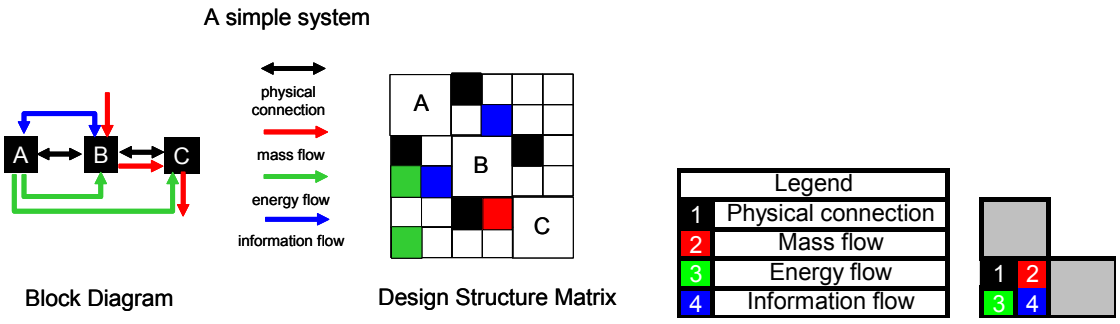
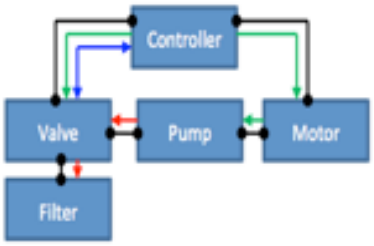


Fig. 5.1: Block Diagram (Left), DSM (Right) of a Simple System and Associated Legends

Figure 5.1 shows how to read a highly simplified DSM matrix for a simple system composed of three components A, B and C. In this example component A physically connects to B which in turn is connected to C. A mass flow occurs from B to C, while energy is supplied from A to B and C, respectively. Additionally A and B exchanges information with each other. Such a DSM forms the basic information upon which the subsequent analysis is build. The aggregated DSM (i.e., binary adjacency matrix) is built considering all four types of connections and if there is any type of connection existing between two components, then they are considered connected.

5.1.2 Structural Complexity calculation for simple fluid flow system

Let us start with the hypothetical fluid flow system described in chapter 3 to illustrate the method (see fig. 5.2 below). The graph energy is $E(A)$ = 5.6 and the topological complexity, C_3 = E(A)/n = 5.6/5 = 1.12. This illustrative example is similar to that used in chapter 3 with differences in component and interface complexities. Now let us differentiate among components and the most likely component complexity vector is given in table 5.2 below.



$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 5.2: (a) Sample system: it shows different connection types amongst components - physical/mechanical connection (black); material/fluid flow (red); energy flow (green) and information/control signal flow (blue); (b) binary adjacency matrix.

Table 5.2: Most likely component complexity for the hypothetical fluid flow system example.

Component	ID	Complexity
Controller	1	1.5
Pump	2	1.0
Valve	3	0.3
Filter	4	0.3
Motor	5	1.2

Please note that the above most likely component complexity is for illustration purpose and only. We define the right-skewedness (or the lack of it) of component complexity distribution using triangular distribution by using parameter p and the following relationship linking the most likely, most optimistic and most pessimistic estimates:

$$(b - \alpha_m) = p(\alpha_m - a) \qquad (5.1)$$

The extent of skewedness is controlled by the parameter p with p = 1 indicating symmetric distribution and p > 1, the extent of right-skewedness. In this example, we have used $p \in [1.0; 3.0]$ and $a \in [0.8\alpha_m; 0.9\alpha_m]$. This is usually the preferred distribution when estimates are made using expert opinion elicitation [Garvey, 2000].

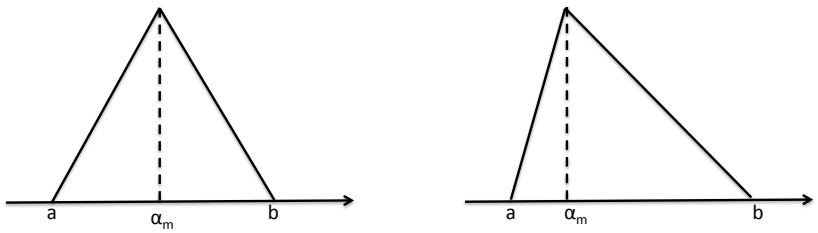


Fig. 5.3: (a) Symmetric triangular pdf of component complexities; (b) right-skewed triangular pdf of component complexities.

Different interfaces have their characteristic interface factor as shown in table 5.3 and we use the model defined in chapter3, for computing the interface complexity. The interface factor for k^{th} interface type expresses the interface complexity as a fraction of component complexity. For example, value of 0.10 means that the interface is only 10% as complex as the interfacing components. A larger value indicates interface complexity for the k^{th} interface type/category.

Table 5.3: Interface factor used for different connection types.

Connection type	**Interface factor, $1/c^{(k)}$**
Mechanical connection	0.05
Information/Control	0.15
Fluid flow	0.10
Energy	0.10

The interface table for this example is given in table 5.4.

Table 5.4: Interface table with interface factors.

Comp. 1	Comp. 2	$1/c^{(k)}$
1	3	0.05
1	3	0.10
1	3	0.15
1	5	0.05
1	5	0.10
2	3	0.05
2	3	0.10
2	5	0.05
2	5	0.15
3	4	0.05
3	4	0.10

Note that two connected components might have multiple types of interfaces between them and they ought to be added up with a resultant interface factor *1/s* such that,

$$\beta_{ij} = \frac{\max(\alpha_i, \alpha_j)}{s} = \max(\alpha_i, \alpha_j) \sum_{k=1}^{l} \left(\frac{1}{c^{(k)}} \right)$$

$$\Rightarrow \quad \frac{1}{s} = \sum_{k=1}^{l} \left(\frac{1}{c^{(k)}} \right) \qquad (5.2)$$

where *k* is the interface type and *l* is the number of interfaces between components (*i,j*). The resulting distributions for total component complexity, C_1 and structural complexity, C for the fluid flow system is shown in fig. 5.4 below.

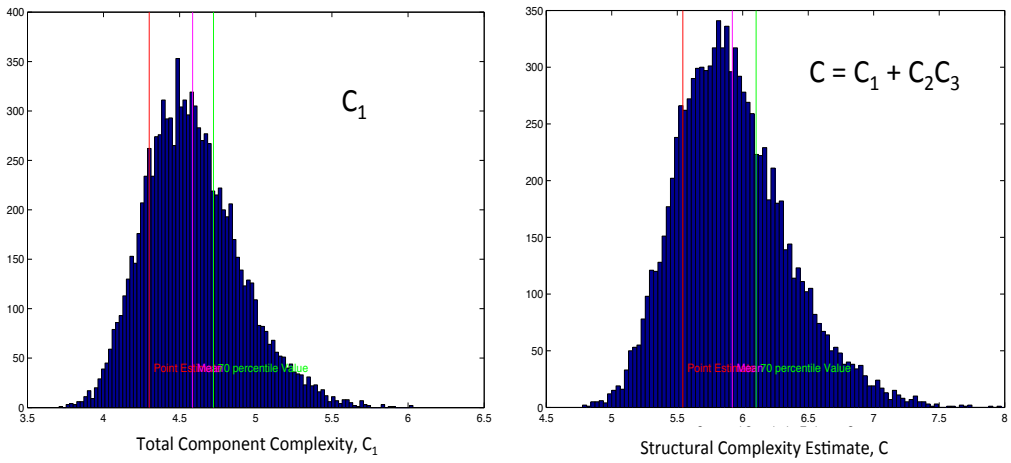


Fig. 5.4: Pdf of total component complexity and structural complexity respectively for the fluid flow system. Both distributions show hint of right-skewedness.

The probability distribution function of both, total component and structural complexity, show hint of right-skewedness with mean being marginally higher than the median and the mode. This is shown pictorially in fig. 5.5. The 70-percentile value is often prescribed as a representative value fro decision-making from a risk perspective [Garvey, 2000].

Table 5.5: Structural Complexity of fluid flow system using most likely estimates of component complexities and the mean, median and 70-percentile values from respective distributions. Here, C_{ML} stands for the Structural Complexity using most likely estimates.

	C_1	C_2	C_3	C	C/C_{ML}
Most Likely	4.3	1.11	1.12	5.543	1
Mean	4.58	1.2	1.12	5.93	1.07
Median	4.56	1.18	1.12	5.89	1.06
70 percentile	4.73	1.235	1.12	6.12	1.11

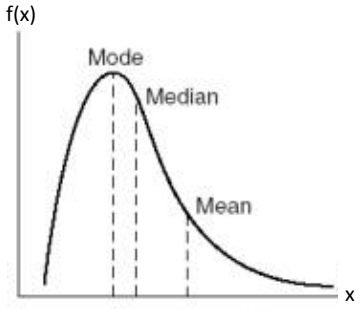


Fig. 5.5: Right-skewed probability density function with mode < median < mean.

Now let us turn our attention to a set of simpler real-world engineered products.

5.1.3 Black & Decker and Innovage Screwdrivers

Now we move on to describe application of structural complexity quantification for two similar engineered products with the same functionality. Both of these are electric screwdrivers, but developed by different entities with slightly different mode of achieving the primary functionality. Here we have two simple engineered systems to serve the same function and compare their structural complexities. These two examples are based on the product dissection data in Andrew Tilstra's thesis [Tilstra, 2010]. The two products with labeled components are shown in Fig. 5.6 and Fig. 5.7 respectively.

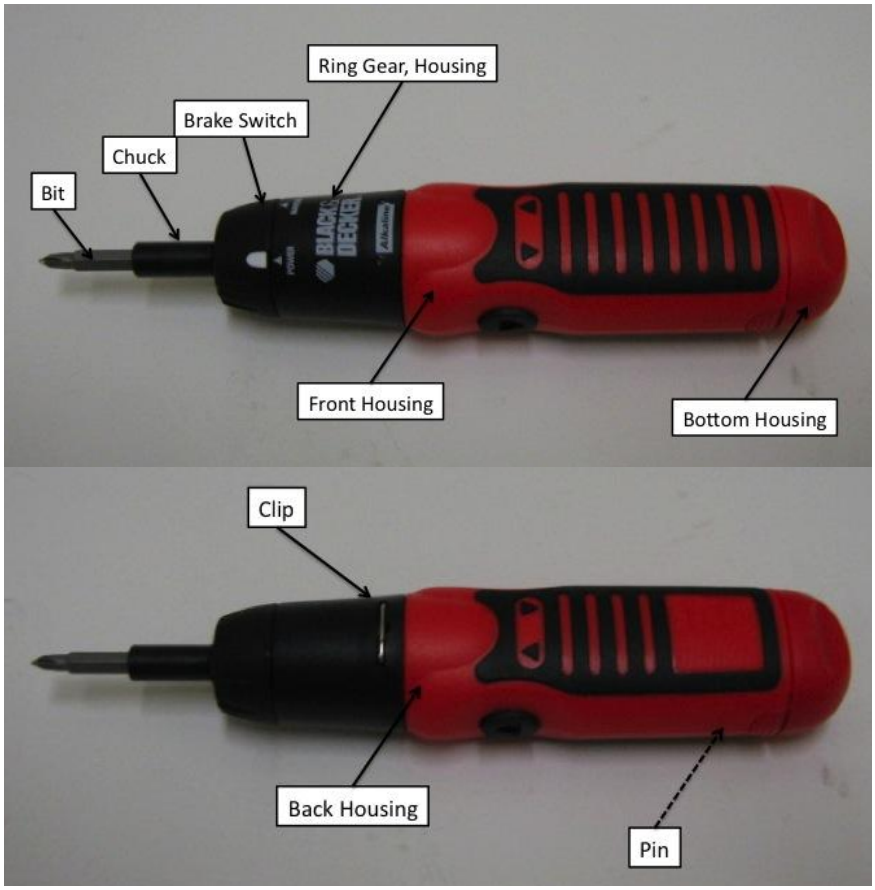


Fig. 5.6: Black & Decker Alkaline Power Screwdriver [Tilstra, 2010].

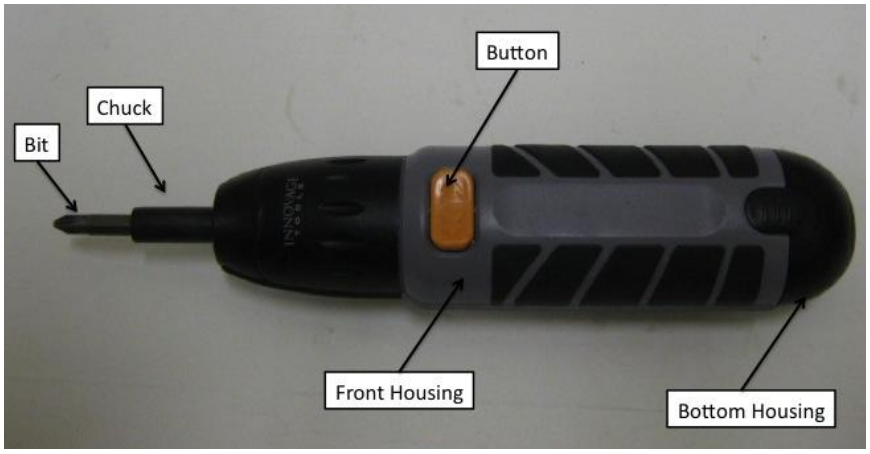


Fig. 5.7: Innovage powered screwdriver [Tilstra, 2010].

Figure 5.8 below shows the exploded view of labeled components of the Black & Decker alkaline screwdriver.

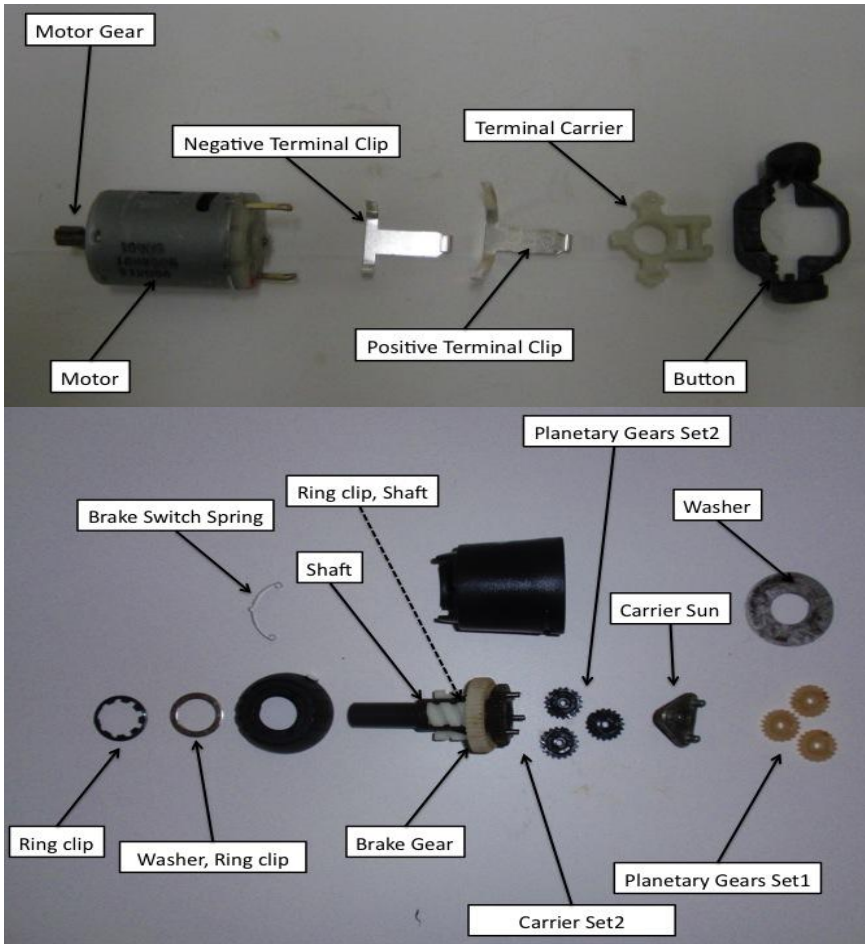


Fig. 5.8: Components of the Black & Decker alkaline screwdriver [Tilstra, 2010].

The developed Product Design Structure Matrices (DSM) for these two systems, based on [Tilstra, 2010], are shown in Fig. 5.9 (a) – (c). Notice that in both cases, the average nodal connection (i.e., average nodal degree) per component is around 5 (Innovage screwdriver has average degree of 5.09 and that of BD screwdriver being 5.0).

The most likely component complexities were estimated using probabilistic estimates [Garvey 2000] on a scale of [0,5]. In all cases, we assumed component complexities to have triangular distribution with most likely estimated being the point estimates (see Fig. 5.10 below).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Bit - 1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Clip - 2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Motor Gear - 3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Washer - 4	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planetary Gear Set1 -5	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carrier Sun - 6	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planetary Gear Set2 - 7	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carrier Set2 -8	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ring Gear, Housing - 9	0	1	0	1	1	1	1	1	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Ring Clip, shaft - 10	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Shaft - 11	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brake Gear - 12	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chuck - 13	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brake Switch - 14	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brake Switch Spring - 15	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ring Clip - 16	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Washer, ring clip - 17	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Motor - 18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
Bottom Housing - 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0
Battery Holder - 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	1	1	0
Series Terminal Clip - 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
Battery Plug - 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0	0	1	0	0
Plug Terminals - 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	0	0	0
Series Terminal Plate - 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0
AA Battery - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	1	0
Negative Terminal Clip -26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0
Positive Terminal Clip - 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1	0
Terminal Carrier - 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	1	1	0
Button - 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1	0
Front Housing - 30	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	1	0	1	1	0	1	1
Back Housing - 31	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	1	1	1	0	1
Pin -32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Fig. 5.9 (a): The adjacency matrix or binary DSM of Black & Decker screwdriver with 32 components.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bit - 1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Motor Plate - 2	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0
Planets Set1 - 3	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planet Carrier Sun - 4	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planets Set2 - 5	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brake Coupler Planet Carrier -6	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ring Gear Case -7	0	1	1	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Coupler -8	0	0	0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brake Drum - 9	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brake Roller - 10	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Shaft - 11	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Spring Clip - 12	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Washer - 13	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chuck - 14	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Motor - 15	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0
Motor Screws - 16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bottom Housing - 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	1	0
Battery Holder - 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1	1	0
Series Terminal Clip - 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Battery Plug - 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0
Plug Terminals - 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0
Series Terminal Plate - 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
AA Battery - 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	1	1	0
Wire switch2motoe -24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
Button - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
Button Base - 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
Wire base2bat - 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
Battery Clip - 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	0
Back Housing - 29	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	1	0	0	1	1	0	1	1
Front Housing - 30	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	0	1
Pins - 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Fig. 5.9 (b): The adjacency matrix or binary DSM of Innovage screwdriver with 31 components.

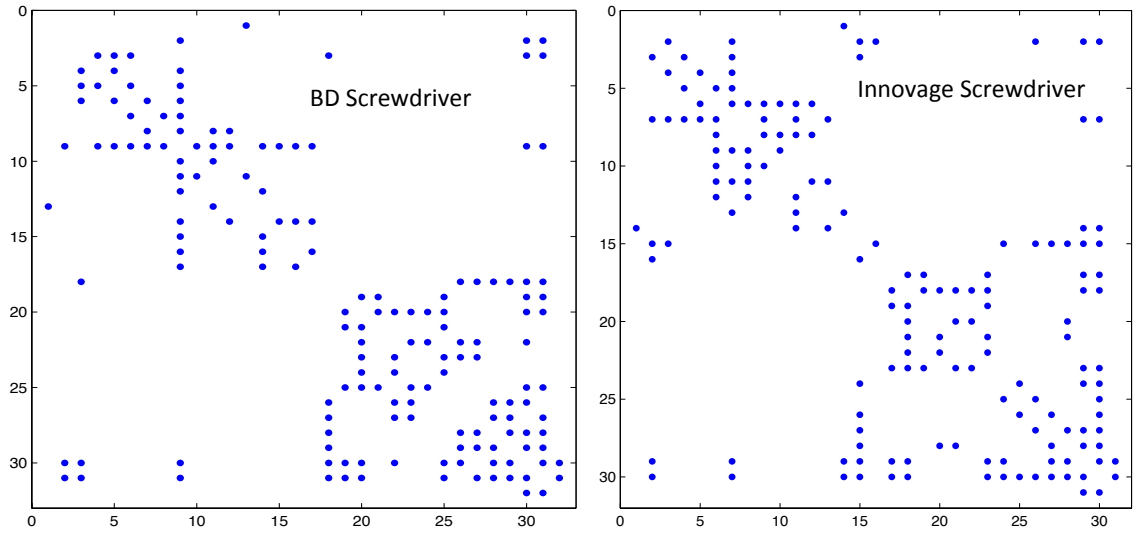


Fig. 5.9 (c): The spy plots of the system connectivity structure for the two powered screwdrivers. Presence of blue dot represents existence of connection between components.

The most optimistic values *a* and the most pessimistic values *b* were set based on the ranking of point estimates of component complexities α_m. The most likely complexity estimates for two systems are shown in Fig. 5.11 below.

The optimistic and pessimistic estimates of component complexity distributions can be characterized using eq. 5.1 with $p \in [1.0; 3.0]$ and $a \in [0.8\alpha_m; 0.9\alpha_m]$. These numbers and associated ranges were chosen to align with the component complexity numbers in fig. 5.11 below. In this example, higher most likely estimates aligned with higher pessimistic estimates. This is a simple and matured system and this fact is reflected in the component complexity numbers in fig. 5.11.

In these simple systems, there are primarily three interface categories as shown below with their interface factors in table 5.6. They are employed consistently across these two products.

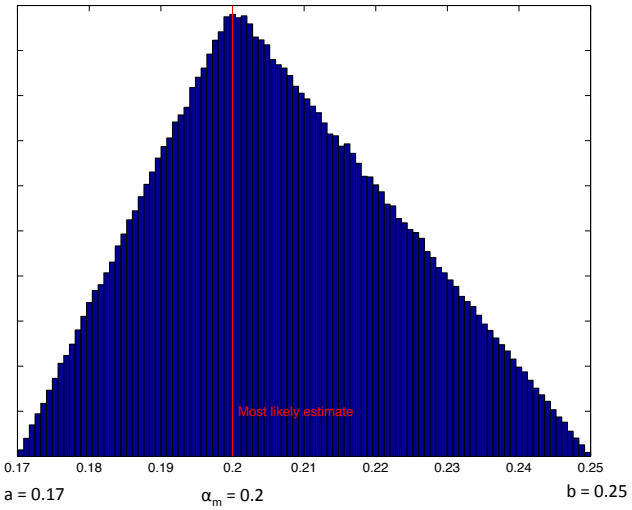


Fig. 5.10: Sample pdf of complexity for housing component.

Component Name	Component Complexity (0 - 5)
Bit - 1	0.1
Clip - 2	0.1
Motor Gear - 3	1
Washer - 4	0.1
Planetary Gear Set1 -5	1
Carrier Sun - 6	1
Planetary Gear Set2 - 7	1
Carrier Set2 -8	1
Ring Gear, Housing - 9	0.3
Ring Clip, shaft - 10	0.5
Shaft - 11	0.5
Brake Gear - 12	0.5
Chuck - 13	0.2
Brake Switch - 14	0.1
Brake Switch Spring - 15	0.1
Ring Clip - 16	0.1
Washer, ring clip - 17	0.1
Motor - 18	1
Bottom Housing - 19	0.1
Battery Holder - 20	0.1
Series Terminal Clip - 21	0.1
Battery Plug - 22	0.1
Plug Terminals - 23	0.1
Series Terminal Plate - 24	0.1
AA Battery - 25	0.1
Negative Terminal Clip -26	0.1
Positive Terminal Clip - 27	0.1
Terminal Carrier - 28	0.1
Button - 29	0.1
Front Housing - 30	0.2
Back Housing - 31	0.2
Pin -32	0.1

Component Name	Component Complexity (0 - 5)
Bit - 1	0.1
Motor Plate - 2	0.1
Planets Set1 - 3	1
Planet Carrier Sun - 4	1
Planets Set2 - 5	1
Brake Coupler Planet Carrier -6	1
Ring Gear Case -7	0.3
Coupler -8	0.5
Brake Drum - 9	0.3
Brake Roller - 10	0.2
Shaft - 11	1
Spring Clip - 12	0.1
Washer - 13	0.1
Chuck - 14	0.2
Motor - 15	1
Motor Screws - 16	0.2
Bottom Housing - 17	0.3
Battery Holder - 18	0.1
Series Terminal Clip - 19	0.1
Battery Plug - 20	0.1
Plug Terminals - 21	0.1
Series Terminal Plate - 22	0.1
AA Battery - 23	0.1
Wire switch2motoe -24	0.2
Button - 25	0.2
Button Base - 26	0.1
Wire base2bat - 27	0.2
Battery Clip - 28	0.1
Back Housing - 29	0.1
Front Housing - 30	0.1
Pins - 31	0.1

Fig. 5.11: Most likely estimates of component complexity for (a) Black & Decker (on left), and (b) Innovage screwdrivers (on right) respectively.

Table 5.6: Interface factor used for different connection types.

Connection type	Interface factor, $1/c^{(k)}$
Mechanical Connection	0.05
Electric Energy	0.10
Rotational Energy	0.10

The propagation of component level uncertainties results in the probability distributions of total component complexity, C_1 and structural complexity as shown in Fig. 5.12 below. Since the associated uncertainties are not very large, the balancing effect of the law of large numbers leads to only marginal right-skewedness in the distributions of aggregated complexity measures.

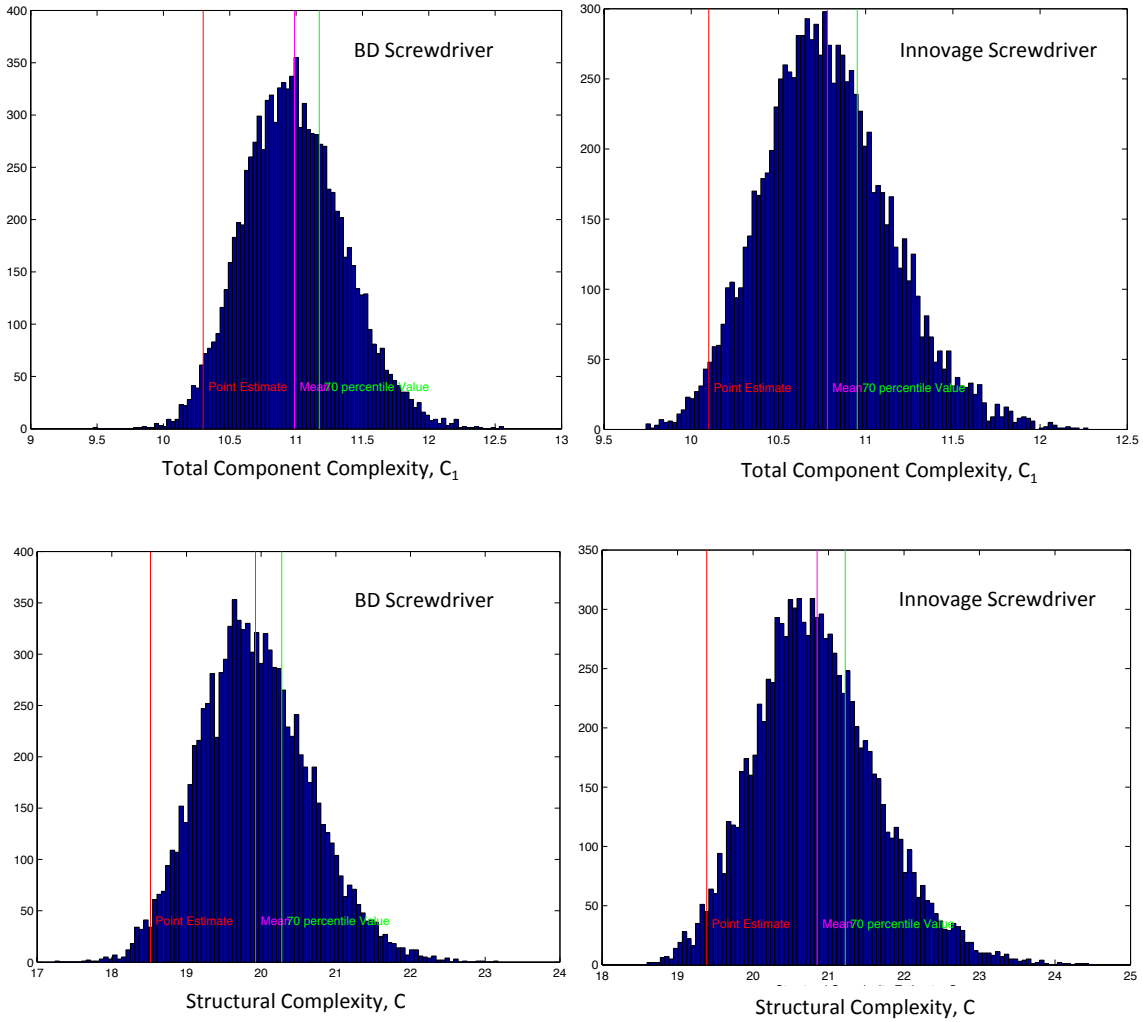


Fig. 5.12: Probability distribution of total component complexity and structural complexity respectively for BD and Innovage screwdrivers. Both distributions show hint of right-skewedness.

The extent of right-skewedness can be observed from table 5.7 where we have the median being smaller than the mean. Existence of right-skewedness indicates that there are some component complexity values that are much higher than the rest of the components. A higher degree of right skewedness indicates that

the most likely estimates are more optimistic and will always underestimate the real complexity estimate. Large right skewedness is often indicative of *novelty* associated to the system being developed where actual complexity is likely to be underestimated and is well documented in the literature [Garvey, 2000]. The extent of right skewedness is to some extent rationalized due to the law of large numbers that tends to drive the aggregated distribution toward the normal distribution. Since these two products are quite standardized with very little *novelty* (*novelty* is usually associated with higher complexity), the distribution of structural complexity is rather narrow and the 70-percentile value is only about 10% higher that the most likely estimate. This changes significantly for systems with significant novelty (leading to enhanced component complexity and likely, either in components or in interfaces or both. In all cases, the topological complexity, C_3 is smaller than 2 and therefore did not reach the corresponding *P* point (see chapter 3).

Table 5.7: Comparison of Structural Complexity of Black & Decker and Innovage screwdrivers.

	C_1		C_2		C_3		C		$C_{Innovage}/C_{BD}$
	BD	**Innovage**	**BD**	**Innovage**	**BD**	**Innovage**	**BD**	**Innovage**	
Most Likely	10.3	10.1	4.94	5.39	1.6635	1.723	18.58	19.39	1.04
Mean	10.99	10.78	5.37	5.84	1.6635	1.723	19.93	20.85	1.046
Median	10.97	10.75	5.35	5.82	1.6635	1.723	19.88	20.79	1.046
70 percentile	11.17	10.95	5.48	5.97	1.6635	1.723	20.27	21.24	1.05

It is interesting to note that the Black & Decker screwdriver has higher component complexity, while Innovage screwdriver has higher topological complexity (also interface complexity). There is very little to choose from the two in terms of their structural complexity, with or without considering uncertainty propagation, as the difference is less than 5%. Even from the pricing perspective in the market, they seem to be quite at par with average price of Innovage screwdriver being slightly higher. But the differentiation in price does not seem to be significant over multiple time periods.

5.2 Printing System

Printing is a process for reproducing text and images, typically with ink on physical paper using a printing system. It is an essential part of publishing and transaction printing and is often carried out as an industrial process. There are different methodologies/processes to enable printing and we are primarily focused on the Xerographic process and systems that implement the process. Xerography is the dry ink marking process underlying *electrophotographic* copiers and printers. Xerographic systems have had commercial success for over 45 years and currently drive a segment of the printing market valued at over $150 billion worldwide [Hamby and Gross, 2004]. The xerographic process consists of: i) charge, ii) expose, iii) develop, iv) transfer, v) fuse, and vi) clean (see fig. 5.13 below). The unifying element in the xerographic process is the photoreceptor—the surface for image generation and transportation that is conductive in the presence of light and insulative otherwise. Photoreceptors typically take the form of either a drum or a belt and must be capable of providing a consistent surface for image creation while cycling through the xerographic process for hundreds of thousands of prints. Details of these sub-processes can be found in the literature [Hamby and Gross, 2004].

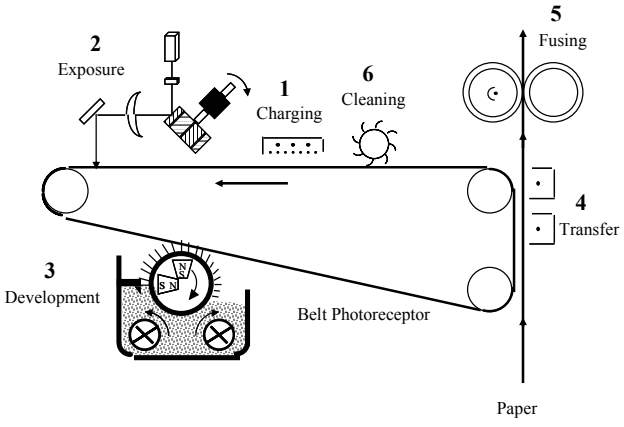


Fig. 5.13: Schematic diagram of a monochrome xerographic process [Hamby and Gross, 2004].

5.2.1 Printing System DSM Development

The first step in applying the developed methodology to complex systems is to develop the product DSM of the system. The product DSM was developed

following the process developed by Suh and co-authors [Suh *et al.*, 2010]. The connection types were generalized to four main categories (i.e., physical connection, mass flow, energy flow and information/signal fow) as shown in table 5.1.

The systems considered for this study are tagged as: (1) *New* and (2) Old. The *New* system is a higher end printing system compared to the *Old*. A view of these two systems with associated DSMs, are shown in Fig. 5.14 and Fig. 5.15 respectively.

Fig. 5.14: *New* Digital Printing System and its DSM (Size: 84x84).

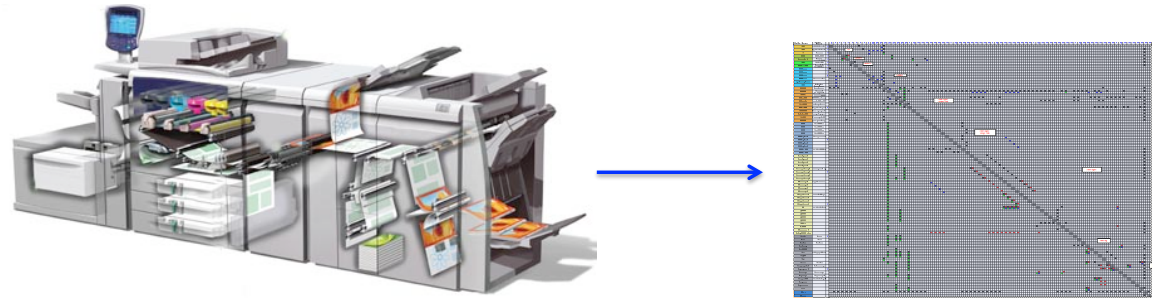


Fig. 5.15: *Old* Digital Printing System (Shown with High Capacity Feeder and Finisher) with its DSM (Size 91x91).

The DSM was developed based on (i) system documentation; (ii) physical inspection and selective disassembly; and (ii) opinions of subject matter experts. The *new* printing system has significant functional advancements that include automatic image quality correction technology and active environmental control of Ozone and NOHAD [Suh *et al.*, 2010].

There are challenges in mapping some connections between components for printing systems, particularly with respect to wire harnesses and software connections. If two components (e.g. PWBs) A and B are connected to each other electrically using wire harness then we aggregated the harnesses within the

components and assumed components A and B are directly connected. Information flow between software components also raises an important representational question. Now software typically resides in the memory of a processor or circuit board. At the physical level, software actions are transformed into electrical signals that flow over wires through the system. While we do represent the physical connection between software and circuit board in the DSM, we also showed information flow between software directly, for clarity of information flow. An interesting situation, peculiar to flow modeling of printing systems is the question of representing only intended interactions within the system (e.g. flow of paper, toner, image information through the system) versus mapping both intended and *unintended* flows. Examples of unintended flows are the intrusion of dirt into the printing engine, leaking of toner, the generation of ozone in the printing engine or the introduction of electronic noise into various sensor signals [Suh *et al.*, 2010]. Only those unintended flows and interactions that the system *would actively deal with* were modeled as a guideline.

After capturing these four types of connections, we can split the DSM in terms of individual connection types or generate an *aggregated* DSM. For this present study, we primarily focused on the physical DSM (i.e., contains physical connections only, other type of connections are neglected) and the aggregated DSM. The aggregated DSM is built considering all four types of connections and if there is any type of connection existing between two components, then they are considered connected. One can observe that aggregated DSM will always have number of different kinds of connections. In graph-theoretic terminology, the physical DSM can be considered to be as a *spanning subgraph* of the aggregated DSM. The *New* printing system DSM (size 84 x 84) was available from the literature [Suh *et al.*, 2010] and the *Old* printing system was decomposed to derive a finer DSM (size 91 x 91). The DSMs were built following the procedure outlined in the literature [Suh *et al.*, 2010]. The conventions outlined there (also earlier in this chapter) for modeling certain types of connections, especially with respect to wire harnesses and mapping of information flow across embedded software components were followed while developing this DSM to ensure consistency. Due to available time constraint, it was

decided to model just the xerographic subsystems in more detail. This meant the ROS assembly; marking elements, fusing area and also paper path elements were expanded in more details.

5.2.2 Structural Complexity Estimation

In all instances, we assumed component complexities to have triangular distribution with most likely estimated being the point estimates as before. The component complexities are assessed and reflect the internal complexity of the individual components in the DSM. Due to unavailability of historical data with adequate level of validity, the component complexities were determined by *expert* review, using a scale [0,5] with 5 being the highest complexity level [Sinha and de Weck, 2013].

In almost all cases, the triangular distribution representing component complexity was right-skewed. For the *new* printing system, the optimistic and pessimistic estimates of component complexity distributions obtained from the experts could be well characterized by using eq. 5.1 with $p \in [1.0;\ 7.0]$ and a $\in [0.8\alpha_m;\ 0.9\alpha_m]$. For the *old* printing system, the estimated ranges were well characterized by $p \in [1.0;\ 4.0]$ and a $\in [0.85\alpha_m;\ 0.95\alpha_m]$. Note that the pessimistic estimates are much higher than the most likely estimates. This indicates existence of significant external uncertainty, but with very low probability of occurrence. This is indicative of fat-tailed distributions of factors that leads to uncertainty in component complexity estimates.

In this example, higher most likely estimates align with higher pessimistic estimates.

The representative interface factors for the four interface types used were determined based on the available programmatic data and are listed in table 5.8 below. For any type of interface, the interface factors listed below were averaged over instances of the same type of interface. These numbers were used consistently across the two printing systems.

Table 5.8: Interface factors used in modeling the two printing systems

Connection type	Interface factor, $1/c^{(k)}$
Physical Connection	0.07
Mass	0.10
Energy	0.15
Information	0.18

We compute the structural complexity for most likely values and list the mean, median and 70 percentile values for structural complexity, C and also its constituents (C_1, C_2, C_3). The values are listed in table 5.9 below.

Table 5.9: Comparison of Structural Complexity for two printing systems.

	C_1		C_2		C_3		C		C_{New}/C_{Old}
	Old	**New**	**Old**	**New**	**Old**	**New**	**Old**	**New**	
Most Likely	110.2	169	55.68	102.78	1.36	1.804	185.93	354.42	1.9062
Mean	125.62	213.6	63.29	130.6	1.36	1.804	211.69	449.2	2.122
Median	124.47	211.84	62.46	128.62	1.36	1.804	209.42	443.88	2.12
70 percentile	127	219	65.82	134.2	1.36	1.804	216.2	461.1	2.133

While using the most likely estimates indicates a 90% increase in structural complexity for the *new* printing system, other representative measure of central tendency like mean shows a little more than 200% increase in structural complexity. If we use the 70-percentile measure as the representative value from a risk management perspective [Garvey, 2000], the increase is about 213%. A higher, right-skewed uncertainty characterization leads to even sharper increase in structural complexity as we consider 70-percentile values for comparison, as opposed to the most likely values for component complexity. Notice that the topological complexity, C_3 is much higher for the new print engine and it approaches the *P* point value of 2 (see chapter 3) and the architecture tends to be more distributed in nature.

Please note that these two printing systems were quite different from a performance centric viewpoint with the *new* system is a much larger one a significantly higher throughput and stricter performance regime, leading to highly engineered components with more distributed architecture. This is the reason for its significantly higher structural complexity. Higher complexity has led to

significant and nonlinear increase in the development effort/cost, but the actual numbers are of course highly confidential information and therefore not public domain.

This methodology is now employed to compare two aircraft engine architectures in the next section.

5.2.3 Effect of the system decomposition level on topological complexity

An important aspect in practical application of the structural complexity is related to the effect the level of system representation or the level of decomposition on the complexity metric. Between a coarse and a finer representation of the same system, we have larger number of components and interfaces at a deeper/finer level of system decomposition. But the basic structure remains nearly the same beyond a level of decomposition that adequately describes and differentiates the system. If the basic architectural patterns remain the same, the topological complexity metric should remain approximately the same. In order to verify this, we use a coarser decomposition of the older digital printing system (see fig. 5.15). This coarser representation aggregated part of the *xerographics* subsystems. This included the ROS assembly, marking elements, and also paper path elements. The difference in terms of number of components modeled in each of these areas is best explained using the table 5.10 below:

Table 5.10: Difference in coarse and finer Decomposition Levels

Functional Area	Coarse DSM (50x50)	Fine DSM (91x91)
ROS Assembly	4	10
Marking elements	16	38
Paper Path	7	12

The result is shown in table 5.11 below.

Table 5.11: Attributes of coarse and finer representations

DSM attribute	Coarse Representation	Finer representation
System size, N	50	91

Graph Energy	67.67	123.73
C_3	1.3534	1.3597

From table 5.11, we can observe that the topological complexity metric, C_3 is almost constant across two architectural representations of the same digital printing system. This shows that the topological complexity metric is robust against levels of system decomposition of engineered systems. This is an important and desirable characteristic of a complexity metric and a larger set of such studies would be required to cement this claim across different classes of engineered systems.

5.3 Aircraft Engine

The underlying principle of a geared turbofan engine, referred here as the *New Architecture,* is to further increase bypass ratio over current dual-spool design, referred here as the *Old Architecture*, in order to improve propulsive efficiency (specific fuel consumption), decreasing noise and hopefully weight at the same time [Riegler and Bichlmaier, 2007]. This can be achieved by reducing fan speed and pressure ratio for high bypass ratio fans, and increasing low-pressure compressor (LPC) and low-pressure turbine (LPT) speeds, thereby achieving very high component efficiency. Propulsive efficiency of a turbofan engine is primarily dependent on bypass nozzle jet velocity for a given flight condition. Propulsive efficiency will be high when bypass jet velocity is low. This can be achieved by low fan pressure ratio which requires a large fan diameter for a given thrust demand. Therefore, the fan rotational speed has to be reduced to keep the fan tip speed below the supersonic level. The final outcome of applying these design criteria is a high bypass ratio turbofan engine with low thrust-specific-fuel-consumption (TSFC) and lower specific thrust [Malzacher et al., 2006; Riegler and Bichlmaier, 2007]. Along with the low bypass jet velocity comes low jet noise, and because of the correspondingly slow fan speed, the fan emitted sound pressure level and therefore the noise level is low for the geared turbofan configuration or the New Architecture. The two architectures compared are two potential embodiments of large

commercial gas turbine aircraft engines. The comparison is of interest because of the potential benefits of the geared turbofan architecture in fuel burn and noise. The impact of this architecture on the organization, design and development process would then stem from the differences in the relative complexity of the architecture, as well as the organizational and process boundaries overlaid on the architecture.

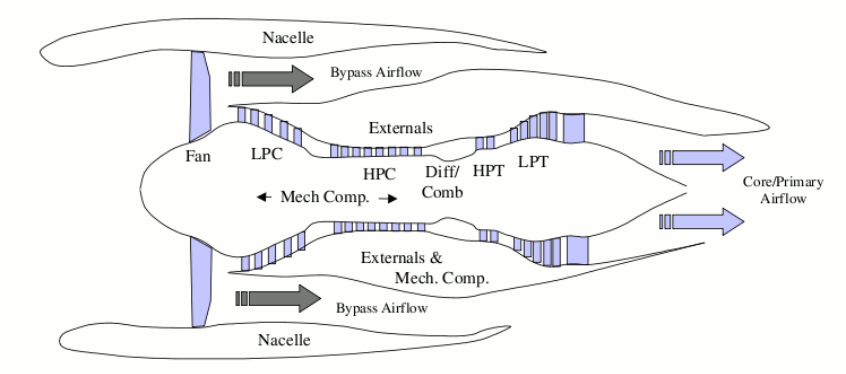


Fig. 5.16: Typical commercial gas turbine engine with major functional modules identified.

Both are axial flow, high bypass ratio gas turbine engines. Both engine architectures fundamentally provide similar primary function like thrust generation to the customer, though magnitude of maximum generated thrust are different (see Fig. 5.17 below).

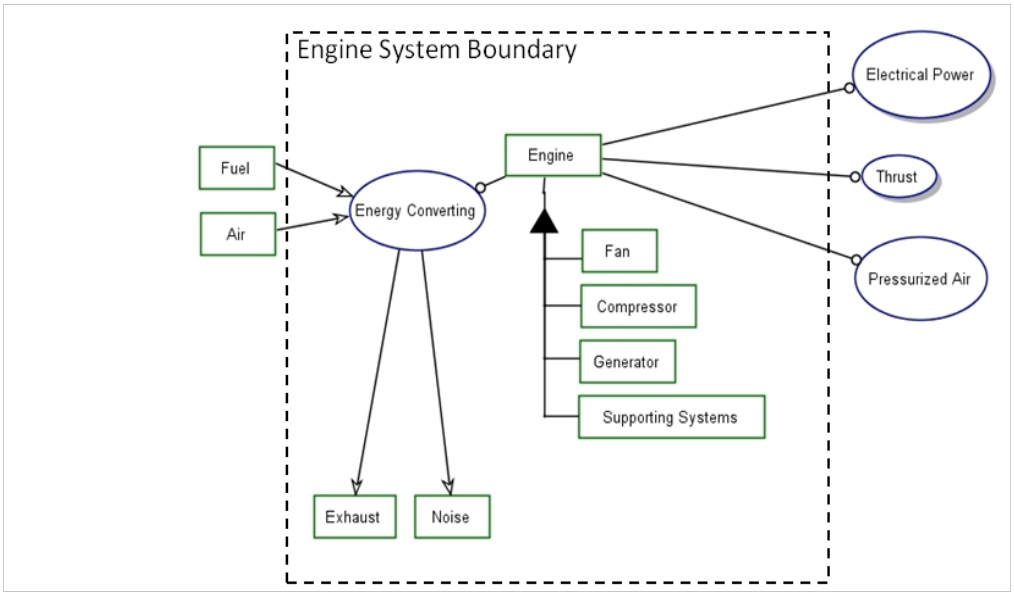


Fig. 5.17: Three primary propulsion system delivered functions

They arrive at that function through different architectural arrangement. In the next sub-section, the Design Structure Matrix (DSM) construction procedure is explained for these two cases.

5.3.1 Aircraft Engine DSM Development

The DSM was constructed for the two engines to allow comparison of the two architectures. This then provides the following benefits [Ulrich, 1995]:

- Measurable framework for comparison of architecture between the GTF and traditional turbofan platforms.
- Provide a platform to perform modularity analysis using the different algorithms
- Provide a platform to overlay the architecture, modules and components on the organizational structure, to determine how the architecture may impact the design and development process through organizational interactions.

Development of a DSM for a complex system requires that a level of abstraction be made. The level of abstraction must align with the analysis being performed [Suh *et al.*, 2010]. The DSM is created with an abstracted view of the components to provide the ability to assess the system complexity as well as the organizational connectivity between teams responsible for the design and development of the engine components. Components for the DSM were selected based on their need for inclusion as a result of the functional decomposition of the engine. While the engines studied are designed for significantly different airframe applications, the degree of abstraction of function allows comparison because of the similarity of the product application. This need is met through addressing both the "scope" and "granularity" of the matrix [Suh *et al.*, 2010]. A balance is needed in having sufficient detail to perform the required analysis, without making the DSM generation process so cumbersome as to be a design and development process in itself. This DSM generation method is reflected in the system level decomposition, which can be seen to clearly apply equally to both platforms. Components

represented in the DSM were selected based on this functional representation of the system. Experience was that a matrix with approximately 75 to 85 components was sufficient to represent complexity of a large scale printing system [Suh *et al.*, 2010]. The DSM generation proceeded without limitation to the number of components. We ended up with 73 components, which adheres the above guideline of approximately 75 to 85 components. In the DSM constructed, the multiple airfoils per stage, and multiple stages per module were simplified to one. Repeated features in the architecture are not believed to add architectural complexity, and are not included. For example, the bearing compartments were aggregated to provide the rotor supporting function and connect the rotors (rotating) to the static (cases), as well as provide all of the internal supporting functions (e.g., lubrication, sealing, and power extraction) [James *et al.*, 2011].

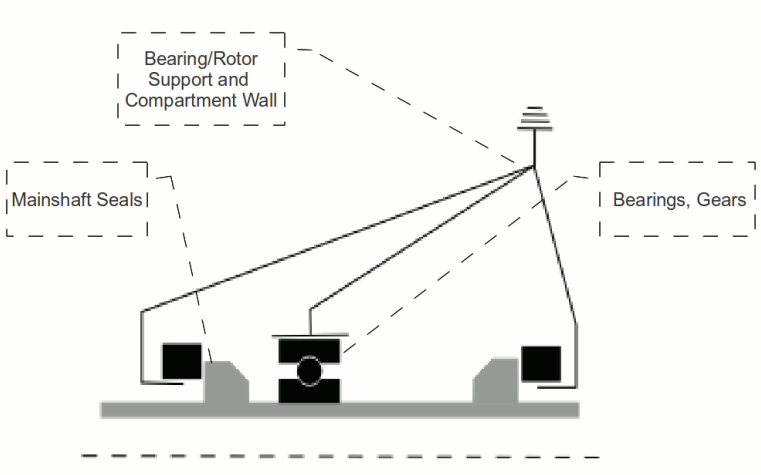


Fig. 5.18: Aggregation of a Typical Bearing Compartment

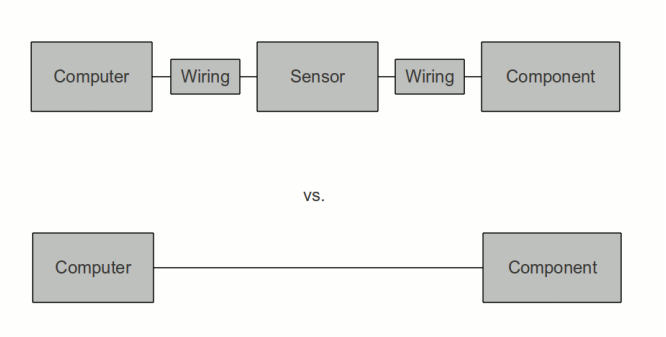


Fig. 5.19: Example of Aggregation of Component Abstraction

With the components that define the system architecture identified, the connectivity of those components to map the value delivery internal to the machine was populated through the matrix using connectivity mapping and encoding scheme. The challenge in creating an effective DSM was to prune the component list to only what was functionally required to represent the architecture of the engines for comparison purposes [James *et al.*, 2011].

5.3.2 DSM Encoding

Many DSM's used to date are binary, and represent connectivity of the components or process simply by indicating if a connection exists or not. In order to develop a deeper understanding of the gas turbine engine, a more detailed approach is taken using a "quad" connection structure is utilized [Suh *et al.*, 2008]. This provides the ability to analyze the network from different views, and to segregate relationships based on connection type – which may have different impacts on the design and development of the machine, and also will likely be represented by different experts in the design process – which will aid in the investigation of the architectural impact on the social layer interactions.

In addition, the different types of flows (core flow, bypass flow, fuel flow, oil flow and secondary flow) are critical to understanding the energy flow through a gas turbine engine, and this refinement is proposed in this thesis as a method of adding further detail to the DSM. To capture the benefit of having this information stored in the DSM, a scheme was developed to "encode" all of the information into a single integer based on a 2^n - 1encoding scheme.

The quad based DSM structure (mechanical connection, flow connection, information, and energy) could then be generated in a spreadsheet such as Excel, and then "encoded" into a square adjacency matrix of connections for network analysis and visualization. Tools to facilitate the encoding and decoding of the matrices were developed in *Perl*. In order to represent the different types of flows in the DSM, each quantity to be represented was given an integer number of the scheme 2^n - 1. Each connection between components has one or more of the basic

encoding types, with additional detail added by using the detail encoding in addition to the basic encoding. The following scheme is used [James *et al.*, 2011]:

Table 5.12: Basic encodings for the gas turbine engine DSM

n	Flag	Flow Type	Description
0	0	None	No connection
1	1	Mechanical	Physical coupling between components. This is by nature symmetric.
2	3	Fluid Flow	Flows of any fluid between two components
3	7	Information	Information transfer between components. Generally assumed to be electronic measurement for sensors, etc.
4	15	Energy	Energy transfer of any energy type.

Table 5.13: Detail level encoding for the gas turbine engine DSM

n	Flag	Flow Type	Description
5	31	Gaspath flow	Flow through the engine "core" which passes through the compressors and turbines
6	63	Bypass flow	Flow through the fan only, bypassing the engine core
7	127	Secondary Flow	Air flow taken off of the gas path or bypass flows and used for component cooling or pressurization
8	255	Fuel flow	Fuel flows through the fuel system. Ends at the fuel nozzles, exhaust products are considered gas path flow.
9	511	Oil flow	Oil flows through the lubrication system.
10	1023	Torque	Transfer of torque between components
11	2047	Electrical Energy	Transfer of electrical energy between components
12	4095	Chemical Energy	Transfer of chemical potential energy between components. Aides in visualization of energy transfer pathways and conversion of chemical potential to thermal energy.
13	8191	Thermodynamic Energy	Transfer of thermodynamic energy between components, including both pressure and temperature, generally considered enthalpy. Used for gaspath flow energy transfer.
14	16383	Hydraulic Energy	Transfer of pressure energy between components. While this could be considered part of thermodynamic energy, this is used for hydraulically actuated systems that operate on pressure differentials.

The DSM's were generated for both the engines with similar levels of aggregation. The traditional turbofan DSM has 69 components, and the new generation geared turbofan DSM has 73 components. The size of the two matrices is close enough for comparison purposes, and since they were developed with the same guidelines for aggregation this is believed to represent the architecture properly for this purpose [James *et al.*, 2011].

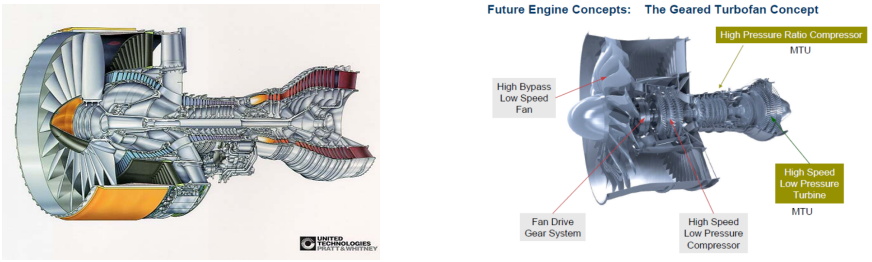


Fig. 5.20: Schematic of old and new aircraft engines.

The engine architectures, captured using DSM, were developed based on (i) system documentation; (ii) physical inspection and selective disassembly; and (ii) opinions of subject matter experts. This is a very intensive, manual process requiring significant time [Suh *et al.*, 2010]. This is primarily due to lack of digital tool support at the moment and we hope this situation will improve in future, leading to quicker and automated generation (at least partly automated) of the system adjacency information from product design and development systems. The aggregated DSMs for the two engine architectures are shown in fig. 5.21 and fig. 5.22 below.

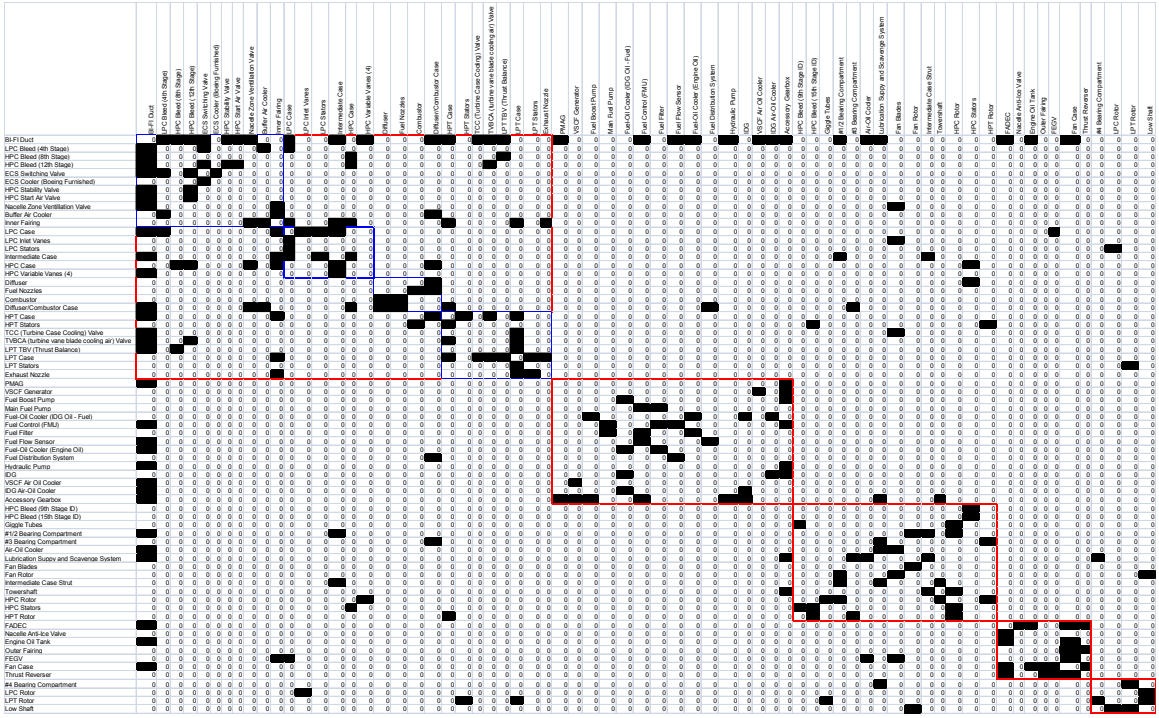


Fig. 5.21: DSM of old architecture. The bordered partitions indicate the 5 modules found by applying the modularity metric [Girvan and Newman, 2002].

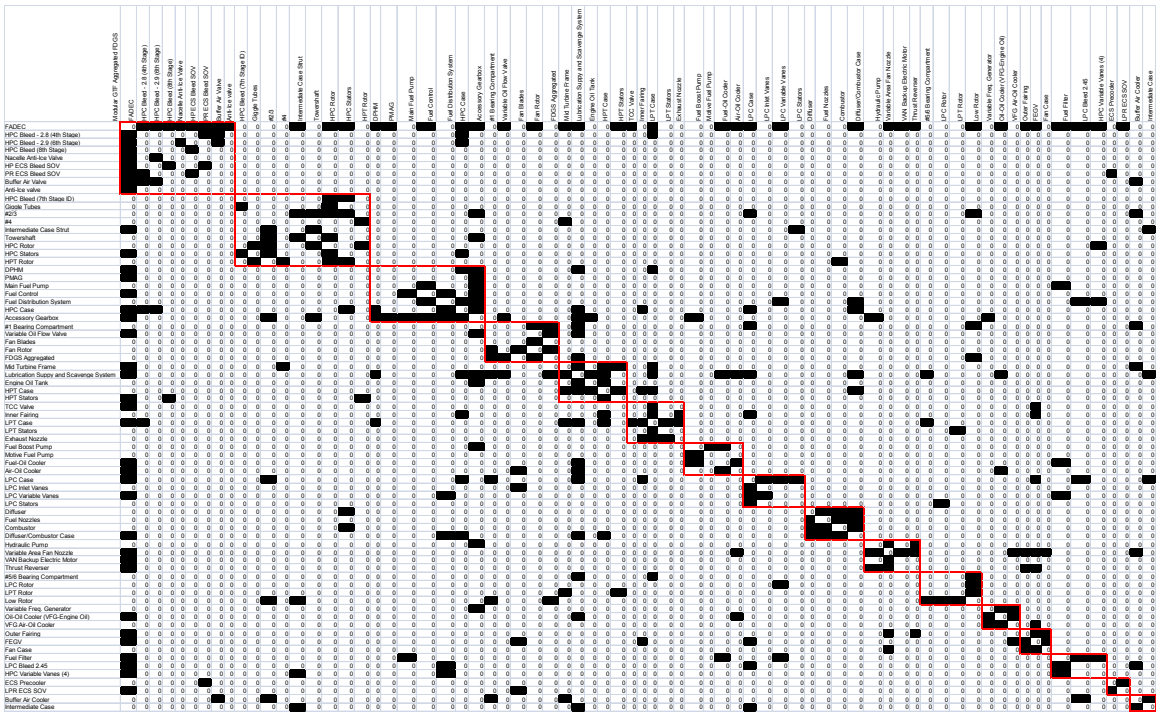


Fig. 5.22: DSM of new architecture. The bordered partitions indicate the 16 modules found by applying the modularity metric [Girvan and Newman, 2002].

5.3.3 Structural Complexity Estimation

The two architectures represented above (see fig. 5.21 – 22) shows some fundamental differences in density and connectivity. This is likely attributed to the higher integration of the geared turbofan, a smaller and modern engine. This increase in structural complexity can be seen visually in a comparison of the two DSM's, and is also demonstrable through computed metrics.

Table 5.14: Comparison of two engine architectures.

	Older Architecture	New Architecture	Change
#Components	69	73	6%
Connection Density	5.73%	6.87%	20%
#Connections (all)	269	361	34%
#Components + #Connections	338	434	28%
Mechanical	240	326	36%
Information	47	48	2%
Energy	58	60	3%
Flow	87	105	21%

Graph Energy, E(A)	104.4	123.3	18%
Topological Complexity, C_3	1.51	1.69	12%
Modularity Index (Q) [Girvan and Newman, 2002]	0.43 (5 Modules)	0.35 (16 Modules)	-20%

The DSM for the new architecture shows significantly more connectivity in all areas measured, with a 20% increase in connection density of the DSM. The individual connection types all increased in number, reflecting a more inter-connected architecture. The largest increase, 36%, is found for the mechanical connections. The increase in graph energy E(A) and the topological complexity C_3, indicates that the system is more distributed than the traditional turbofan architecture represented. The topological complexity measure is still far from the critical value 2, characterized as the *P* point (see chapter 3).

The modularity analysis performed for both engine architectures using the total connectivity of the DSM showed that there are many potential modules. Some of the modules have relatively few components, because of their high relative internal connectivity strength. There is a strong indication that the geared engine architecture is more distributed and the modules are not very prominent (see fig. 5.21 and table 5.14) and have smaller size, in comparison to previous generation dual-spool engine.

Apart from considering the connectivity structure, we have to estimate the component and interface complexities in order to compute the overall Structural Complexity for these two architectures. The component complexities are assessed and reflect the internal complexity of the individual components in the DSM. Due to unavailability of historical data with adequate level of validity, we resorted to an expert opinion based estimation to come up with the most likely estimates of component complexities, α . These were determined by *expert* review, using a scale [0,5] with 5 being the highest complexity level. The experts were the senior engineers associated with the engine development programs. In all cases, the triangular distribution representing component complexity was right-skewed as shown in Fig. 5.22 below.

This is expected as experts tend to be more optimistic in general and their most likely estimates are influenced by an inherent bias. There could be various socio-political reasons that shape such behavior and are likely shaped by the expert's operating environment [Garvey, 2000]. In case of the *new* engine architecture, the range of pessimistic estimates of component complexity for the new engine architecture tended to be about 1.1 to 4 times that of the most likely estimates, while the optimistic estimates were about 0.8 to 0.9 times that of the most likely value. Such right-skewed estimates can be represented diagrammatically as shown in fig. 5.23 below. There were cases where optimistic values were only 10% lower than the most likely estimate while the pessimistic value was 4 times the most likely estimate.

This indicated a very high external uncertainty about certain novel components used in the new engine architecture. There is a possibility of significantly exceeding the most likely estimate, but the probability of such occurrence is thin. It hints at existence of heavy-tailed, one-sided uncertainties that influences component complexity. This leads to extreme values of component complexity.

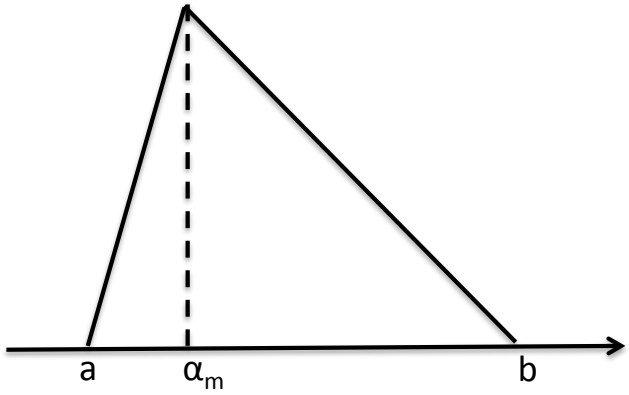


Fig. 5.23: Right-skewed triangular pdf: $(b-\alpha_m) = p(\alpha_m-a)$ where $a \in [0.8\alpha_m ; 0.9\alpha_m]$ and $p \in [1.0; 15.0]$. This means that $b \in [1.1\alpha_m; 4.0\alpha_m]$

Based on the data collected from experts, the following relationship linking the three estimates (see eq. 5.3 below) required to define a triangular distribution

can be well approximated by using the parameter ranges $p \in [1.0;\ 15.0]$ and $a \in [0.8\alpha_m;\ 0.9\alpha_m]$.

$$(b - \alpha_m) = p(\alpha_m - a) \qquad (5.3)$$

The range of values for parameter *p* reflects the range of values obtained for the most pessimistic component complexity values.

For computing interface complexity, we used the following relation,

$$\beta_{ij}^{(k)} = \frac{\max(\alpha_i, \alpha_j)}{c^{(k)}} \qquad (5.4)$$

where $1/c^{(k)}$ stands for the interface factor for connections of type/category, *k*. For the engine examples, the connection sub-types, as described before, are listed with their corresponding interface factors in table 5.15 below. The above relationship means that the interface complexity is only $[1/c^{(k)}]$ times that of the maximum of interfacing component complexities. For example, interface factor of 0.05 indicates that the interface complexity is only 5% of the largest complexity of the interfacing components. In this case study, we relied on expert opinion and additional anecdotal data to come-up with these estimates since historical data were not available. The same interface factors were used in computing the structural complexity for both, the old and the new engine architecture.

Table 5.15: Interface factor for different connection sub-categories used in jet engine example.

Connection type/sub-type	**Interface factor, $1/c^{(k)}$**
Mechanical connection	0.05
Information/Control	0.20
Gaspath flow	0.25
Bypass flow	0.2
Secondary flow	0.10
Fuel flow	0.15
Oil flow	0.10
Electric energy	0.15
Chemical energy	0.25
Thermodynamic energy	0.25
Hydraulic energy	0.2
Rotational energy	0.15

The computed structural complexity numbers for the new engine architecture are shown in table 5.16 below. Due to this highly asymmetric, right-skewed distribution of component complexities, the total component complexity, C_1 has a large right-skew. The total interface complexity, C_2 was computed using the interface complexity computation eq. 5.2 in conjunction with interface factors from table 5.13. Here the system connectivity is deterministic with C_3 = 1.69. The computed structural complexity therefore has a distribution that is also right skewed to an extent. We compute the structural complexity for most likely values and list the mean, median and 70 percentile values for structural complexity, C and also its constituents (C_1, C_2, C_3). The values are listed in table 5.14 below.

Table 5.16: Structural Complexity and its constituents using most likely estimates of component complexities and the mean, median and 70 percentile values from respective distributions for the new engine architecture. Here, C_{ML} is the Structural Complexity using most likely estimates.

	C_1	C_2	C_3	C	C/C_{ML}
Most Likely	188	184	1.69	499	1
Mean	244	240.4	1.69	650.3	1.30
Median	242	238.9	1.69	646.8	1.29
70 percentile	247.9	246.2	1.69	663.94	1.33

In case of the *old* engine architecture, the uncertainties around most likely estimates of component complexity were less striking with the most pessimistic values being 10% to 100% higher than the most likely estimates. Also, higher uncertainties were both sides (i.e., component with higher pessimistic value also had a lower optimistic value than the most likely estimates). The parameter range in case of the older engine architecture were $p \in [1.0;\ 5.0]$ and $a \in [0.8\alpha_m;\ 0.9\alpha_m]$. Using eq. 5.3, we compute the ranges for the most pessimistic estimate, $b \in [1.1\alpha_m;\ 2\alpha_m]$. Hence, in case of the old engine architecture, the range of pessimistic estimates of component complexity varied between 1.1 to 2 times that of the most likely estimates only. Compare this to the case of new engine architecture where the variation was more than 3 times. This is expected due to the nature and

novelty of components in the new engine architecture. The optimistic estimates were still about 0.8 to 0.9 times that of the most likely value (see fig. 5.22).

In table 5.17 below, we compare the structural complexity for the two engine architectures. While using the most likely estimates indicates a 42% increase in structural complexity for the new engine architecture, other representative measures of central tendency like mean and median shows more than 65% increase in structural complexity. Using the 70-percentile measure as the representative value, we observe an increase by factor of nearly two-thirds. This clearly reflects the influence of higher, asymmetric uncertainty in quantifying component and interface complexities. A higher, right-skewed uncertainty characterization leads to even sharper increase in structural complexity as we consider 70-percentile values for comparison, as opposed to the most likely values for component complexity. Higher complexity has led to a nonlinear increase in the development effort/cost, but the actual numbers are highly confidential and not publicly available.

Table 5.17: Comparison of Structural Complexity for two engine architectures. Here, C_{ML} is the structural complexity based on most likely estimates.

	C_1		C_2		C_3		C		C/C_{ML}		C_{new}/C_{old}
	Old	**New**	**Old**	**New**	**Old**	**New**	**Old**	**New**	**Old**	**New**	
Most Likely	161	188	126	184	1.51	1.69	351	499	1	1	1.42
Mean	179	244	141	240.4	1.51	1.69	392	650.3	1.12	1.30	1.65
Median	178	242	139	238.9	1.51	1.69	388	646.8	1.10	1.29	1.66
70 percentile	181	247.9	145	246.2	1.51	1.69	399.6	663.94	1.14	1.33	1.66

The detailed architectural analysis including the sensitivity analysis procedure, outlined in chapter 3, reveals that primary functionality generators (e.g., generating thrust) are significant contributors to component complexity while supporting systems (e.g., lubrication systems, accessory gearbox, robust control systems) are the primary contributors to architectural complexity and having significant impact on system integration efforts. Most of the rotating components showed low sensitivity to the structural complexity. Supporting systems like the

engine control system, lubrication system, and accessory gearbox are found to be more sensitive to structural complexity.

Chapter Summary

In this chapter, application of the methodology, developed in chapter 3, was demonstrated. The applications included simpler systems to very complex, current generation aircraft engines. Basic inputs required before we can compute the structural complexity included generation of the system connectivity structure using the Design Structure Matrices (DSM), and estimation of component complexity and interface factors.

Lack of historical data (or accessibility to the data) prevented us from applying the data analytics based method of component and interface factor estimation for the real-world case studies and as a result, expert opinion based methodology were employed to achieve the goal. Application of the methodology demonstrated the impact of uncertainty propagation in structural complexity estimation. The right-skewedness of the component complexity distribution manifested itself by magnifying the structural complexity as we go from most likely value based estimate to mean value of the structural complexity distribution or the 70-percentile measure. Please note that the effect of right-skewedness actually mellows down as we sum the component complexities (they are random variables). This is manifestation of the law of large numbers.

In real-world engineered systems, there is significant possibility of exceeding the projected complexity of components, while the optimistic estimates are likely to be quite close to the projected (i.e., most likely) estimate of component complexity. This hints at existence of significant external uncertainty, but with small probability of occurrence. This makes us believe that there are external factors with fat-tailed distributions that influences component complexity. This appears to be typical of engineered complex systems and well supported by the data found in the existing literature [Garvey, 2000]. The uncertainty propagation aspects have been demonstrated empirically in this chapter. Given the analytical probability

distribution functions for component and interface complexities, it is possible to derive the probability distribution function for structural complexity.

The effect of the system decomposition level was investigated using a coarser and a finer representation of the same digital printing system and showed that the topological complexity metric remains nearly invariant to the system decomposition level.

Also notice that in no engineered system covered herein, the topological complexity measure, C_3 reached the critical value of 2, characterized by the *P* point. This meant that none of them lie within the topologically hyperenergetic regime (see chapter 3). But some of them (e.g., the new print engine) could be seen approaching that regime, which is predicted to represent significantly higher system integration challenges and likely other associated effects including system robustness against structural failure in system components.

References:

Danilovic, M., & Browning, T. R. (2007), "Managing complex product development projects with design structure matrices and domain mapping matrices",International Journal of Project Management, 25(3), 300-314.

Eun Suk Suh, Michael R. Furst, Kenneth J. Mihalyov and Olivier L. de Weck "DSM Model Building Instructions" – working paper, 2009.

Suh E.S., Furst M.R., Mihalyov K.J. de Weck O.L., "Technology infusion for complex systems: A framework and case study", Systems Engineering, 13: 186–203, 2010.

James D., Sinha K., de Weck O.L, "Technology Insertion in Turbofan Engine and assessment of Architectural Complexity", 13th International Dependency and Structure Modeling Conference, 2011.

Riegler, C., and Bichlmaier, C., "The Geared Turbofan Technology – Opportunities, Challenges, and Readiness Status," 1st CEAS European Air and Space Conference, 10-13 September 2007, Berlin, Germany.

Malzacher, F.J., Gier, J., and Lippl, F., "Aerodesign and Testing of an Aeromechanically Highly Loaded LP Turbine", Journal of Turbomachinery, 2006, Vol. 128, No. 4, pp 643-649.

Sinha, K., & de Weck, O. (2009). Spectral and Topological Features of 'Real-World' Product Structures. Proceedings of 11th International DSM Conference, 2009.

Sinha K., de Weck O.L, "Structural Complexity Metric for Engineered Complex Systems and its Application", 14th International DSM Conference, 2012.

Ulrich K. T., "The role of product architecture in the manufacturing firm", Research Policy, 24, 1995, pp. 419-441.

Girvan M. and Newman M. E. J., "Community structure in social and biological networks", Proc. Natl. Acad. Sci. USA **99**, 2002.

Whitney D. E., "Physical limits to modularity", Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division, 2004.

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Tilstra A.H., "Representing Product Architecture and Analyzing Evolvable Design Characteristics", PhD Thesis, University of Texus, Austin, 2010.

Chapter 6

Complexity Management and Systemic Implications

In the previous chapters, we looked at the formulation and important properties of the structural complexity metric; its theoretical verification and empirical validation; and its application to large, real-world engineered complex systems. In this chapter, we introduce the notion of *complexity budgeting* using value function, look at the aspect of granularity of system representation and its impact on structural complexity estimation, introduce the notion of distribution of complexity across the system, including an analytical look at the relationship between structural complexity and structural modularity and discuss the associated systemic implications through some real-world examples.

6.1 Complexity Management

Until now we have looked computing structural complexity. Once we have a quantification process and methodology in place, the next question is what to do with this quantity. The structural complexity has to be estimated and tracked throughout the system development activity and any deviation serves as an indicator for possible overshooting in terms of programmatic parameters like cost or effort.

From a programmatic perspective, one question that remains is about fixing the desired level of complexity and how should we go about fixing a *complexity budget*, similar in spirit to the mass budget or power budget used, for example, in aerospace system development projects.

We describe below a suggested methodology for fixing the *complexity budget* for engineered system development, based on the value function that links system performance and Non-Recurring Engineering cost (NRE).

6.1.1 Complexity Budget

Much like the mass or power budgets used in traditional engineering system development, we can think about a notion of *complexity budget*. *Complexity budget* refers to a level of complexity that is most beneficial for the project from a value perspective where we gain performance while keeping NRE cost/effort within prescribed/manageable limits.

The relationship between system performance and complexity can be modeled using S-curve with parameters (n,k) as shown below,

$$P = P_{\max}\left(\frac{kC^n}{1+kC^n}\right) \qquad (6.1)$$

Please note that the system performance is usually a composite measure, using a linear combination of multiple performance parameters. The system performance level is assumed to saturate at P_{max} (see fig. 6.1). As can be seen by analyzing eq. 6.1, higher *n* indicates higher rate of performance gain and saturation at a lower complexity level, while lower *k* shifts the curve towards the right (see fig. 6.2). The relative impacts of parametric variation in performance-complexity profile is explained in fig. 6.2. Here, *n* is defined as the *rate of performance gain* and *k* is the *shift* parameter. As seen in chapter 4, the NRE cost/effort can be well estimated by a monotonically increasing, nonlinear curve,

$$NRE = aC^m \qquad (6.2)$$

where (a,m) are the parameters (see fig. 6.1). Please note that, the parameter *m* is defined as the *rate of complexity penalty* and captures associated *complicatedness*.

We define a value function, V that expresses the performance gain per unit NRE expenditure. This is much like a price for enhanced performance and this price

is being paid to counter increased complexity. Hence, we can interpret the value function as the complexity price for performance gain. Once the performance gain saturates, any increase in complexity is counter productive as we have to pay the complexity penalty in terms of NRE, without extracting any performance benefits. This leads to erosion in system value function.

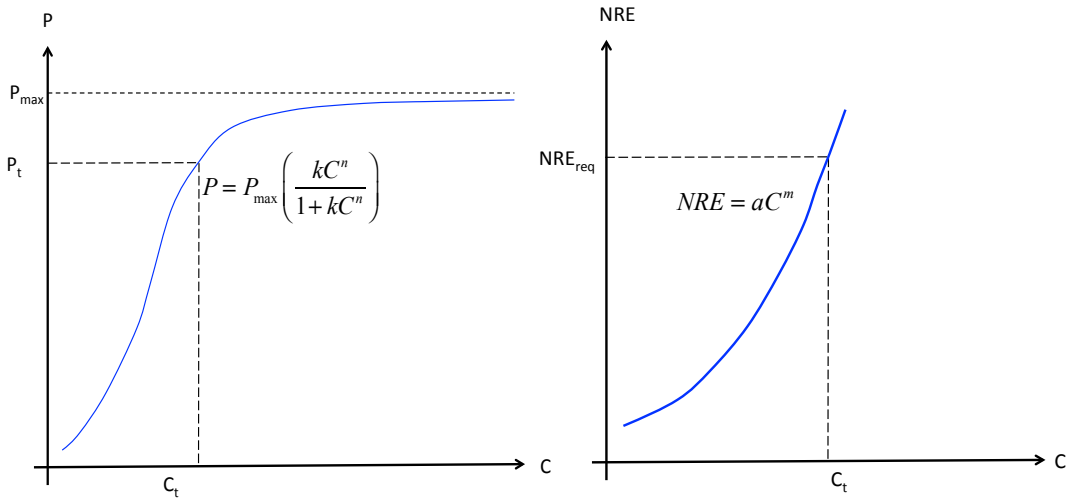


Fig. 6.1: The relationship between complexity and (i) performance; and (ii) NRE cost/effort.

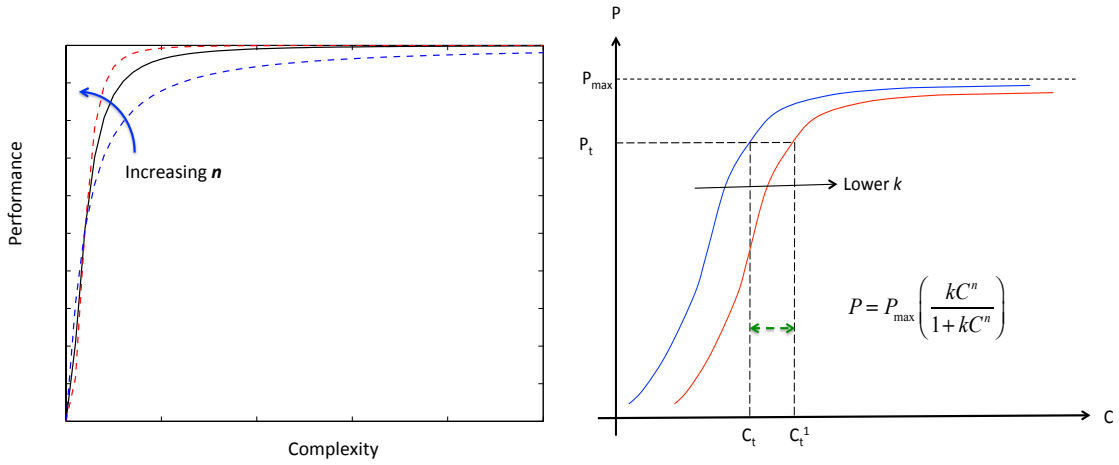


Fig. 6.2: (i) Increasing *n* means higher rate of performance gain and performance saturation at a lower complexity level; (ii) lower *k* shift the performance-complexity curve to the right, indicating higher complexity level to attain the same performance level.

The effect of parameter *m* on the NRE in eq. 6.1 is shown in fig. 6.3 below. We can interpret the exponent *m* as the rate of NRE penalty for increased complexity. As

seen in Chapter 4, a smaller value of *m* indicates better complexity handling capability (i.e., lower *complicatedness*). The other parameter, *a* acts as a scaling factor.

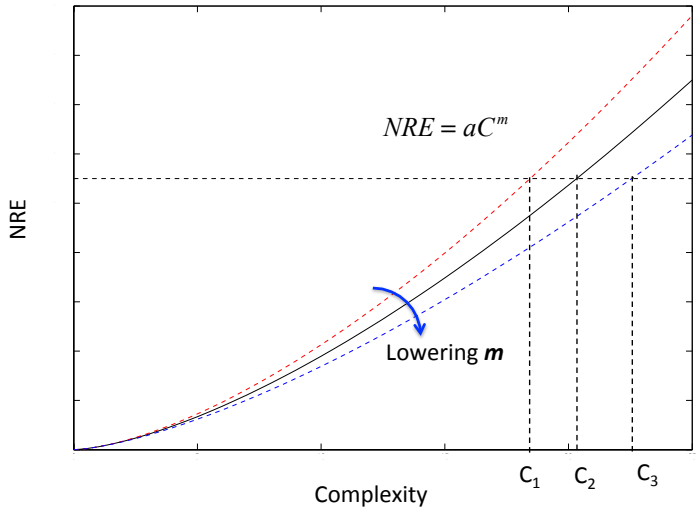


Fig. 6.3: Lower *m* value indicates smaller NRE for the same complexity level, indicating a lower rate of NRE penalty for increased complexity. A lower *m* value indicates better complexity management and lower perceptual complexity/ complicatedness for the development team (see chapter 4 for details).

Now, using the above functional forms, the value function V can be written as,

$$V = \frac{P}{NRE} = \underbrace{P_{\max}\left(\frac{k}{a}\right)}_{S}\left[\frac{C^{(n-m)}}{1+kC^n}\right]$$

$$= S\left[\frac{C^{(n-m)}}{1+kC^n}\right] \qquad (6.3)$$

We can observe that, for $n \le m$, the value function decreases monotonically for increasing complexity, C. If we compute its first derivative with respect to complexity, we have

$$\frac{dV}{dC} = -\frac{S}{(1+kC^n)^2 C^{(m-n+1)}}\left[(m-n)+kmC^n\right] \qquad (6.4)$$

It is clear that $\frac{dV}{dC} < 0$ for $n \leq m$. The rate of decreasing value is higher for $n < m$ and can be seen in fig. 6.4. This means that, if this condition is satisfied, the value will always decrease and it does not pay (from a system value standpoint defined here) to increase the complexity level of the system.

The interesting case is when $n > m$, indicating that the initial rate for performance gain outweighs the rate of NRE penalty for increased complexity. Writing the optimality condition below, we compute the complexity level, C_* for value maximization,

$$\frac{dV}{dC} = 0$$

$$\Rightarrow \frac{SC_*^{(n-m-1)}}{(1+kC_*^n)^2}\left[(n-m)+kmC_*^n\right] = 0$$

$$\Rightarrow \frac{n-m}{C_*} = \frac{knC_*^{n-1}}{1+kC_*^n}$$

$$\Rightarrow C_*^n = \frac{n-m}{km}$$

$$\therefore C_*^n = \frac{\left(\frac{n}{m}\right)-1}{k} \qquad (6.5)$$

It can be shown that $\left.\frac{d^2V}{dC^2}\right|_{C_*} < 0$ (see appendix L for details), guaranteeing that the value function V is indeed maximized at $C = C_*$. This result is graphically demonstrated in fig. 6.4. This means that, in contrast to the previous case, there is value enhancement in increasing the complexity up to a point. The corresponding performance level P_* is given by,

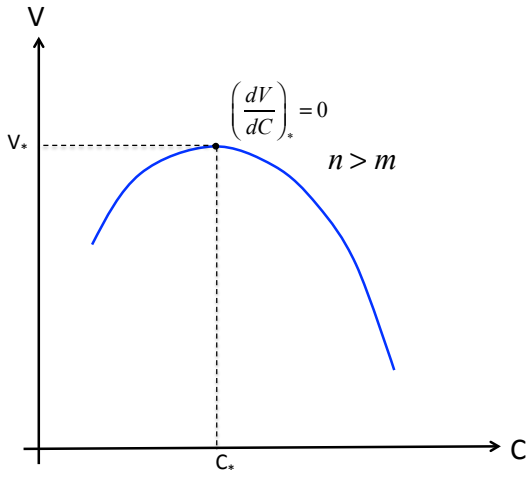
$$P_* = P_{\max}\left(\frac{kC_*^n}{1+kC_*^n}\right)$$
$$= P_{\max}\frac{\left(\frac{n}{m}-1\right)}{\left(\frac{n}{m}\right)}$$
$$= P_{\max}\left(1-\frac{m}{n}\right) \qquad (6.6)$$

Eq. 6.6 indicates that P_* reaches $P_{\max}$ as n/m ratio increases. Large n/m ratio indicates sharper performance gain with better complexity handling/management capability. Also a smaller k combined with larger n/m ratio leads to higher complexity level, C_* at maximal value function.

The corresponding NRE value at this complexity level is given by,

$$NRE_* = aC_*^m = a\left(C_*^n\right)^{\frac{m}{n}}$$
$$= a\left[\frac{\left(\frac{n}{m}\right)-1}{k}\right]^{\frac{m}{n}} \qquad (6.7)$$

This expression is more complicated as m/n ratio appears on the exponent, but a larger n/m ratio usually leads to a smaller NRE.



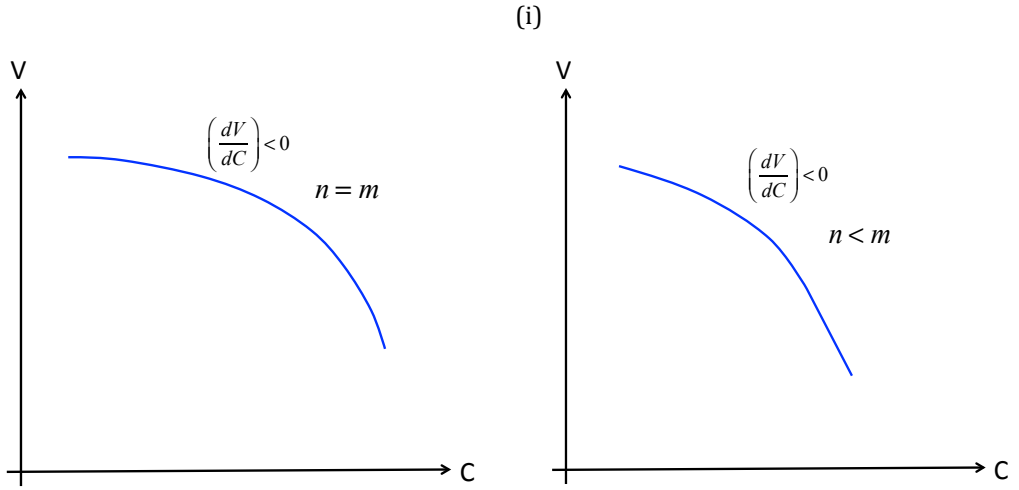


Fig. 6.4: Plots showing the value-complexity curves for (i) $n > m$; (ii) $n = m$; and (iii) $n < m$. For $n \leq m$, the value function is monotonically decreasing with complexity

The maximized system value, $V_{\max}$ is given by,

$$V_{\max} = \frac{SC_*^{n-m}}{1+kC_*^n} = \left(\frac{m}{n}\right)SC_*^{n-m} = \left(\frac{m}{n}\right)S\left[\frac{\left(\frac{n}{m}\right)-1}{k}\right]^{\left(1-\frac{m}{n}\right)} \qquad (6.8)$$

A large n/m ratio and low k value usually helps attain a larger maximum value function, $V_{\max}$.

For any required level of system performance, P_t to be achieved given the Performance-Complexity and NRE-Complexity curves, we can trace NRE_{req} as the expected NRE cost/effort as shown in fig. 6.5 below. If the budgeted NRE, NRE_{budget} is smaller than the estimated NRE_{req}, then we have to look for ways to reduce the value of exponent m and shift the NRE-Complexity curve. This calls for improved

complexity management strategies and this might prove to be unachievable under the circumstances.

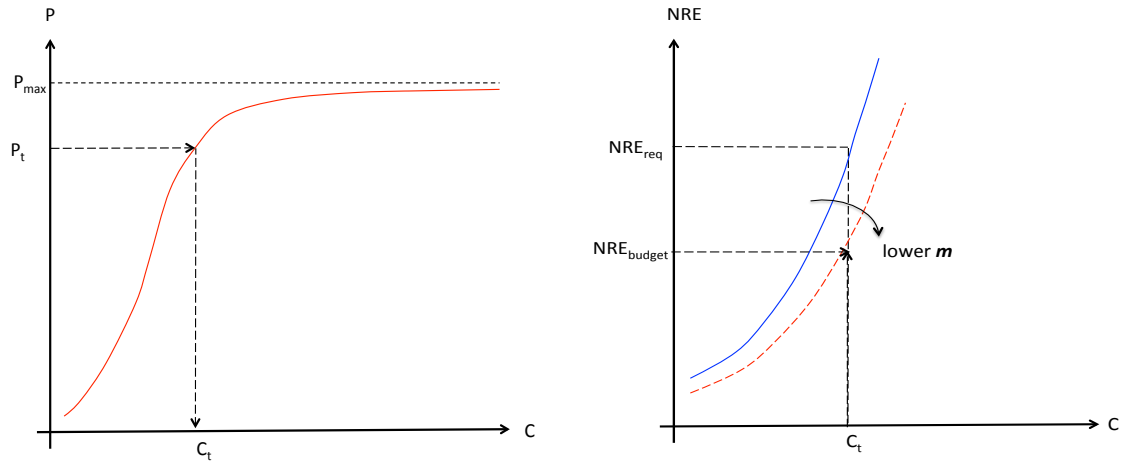


Fig. 6.5: Trace the trade-space for a given system performance target, P_t : (i) find the requisite complexity, C_t and (ii) find the NRE_{req} to achieve the specified level of performance for a given NRE-Complexity curve. If $NRE_{reg} > NRE_{budget}$, then improved complexity management is mandated to lower the *m* value and influence a shift in NRE-Complexity profile.

In such case, one needs to explore other options including compromising on the system performance targets or to look for different system architectures for which a higher rate of performance gain (i.e., exponent *n* in eq. 6.1) is achievable.

Now once we set a budget for structural complexity following the procedure above, there are different ways to distribute this total structural complexity into its three components. As discussed in chapter 3 under *isocomplexity*, there are trade-offs based on complexity handling capabilities at the organizational level. For example, an organization with excellent system integration capability might opt for a complex or more distributed architecture with simpler components as opposed to one with complex components and simpler architecture (requiring higher component engineering capability), provided both forms are capable of achieving the system goals. Also, there may be significant differences in terms of complexities of individual components and one organization may be better equipped to handle it, compared to a different organization. In general, complexity consolidated at the component level gives an impression of reduced complexity (i.e., perceptive complexity or complicatedness is reduced, not necessarily the actual complexity).

We tend to perceive the overall interface design and management as more complex, although individual interfaces may be much simpler than the connecting components. It may be due to the higher number of interfaces and the ability to understand/interpret their topological connectivity patterns. Why does this happen? Is it due to the enhanced focusing on a fewer number of component that might have high complexity? We believe this is where complexity management meets the cognition science.

6.1.2 Complexity and Cognition

In the process of building simple ball and stick models (see chapter 4), we observe significant differences in individual model building times. The build time profiles vary quite significantly at the level of individual subjects. What is simpler for one subject might turn out to be difficult or even intractable for another. One would expect significant difference in build times at individual level and that such variance will increase with increasing structural complexity levels and this was validated using the experimental data in chapter 4.

This perceived complexity or *Complicatedness* is an observer-dependent property that characterizes an actor's / observer's ability to unravel, understand and manage the system under consideration. In contrast, *complexity* is an inherent system property and a given system may represent different degrees of *complicatedness* depending on the observer.

For example, the complexity of an automobile's automatic transmission may be hidden from a user and is perceived to be less complex. Therefore separating *complicatedness* from *complexity* improves the clarity by which systems can be described, analyzed and certain class of system observables (e.g., development cost, extent of reworks) be predicted.

We can think of *complicatedness* as a conduit through which *complexity* manifests itself at the level of system-level observables like the *system development cost* [Tang and Salminen, 2001]. Complicatedness provides insights to the cognitive aspects of the observer and his/her ability to handle a certain level of complexity.

We list the five main factors affecting complicatedness as (i) complexity; (ii) modularity or design encapsulation; (iii) novelty; (iv) cognitive capability and (v) cultural/organizational effects. Effects of each of these five factors are as follows:

(i) Complexity: Complicatedness is the degree to which an actor or decision unit for the system is able to manage the level of complexity presented by the system. Assuming other factors being equal, complicatedness ***K*** can be written as a function of complexity, ***K=g(C)***. We expect monotonicity of complicatedness with respect to complexity and at ***C=0, K=0***.

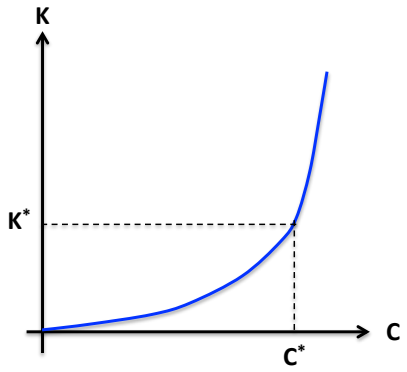


Fig. 6.6: Complicatedness vs. complexity: two different behavioral patterns, both with *knee* points defined by (C^*, K^*).

Intuitively, there is a level of complexity beyond which the observer can barely cope with the system and the system complicatedness becomes unmanageable (see Fig. 6.6). Hence, $\mathbf{K} \rightarrow \infty$ for $\mathbf{C > C^*}$.

(ii) Modularity or design encapsulation: Structural modularity or design encapsulation is a means of containing the complicatedness of a system. A well-architected system may hide inherent complexity in an effective fashion such that it appears less complicated. A good design or architecture always presents a less complicated system image to the actor's or decision units [Tang and Salminen, 2001]. Design encapsulation (notice that it also leads to information hiding) helps focusing attention on a subset of the system at a time. This is similar to ''chunking'' of information to circumvent the human cognitive span limitation [Hirschi and Frey, 2002].

(iii) Novelty: As an observer gains experience with a system, he/she starts developing knowledge and intuition about the system. A user can get more exposure with a system over time and deems it to be less complicated with passage of time, although the internal system remains unaltered. This seems quite natural if we view humans as *adaptive* systems. Humans continually update and adapt themselves as additional knowledge becomes available to new boundaries / constraints are discovered. This also applies in case of component or subsystem re-use in the new system, which drives the complicatedness down.

(iv) Cognitive bandwidth: Some actor's or decision units (i.e., group of individuals / team) may relate better to a more complex system than other actor's. This is reflective of the innate cognitive capability of an individual or a group of individuals to unravel the system, understand and manage the system. A high cognitive bandwidth on the part of the decision unit helps reduce complicatedness of a system for that decision unit.

(v) Cultural/organizational effects: This is a subtle factor. This factor includes organizational culture and also more broad based societal cultures. Some organizations or societies may have better ability to manage certain classes of complexities better than others, and thereby reduce the complicatedness of the system.

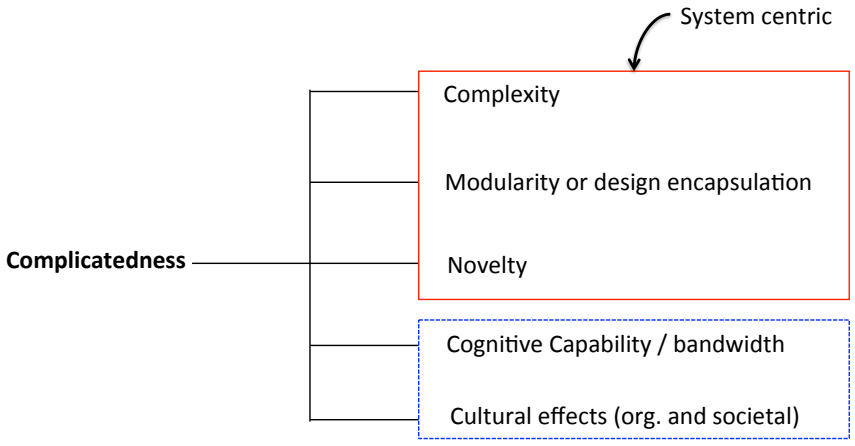


Fig. 6.7: List of factors influencing the *complicatedness function* or the perceived complexity – an observer dependent property that links structural complexity to development cost. It is a monotonically increasing function of structural complexity.

Looking closely at the factors listed above, we observe that factors (i), (ii) and (iii) are related to the system architecture and design, while cognitive bandwidth relates to human ability to handle a given complexity (see fig. 6.7). One might argue that cultural effects has more indirect influence through its impacts on factors (iii) and (iv), but it has direct effect in terms of providing a more encouraging environment for enhancing complexity handling capabilities.

Note on evolution of complexity handling capability: The complicatedness versus structural complexity curve (see fig. 6.6) reflects the complexity handling capability of an entity (i.e., an individual or a group). This curve suggests that the complexity handling capability diminishes quickly as we move up the complexity regime.

Beyond a certain level, complexity cannot be effectively handled. This behavior conveys that human's complexity handling capability is *bounded.* While there seem to be a fundamental limit, are the individual complexity handling curves as simple without any hint of human *adaptatio*n and *innovation* that in a sense stretches this *bound* and alters the individual complexity handling capability?

Our initial experimental data with ball and stick models in chapter 4 suggest that human complexity handling capability is indeed more complex and ability to adapt and innovate leads to stretching of the bound as shown in fig. 6.8. This profile is similar to what is observed in case of cities and how cities adapt and innovate and avoid the apparent *death trap* [Bettencourt at al., 2007; Bettencourt and West, 2010] with explosion in population. As structural complexity level increases, the individual adapts and by exploiting higher-level patterns. This, in turn, reduces the complicatedness or perceived complexity for this individual (see fig. 6.8).

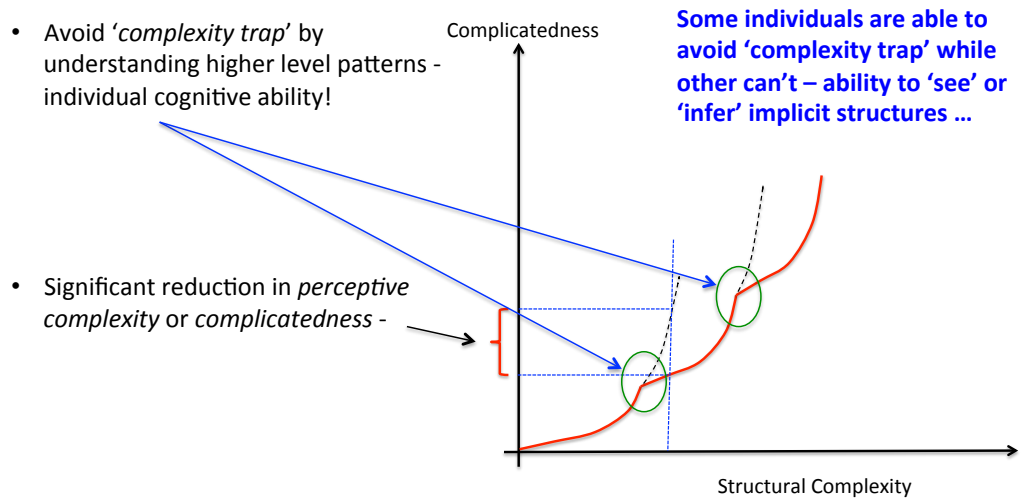


Fig. 6.8: Complexity and Complicatedness – notion of complexity trap and regime changes in perception of complexity at individual level.

We term is phenomena as avoidance of *complexity trap.* Note that this is a preliminary observation and needs to be investigated in future.

6.1.3 Linking Structural Complexity to system development project duration and scheduling

In chapter 4, we have developed an aggregate level regression-based model linking structural complexity, C to the total development effort, E (expressed through the averaged build time). The functional form of this relationship is given below,

$$E = aC^b \qquad (6.9)$$

Now we look for establishing an analytical relationship linking structural complexity with system development project duration and scheduling. The existing literature investigating regular patterns of effort utilization in research and development of engineered systems reveals patterns that are independent of the type of work and is fundamentally related to the way people solve problems [Norden 1964; Putnam 1978; Bashir and Thomson 2001].

Norden developed a model describing the effort utilization, approximating the entire project cycle based on a number of system development projects across

different disciplines. This model links the estimated total development effort *E* to the effort utilization in each time period y_t:

$$y_t = 2E\alpha t e^{-\alpha t^2} \qquad (6.10)$$

where, y_t is the effort spend during period *t*; *E* is the total development effort and α is a "*shape parameter*" that depends on the time to reach the maximum effort expenditure during the entire project cycle. The effort spend till time *t* is computed by integrating over eq. 6.10 and is given by:

$$y(t) = \int_0^t y_\tau d\tau = E(1 - e^{-\alpha t^2}) \qquad (6.11)$$

Representative plots of y_t and $y(t)$ are shown in fig. 6.9 below. Note that the effort spent during time period *t* tails out to infinity.

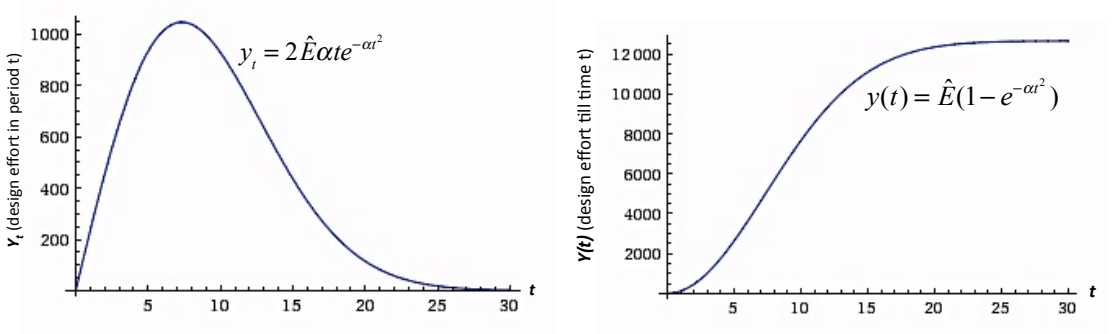


Fig. 6.9: Plots of effort spend during a time period y_t and cumulative effort spend y(t)

We can see that the Norden's model satisfies the essential boundary conditions with

$$y_0 = 0 \quad y_\infty = \lim_{t->\infty}\left[2E\alpha t e^{-\alpha t^2}\right] = 0, \text{ and } y(\infty) = \int_0^\infty y_\tau d\tau = E\underbrace{\int_0^\infty (2\alpha\tau)e^{-\alpha\tau^2} d\tau}_{=1} = E.$$

To compute the development effort spent at any point in time, we need to estimate the *shape parameter,* α. It is related to the time of peak effort utilization and found by maximizing y_t:

$$\frac{dy_t}{dt} = 0$$
$$=> \; 2E\alpha e^{-\alpha t^2}[1 - 2\alpha t^2] = 0$$

The *shape parameter*, α is related to the time t_* of peak effort utilization as shown below:

$$\alpha = \frac{1}{2t_*^2} \qquad (6.12)$$

Hence, the *shape parameter* α can be estimated based on the time-to-peak development effort for similar class of development projects. As proposed by Bashir and Thomson [2001], we can compute an effort spending rate measure, *r* from historical system development projects of similar nature such that,

$$r = \frac{E}{t_*} \qquad (6.13)$$

Here, the rate measure, *r* represents the *average* effort expenditure rate if the development project were to be completed in time t_*. Hence, *r* is a measure of productivity. We can express the time-to-peak effort expenditure as a function of Structural Complexity, *C* by linking eq. 6.9 and eq. 6.13. This is a power-law type functional form, so prevalent in the cost and effort estimation literature [Garvey, 2000].

$$t_* = \frac{E}{r} = \frac{aC^b}{r} = \left(\frac{a}{r}\right)C^b \qquad (6.14)$$

As shown before, Norden's model is *right-skewed* and tails out to infinity. For practical purposes, let us assume that project duration is given as t_d such that,

$$y(t_d) \approx E \qquad (6.15)$$

Now the effort spent during the last period can be expressed as the difference between $y(t_d)$ and $y(t_d - 1)$,

$$\begin{aligned} & y(t_d) - y(t_d - 1) \\ &= E - E[1 - e^{-\alpha(t_d - 1)^2}] \\ &= Ee^{-\alpha(t_d - 1)^2} \qquad (6.16) \end{aligned}$$

Let us define a parameter, p as the fraction of effort spend during the last period. Using eq. 6.16, we find the following expression for p:

$$p = \frac{effort\ spend\ during\ last\ period}{total\ effort}$$
$$= e^{-\alpha(t_d-1)^2} \qquad (6.17)$$

Taking natural logarithm of both sides in eq. (6.17), we get:

$$\ln(p) = -\alpha(t_d - 1)^2$$
$$t_d - 1 = \sqrt{\frac{\ln(p)^{-1}}{\alpha}} \qquad (since\ t_d > 1)$$

Using eq. 6.12 in the above expression, we arrive at the final expression for estimated project duration is:

$$t_d = 1 + t_* \sqrt{\ln(1/p^2)} \qquad (6.18)$$

Using eq. 6.9 for t_* in eq. 6.14, we derive en explicit relationship linking structural complexity, C to the engineering system development project duration:

$$t_d = 1 + \left(\frac{a}{r}\right) C^b \sqrt{\ln\left(\frac{1}{p^2}\right)} \qquad (6.19)$$

Here, we observe that the estimated development project duration is dependent on Structural Complexity, C and involves estimation of four parameters {a; b; r; p}. The first two parameters are estimated from the development effort versus structural complexity model, while the other two parameters are derived from a set of historical projects of similar type. We can observe that the estimated project duration also has a super-linear dependence on structural complexity for b>1. This illustrates explicit dependence of project duration and scheduling on the Structural Complexity of the engineered system being developed, leveraging Norden's model.

6.1.4 Granularity of System Representation and the *Complexity Gap*

One interesting question in the area of engineering systems is about the level of detail at which a system should be represented. This depends on the questions we are trying to answer with the analysis. With respect to computation of structural complexity, the question is about the ability to seamlessly move across different levels of abstraction (i.e., system representation) and its impact on the estimates of structural complexity as we move across levels of representation. In practice, it appears that the hierarchical L3 (i.e., level 3 system decomposition) decomposition is usually good enough for comparing different engineering systems with similar end goals.

As we will see later in this chapter, the structural complexity evolves as we have more details about the system architecture and then settles down once we have the fundamental architecture captured (that tends to happen at L3 decomposition level). This is illustrated in fig. 6.10 below.

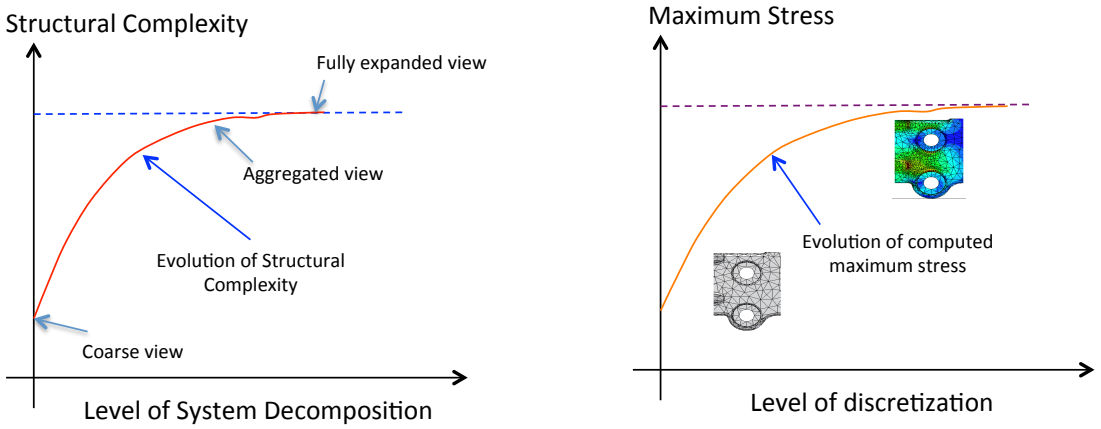


Fig. 6.10: Evolution of maximum stress in a structure with increasing level of discretization in Finite Element (FE) model is analogous to the evolution of structural complexity with increasing level of system decomposition.

Another interesting theoretical question relates to the *complexity gap*. The complexity gap is similar in spirit to the notion of information gap [Thomas and Cover, 2006] in information theory.

Let us consider a system using a finer level representation with binary adjacency matrix, $\Lambda_{(n+m)\times(n+m)}$ since there are $(n+m)$ components in the finer level system representation. For the same system, assume that the finer level details are *abstracted* above at a coarser level where we observe two subsystems with adjacency matrices $A_{1(n\times n)}$, $A_{2(m\times m)}$ respectively and their interconnections are captured through rectangular matrices $K_{1(n\times m)}$ and $K_{2(m\times n)}$. The complexity gap, δ, if any, is defined as the difference in the structural complexities computed using these two representations of the same system.

At the finer level of decomposition, the binary adjacency matrix for the system can be written as,

$$\Lambda = \begin{bmatrix} A_1 & K_1 \\ K_2 & A_2 \end{bmatrix} = \underbrace{\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} 0 & K_1 \\ K_2 & 0 \end{bmatrix}}_{B}$$

$$= A + B$$

where,

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & K_1 \\ K_2 & 0 \end{bmatrix}$$

The system adjacency matrix, Λ is expressed here as the subsystem adjacency matrices A and the inter-subsystem connectivity matrix B, which is a bipartite matrix. The graph energy inequality (see chapter 3 for details) in terms of graph energies of the subsystems and the bipartite matrix is given by,

$$E(A) \le E(\Lambda) \le E(A) + E(B)$$
$$\therefore E(A_1) + E(A_2) \le E(\Lambda) \le E(A_1) + E(A_2) + E(B) \qquad (6.20)$$

The graph energy at the final level can be written as,

$$E(\Lambda) = E(A_1) + E(A_2) + \Delta$$
$$= E(A_1) + E(A_2) + pE(B) \quad where,\ p = \frac{\Delta}{E(B)} \in [0,1] \qquad (6.21)$$

Let us assume that the sum of component complexity is C_1 and the sum of interface complexities is β. The structural complexity at this finer level of decomposition as,

$$SC = C_1 + \beta\left(\frac{E(\Lambda)}{m+n}\right) \qquad (6.22)$$

Now, notice that, the interface complexity can be expressed as the sum of interface complexities of the subsystems and the inter-subsystem,

$$\beta = \beta_1 + \beta_2 + \beta_k \qquad (6.23)$$

Let us consider the finer level subsystems, brought-up at the coarser level. The structural complexity at this level is the sum of structural complexities of the individual subsystems plus the complexity stemming from the inter-subsystem interactions. Using this representation, the aggregated structural complexity is given as,

$$SC_{agg} = \underbrace{\left(C_{A_1} + C_{A_2}\right)}_{C_1} + \beta_1\left(\frac{E(A_1)}{n}\right) + \beta_2\left(\frac{E(A_2)}{m}\right) + \beta_k\left(\frac{E(B)}{m+n}\right)$$
$$= C_1 + \beta_1\left(\frac{E(A_1)}{n}\right) + \beta_2\left(\frac{E(A_2)}{m}\right) + \beta_k\left(\frac{E(B)}{m+n}\right) \qquad (6.24)$$

where C_{A_1} and C_{A_2} stands for the component complexities from the two subsystems. The *complexity gap*, δ is defined as the difference between the structural complexities computed by eq. 6.22 and 6.24 and is given by,

$$\delta = SC - SC_{agg}$$
$$= \left(\frac{\beta}{n+m}\right)E(\Lambda) - \frac{\beta_1}{n}E(A_1) - \frac{\beta_2}{m}E(A_2) - \frac{\beta_k}{n+m}E(B) \qquad (6.25)$$

Let us expand the first term on the RHS of eq. 6.25 using relation 6.21,

$$\left(\frac{\beta}{n+m}\right)E(\Lambda)=\left(\frac{\beta}{n+m}\right)E(A_1)+\left(\frac{\beta}{n+m}\right)E(A_2)+p\left(\frac{\beta}{n+m}\right)E(B)$$
$$=\left(\frac{\beta_1}{n+m}\right)E(A_1)+\left(\frac{\beta_2}{n+m}\right)E(A_2)+p\left(\frac{\beta}{n+m}\right)E(B)$$
$$+\left(\frac{\beta_2+\beta_k}{n+m}\right)E(A_1)+\left(\frac{\beta_1+\beta_k}{n+m}\right)E(A_2)$$
$$=\left(\frac{\beta_1}{n}\right)E(A_1)\left[\frac{n}{n+m}\right]+\left(\frac{\beta_2}{m}\right)E(A_2)\left[\frac{m}{n+m}\right]+p\left(\frac{\beta}{n+m}\right)E(B)$$
$$+\left(\frac{\beta_2+\beta_k}{n}\right)E(A_1)\left[\frac{n}{n+m}\right]+\left(\frac{\beta_1+\beta_k}{m}\right)E(A_2)\left[\frac{m}{n+m}\right] \qquad (6.26)$$

Let us define, $\frac{n}{m}=r>1$ without loss of generality as we can always assign graph A_1 to the larger subsystem. Using the ratio, $r\in[1,\infty)$ and arranging terms using 6.25 and 6.26, we can express the complexity gap as,

$$\delta=\frac{\beta_1}{n}E(A_1)\left[\frac{r}{1+r}-1\right]+\frac{\beta_2}{m}E(A_2)\left[\frac{1}{1+r}-1\right]+p\left(\frac{\beta}{n+m}\right)E(B)-\left(\frac{\beta_k}{n+m}\right)E(B)$$
$$+\left(\frac{\beta_2+\beta_k}{n}\right)E(A_1)\left[\frac{r}{1+r}\right]+\left(\frac{\beta_1+\beta_k}{m}\right)E(A_2)\left[\frac{1}{1+r}\right]$$
$$=\frac{E(A_1)}{n}\left[\left(\frac{r}{1+r}\right)(\beta_2+\beta_k)-\left(\frac{1}{1+r}\right)\beta_1\right]+\frac{E(A_2)}{m}\left[\left(\frac{1}{1+r}\right)(\beta_1+\beta_k)-\left(\frac{r}{1+r}\right)\beta_2\right]$$
$$+\frac{E(B)}{n+m}\left(p\beta-\beta_k\right)$$
$$=\frac{E(A_1)}{n}\beta\left[\left(\frac{r}{1+r}\right)\left(1-\frac{\beta_1}{\beta}\right)-\left(\frac{1}{1+r}\right)\left(\frac{\beta_1}{\beta}\right)\right]$$
$$+\frac{E(A_2)}{m}\beta\left[\left(\frac{1}{1+r}\right)\left(1-\frac{\beta_2}{\beta}\right)-\left(\frac{r}{1+r}\right)\left(\frac{\beta_2}{\beta}\right)\right]$$
$$+\frac{E(B)}{n+m}\beta\left(p-\frac{\beta_k}{\beta}\right)$$
$$=\frac{E(A_1)}{n}\beta\left[\left(\frac{r}{1+r}\right)-\frac{\beta_1}{\beta}\right]+\frac{E(A_2)}{m}\beta\left[\left(\frac{1}{1+r}\right)-\frac{\beta_2}{\beta}\right]+\frac{E(B)}{n+m}\beta\left(p-\frac{\beta_k}{\beta}\right) \qquad (6.27)$$

In the above expression for complexity gap, δ, the first term is positive if,

$$\left(\frac{r}{1+r}\right)\left(1-\frac{\beta_1}{\beta}\right)>\left(\frac{1}{1+r}\right)\left(\frac{\beta_1}{\beta}\right)$$
$$\Rightarrow\left(\frac{r}{1+r}\right)>\frac{\beta_1}{\beta}$$
$$\therefore r>\frac{\beta_1}{\beta_2+\beta_k} \qquad (6.28)$$

For the second term to be positive, we must have,

$$\left(\frac{1}{1+r}\right)\left(1-\frac{\beta_2}{\beta}\right)>\left(\frac{r}{1+r}\right)\left(\frac{\beta_2}{\beta}\right)$$
$$\Rightarrow\left(\frac{1}{1+r}\right)>\frac{\beta_2}{\beta}$$
$$\therefore r<\frac{\beta_1+\beta_k}{\beta_2} \qquad (6.29)$$

Combining relations 6.28 and 6.29, we have the following condition that guarantees positivity of the first two terms,

$$\frac{\beta_1}{\beta_2+\beta_k}<r<\frac{\beta_1+\beta_k}{\beta_2} \qquad (6.30)$$

The third term is positive if,

$$p>\frac{\beta_k}{\beta} \qquad (6.31)$$

To simplify and operationalize better, let us introduce the triple $(u,v,w)\in(0,1)$ where $u=\frac{\beta_1}{\beta};\ v=\frac{\beta_2}{\beta};\ w=\frac{\beta_k}{\beta}$ and eq. 6.23 enforces the constraint, $u+v+w=1$. Using these ratios, we can express the conditions such that all terms in the *complexity gap*, δ expression, are positive,

$$\left.\begin{array}{l}\frac{u}{1-u}<r<\frac{1-v}{v}\\ p>w\end{array}\right\} \qquad (6.32)$$

Let us consider the balanced case, $r=1$. Using relation 6.27 and assuming 6.32 holds, we get the *complexity gap*,

$$\begin{aligned}\delta &= \frac{1}{2}\frac{E(A_1)}{n}\beta\left(1-2\frac{\beta_1}{\beta}\right)+\frac{1}{2}\frac{E(A_2)}{m}\beta\left(1-2\frac{\beta_2}{\beta}\right)+\frac{E(B)}{n+m}\beta\left(p-\frac{\beta_k}{\beta}\right)\\ &= \frac{1}{2}\frac{E(A_1)}{n}\beta\underbrace{\left(1-2u\right)}_{\geq 0}+\frac{1}{2}\frac{E(A_2)}{m}\beta\underbrace{\left(1-2v\right)}_{\geq 0}+\frac{E(B)}{n+m}\beta\left(p-w\right) \qquad (6.33)\end{aligned}$$

In general, we have $p>w$ and therefore all the terms in 6.33 are greater than zero and hence $\delta>0$, indicating the complexity gap is positive. In case, $r\to\infty$, we have $v\to 0$. The *complexity gap* is expressed as,

$$\begin{aligned}\delta &= \frac{E(A_1)}{n}\beta(1-u)-\frac{E(A_2)}{m}\beta v+\frac{E(B)}{n+m}\beta\left(p-w\right)\\ &\approx \frac{E(A_1)}{n}\beta(1-u)+\frac{E(B)}{n+m}\beta\left(p-w\right) \qquad (6.34)\end{aligned}$$

In general, we have $p>w$ and therefore all the terms in 6.34 are greater than zero and hence $\delta>0$, indicating the complexity gap is positive.

Using the triple $(u,v,w)\in(0,1)$, the complexity gap in eq. 6.27 can be written as,

$$\begin{aligned}\delta &= \frac{E(A_1)}{n}\beta\left[\frac{r}{1+r}-u\right]+\frac{E(A_2)}{m}\beta\left[\frac{1}{1+r}-v\right]+\frac{E(B)}{n+m}\beta\left(p-w\right)\\ &= \frac{E(A_1)}{n}\beta\left\{\left[\frac{r}{1+r}-u\right]+r\frac{E(A_2)}{E(A_1)}\left[\frac{1}{1+r}-v\right]+\frac{E(B)}{E(A_1)}\left(\frac{r}{1+r}\right)\left(p-w\right)\right\} \qquad (6.35)\end{aligned}$$

In the expression 6.35 above, please note that, r$\frac{E(A_2)}{E(A_1)}$ is a finite quantity of order 1 (see the dimensional analysis in chapter 3) and could well be a small number depending on the actual topology of the subsystems. Now, $\frac{E(B)}{E(A_1)}$ is a small number since the bipartite matrix, connecting the two subsystems is usually very sparse (i.e., think of the subsystems as modules and B representing inter-module

links). In addition, the set of ratios (p,r,u,v,w) are not independent, but correlated to varying degrees with the constraint $u+v+w=1$. In general, inequalities 6.32 are satisfied and all the terms on the RHS of eq. 6.35 remain positive, leading to a positive complexity gap, δ. This result is influenced by the existing correlations among the ratios in eq. 6.35 and needs to be investigated in future for any non-hypothetical exceptions.

The interpretation of the *complexity gap* is similar to the *information* gap, which is always positive or zero [Thomas and Cover, 2006]. This gap can be interpreted as follows: if a system is reconstructed at a higher level, using a lower level information, there is a physical loss of information due to this reconstruction and we can only get back a very well-approximated version (but not the *exact* replica of the system). For example, let us think of a structural system. If we use the details of lower level sub-structures in a higher, system level analysis, we lose some details on localized load sharing mechanism [Engel, 2007] when compared to lower level complete system model. This is essentially due to the load redistribution in the structural system [Engel, 2007]. The extent of this *redistribution* of load and the subsequent loss of information/detail depends on the details of the structural system.

This is a fundamental principle in information theory [Thomas and Cover, 2006] and seems to be applicable to *complexity gap* as well and it is also physically consistent. We believe that the positivity of the *complexity gap* could serve as a crucial *construct validity* criterion for any proposed complexity metric.

6.2 Distribution of Structural Complexity

In this section, we look at the factors affecting distribution of complexity and structural modularity. The distribution of structural complexity over the regions of the system (i.e., subsystems) and its implications on system development are discussed. It is shown that modularity and topological complexity are not necessarily negatively correlated, and we can have architectural configurations where both complexity and modularity increase, building upon a simple toy model

involving two modules. Observations from large real-world engineered systems are explained using the macro level predictions from the toy model.

6.2.1 Implication of complexity distribution on system development

Distribution of structural complexity across the system elements plays a very significant role in achieving a set of system properties and often to programmatic success of the system development project. Knowledge of overall system architecture is absolutely critical to be able to quantify and track the complexity during the system development activity. There may be subsystems that are significantly more complex and respective development team should be able to handle such high complexity in order to be successful. Therefore, knowledge of the complexity distribution plays a crucial role in selection or composition of the subsystem development team. We might view the system from a somewhat higher level where the modules of subsystems are treated as super-components. Each super-component has an internal system complexity (that represents complexity of the super-component) and this fact should not be overlooked (see fig. 6.11 below).

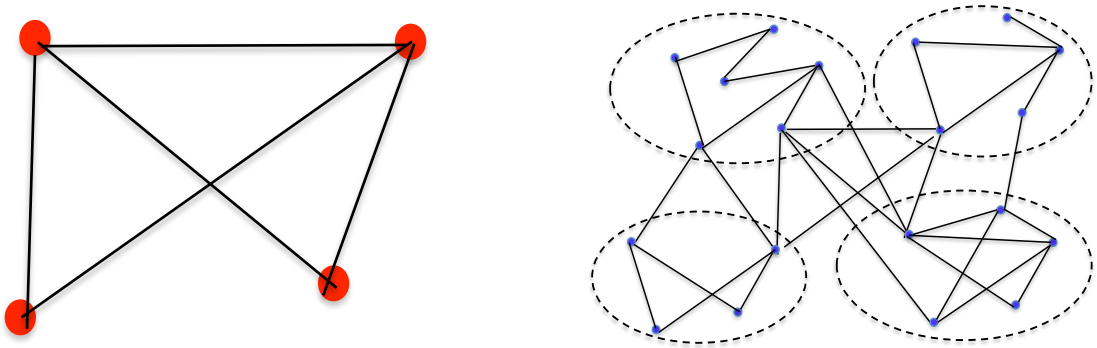


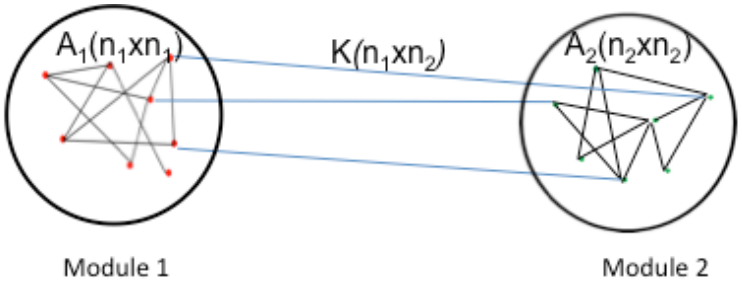
Fig. 6.11: A higher vis-à-vis a lower level architectural view of the same system. At higher level, we have subsystems as nodes. At the lower level, it is broken down into components.

We cannot treat the complexity of this aggregated component the same way as we would for a simple component. The total structural complexity is now distributed within and across the subsystems. In order to extract information on the complexity distribution over the entire system, we do need complete information

about the internal structure of subsystems. This information is crucial for tracking and management of large, engineered system development efforts. Implication of the complexity distribution on system development effort and associated decision-making can be best explored using simplified examples.

6.2.2 Structural Complexity and Modularity

Let us consider the following example of a system structure consisting of two modules of subsystems as shown below.



$$A = \begin{bmatrix} A_1 & K_2 \\ K_1 & A_2 \end{bmatrix} \Rightarrow E(A) = E(A_1) + E(A_2) + \Delta$$

Let us represent the component and interface complexities as α and β, we arrive at the following expression for Structural Complexity:

$$C = \left(\sum_{i=1}^{n_1} \alpha_i^{A_1} + \sum_{i=1}^{n_2} \alpha_i^{A_2} \right) + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma E(A)$$

$$= \left(\sum_{i=1}^{n_1} \alpha_i^{A_1} + \sum_{i=1}^{n_2} \alpha_i^{A_2} \right) + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma \left(E(A_1) + E(A_2) + \Delta \right)$$

$$= \left[\sum_{i=1}^{n_1} \alpha_i^{A_1} + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} \right) \gamma E(A_1) \right] + \left[\sum_{i=1}^{n_2} \alpha_i^{A_2} + \left(\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} \right) \gamma E(A_2) \right]$$

$$+ \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} \right) \gamma E(A_2) + \left(\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} \right) \gamma E(A_1) + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \beta_{i,j}^{A_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} \beta_{i,j}^{A_2} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma\Delta$$

$$+ \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma E(A_1) + \left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \beta_{i,j}^{K_1} + \sum_{i=1}^{n_2}\sum_{j=1}^{n_1} \beta_{i,j}^{K_2} \right) \gamma E(A_2), \text{ where } \gamma \text{ is a normalization factor}$$

Hence, we can write, *Structural Complexity = sum of module structural complexities + integrative structural complexity*. Let us apply the above result to a modular system structure with two modules:

$$E(A_1) = x_1 E(A);\ E(A_2) = x_2 E(A);\ \Delta = x_3 E(A)$$

$$where,\ x_1 + x_2 + x_3 = 1$$

$$m = \#\,interactions;\ \Omega = \sum_{i=1}^{n_1} \alpha_i^{A_1} + \sum_{i=1}^{n_2} \alpha_i^{A_2}$$

$$\sum_i^{n_1}\sum_j^{n_1} \beta_{i,j}^{A_1} = y_1 m;\ \sum_i^{n_2}\sum_j^{n_2} \beta_{i,j}^{A_2} = y_2 m;\ \sum_i^{n_1}\sum_j^{n_2} \beta_{i,j}^{K_1} = y_3 m;\ \sum_i^{n_2}\sum_j^{n_1} \beta_{i,j}^{K_2} = y_4 m$$

$$where,\ y_1 + y_2 + y_3 + y_4 = 1$$

Using the earlier derivation of structural complexity for two sub-system example, we can write the structural complexity in this case as:

$$Structural\ Complexity,\ C = \Omega + \gamma m E(A)\underbrace{[x_1 y_1 + x_2 y_2 + x_1 y_2 + x_2 y_1 + x_3 + (x_1 + x_2)(y_3 + y_4)]}_{=1}$$

$$= \Omega + \gamma m E(A) \qquad (6.36)$$

There is a conventional wisdom that complexity and modularity are negatively correlated and increased modularity brings down the complexity. This is definitely not a causal relation and what we actually have is a set of seven fractions - $\{x_1, x_2, x_3\}$ and $\{y_1, y_2, y_3, y_4\}$ that determines how the overall structural complexity and modularity plays out. We can very well have increases in complexity alongside an increase in modularity as shown below in fig. 6.12.

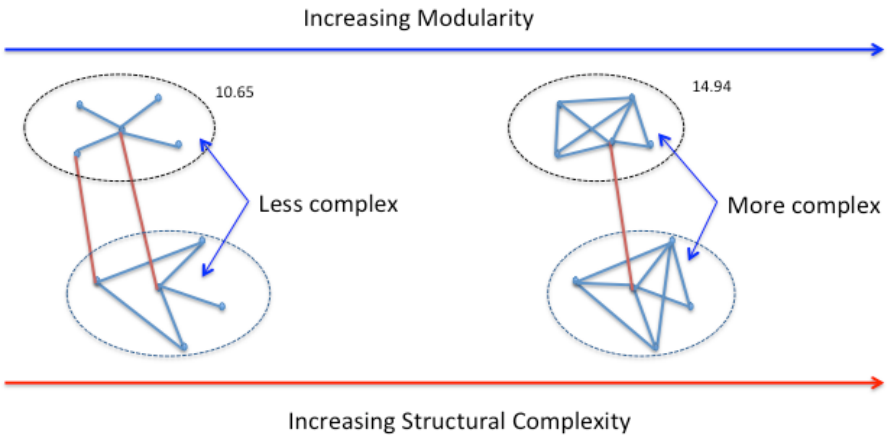


Fig. 6.12: Topological Complexity and Structural Modularity are not necessarily negatively correlated.

These seven parameters have only two constraints: $x_1 + x_2 + x_3 = 1$ and $y_1 + y_2 + y_3 + y_4 = 1$. They shape how the overall complexity is distributed within and across the modules and the associated structural modularity.

Now let us look at the computation of modularity index, given the set of modules. Assume that the modules are somehow given to us and we need to compute modularity index based on the given modules. Modularity index, Q is then defined as the fraction of edges that fall within module 1 or 2, minus the expected number of edges within module 1 and 2 for a *random graph* with same node degree distribution as the given network [Newman, 2010].

Expanding this basic definition and after unpacking different terms involved, we arrive at the final form of the modularity index for this two module system as, $Q = \sum_{c=1}^{2} (e_{ii} - a_i^2)$ where e_{ii} is the fraction of edges with both end vertices in the same module i and a_i is the fraction of edges with at least one end vertex inside module i. Expanding the above for Q, we arrive at the modularity index:

$$\begin{aligned} Q &= (y_1 + y_2) - \left[(y_1 + y_3)^2 + (y_2 + y_4)^2 \right] \\ &= (y_1 + y_2) + 2(y_2 + y_4) - 2(y_2 + y_4)^2 - 1 \end{aligned} \qquad (6.37)$$

Notice that there is no element of $\{x\}$ vector in eq. 6.37 above. The elements of $\{x\}$ and $\{y\}$ vectors are coupled through the topological complexity expression and more precisely, through the equality constraint, $x_1 y_1 + x_2 y_2 + x_1 y_2 + x_2 y_1 + x_3 + (x_1 + x_2)(y_3 + y_4) = 1$ (see eq. 6.36). Otherwise, these are free parameters that can be varied to traverse the structural complexity vs. structural modularity space.

Here, we can observe that the expression of modularity does not depend on how many nodes belong to each module and detailed topological features within each module. Once the modules are '*defined*', there is no role for topological features within the modules in the modularity index expression. This is not the case for

topological complexity though. It inherently captures the global effect of relative module size (i.e., number of nodes within each module).

Numerical Example: Let us consider a graph with 100 nodes and 251 links with only 1 link connecting the modules. Each module has 125 links each while 100 nodes are distributed between the two modules based on the ratio n_1/n_2. Otherwise, the modules are random networks. The adjacency matrix of this modular network is *A*. For comparison, we construct a random network *B* with the same number of nodes and links (i.e., 100 nodes and 251 links). We perform a few simulation studies to demonstrate the relationship between the topological complexity and structural modularity.

Case 1: We vary the ratio n_1/n_2 with $n_1+n_2 = 100$ and the observed behavior is shown in Fig. 6.13 below.

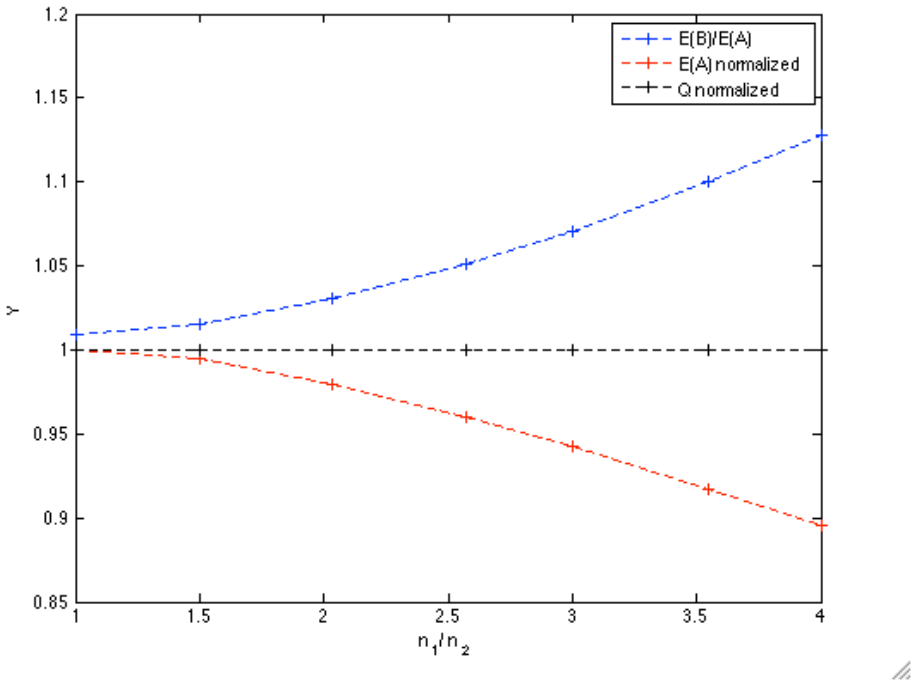


Fig. 6.13: A high n_1/n_2 value indicates increasing 'centralization' lead to reduced topological complexity (red lines). The blue lines indicate the ratio of topological complexity of a random network, A with same number of nodes and links to the topological complexity of our hypothetical modular architecture, B. Here, $N = n_1 + n_2 = 100$, $m_1 = m_2 = 125$, $k = 1$, $m = m_1 + m_2 + k = 251$.

We can observe that a relative centralization (as the size of module 1 grows much bigger than that of module 2) brings down the overall structural complexity, but has

no impact on the modularity index. This clearly negates the notion that reduced topological complexity will drive down structural modularity and vice-versa. Also note that the ratio, E(B)/E(A) has increased, indicating a relative increase in topological complexity of a corresponding non-modular structure with the same number of components and edges.

Case 2: Now we keep the module size constant and change the number of intra-module connections. In this case, both modules have 50 components each. Here we observe structural complexity decreasing as one module becomes dominant and the architecture tends toward centralization. Interestingly, the modularity index also decreases (see fig. 6.14 below). This shows a case where modularity and complexity are positively correlated, as against the perceived notion that they are essentially negatively correlated (i.e., increasing modularity leads to a decrease in complexity).

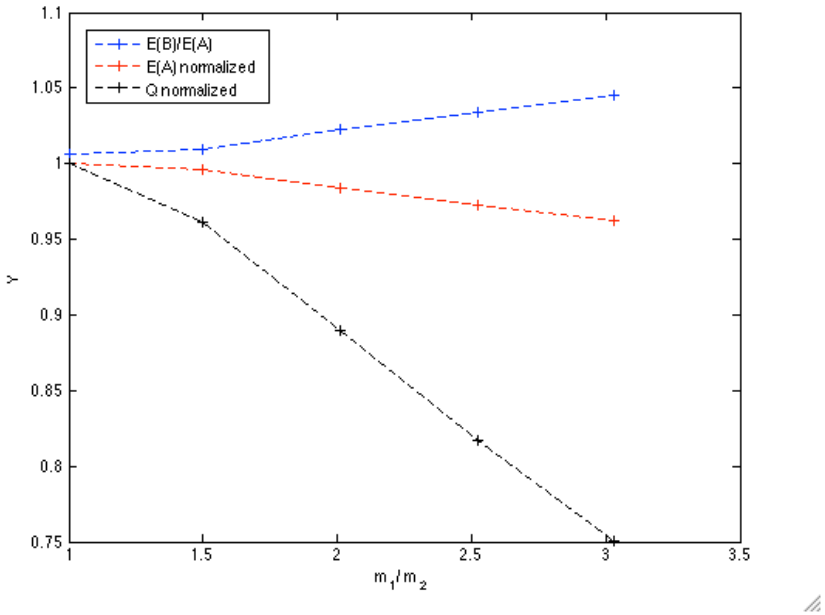


Fig. 6.14: m_1/m_2 gives the relative density of two modules. A high number indicates increasing 'centralization' leading to reduced structural complexity (red lines). The blue line indicates the ratio of topological complexity of a random network with same number of nodes and links to the topological complexity of our hypothetical modular architecture (based on an ensemble size of 50). Based on this imposed modules, the modularity index, Q decreases from 0.5 to 0.38. Here, $N = n_1 + n_2 = 100$, $n_1 = n_2 = 50$, $k = 1$, $m = m_1 + m_2 + k = 251$.

This result clearly brings forth the point that topological (by extension structural) complexity does not necessarily have an inverse relationship to modularity. Note that ratio E(B)/E(A) has increased in this case as well, indicating a relative increase in topological complexity of a corresponding non-modular structure with identical number of components and edges.

In this section, we observed the following main points:

1. Structural Complexity and Modularity measure two different structural aspects of the overall system architecture.

2. They are not necessarily negatively correlated as often presumed and we can have systems where increased complexity can manifest alongside increased modularity. Complexity encapsulated inside modules does not necessarily imply reduction in overall complexity [Hirschi and Frey, 2002].

3. In case of simple, two-module system, we have 7 parameters that shape the distribution of structural complexity and modularity of the system. They form an under-constrained set of equations and many feasible solutions exist. This, in theory, enables us to trade structural complexity with modularity.

A highly modular architecture with very complex modules can result in a complex architecture, but may still be desirable if system decomposability is of primary importance. This fundamentally relates to the usefulness of using *reductionist* strategies, which humans have reasonably mastered over the last century, for developing a complex, engineered system. Increasing modularity helps application of reductionist strategies while higher complexity makes reductionism less applicable/effective.

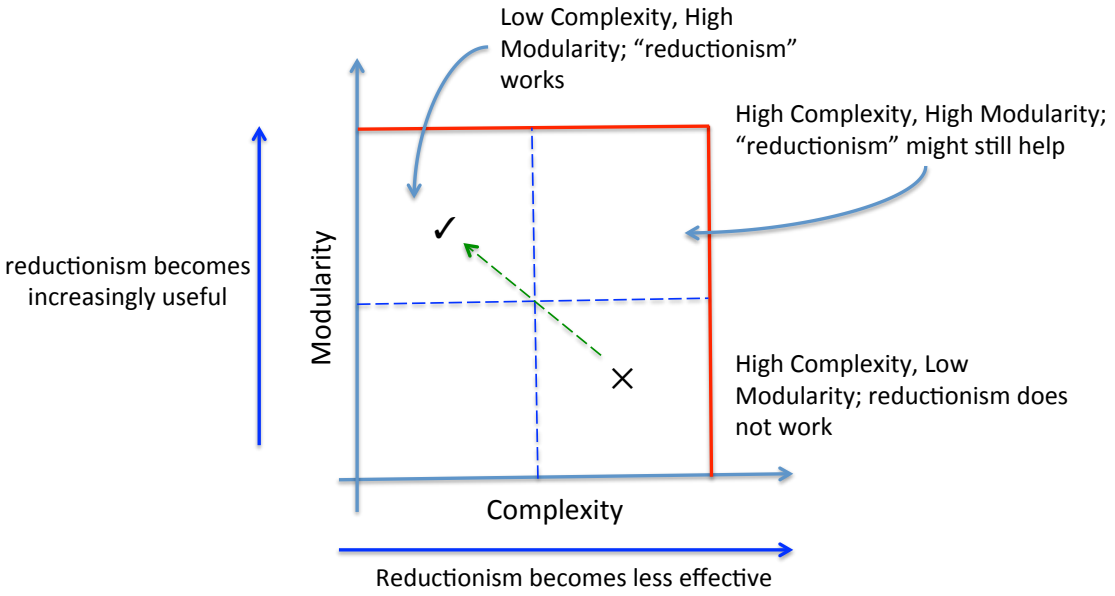


Fig. 6.15: Complexity-Modularity trade-space and their impact on effectiveness of reductionist strategies for system development.

Looking at the above trade-space in fig. 6.15, the ideal quadrant is the low complexity, high modularity zone. This is where reductionist strategies work the best and one can use decomposition to better handle the system design and development. Higher modularity can aid *incremental evolvability* of the system by enabling exploration of the neighboring areas in the design space.

6.2.3 Why is modularity important?

Modularity is the degree to which a system's components may be separated and recombined. Modularization or design encapsulation is not necessarily a means of reducing intrinsic complexity of the system, but it is a means of effectively re-distributing the total complexity across the system such that the *Perceived Complexity* or *Complicatedness* [Tang and Salminen, 2001; Ramasesh and Browning, 2012] is contained. This aligns perfectly with our deep-rooted capability to '*divide and conquer*'. Structural complexity is an *intrinsic* property of the system and therefore observer-independent. Complicatedness is an extrinsic, observer-dependent property of the system. The impact of intrinsic property of complexity is

manifested in development cost through this intermediate, system-dependent complicatedness function, as depicted in the fig. 6.16 below.

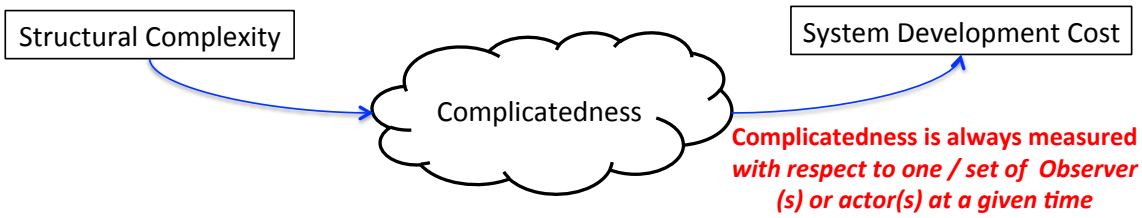


Fig. 6.16: Manifestation of Structural Complexity through the observable, system development cost/effort.

This intermediate property of complicatedness brings in the aspect of human cognition into the scheme of things and is intimately related to any observable from a human executed activity. This is the sole reason for differences/spread in rating of component complexities by different subject matter experts. Each expert's view of component complexities is likely to be a juxtaposition of both intrinsic complexity and his/her own complicatedness function. This is what leads to the spread of component complexity estimates obtained from expert opinion.

We, as humans, tend to behave as *adaptive systems* and adapt with extended exposure and therefore, personal complicatedness function will also evolve with the knowledge and the extent of exposure. We hope to find refuge under modularization strategy as a mean of design encapsulation that aids the system development effort by *divide and conquer*. This is likely to find a strong link with aspects of human cognition in general. In future, it might be instructive to link product development with human cognition and to investigate their interaction in a rigorous way.

6.3 Examples of Systemic Implication

The development of *Boeing 787* (e.g., the *Dreamliner*) is a good example of systemic implications that stem from complexity of the system and their management [Heimsch, 2011; Cohan, 2011; Hiltzik, 2011; Zhao, 2012].

The Boeing Company announced in 2004 that it was embarking on an ambitious commercial airplane development project in order to bring the 787

Dreamliner to market. Boeing had launched its last major commercial airplane, the 777, a decade ago and was under pressure from Airbus, its primary competitor who was stealing market share away from Boeing. Recognizing the need for speed to market for the 787, along with increased quality standards and reduced production costs, Boeing focused on an innovation strategy and decided to outsource about 70% of the design and manufacturing for this plane (see fig. 6.17 below) [Heimsch, 2011; Cohan, 2011].

The development was done at 17 different companies spread over 10 different countries worldwide and there was an implicit understanding or hope that this strategy will be effective in persuading different countries / airlines to buy into this initiative as they also have a share of the pie [Cohan, 2011; Hiltzik 2011].

Boeing felt that the 787 Dreamliner represented groundbreaking innovation, with benefits that would resonate with customers and enable Boeing to regain its leadership position and the 787's composite material design, aerodynamics, fuel efficiency and propulsion systems has redefined how commercial aircraft are designed and manufactured, and will impact the broader aviation industry [Cohan, 2011; Cohan, 2008].

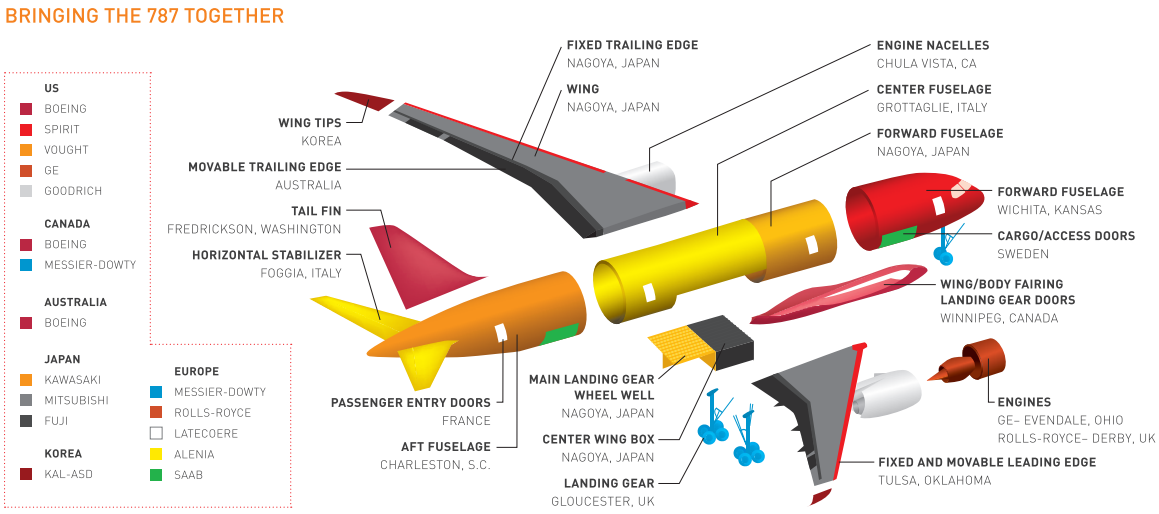


Fig. 6.17: Boeing partnered with organizations from 10 different countries to share and manage the risk while scouting for best in class technology.

Recognizing that they may not have the internal expertise to design first in class systems for every design element, Boeing sought out leaders for each of these areas to bring the best together and build the 787. A major benefit of the 787 is its fuel efficiency. Because of weight-saving fuselage design, which is 80% lightweight carbon-fiber composite material, the 787 uses 20% less fuel than other planes its size [Heimsch, 2011; Cohan, 2009]. The composite material is expected to last longer than the aluminum alternative, which should result in lower maintenance costs for the airlines. Additional benefits include more passenger legroom, cargo space, significant noise reduction and much improved passenger ergonomics.

Historically, Boeing has outsourced much of its manufacturing, considering it non-core to its operations. But Boeing has focused on owning its core design work, viewing it as its competitive advantage. However, with the 787, Boeing opted to also outsource much of the development work, recognizing that the detailed design and production went hand in hand, and were needed to be done by the partner in order to capitalize on their industry leading expertise [Zhao, 2012]. For the first time in history, Boeing decided to outsource both the design and the manufacturing of the 787 to shift the economic risk to the suppliers [Cohan, 2011].

This was very aggressive outsourcing strategy that has never been pursued by Boeing or any other aircraft developers. In this particular program, Boeing adopted a risk-sharing partnership with suppliers where the tier 2 suppliers were chosen by tier 1 suppliers and not by Boeing. The non-recurring development cost was shared with the suppliers and Boeing bore only 45% of planned 787 development cost of about $10 billion. The detailed design of sub-systems was entrusted with the suppliers. This might have led to *masking* of the detailed system architecture from Boeing to an extent.

Unfortunately, the 787 development project was never successfully managed and led to a 40 month delay in launching the first aircraft for operation (planned duration of 64 months vis-à-vis ac actual duration of 104 months) and resulted in significant cost overrun of more than $10 billion and erosion of brand value for Boeing.

Earlier, Boeing outsourced only the manufacturing and maintained tight control over the system architecture and design (e.g., Boeing 777 development program), providing those suppliers with extremely detailed specifications of what each aircraft component should do [Zhao, 2012]. But by outsourcing both the design and the manufacturing, Boeing lost control of the development process. This is to say that Boeing had a view at the level of subsystems as shown in Fig. 6.7 (e.g., large red nodes). This view completely hides what is inside the subsystems and there is no way for Boeing to judge the total structural complexity of the system as it evolved and thereby no information about the relative complexity sharing within those subsystems.

If a subsystem or module starts to become too complex, it is possible that the outsourcing partner does not have the adequate capability for handling that level of complexity and this may jeopardize the overall system development effort [Hiltzik, 2011]. But following their strategy of outsourcing the detailed design, Boeing was perhaps not in a position to even know about such events in a timely fashion, let alone intervening on its own.

There is another lurking danger in following the approach of '*outsource and forget'.* At the top level, there is a possibility to assign a lower complexity to the subsystems as some details may not be available at that stage and anyway, there might be a temptation to think that subsystems are "no longer my headache" as this has been outsourced to another entity (see fig. 6.18 below). But it is the controlling organization (e.g., Boeing Corporation in this case) that is ultimately responsible for delivering system properties and that responsibility cannot be outsourced. It was shown [Zhao, 2012] that the 787 development program and the risk-sharing partnership mechanism that was put in place suffered from what is known as the *prisoner's dilemma* [Xu and Zhao, 2012]. This meant that what was best for each individual firm did not align with what was best for the overall program. The OEM (i.e., Original Equipment Manufacturer, that is Boeing in this case) was more concerned about its own share of the development cost and apparently lost sight of the lurking delays of the overall program. A closer control and monitoring with

transparency into the detailed system architecture by the OEM was a necessary enabler for success of such large engineering system development programs.

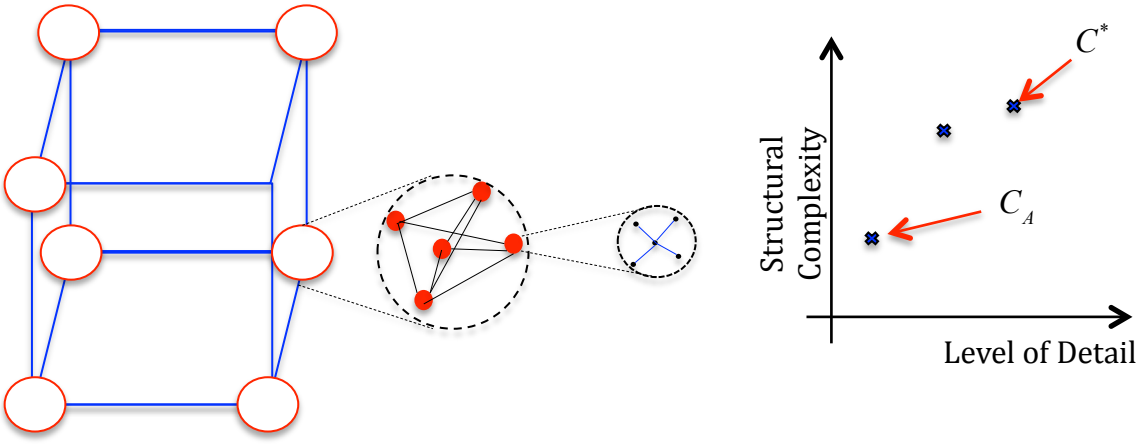


Fig. 6.18: Evolution of structural complexity in case of an evolving design. At top level, each node actually represents a subsystem or module and their lower level details are shown on the left. Complexity estimates are performed at each level of detail without considering the fact that at the top level, we do not have component, but we have subsystems and we cannot treat them as simple components. Doing that one might get the impression that structural complexity is only $\boldsymbol{C_A}$ while in reality, it is $\boldsymbol{C^*}$.

We can observe from fig. 6.18 that Boeing might have predicted complexity as only C_A but what they are responsible for is C^*, which is much higher. It appears that this lesson has been learnt and remedial actions are being taken. Boeing has decided to have tight control over the system architecture and design for development of future Dreamliner variants as exemplified by the flowing quote from a senior Boeing executive - "We fully recognize that we made some mistakes in that regard… on the 787-9, we are pulling more of the engineering back inside to try and alleviate some of the issues we've had on the 787-8 … " [Weber, 2009].

Such instances can also be found elsewhere like the space program. Consider the case of the *Columbia* disaster (2003) and for that matter, the *Challenger* disaster (1986). National Aeronautics and Space Administration (NASA) was the controlling organization. But the detailed design, build and launch were performed by the United Space Alliance partners that included Lockheed Martin, Boeing and many others [Goldsmith and Eggers, 2004]. In both cases, it was found that NASA did not have the capacity to assess the situation on its own. So NASA had to rely on the conclusions from the contracted organizations only and had to act as mere

communicator. Lack of complete knowledge of the overall system architecture had left NASA in a rather unenviable position as it lost its ability to judge the system properties independently [Goldsmith and Eggers, 2004]. This is also a likely contributor to programmatic delays that have almost become a norm.

Significant increase in complexity of some subsystems often derails the whole system development effort. The inability to track and manage such complexity growth actively at both subsystem and system level, have led to suboptimal result and sometimes complete programmatic failure. From the discussion above, we could extract the following key points:

1. Knowledge, adequate visibility of the overall system architecture and distribution of total complexity across different subsystems is absolutely essential for matured complexity management capability. This leads to eventual success of the system development activity. Implementation of *divide and conquer* strategy at system architecture level has to be be kept under tight control – never outsource this aspect of the system to partners.

2. Distribution of overall complexity is a critically important input to the '*divide and conquer*' strategy. It has a big impact on the '*division*' of architecture into chunks.

3. If there are subsystems that significantly more complex, the development team should have the capability to handle this high complexity. Knowledge of the relative subsystem complexities influence selection / composition of the subsystem development team. To do this effectively, visibility to the overall system architecture is absolutely critical. Any asymmetry in complexity distribution is usually counter-productive unless there is asymmetry in complexity management capabilities of responsible development teams.

4. In case of evolving system architecture, one should keep track of the overall evolving architecture to make sure subsystem complexities are contained within sustainable limits. If not, there might be a need to re-structure the

subsystem development team to address the evolving reality. Otherwise, expect negative surprises at the time of overall system integration. It is imperative that every large-scale system development effort does active complexity distribution and management.

The ability to embrace system complexity effectively requires good complexity management. Good strategies for complexity management are imperative for successful execution of complex system development. It is not only about managing the entities (or nodes), but also about managing the network (or the interaction structure amongst entities). Effective management at the entity level does not translate to effective management of interaction structure. Effective complexity management starts from choosing the system architecture.

Chapter Summary

In this chapter, we formulated a methodology for estimating a complexity budget, similar to the notion of mass or power budgets in conventional systems engineering. Since increased complexity often enables improved system performance, we formulated a value function as amount of performance gain per unit NRE penalty due to increased complexity. We found that there exists a maximum system value if the rate of performance gain (with increasing complexity) is greater than the rate of NRE cost/effort penalty due to the increased complexity.

In all other cases, the system value is found to decrease monotonically with increasing complexity. We also addressed the issue of system representation and introduced the notion of *complexity gap,* similar to the notion of information gap in information theory. It was shown that this complexity gap is positive and plays a role similar to the role played by information gap in information theory.

We also linked project duration and scheduling to structural complexity using Norden's model and shown that the estimated development project duration varies super-linearly with structural complexity for any exponent greater than 1.

We discussed the implication of complexity distribution on the system development effort and looked at the interplay between structural complexity and modularity. It is shown analytically and using simulation that complexity and modularity represents two different structural properties associated to the system architecture and need not be negatively correlated as widely believed. They can be controlled using a set of parameters and influences the distribution of complexity across the system architecture.

In this light, we argue that the importance of complete knowledge of the overall system architecture is crucial for decision-making during the development process and constitute a core capability for the primary development organization. This capability is essential for complexity to be tracked and actively managed during the process. It is argued that delivering the system properties is ultimately the responsibility of primary system development organization with an appropriate integrative mechanism where the overall system architecture remains visible and cannot be outsourced to partners.

Complexity management has implications across the spectrum of human endeavor, stretching from engineered system development to coordinating and maintaining our healthcare systems, to name a few.

References:

Bashir, H.A. and Thomson, V., "Models for estimating design effort and time", Design Studies 22 (2001), pp 141–155.

Putnam L.H., "General empirical solution to the macro software sizing and estimating problem", IEEE transactions on Software Engineering, Vol SE4, No 4 (1978), pp 345-361.

Norden P.V., "Manpower utilization patterns in research and development projects", Ph.D Dissertation, Columbia University, 1964.

Bettencourt L.M.A., Lobo J. and Strumsky D., "Invention in the city: Increasing returns to patenting as a scaling function of metropolitan size", Research Policy, Volume 36, Issue 1, 2007, pp 107–120.

Bettencourt L.M.A and West G.," A Unified Theory of Urban Living", *Nature* **467**, 2010, pp 912–913.

Tang, V, and Salminen, V., "Towards a Theory of Complicatedness: Framework for Complex Systems Analysis and Design", 13th International Conference on Engineering Design, Glasgow, Scotland, August 2001

Ranga V. Ramasesh and Tyson R. Browning, "Toward A Theory Of Unknown Unknowns In Project Management", preprint, 2012.

Heimsch D., "http://www.scribd.com/doc/59779160/Boeing-Dream-Liner-Outsourcing-Study"

Cohan P.S., "http://www.dailyfinance.com/2011/01/21/boeing-dreamliner-delays-outsourcing-goes-too-far/"

Hiltzik M., "http://articles.latimes.com/2011/feb/15/business/la-fi-hiltzik-20110215"

Goldsmith S., Eggers W.D., "Governing by Network: The New Shape of the Public Sector", Brookings Institution Press, ISBN-10: 0815731299, 2004.

Weber J., "http://www.businessweek.com/bwdaily/dnflash/content/jan2009/db20090116_971202.htm"

Cohan P.S., "You Can't Order Change: Lessons from Jim McNerney's Turnaround at Boeing", Portfolio Hardcover, ISBN-10: 1591842395, 2008.

Zhao Y., "Why 787 Slips Were Inevitable?", Working Paper, http://zhao.rutgers.edu/787-paper-12-02-2013.pdf, 2012.

Xu, X. and Zhao Y., "Incentives and Coordination in Project-Driven Supply Chains", Working Paper, http://zhao.rutgers.edu/Xu-Zhao-11-23-2013.pdf, 2012.

Hirschi N.W, and Frey D.D., "Cognition and complexity: An experiment on the effect of coupling in parameter design", Research in Engineering Design 13, pp 123 – 131, 2002.

Newman M.E.J., "Networks: An Introduction", Oxford University Press, ISBN-10: 0199206651, 2010.

Cover T.M and Thomas J.A., "Elements of Information Theory", Wiley Interscience; 2nd edition, 2006.

Engel H., "Structure Systems", Hatje Cantz; 3rd edition, ISBN-10: 3775718761, 2007.

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Chapter 7

Dynamic Complexity Quantification

As we have discussed before, there are three primary dimensions of complexity that emerged in the context of system design and development can be grouped as (1) *Structural Complexity*; (2) *Dynamic Complexity* and (3) *Organizational Complexity* [Sheard and Mostashari 2010].

Now, we turn our focus to estimation of dynamic complexity of an engineered complex system. Dynamic complexity refers to the complexity of the dynamical behavior of the system and we will focus primarily on the steady state dynamics, but the methodology is extensible to include transient dynamics as well. Dynamic complexity is a form of complexity that stems from the underlying physics and interrelationships that govern the performance of the system. A highly sophisticated or complex system can behave in a multitude of different ways or modes with rather spontaneous switching of modes [Marczyk 2008]. In this context, higher dynamic complexity implies increased ability to deliver surprises. High dynamic complexity does not necessarily imply many interconnected components.

The dynamic complexity is implicitly a function of system's *operational envelope*. By *operational envelope*, we mean the environments under which the system has to operate. For any system, a significant departure in the operational envelope might alter its dynamic complexity, depending on the extent of deviation in its behavior due to the altered operational envelope.

A Multi-input, multi-output (MIMO) system with very few components can be extremely difficult to understand and control, largely due to the associated interaction uncertainties between responses of the system. Apart from high uncertainty in its behavioral responses, a system can have high dynamical

complexity due to: (i) behavioral dependency structure, (ii) interaction uncertainty between behavioral responses or both of them. Uncertainty in behavioral relationships acts as a form of interaction complexity.

The interaction structure among the associated behavioral responses of the system is represented using a network of interactions among the behavioral responses of a system. We can view the system behavior as a physical process and dynamic complexity is the complexity of that physical process. If this sense, process complexity is a synonym for dynamic complexity. The structural complexity is a measure of complexity of the system architecture, while dynamic complexity develops due to uncertainties in the system during its operation and the direct dependency structure amongst the system responses, in line with the definition put forth by Frizelle and Woodcock [Frizelle and Woodcock, 1995] while characterizing complexity of manufacturing systems.

In this chapter, we develop a dynamic complexity metric having a similar mathematical structure to that of the structural complexity metric developed earlier and apply the proposed dynamic complexity metric to different dynamical systems for validation. Finally the dynamic complexity metric is applied to different categories of jet engines as examples of large MIMO systems. The results corroborate the positive correlation between structural complexity and dynamic complexity, discussed earlier in this thesis (see chapter 1).

7.1 Dynamic Complexity

To successfully manage dynamic complexity throughout system design, development, and deployment, a proper definition of dynamic complexity is essential. In the context of engineered systems, dynamic complexity refers to complexity inherent in the system and is influenced by physical laws. Therefore, we can think of dynamic complexity as the complexity of physical processes governing the system behavior. It is conjectured that higher dynamic complexity implies increasing number of possible failure modes and the effort necessary to cause failure decreases [Marczyk, 2008]. Dynamic complexity captures the notion of

emergent behavior and nonlinear interaction phenomena characteristic of complex engineered systems. Please bear in mind that it is a function of system behavior over the operational envelope of the system.

Dynamic complexity is a function of three fundamental components: (i) inherent uncertainty in system responses; (ii) inherent uncertainty in the pair-wise dependency relationships among system responses; and (iii) dependency structure among those system responses (see Fig. 7.1 below). The inherent uncertainty in system responses can be further grouped into two kinds: (i) uncertainty in pair-wise interaction of system responses; and (ii) uncertainty in system responses themselves.

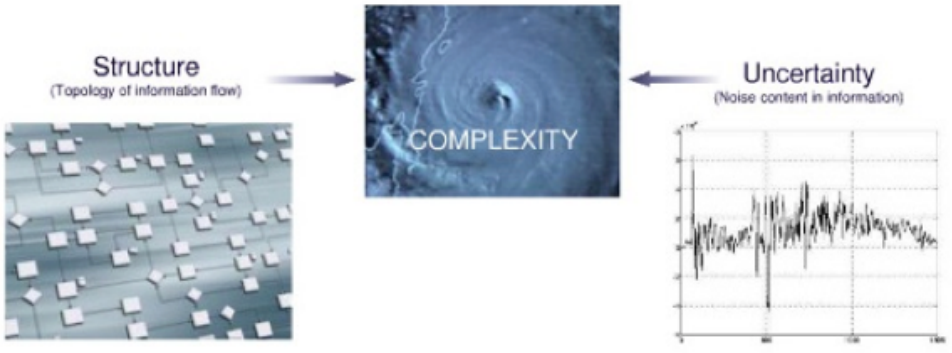


Fig. 7.1: Aspects of dynamic complexity for engineered complex systems [Marczyk, 2008].

Taking an information-theoretic view, we use Claude Shannon's information entropic measures [Cover and Thomas, 2006] for characterization of uncertainty in system responses and also interaction uncertainties, details of which will be elaborated in subsequent sections.

We use the same functional form as in the structural complexity metric (see chapter 2 and 3) for estimating the dynamic complexity of an engineered complex system:

$$Dynamic\ Complexity,\ C = C_1 + C_2 C_3 \qquad (7.1)$$

The individual terms have a different meaning as compared to those in the structural complexity metric. Here the first term, C_1 represents the sum of

complexities of individual system responses. The second term has two factors: (i) sum of complexities of each pair-wise interaction between system responses (C_2) and (ii) effect of the interaction structure among the system responses (C_3).

We use information entropic measures to represent the complexities associated to individual system responses and their pair-wise interactions. Before we move further, let us introduce the information entropic measures that we will be using in formulating the dynamic complexity metric.

7.1.1 *Information Entropy*

Information entropy or the Shannon information entropy, to be precise, is an averaged measure of uncertainty of a random variable. Let Y represent a system response, which is a random variable.

Case 1:

If Y is a discrete random variable, then entropy H(Y) is given by,

$$H(Y) = -\sum_{y \in S} p(y) \ln[p(y)]$$
$$= \sum_{y \in S} p(y) \ln\left[\frac{1}{p(y)}\right] \qquad (7.2)$$

where p(y) is the probability mass function and S is the support of random variable Y, respectively. Comparing eq. 7.2 to the definition of the *expectation* operator $E(.)$, we find,

$$H(Y) = E\left[\ln\left(\frac{1}{p(y)}\right)\right] \qquad (7.3)$$

This form expresses the Shannon information entropy as averaged value of $\left[\ln\frac{1}{p(y)}\right]$. Therefore, the information entropy is in a sense, an averaged measure of uncertainty associated to the random variable under consideration. The inherent uncertainty associated to each behavioral response can be measured using the Shannon information entropy. It is used as a surrogate measure of dynamic

complexity of each individual behavioral response. Notice that this quantity does not consider any relationship to other behavioral responses associated with the system under consideration. The Shannon information entropy is a common indicator of diversity or extent of emergence associated to a behavioral response, Y [Cover and Thomas, 2006, Willcox et al., 2011].

Case 2:

For a continuous random variable Y with probability density function f(y), the entropy (or the *differential entropy*) is given as,

$$h(Y) = -\int_S f(y)\ln[f(y)]dy \qquad (7.4)$$

where *S* represents the support set of the random variable Y.

In practice, we compute the above quantities numerically, using discretization. I that case, the discretized information entropy and the differential entropy are related as [Cover and Thomas, 2006],

$$H(Y^{\Delta}) = h(Y) - \ln(\Delta) \qquad as\ \Delta \to 0 \qquad (7.5)$$

where Δ is the bin size used in discretization of Y (see Fig. 7.2 below).

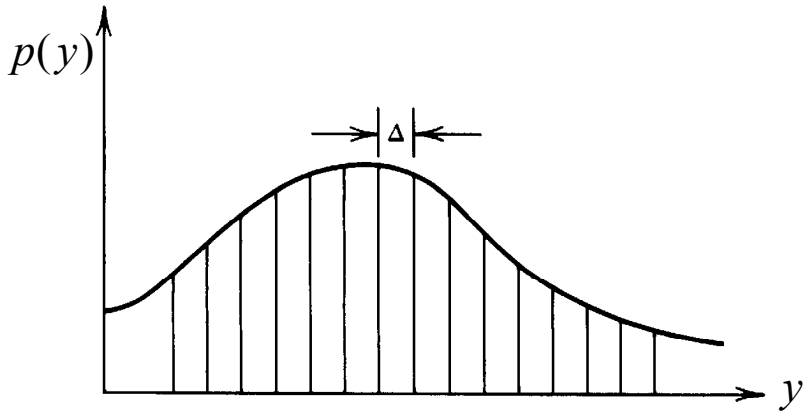


Fig. 7.2: Discretization of a continuous random variable [Cover and Thomas, 2006].

We define the complexity associated to individual system responses Y_i as,

$$\alpha_i = H(Y_i) \qquad (7.6)$$

7.1.2 *Mutual Information*

Mutual information is a measure of the amount of information that one random variable contains about another random variable. Therefore, it is the reduction in uncertainty of one random variable due to the availability of the knowledge of the other. Let us consider two random variables Y_i and Y_j with a joint probability mass function $p(y_i, y_j)$ and marginal probability mass functions $p(y_i)$ and $p(y_j)$. The mutual information $I(Y_i; Y_j)$ is the relative entropy between the joint distribution $p(y_i, y_j)$ and the product of the distributions, $p(y_i).p(y_j)$. In a sense, it is the "*distance*" between the joint probability distribution $p(y_i, y_j)$ and the product of distributions $p(y_i).p(y_j)$. Mutual information is defined as:

$$I(Y_i;Y_j) = \sum_{y_i \in S_i} \sum_{y_j \in S_j} p(y_i, y_j) \ln\left[\frac{p(y_i, y_j)}{p(y_i).p(y_j)}\right] \qquad (7.7)$$

where S_i and S_j represents the support of random variables Y_i and Y_j respectively. Also the joint information entropy of random variables Y_i and Y_j is defined as,

$$H(Y_i, Y_j) = -\sum_{y_i \in S_i} \sum_{y_j \in S_j} p(y_i, y_j) \ln\left[p(y_i, y_j)\right] \qquad (7.8)$$

We can express the mutual information in terms of the individual and joint information entropy as follows:

$$\begin{aligned} I(Y_i;Y_j) &= \sum_{y_i, y_j} p(y_i, y_j) \ln\left[\frac{p(y_i, y_j)}{p(y_i).p(y_j)}\right] \\ &= \sum_{y_i, y_j} p(y_i, y_j) \ln[p(y_i, y_j)] - \sum_{y_i, y_j} p(y_i, y_j) \ln[p(y_i)] - \sum_{y_i, y_j} p(y_i, y_j) \ln[p(y_j)] \\ &= -H(Y_i, Y_j) - \underbrace{\sum_{y_i} p(y_i) \ln[p(y_i)]}_{-H(Y_i)} - \underbrace{\sum_{y_j} p(y_j) \ln[p(y_j)]}_{-H(Y_j)} \\ &= H(Y_i) + H(Y_j) - H(Y_i, Y_j) \qquad (7.8) \end{aligned}$$

The above expression is similar to set theoretic operations described using the Venn diagram (see Fig. 7.3 below).

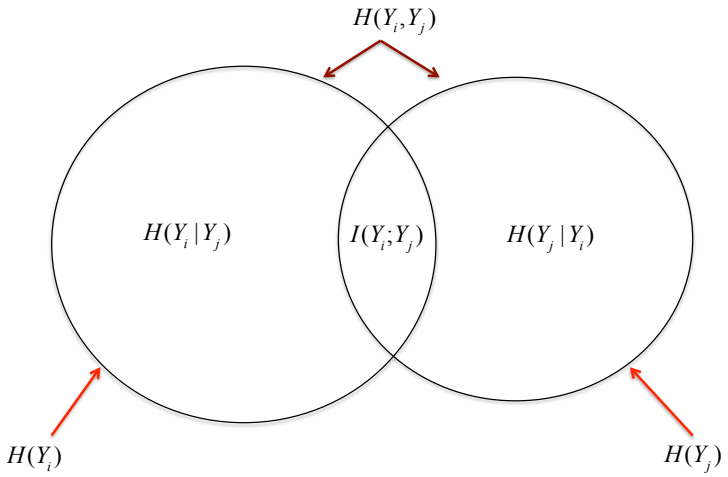


Fig. 7.3: Relationship between entropy and mutual information using Venn diagram [Cover and Thomas, 2006].

The mutual information $I(Y_i; Y_j)$ corresponds to the intersection of the information entropy in Y_i with the information entropy in Y_j. Also notice that the mutual information between two continuous random variables is independent of their discretization,

$$I(Y_i^{\Delta};Y_j^{\Delta}) = I(Y_i;Y_j) \qquad (7.9)$$

In fact, mutual information is invariant under linear transformation and arbitrary re-parameterization using invertible maps [Schlick, 2012] and therefore, is a robust measure of statistical dependence. This is a helpful property from a numerical perspective. In complex engineered systems, there are usually multiple behavioral responses that are of primary interest. These responses may or may not be correlated in a pair-wise sense. If any pair of behavioral responses is strongly correlated, there is a pair-wise dependency between the responses and inherent uncertainty of this interaction over the operational modes of the system influence the dynamic complexity of the system. Mutual information expresses the amount of uncertainty one can reduce for a response Y_j, given the knowledge of response Y_i for two correlated responses. Therefore lower mutual information indicates higher uncertainty for any pair-wise interaction. The pair-wise interaction complexity is given as the inverse of mutual information. We define interaction complexity as,

$$\left.\begin{aligned}\beta(i,j) &= \frac{1}{I(Y_i;Y_j)},\ if\ I(Y_i;Y_j) \neq 0 \\ &= 0,\quad otherwise\end{aligned}\right\} \qquad (7.10)$$

Hence, higher mutual information implies lower behavioral interaction complexity and vice versa (see figure 7.4).

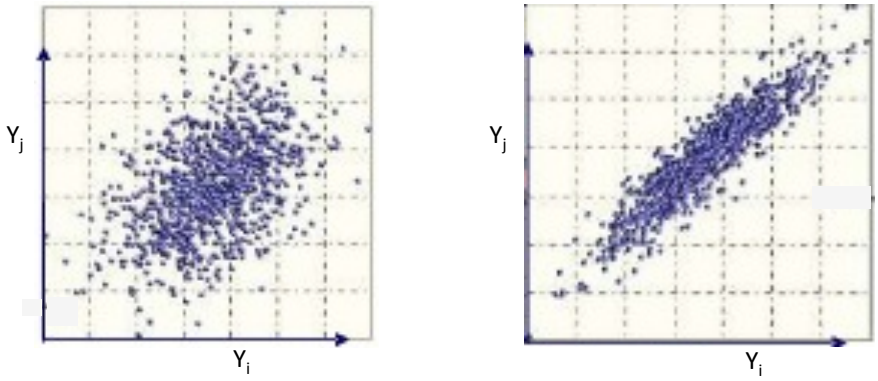
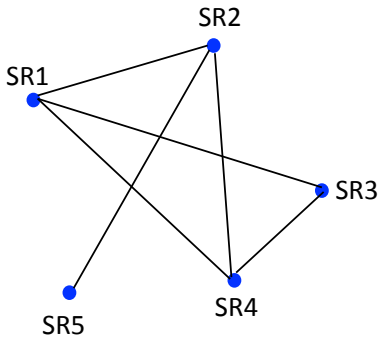


Fig. 7.4: Interaction complexity over the operational envelope: (a) higher interaction complexity, and (b) lower interaction complexity [Marczyk 2008].

7.1.3 *Interaction Structure*

The interaction structure among the behavioral responses of a system is represented via a graph in which the behavioral responses are the nodes (vertices) of the graph. Links or edges of this graph represent pair-wise interaction between behavioral responses. The generation of the binary (0/1) dependency structure matrix of representative behavioral responses can be authored from the analytical model if all relationships are very well understood (unlikely for a large complex system); or correlation structure generated from studying the system at different states (more applicable to large, complex system) or a combination of these two methods. Here we opt for data-driven approach using correlation structure, where data may be generated from simulation or observation (i.e., experiment). Generation of the dependency structure matrix or the *A* matrix (i.e., connectivity map of behavioral responses) hinges on the computation of correlation among behavioral responses. The diagonal elements of the *A* matrix are zero. An example of this behavioral dependency structure matrix or the *A* is shown below in fig. 7.5. There

are 5 system responses labeled SR1 to SR5 and the corresponding *A* matrix is shown below. This is same as the adjacency matrix applied to the system behavioral domain where each node stands for a system response of interest.



$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 7.5: A hypothetical system with 5 system responses labeled SR1 to SR5 with the dependency structure shown as a network and the corresponding *A* matrix. The interdependent system responses are assumed to be bi-directional for illustration.

Obtaining the correlation structure, and subsequent generation of the *A* matrix, is a very active research area [Kolar *et al.*, 2013] and there are significant recent methodological developments that help resolve existence of intricate, nonlinear correlations from observed data [Feizi *et al.*, 2013]. For details, please refer to appendix M.

Once the dependency structure matrix is generated, next step is to compute a measure of topological complexity of this dependency structure. This is essentially the complexity of the correlation structure among the system responses observed over the operational envelop of the system. We use the same fundamental notion of topological complexity as in the case of developing the topological complexity measure for the structural complexity metric (see chapter 3). The associated *matrix energy* of *A* matrix is defined as the sum of its singular values σ_k:

$$E(A) = \sum_{k=1}^{n} \sigma_k \qquad (7.11)$$

The *matrix energy* also expresses the minimal effective dimension embedded within the connectivity pattern represented through the binary dependency matrix. Notionally, this quantity encapsulates the *structure* of dependency structure among

the behavioral responses. Using singular value decomposition (SVD), we can express matrix A as:

$$A = \sum_{k=1}^{n} \sigma_k \underbrace{u_k v_k^T}_{E_k} = \sum_{k=1}^{n} \sigma_k E_k \qquad (7.12)$$

where E_k represents simple, building block matrices of unit matrix energy and unit norm. Using this view, we observe that *matrix energy* or *graph energy* express the sum of weights associated with the building block matrices required to represent or reconstruct the binary dependency matrix *A*. In this context, it represents the difficulty or complexity in understanding and thereby intervening in the operational behavior of the system. It serves as an indicator/predictor of inherent structure in the dependency relationships among behavioral responses [Recht et al., 2010]. Lower matrix energy implies that we can effectively reduce the system dimension without greatly jeopardizing our understanding of the system behavior. Extent of such understanding reduces as the dependency structure become more distributed, leading to larger values of matrix energy and thereby increasing complexity.

Combining the individual terms in relation 7.1, we arrive at the final form of dynamic complexity metric,

$$C = \underbrace{\sum_{i=1}^{n} H(Y_i)}_{C_1} + \underbrace{\left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta(i,j).A(i,j) \right)}_{C_2} \underbrace{\left(\frac{E(A)}{n} \right)}_{C_3} \qquad (7.13)$$

where *A* is the binary (0/1) dependency matrix of *n* behavioral responses of the system. Note that this functional form is identical to that of the structural complexity metric, the difference being what the individual terms mean and how they are estimated.

7.1.4 *A notional example with two system responses*

The uncertainties in system outputs/responses stems from the input uncertainties in design variables and system parameters. The uncertainty in system responses and their interactions are reflections of propagation of input uncertainties. Let us consider a very simple interacting system with just two system

responses, {Y_1, Y_2}. Let *X* be the vector of all design variables and system parameters that dictates the system behavior and the responses are linked to the input vector as follows:

$$Y_1 = f_1(X);\ Y_2 = f_2(X) \qquad (7.14)$$

Now the information entropic measures for the system responses are:

$$\begin{aligned} H(Y_1) &= -\int_S p(Y_1)\ln[p(Y_1)]dy_1 \\ &= -\int_S p\{f_1(X)\}\ln[p\{f_1(X)\}]d[f_1(X)] \\ &= g[p\{f_1(X)\}] \end{aligned} \qquad (7.15)$$

Similarly, we can express,

$$H(Y_2) = g[p\{f_2(X)\}] \qquad (7.16)$$

As shown in eq. 7.8, computation of mutual information, $I(Y_1;Y_2)$ will require the knowledge of joint probability distribution of the system responses, $p(Y_1,Y_2)$, which is a derived joint distribution involving the system vector, *X*. The interaction complexity can be expressed as,

$$\beta = \frac{1}{I(Y_1;Y_2)} = h[p\{f_1(X)\}, p\{f_1(X)\}] \qquad (7.17)$$

For the simple system described here, the behavioral dependency matrix is,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix};$$

$$\Rightarrow E(A) = 2,\ n = 2,\ C_3 = \frac{E(A)}{n} = 1$$

The dynamic complexity for this simple system, in terms of input uncertainties can be represented as,

$$C = H(Y_1) + H(Y_2) + \beta.1$$
$$= g[p\{f_1(X)\}] + g[p\{f_2(X)\}] + h[p\{f_1(X)\}, p\{f_2(X)\}] \qquad (7.18)$$

It is quite clear that we have to depend on computation procedures for computing the joint probability distribution and subsequently mutual information and dynamic complexity. The computation procedure for dynamic complexity calculation is detailed in the following section

7.2 Computational Procedure

For computation of information entropy of individual behavioral responses, its probability density function is the primary input and is computed using the procedure in [Botev et al. 2010]. For this study, we will not focus on the complexity of individual behavioral responses. The generation of binary dependency structure matrix A is based on the correlation structure amongst system responses and the computational procedure is as follows:

1. Identify main behavioral responses n and compute them for different operational modes s over the entire operational envelope of the system. This generates a $s \times n$ data matrix. Each entry of this data matrix is a number. The behavioral response values are normalized to lie in the interval [0,1] before performing the correlation analysis.
2. Compute p-values for testing the null hypothesis of no correlation. Each p-value represents the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero. Consider a pair of responses (i, j). If p(i,j) is small, say less than 0.05, then the correlation r(i,j) is significant. Extract the entire correlation structure using the data from step 1.
3. In addition to statistical significance, a correlation is deemed *important* if abs[r(i, j)] > *threshold* value. A practical threshold value to use is found to be 0.25 and has been used in all examples studied here. This is a parameter to

be chosen before embarking on the correlation analysis and can vary across different application areas.

4. The correlation structure thus computed could still have indirect dependencies based on the observed data. That is, if two system responses are not directly dependent, we can still get a false dependency if there is an indirect path linking through other system responses. Extract the primary correlation structure or the direct dependency structure by applying the spectral de-convolution methodology described in [Feizi *et al.*, 2013]. This will give us a weighted behavioral dependency matrix, which is then converted to a binary adjacency matrix in system behavioral space as defined by the systems operational envelope (e.g., where nodes represent system responses and edges reflect pair-wise dependencies between them). The details of the adopted computational procedure can be found in the appendix M.
5. Populate the behavioral dependency matrix, A with 0's and 1's. If significant the correlation exists between response-pair (i, j) and deemed important based on the filters used in step 2, assign A(i,j) = 1, otherwise A(i,j) =0.

Depending on data, one can use a combination of different correlation coefficients (i.e., Pearson correlation, Spearman correlation). Pearson's method applies best for linear correlations while Spearman's rank correlation works well for nonlinear, but monotonic relationships among responses. Most often, both of these methods lead to the same or very similar outcome with Pearson's method resulting in more conservative estimates. It is possible to include advanced methods for correlation structure detection [Fan and Liu, 2013], depending on the problem specific characteristics of data. Choice of correlation structure detection method has no fundamental bearing on the overall methodology for dynamic complexity computation procedure. Estimation of connectivity or dependency structure from available data using correlation structure extraction is a growing research area with significant computational methodological advances [Kolar *et al.,* 2013]. Recent

methodological developments using spectral approaches [Feizi *et al*., 2013] helps decipher primary correlation structure from observed system behavioral data.

The computation process for mutual information hinges upon the computation of joint probability distribution function for a pair of system responses under consideration. This procedure is performed for all pairs of interacting behavioral responses found in the dependency matrix. The joint probability distribution function is constructed based on behavioral responses computed at different operational modes over the entire operational envelope, and use the nonparametric kernel density estimation approach. The advantage of the nonparametric approach is that it offers a far greater flexibility in modeling a given dataset and, unlike the classical parametric approach.

We use an adaptive kernel density estimation method based on the smoothing properties of linear diffusion processes that leads to a simple and intuitive kernel estimator with substantially reduced asymptotic bias and mean square error [Botev et al. 2010]. Once the joint probability density function $p(y_i,y_j)$ has been constructed, we compute the individual marginal density functions $p(y_i)$ and $p(y_j)$ for the respective behavioral responses [Cover and Thomas 2006]. These probability density functions are used to compute the mutual information $I(Y_i;Y_j)$. The interaction complexity is defined as the inverse of mutual information. The procedure for computing interaction complexity is as follows:

- If (Y_i, Y_j) are statistically independent, A(i,j) = 0 and $\beta(i,j) = 0$
- Otherwise, A(i,j) = 1 and $\beta(i,j) = \dfrac{1}{I(Y_i;Y_j)}$

We demonstrate operationalization and validation of the proposed methodology using examples of simple dynamical systems in the following subsection and finally apply this methodology to more complicated examples of large jet engines with different architectures. The same methodology can be extended to handle dynamic complexity in transient regime where dynamic complexity becomes a time-dependent quantity.

7.3 Illustrative Examples

We present a set of simpler examples to demonstrating the mechanics of the methodology and to validate the appropriateness of the proposed dynamic complexity metric. We start with a simple dynamical system, a damped, driven pendulum, to validate the methodology. Subsequently, we apply this methodology on increasingly involved dynamical systems, starting with a double pendulum. In all these simpler cases, we mostly have system responses that are all interacting to each other (i.e., form a fully connected network of system responses). These dynamical systems are simulated under *non-chaotic* and *chaotic* regimes [Taylor, 2005; Strogatz, 2001] and their dynamic complexity is computed under both scenarios. For each dynamical system, it is intuitive that dynamics under chaotic regime is more complex and any dynamic complexity metric should validate the same. Also, as the dynamics of these systems are progressively more complex and the dynamic complexity metric should reflect the same. We use these examples to validate the proposed dynamic complexity metric.

In all cases, the levels of significance for each correlation component were set to 0.99 (i.e., 99% level of significance). The threshold value for correlation coefficient is fixed at 0.30 for all examples.

(a) Damped, driven pendulum

Let us consider a damped, driven pendulum of mass m and length l. The setup is shown in fig. 7.6 below. The equation of motion about the pivot (see fig. 7.6) can be written as, $I\ddot{\theta} = \Gamma$ where I stands for the rotational moment of inertia and Γ stands for the net torque about the pivot. In this case, we have $I = ml^2$ and the damping force is given by $bv = bl\dot{\theta}$. The driving force is assumed to be sinusoidal, $F(t) = F_0 \sin(\omega_d t)$ with F_0 is the driving amplitude and ω_d is the driving frequency.

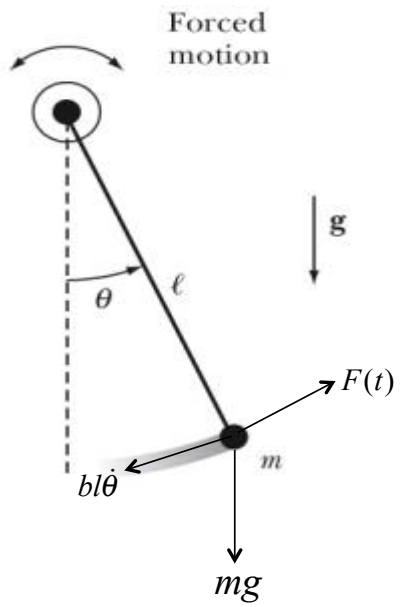


Fig. 7.6: Damped, driven pendulum with forces acting on the bob.

The equation of motion of a damped, driven pendulum is given as:

$$ml^2\ddot{\theta} + bl^2\dot{\theta} + mgl\sin\theta = lF_0\sin(\omega_D t)$$

$$\Rightarrow \ddot{\theta} + \frac{b}{m}\dot{\theta} + \frac{g}{l}\sin\theta = \frac{F_0}{ml}\sin(\omega_D t)$$

$$\Rightarrow \ddot{\theta} + 2\beta\dot{\theta} + \omega_0^2\sin\theta = \gamma\omega_0^2\sin(\omega_D t) \qquad (7.19)$$

where,

$$\left.\begin{aligned} 2\beta &= \frac{b}{m} \\ \omega_0^2 &= \frac{g}{l} \\ \gamma &= \frac{F_0}{mg} \end{aligned}\right\}$$

We can think of γ as a normalized driving force amplitude. If $\gamma < 1$, the driving force amplitude (F_0) is smaller than the weight of the pendulum bob (mg) and we expect the motion to be small and periodic. For $\gamma > 1$, the driving force is higher than weight of the pendulum, and we expect the ensuing motion to be large. We will see

that a progressively larger value of γ, while keeping all other parameters in eq. 7.19 fixed, may drive the system to the chaotic regime.

A geometric tool that we use for illustrating and differentiating between non-chaotic and chaotic regimes is called the *Poincare section*. This is a way to visualize the phase space of the dynamical system and provides glimpses into the trajectories in a lower dimensional space. Think of placing a plane in space, perpendicular to the plane of pendulum motion such the pendulum always crosses this plane. On this plane, if we plot the phase space of the system, what we get is a Poincare section. For the damped, driven pendulum example, the Poincare section can be obtained by sampling at regular time intervals.

By manipulating the system parameters we can simulate its behavior in non-chaotic region and in chaotic region. There are different operating conditions and combinations of parameters that can lead to chaotic dynamics. In this example, we keep the same initial conditions and set $2\beta = 0.5$, $\omega_0 = 1$ and $\omega_D = 0.667$. The driving force amplitude, γ is varied. In Fig. 7.7, the corresponding Poincare sections [Taylor 2005] for $\gamma = 0.8$ and $\gamma = 1.2$ are shown for illustration. As we can see, the damped, driven pendulum shows periodic behavior at low γ value and turns chaotic at $\gamma = 1.2$.

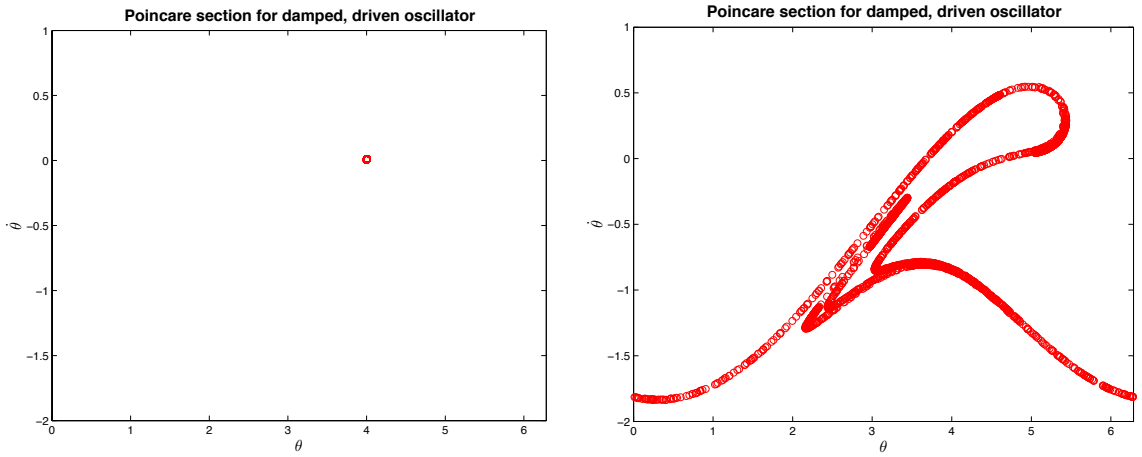


Fig. 7.7: Poincare Section for (a) non-chaotic regime with $\gamma = 0.8$, and (b) chaotic regime with $\gamma = 1.2$ for damped, driven pendulum system.

The periodicity of the solution for $\gamma = 0.8$ is visible in the Poincare section, plotted above. At any given section (or any given time instance), the pendulum bob possesses nearly identical phase-space (this can change somewhat for very high damping values).

In this example, there are two system responses (i.e., $n = 2$) that are related and we have the binary behavioral dependency matrix, $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ with $E(A) = 2$ and $\frac{E(A)}{n} = 1$. Under all cased considered here, the behavioral dependency remains the same. We simulated the pendulum system at $\gamma = 0.8; 1.05; 1.10$ and 1.20 using MATLAB™ with *ode113* as the solver. We subsequently computed the dynamic complexity of the system at each value of γ and the result is shown in table 7.1 below:

Table 7.1: Dynamic Complexity of damped, driven pendulum moving from non-chaotic (low driving force) to chaotic regime (high driving force).

γ	**Dynamic Complexity**
0.8	0
1.05	0.248
1.10	0.466
1.20	0.667

At low value of the driving force amplitude, the motion is small and periodic. There is no uncertainty as such and we get a dynamical complexity that is zero. As the motion starts to become larger, there is an increasing degree of non-periodicity and that results in a small, but finite dynamic complexity. The dynamic complexity increased by a factor of 2.8 as the system moved from the boundary of non-periodic behavior (i.e., $\gamma = 1.05$) to the chaotic region (i.e., $\gamma = 1.20$). This corroborates with our intuitive notion of complexity of the dynamical process observed [Taylor, 2005]

and validates the proposed dynamic complexity metric, at least on this simple dynamical system.

(b) Double pendulum

A planar double pendulum is a mechanical system with two simple pendulums as shown in Fig. 7.8 below. Here, m_1, L_1, and θ_1 represent the mass, length and the angle from the normal of the inner bob and m_2, L_2, and θ_2 stand for the mass, length, and the angle from the normal of the outer bob. The position of mass m_1 is (x_1, y_1) and that of mass m_2 is (x_2, y_2). With the origin being the pivot point at the top of the double pendulum, the positions of the two masses are given as,

$$\left.\begin{aligned} x_1 &= L_1 \sin\theta_1 \\ y_1 &= -L_1 \cos\theta_1 \\ x_2 &= L_1 \sin\theta_1 + L_2 \sin\theta_2 \\ y_2 &= -L_1 \cos\theta_1 - L_2 \cos\theta_2 \end{aligned}\right\}$$

In this example, we assign the following parameters and use the same for all simulations, $m_1 = m_2 = 1$; and $L_1 = L_2 = 1$.

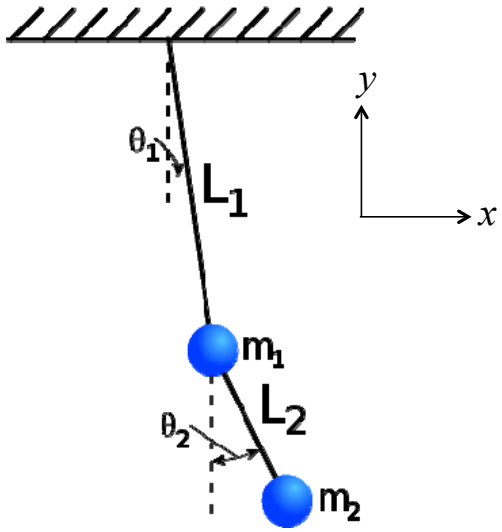


Fig. 7.8: Undamped, double pendulum example [Roja, 2009].

The equation of motion of a double pendulum in phase-space is given as:

$$M(t,y)\dot{y} = f(t,y) \qquad (7.20)$$

where,

$$M(t,y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (m_1+m_2)L_1 & 0 & m_2 L_2 \cos(y_3 - y_1) \\ 0 & 0 & 1 & 0 \\ 0 & m_2 L_1 \cos(y_3 - y_1) & 0 & m_2 L_2 \end{bmatrix}$$

$$f(t,y) = \begin{Bmatrix} y_2 \\ -(m_1+m_2)g\sin(y_1) + m_2 L_2 y_4^2 \sin(y_3 - y_1) \\ y_4 \\ -m_2 g \sin(y_3) - m_1 L_1 y_2^2 \sin(y_3 - y_1) \end{Bmatrix}$$

$$y = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]$$

This dynamical system is capable of exhibiting chaotic behavior under certain input conditions. The behavior of this double pendulum varies from *non-chaotic* motion at low energies, to *chaotic* at intermediate energies, and back to *non-chaotic* motion at high energies [Roja 2009]. Simulations show that the behavior of a double pendulum is regular at low energies, chaotic at intermediate energies, and back to regular at high energies. In figure 7.9 below, we plot the phase-space of the second pendulum mass, m_2 corresponding to a *quasiperiodic* motion (i.e., nearly periodic) and *chaotic* motion respectively. This can be achieved by varying the initial condition. The quasiperiodic solution corresponds to $y = [\pi, 0.5, 0, 0.5]$ and the chaotic solution corresponds to $y = [\pi, 0.5, \pi, 0]$. We use the quasiperiodic motion as a representative non-chaotic solution.

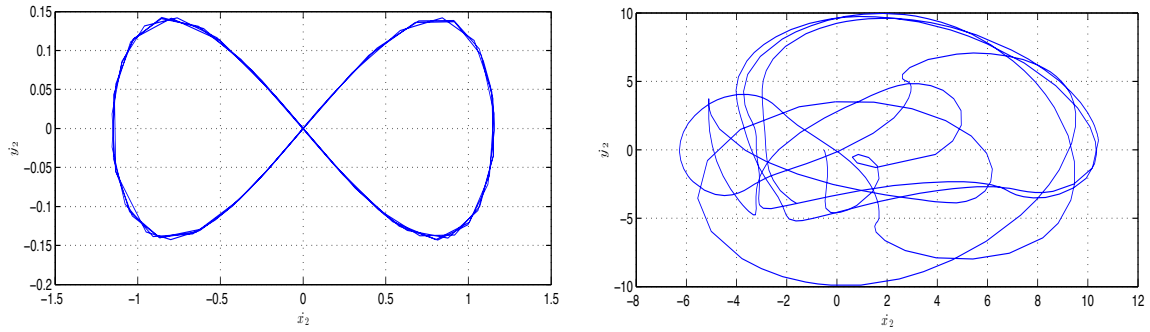


Fig. 7.9: Velocity profile of the second pendula mass m_2 for (a) non-chaotic and (b) chaotic regimes respectively. They correponds to initial conditions $y=[\pi,0.5,0,0.5]$ and $y=[\pi,0.5,\pi,0]$ respectively.

The variation in dynamic complexity for this double pendulum as it moves from non-chaotic to chaotic regime is shown in the table below:

Table 7.2: Dynamic Complexity of a double pendulum

Regime	**Dynamic Complexity**
Non-chaotic	3.2
Chaotic	10.4

The dynamic complexity increased by a factor of 3.25 as the system moves from a non-chaotic to chaotic regime.

(c) Variations of Double pendulum on cart type systems

Now we consider variants of "double pendulum on cart" type systems with increasingly complex dynamics. The first one is a simple double pendulum on cart as shown in Fig. 7.10. The pendulums are considered as a point mass and we neglect the inertial properties. Physically, the angles subtended by the pendulums (θ_1,θ_2) are not coupled and can move independently. The mass of the cart is large as compared to the masses of the pendulum. In subsequent simulations of the double pendulum system shown in fig. 7.10, we use the following parameters: $m_0=20$; $m_1=m_2=1$ and $l_1=l_2=1$.

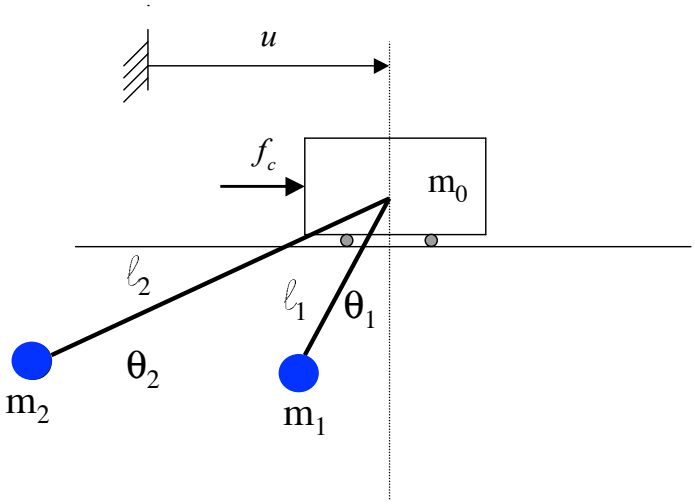


Fig. 7.10: Double pendulum on a cart

The equation of motion in phase-space can be written in the form:

$$M(t,y)\dot{y} = f(t,y) \qquad (7.21)$$

where,

$$M(t,y) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & (m_0+m_1+m_2) & 0 & -m_1 l_1 \cos(y_3) & 0 & -m_2 l_2 \cos(y_5) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -m_1 l_1 \cos(y_3) & 0 & m_1 l_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -m_2 l_2 \cos(y_5) & 0 & 0 & 0 & m_2 l_2^2 \end{bmatrix}$$

$$f(t,y) = \begin{Bmatrix} y_2 \\ f_c - m_1 l_1 y_4^2 \sin(y_3) - m_2 l_2 y_6^2 \sin(y_5) \\ y_4 \\ -m_1 g l_1 \sin(y_3) \\ y_6 \\ -m_2 g l_2 \sin(y_5) \end{Bmatrix}$$

$$y = [u, \dot{u}, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]$$

This dynamical system is also capable of exhibiting chaotic behavior under some initial conditions. For the assigned set of parameters, we observed quasiperiodic behavior for the initial condition, $y = [0,0,0.2,0,0.28,0]$. Simulating the system, we extract the following binary dependency matrix,

$$A_{non-chaotic} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The graph energy of this behavioral dependency matrix, $E(A_{non-chaotic}) = 6.16$.

Now let us look at the chaotic regime observed for a different initial condition. Chaotic regime was observed for the following initial condition, $y = [0,0,0.5\pi,0,\pi,0.1]$ and subsequent simulation led to the following binary dependency matrix if the system is operating under chaotic regime,

$$A_{chaotic} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The graph energy of this behavioral dependency matrix, $E(A_{chaotic}) = 8.82$. Notice that there are additional behavioral dependencies that have shown up. This is an interesting observation and implies that there is enough evidence of additional interdependency between system responses statistically in this operating region. Similar behavior has been reported elsewhere in the literature [Gassmann, 1997]. The result for the double pendulum on cart is summarized in table 7.3 below:

Table 7.3: Dynamic Complexity of a double pendulum on cart

Regime	**Dynamic Complexity**
Non-chaotic	14.37

Chaotic	55.35

We observe a significant increase in dynamic complexity (by a factor of 3.85) as the system turned chaotic, as one would expect. Notice that, the increase due to topological differences in the response dependency network is only about 43%. This means that the significant changes in dynamic complexity stems from increasing uncertainty in the dynamics of the system as one would expect as move into a chaotic regime.

The second variant is an inverted double pendulum on a cart. In this case, the pendulums are moeled as rigid body and we no longer neglect the inertial properties of the pendulums. The angles subtended by the pendulums (θ_1,θ_2) are coupled in this setup (see Fig. 7.11 below).

This cart-pendulum systems is an example of an underactuated mechanical system and has been studied extensively for investigating effectiveness of various control schemes and demonstrating ideas emerging in the area of nonlinear control [Xin, 2008; Bogdanov, 2004].

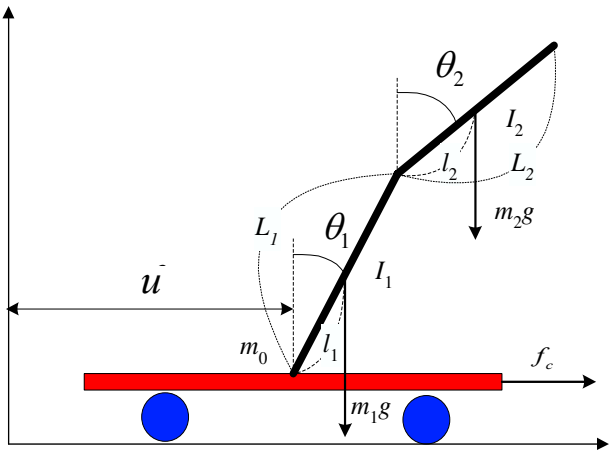


Fig. 7.11: Inverted Double pendulum on a cart [Bogdanov, 2004]

The equation of motion in phase-space can be written in the form:

$$M(t,y)\dot{y} = f(t,y) \qquad (7.22)$$

where,

$$M(t,y)=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_0 & 0 & \frac{\beta_1}{g}\cos(y_3) & 0 & \frac{\beta_2}{g}\cos(y_5) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{\beta_1}{g}\cos(y_3) & 0 & \alpha_1 & 0 & \alpha_3\cos(y_3-y_5) \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{\beta_2}{g}\cos(y_5) & 0 & \alpha_3\cos(y_3-y_5) & 0 & \alpha_2 \end{bmatrix}$$

$$f(t,y)=\begin{Bmatrix} y_2 \\ \frac{\beta_1}{g}y_4^2\sin(y_3)+\frac{\beta_2}{g}y_6^2\sin(y_5)+f_c \\ y_4 \\ -\alpha_3 y_6^2\sin(y_3-y_5)+\beta_1\sin(y_3) \\ y_6 \\ \alpha_3 y_4^2\sin(y_3-y_5)+\beta_2\sin(y_5) \end{Bmatrix}$$

$$y=[u,\dot{u},\theta_1,\dot{\theta}_1,\theta_2,\dot{\theta}_2]$$

$$\begin{cases} \alpha_0=m_0+m_1+m_2 \\ \alpha_1=m_1l_1^2+m_2L_1^2+I_1=\left(\frac{1}{3}m_1+m_2\right)L_1^2 \\ \alpha_2=m_2l_2^2+I_2=\frac{1}{3}m_2L_2^2 \\ \alpha_3=m_2L_1l_2=\frac{1}{2}m_2L_1L_2 \\ \beta_1=m_1gl_1+m_2gL_1=\left(\frac{1}{2}m_1+m_2\right)gL_1 \\ \beta_2=m_2gl_2=\frac{1}{2}m_2gL_2 \end{cases}$$

Characterization of the chaotic regime followed the procedure as in [Roja, 2009]. We used quasiperiodic regime as a representative non-chaotic region and look at a weakly chaotic regime (based on the value of the Lyapunov exponent) as

representative of the chaotic regime for this inverted double pendulum on cart system. In subsequent simulations of the system shown in fig. 7.11, we use the following parameters: $f = 0$; $m_1 = m_2 = 1$ and $l_1 = l_2 = 1$. Hence, this is not a forced motion of the inverted double pendulum system. For the assigned set of parameters, we observed quasiperiodic behavior for the initial condition, $y = [0,0,\pi,0.5,\pi,0]$. Simulating the system, we extract the following binary dependency matrix,

$$A_{non-chaotic} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The graph energy of this behavioral dependency matrix, $E(A_{non-chaotic}) = 8.29$.

The mildly chaotic regime was observed for the following initial condition, $y = [0,0,0.5\pi,0,\pi,0.1]$ and subsequent simulation led to the following binary dependency matrix if the system is operating under chaotic regime,

$$A_{chaotic} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The graph energy of this behavioral dependency matrix, $E(A_{chaotic}) = 8.37$. Notice changes in the dependency structure of the system responses. In this case, the differences were not as large as in the previous case, but note that the operating region is only mildly chaotic with a comparatively smaller Lyapunov exponent. Notice that, the topological difference in the response dependency network is negligible and is only about 1%.

The result is summarized in table 7.4 below:

Table 7.4: Dynamic Complexity of an inverted double pendulum on cart

Regime	Dynamic Complexity
Non-chaotic	24.04
Chaotic	75.05

We observe a significant increase in dynamic complexity (by a factor of 3) as the systems turned chaotic. This means that the significant changes in dynamic complexity stems from increasing uncertainty in the dynamics of the system only as the difference due to topology of dependency structure of the system responses is minimal. Compared to the previous example, the extent and impact of uncertainty is greater in this example.

If we look at different kinds of double pendulum systems in non-chaotic regime, the dynamic complexity is found to increase as we move from simple double pendulum to inverted double pendulum on cart (see table 7.5). This matches with our intuitive understanding of evolution of dynamic complexity.

Table 7.5: Dynamic Complexity of dynamical systems in non-chaotic regime

Dynamical System	Dynamic Complexity
Double pendulum	3.2
Double pendulum on cart	14.37
Inverted double pendulum on cart	24.04

Across the different dynamical systems considered, we observe that the proposed dynamic complexity metric exactly matches with our intuitive view of dynamic complexity for such systems. We also noticed that, within the same physical system, the primary source of dynamic complexity here stems from the interaction uncertainty.

7.4 Case Study: Dynamic Complexity of Aircraft Engines

A jet engine is an air-breathing reaction engine that discharges a fast moving jet that generates thrust by jet propulsion. In general, most jet engines are internal combustion engines [Mattingly, 1998]. Jet engine architectures considered here

includes turbojets, turbofans and geared turbofans. We will now apply the developed methodology on a set of aircraft engines with different architectures (i.e., turbojet, dual spool turbofan and a dual spool geared turbofan) to compute and compare their dynamical complexities.

Early jet aircraft used turbojet engines (see Fig. 7.12(a)) that were relatively inefficient for subsonic flight and noisy. Turbojets consist of an air inlet, an air compressor, a combustion chamber, a gas turbine that drives the air compressor and a nozzle. The air is compressed into the chamber, heated and expanded by the fuel combustion and then allowed to expand out through the turbine into the nozzle where it is accelerated to high speed to provide propulsion. Modern subsonic jet aircraft use high-bypass turbofan engines that offer high speed with fuel efficiency (see figure 7.12b). Modern high-bypass turbofans evolved from the two-spool axial-flow turbojet engine, essentially by increasing the relative size of the low-pressure (LP) compressor to the point where most of the air exiting the unit actually bypasses the core (or gas-generator) stream passing through the main combustor. Since most of the airflow through a high-bypass turbofan is low-velocity bypass flow, even when combined with the much higher velocity engine exhaust, the net average exhaust velocity is considerably lower compared to a pure turbojet. Jet engine noise is a function of exhaust velocity and therefore turbofan engines are significantly quieter than a turbojet of the same thrust.

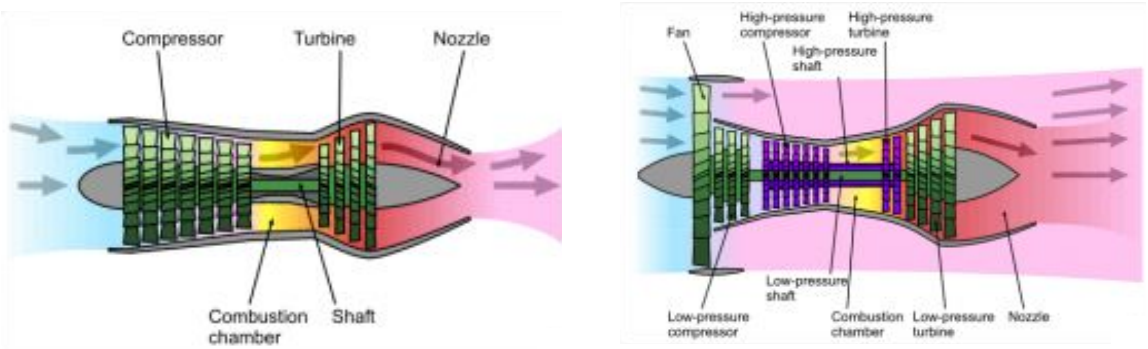


Fig. 7.12: (a) Turbojet, and (b) Dual-spool turbofan engine schematics.

The geared turbofan (GTF) engine concept has existed in smaller class turbofan engines for several decades, but many engine manufacturers remained reluctant to

invest in the technology until recently [Riegler et al. 2007] due to perceived concerns about engine maintainability and system complexity, often on the part of aircraft manufacturers or the airlines themselves. The underlying principle of the GTF engine, shown in Fig. 7.13, is to further increase bypass ratio over current designs in order to improve propulsive efficiency, decreasing noise and hopefully weight at the same time. This can be achieved by reducing fan speed and pressure ratio for high bypass ratio fans, and increasing low-pressure compressor (LPC) and low-pressure turbine (LPT) speeds, thereby achieving very high component efficiency. Propulsive efficiency of a turbofan engine is primarily dependent on bypass nozzle jet velocity for a given flight condition. High propulsive efficiency can be achieved by low fan pressure ratio which requires a large fan diameter for a given thrust demand. Therefore, the fan rotational speed has to be reduced to keep the fan tip speed below the supersonic level. The final outcome of applying these design criteria is a high bypass ratio turbofan engine with low thrust-specific-fuel-consumption (TSFC) and lower specific thrust. Along with the low bypass jet velocity comes low jet noise and because of the correspondingly slow fan speed, the fan emitted sound pressure level and therefore the noise level is low for the geared turbofan (GTF) configuration.

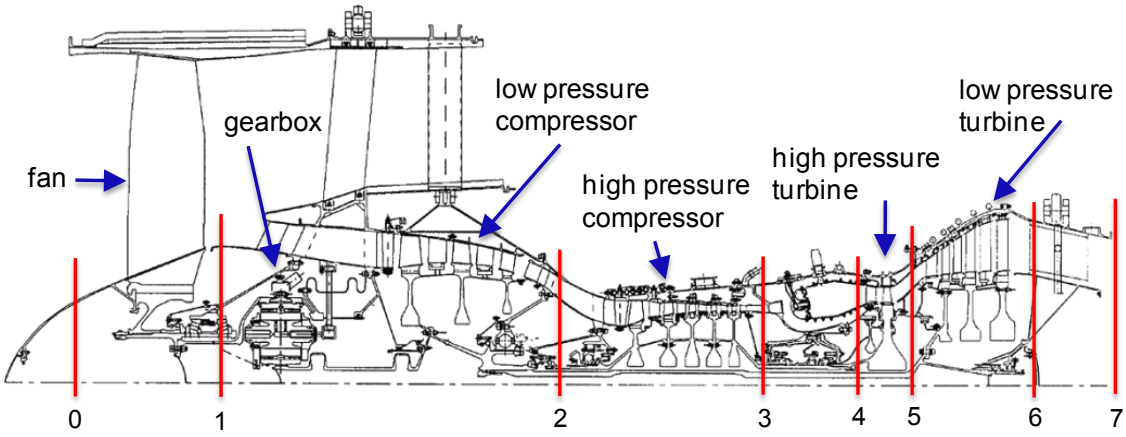


Fig. 7.13: Geared turbofan engine schematic.

The geared turbofan (GTF) engine boasts of significant performance benefits involving reduced fuel burn, engine noise and emissions. The associated changes in the engine architecture resulted in increased structural complexity [Sinha and de

Weck 2012] with increasing criticality of supporting components like lubrication systems. Here we look the dynamic complexity aspect of these three different jet engine architectures and expect a larger dynamic complexity for the geared turbofan configuration. We perform the comparative analysis over an operational envelope or mission profile for a typical jet aircraft [Roskam, 1999]. The operational envelope was chosen based on various environmental conditions under which the aircraft will have to perform. They included operating under extreme environmental conditions (e.g., operating in a desert climate, takeoff at very high altitude, etc) as well as inlet temperature and pressure distortions. The mission profile was based on a typical aircraft and jet engines with different configurations were used to accomplish them. The profile included takeoff and a 2500 ft/min climb to 35000 ft as well as a 2500 ft increase in altitude during cruise. The system responses considered were a combination of overall system responses like TSFC, specific thrust, engine mass-flow, NOx severity index in addition to internal thermodynamic responses like pressure/temperature at each of the important stations along the gas-path. The results are tabulated in table 7.6. All simulations were carried out using GasTurb™ [Kurzke 1995] and based on the default engine examples provided therein. To make the simulations consistent, we set the design points of all engines to a Mach number of 0.82 and altitude of 36250 ft [Roskam 1999].

Table 7.6: Dynamic Complexity of different type of aircraft engines over an operational envelope

Engine Architecture	**Dynamic Complexity**
Turbojet	342.05
Dual-spool Turbofan	857.77
Geared Turbofan	1341.9

We observe that the geared turbofan came out to be the most complex in terms of dynamic complexity with pure turbojet being the simplest. If one looks deeper at the numbers, it appears that the dual-spool turbofan is about 2.5 times more dynamically complex than the pure turbojet and geared turbofan is about 1.6 times more dynamically complex than the low-bypass dual spool turbofans. The geared turbofan is about 4 times more complex than the pure turbojets. This

matches well with our intuitive feeling about dynamic complexities of these jet engine architectures. The geared turbofan boasts of impressive performance gain over current high-bypass dual spool turbofans [Riegler et al. 2007]. The geared turbofan engine has significant architectural differences compared to the conventional dual spool turbofan engine. The table 7.7 below shows the structural and dynamic complexities of the dual-spool and geared turbofan engines.

Table 7.7: Comparison of structural and dynamic complexities for the 2-spool and geared turbofan architectures.

	2-spool Turbofan	**Geared Turbofan**
Structural Complexity	351	499
Dynamic Complexity	857.77	1341.9

We can see that the structural complexity of the geared turbofan architectures has increased by 42 % over the older dual-spool architecture. For the same two architectures, we see a 56 % increase in dynamic complexity of geared turbofan over the dual-spool architecture. The vastly improved performance gains in case of the geared turbofan architecture does not come for free [Sinha and de Weck 2012]. The improved performance has led to higher complexity, both structural and dynamic. This result also corroborates the assertion that the structural and dynamic complexities are positively correlated.

Chapter Summary

In this chapter, we formulated a dynamic complexity metric for engineered systems. Dynamic complexity refers to the complexity of the dynamical behavior of the system and stems from a combination of complexities due to system behavior and system structure. The dynamic complexities of individual system responses were based on their information content measured using Shannon entropy. The pair-wise interaction complexities between system responses were measured based on their mutual information. Lower mutual information signifies increased dynamic complexity of the pair-wise interaction.

We introduced the notion of *matrix energy* as a measure of topological complexity of system behavioral architecture. Topological complexity metric also shares properties similar to and found to correlate strongly with information-theoretic complexity metrics for networks.

We demonstrated operationalization and validation of the proposed methodology using examples of simple dynamical systems and to more complicated examples of large jet engines with different architectures. Here we concentrate on the dynamic complexity arising due to interactions among the behavioral responses. These examples served as test bench for intuitive validation of the proposed dynamic complexity. We demonstrated that the proposed dynamic complexity measure matches with our intuitive view of dynamic complexity for these systems. The geared turbofan architecture is shown to be about 56% more dynamically complex than the current high-bypass turbofans.

Although the geared turbofan boasts of impressive performance gain over current high-bypass dual spool turbofans, it does not come for free. The geared turbofan engine has significant architectural differences compared to the conventional dual spool turbofan engine and that has higher complexity, both structural and dynamic.

In canonical dynamical systems, we observed three fold or higher increase in dynamic complexity as the systems turned chaotic. Within the double pendulum type systems in non-chaotic regime, the dynamic complexity is found to increase steadily from simple double pendulum to inverted double pendulum on cart. We can use the same methodology in a transient scenario where the dynamic complexity becomes a function of time.

As conjectured from observations, increase in dynamic complexity correlate with structural complexity quite strongly in case aircraft engines. We can view the system behavior as a physical process. This physical process runs on top of an infrastructure that is the underlying system architecture, enabling the process. The structural complexity is related to this underlying infrastructure while dynamic complexity is the complexity of the physical process. In general, a more complex physical process would require a more complex infrastructure to run on and

therefore one would expect the dynamic complexity (or process complexity) to strongly correlate with structural complexity or the complexity of the infrastructure.

Going forward, we anticipate the proposed dynamic complexity metric to help explore important questions related to growth of complexity and its impact on engineered complex systems.

References

Bogdanov A., "Optimal Control of a Double Inverted Pendulum on a Cart", Technical Report CSE-04-006, 2004, Department of Computer Science & Electrical Engineering, OGI School of Science & Engineering, OHSU.

Strogatz S.H., "Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering", ISBN-10: 0738204536, Westview Press, 2001.

Botev Z. I., Grotowski J. F. and Kroese D. P., "Kernel Density Estimation via Diffusion", The Annals of Statistics, Vol. 38, No. 5, pp 2916 – 2957, 2010.

Cover T.M and Thomas J.A.: "Elements of Information Theory", Wiley Interscience; 2nd edition, 2006.

Jacek Marczyk (2008): "Complexity Management: New Perspective and Challenges for CAE in the 21st Century", www.oensys.com/articles

Kurzke J., "Advanced User-Friendly Gas Turbine Performance Calculations on a Personal Computer", ASME 95-GT-147, 1995.

Mattingly, J. D., "Elements of Gas Turbine Propulsion", ISBN 0-07-912196-9, McGraw Hill, 1998.

Raymer, D. P., "Aircraft Design: a Conceptual Approach, 3rd Edition". AIAA Education Series. New York, NY, 1999.

Recht et al., "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization", SIAM Rev. 52, pp. 471-501, 2010.

Riegler, C., and Bichlmaier, C., "The Geared Turbofan Technology – Opportunities, Challenges, and Readiness Status," 1st CEAS European Air and Space Conference, 10-13 September 2007, Berlin, Germany.

Roja N., "Numerical Analysis of the Dynamics of a Double Pendulum", project report, 2009.

Roskam J., "Airplane Design, part II", ISBN 978-1884885433, Design Analysis & Research, 1999.

Sheard S.A and Mostashari A., "A Complexity Typology for Systems Engineering", INCOSE International Symposium 2010.

Sinha K., de Weck O., "Structural Complexity Metric for Engineered Complex Systems and its Application", *14th International DSM Conference*, 2012.

Taylor J. R., "Classical Mechanics", ISBN 978-1891389221, 2005.

Willcox K., Allaire D., Deyst J., He C., and Sondecker G., "Stochastic Process Decision Methods for Complex-Cyber-Physical Systems", Final Report, DARPA META 2011.

Xin X., "Analysis of the Energy Based Swing-up Control for a Double Pendulum on a Cart", Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

Fan J., Liu H., "Statistical analysis of big data on pharmacogenomics", Advanced Drug Delivery Reviews 65 (2013) 987–1000.

Kolar M., Liu H., Xing E., "Graph Estimation From Multi-attribute Data", Preprint, arXiv:1210.7665, v2, 2013.

Feizi S., Marbach D., Médard M., and Kellis M., "Network deconvolution as a general method to distinguish direct dependencies in networks", Nature Biotechnology, July 2013.

Schlick C.M., Duckwitz S., Schneider S., "Project dynamics and emergent complexity", Computational and Mathematical Organization Theory, Springer, July 2012.

Frizelle G. and Woodcock E., "Measuring complexity as an aid to developing operational strategy", International Journal of Operations & Production Management, Vol. 15, Issue 5, 1995.

Gassmann F., "Noise-Induced Chaos-Order Transitions", Physical Review E **55**, Number 3, 1997.

Chapter 8

Conclusions and Future Work

Today's large-scale engineered systems are becoming increasingly complex due to numerous reasons including increasing demands on performance, and improved lifecycle properties. As a consequence, large product development projects are becoming increasingly challenging and are falling behind in terms of schedule and cost performance. A complex engineered system is difficult to describe and predict effectively. Keeping complexity under control is paramount as overly complex systems carry a variety of costs and risks.

Over the last century, increasingly complex machines and infrastructures have provided new capabilities that were previously unimaginable. While complexity can be costly, a higher complexity system may very well be worth the price of this additional complexity only if the performance gains outweigh the negatives. A natural tradeoff therefore exists between enabling valuable functionality or performance characteristics and keeping complexity under control.

To enable this tradeoff as part of system architecting, we ought to have a rigorous and computable measure of complexity. In the context of a complex engineered system design and development, we can categorize the associated complexities into (i) *internal* and (ii) *external*. There are three main dimensions of *internal* complexity that emerged in the **context of system design and development** - (1) **Structural Complexity**; (2) **Dynamic Complexity** and (3) **Organizational Complexity**.

In general, structural complexity strongly correlates with organizational complexity [Conway 1968; MacCormack et. al, 2011] and dynamic complexity. In

this thesis, the focus was on *structural complexity quantification* with a preliminary development of *dynamic complexity metric*.

Structural complexity is characteristic of system architecture and is dependent on the physical design of the system. Structural complexity relates to the notion of the architecture of a system, which is a skeleton that connects the components of the system and represents complexity induced by the *form*.

Dynamic complexity refers to the behavioral or functional complexity of the system. A system is deemed dynamically complex if its external behavior/dynamics is difficult to describe and predict effectively. We can view the system behavior as a physical process that runs over an underlying infrastructure. This infrastructure is nothing but the system architecture.

While structural complexity captures the complexity of the underlying infrastructure, the dynamic complexity captures its behavioral or process complexity. The system behavior (or the physical process) is bounded by the underlying system architecture (or the underlying infrastructure) and therefore, dynamic complexity possesses a strong positive correlation with structural complexity.

We have not explicitly dealt the organizational complexity part, but using the network representation, we can at least estimate its topological complexity directly. Please note that *external* sources of complexity like those related to the stakeholders, funding for the development effort, are not considered in this work. The contributions of this thesis are discussed in the following section.

8.1 Thesis Contributions

In this thesis, we have formulated a rigorous, quantitative structural complexity metric. In the process, a novel topological complexity metric capturing the interaction structure amongst system components has been proposed and shown to satisfy strict qualification criteria (i.e., Weyuker's criteria). This established a strong notion of construct validity. This thesis provides a methodological framework for structural complexity calculation, including data-

driven estimation of component and interface complexities, such that this metric is tractable throughout the system development endeavor. The topological complexity metric helped characterization of different architectural regimes. This helps open up the trade-space for system architecture choices involving combination of complex topology and simpler components vis-à-vis simpler topology with complex components.

An empirical validation of the proposed metric using simple experiments showed that the system development effort increased super-linearly with increase in structural complexity. It is possible to have sub-linear development cost vs. structural complexity relationship in theory if we have $b<1$, but is very unlikely to have such a relationship in practice. Such a scenario indicates that the system development entity's cognitive capability outstrips the enhanced complexity of the system. The exponent b reflects the perceived complexity or complicatedness. It is an observer dependent property unlike structural complexity, which is a system characteristic. Using the *cognitive demand ratio s*, it was observed that indeed the *cognitive demand* at lower complexity regime is much smaller. This indicated that at the lower regime of structural complexity, it might not be a prime driver of developmental cost/effort. It is possible to influence complicatedness by external means like advanced tools that aids cognitive capability and by introducing better complexity management techniques. This is where complexity meets cognitive science and systematic data collection on engineered system development in future will aid human cognitive model building exercise.

Structural complexity estimation was applied to multiple real-world engineered systems ranging from simple power screwdrivers to aircraft engines. It was observed that most engineered systems considered here lies in a transitional topological complexity regime with $1<C_3<2$. They all lie to the left of corresponding P points. The impact of system decomposition level on the topological complexity metric was investigated using a coarser (with 50 components) and finer (with 91 components) of the same digital printing system. The results showed near invariance (within numerical tolerance) of the topological complexity metric to system decomposition level for this example. On incorporating uncertainty in

component and interface complexity estimation, a hint of right-skewedness in structural complexity distribution was observed. This hints at existence of external uncertainty, but with small probability of occurrence. This is typical of engineered complex system development efforts and well supported by empirical data [Garvey, 2000].

The notion of *complexity budget* was introduced in a vein similar to more conventional notions of mass and power budgets for engineered systems. This brings together the trade space comprising of complexity, performance and development cost/effort. This work also contributes in the complexity - modularity debate by showing they are not necessarily negatively related and increased modularity can result in an increase of structural complexity and opens up the complexity-modularity trade-space.

This thesis also brings up the notion of *complexity gap* that appears when we attempt to represent a system at higher or coarse level of decomposition. This is similar to the established notion of *information gap/loss* in information theory when certain details are masked in order to abstract or simplify the system.

The notion of *complexity budget* is important as a practitioner's view as this is a methodical way to link complexity with observable quantities like performance and development cost/effort.

Finally a corresponding dynamic complexity metric is proposed for complex engineered systems that adapts the same functional form as the structural complexity metric with component and interface complexities replaced by the degree of uncertainty in specifying/estimating individual system responses and their pair-wise interactions respectively. Effect of system behavioral structure is captured by estimating the graph energy of the binary adjacency matrix in the system behavioral space with each system response as a node and their possible interaction as an edge. Application of the dynamic complexity metric to aircraft engines shows strong positive correlation between structural and dynamic complexities.

8.2 Challenges and Limitations

This thesis is an attempt at formalizing a rigorous and repeatable structural complexity quantification framework. There are still some challenges and associated limitations before this framework can be instituted with adequate fidelity in real-world, large engineered system development programs.

Since complexity cannot be observed physically using measuring devices unlike mass, power etc, we verified and validated the proposed structural complexity metric using construct validity based mathematical verification strategy and empirical validation respectively. We leveraged Weyuker's criteria for mathematical verification and used simple experiments and a set of real-world engineered systems for which data were available/gathered for empirical validation.

The first and probably the foremost challenge in about availability and obtaining data for a reasonably wide and heterogeneous spectrum of real-world, engineered systems for reasonably exhaustive empirical validation of theoretical predictions. This thesis applied the proposed methodology to a small set of real-word engineered systems and also augmented the available data with simple experiments for empirical validation purposes. A sustained effort in systems archeology and structured data collection will help address part of the current challenges that exist due to lack of validation data. Collection of historical data on a given category of engineered systems is required to build a high fidelity structural complexity vs. development cost model that can be used for quantitative prediction.

This will also address the issue of wider generalizability of the approach across a larger swath of engineered complex systems. We argue here that the basic super-linear nature of the relationship between development cost and structural complexity will hold. The degree of super-linearity (expressed by parameter, b) is expected to be only higher for increasing complex engineered systems, but with variation in model parameters {a, b} that depends on additional factors.

Another limitation is the lack of structured data collection and warehousing of system architectural data along with information at component and interface

level for improved and quick estimation of structural complexity. For example, development of PLM systems that maintains detailed system architectural information and specific data on components and interfaces for extraction at any point in time will aid in the complexity computation and tracking process troughout the project/program lifecycle. This will reduce our reliance on expert opinion for estimation of component and interface complexities. That being said, we must note that expert opinion remains a very important source of information that should ve leveraged for complexity estimation of novel components and interfaces, where enough data may not exist. We can treat expert opinion as out primary belief which is subsequently updated using available data (similar to a Bayesian updation) in order to build component complexity models for engineered systems. Please note that the proposed, bottom-up models for component and interface complexities are preliminary suggestions and might require augmentation, depending on domain specific features/requirements.

Similar data collection efforts are also required for operationalization of the complexity budgeing approach where the historical complexity – performance mapping for given category of systems can be leveraged to establish effective complexity budget assignment and tracking during system development efforts, similar to the traditional mass or power budgeting procedures.

8.3 Future Work

In order to address the challenges and limitations expressed above, we list a number of avenues for future research.

Expanding the ball and stick experiment – Repeating the ball and stick experiment with a larger and diverse pool of test subjects and an expanded set of molecular structures, while attempting to disaggregate the total build time into (i) physical assembly time; (ii) cognitive processing time; and (iii) rework. This could be done via posterior video motion analysis, preferably in conjuction with cognitive scientists, to understand the process of building on a finer level. A larger and diverse

pool of test subjects could be segmented to understand the effect of human profiles on complexity management capability.

Continued collection of real-world engineered systems for empirical validation – This is evident from the preceeding discussion. Such systems archeology efforts require extensive collaboration with industrial partners and help develop (i) data-driven methods for estimating component and interface complexity models; (ii) high fidelity quantitative guidance on development effort, given the structural complexity. Such data collection efforts could leapfrog the application of data analytics to engineering system development and might lead to statistically significant causal models of system developmemt efforts (using frameworks like Structural Equation Modeling) in future. In addition, it is predicted that the extent of system integration challenges increase significantly if topological complexity metric, $C_3>2$. This is an area requiring empirical data to validate this theoretical prediction for systems where $C_3>2$.

Effect of System Decomposition level on the structural complexity metric – Investigate the effect of system decomposition level on structural complexity metric by using representations of the same system at different levels. We have shown that the topological complexity remains invariant between two different levels of system decomposition for a digital printing system. It is of important to develop application specific guideline on the system decomposition level.

Structure-function interactions in networked systems – Exploration of trade-offs that exists between system performance measures vis-à-vis system lifecycle properties is of significant academic and practical significance. Does enhanced reliability come at the expense of performance? Can we enhance both by leveraging novel architectures? Does this improvement lead to added complexity or even that can be contained?

Knowledge of such interactions will help establish apriori the level of *essential* (or required) structural complexity of a system, given the set of required functions or the target performance level. It appears that system complexity tends

to enhance system performance, but the curve tapers down beyond a certain level of complexity and saturates thereafter. Any guideline on characterizing the level of this optimal complexity, beyond which performance gains saturate, has enormous practical implications for system architecting and design.

This is also expected to help exploit novel system structures to conceive systems that balance lifecycle properties (e.g., flexibility, complexity, reliability, etc) with system performance.

Impact of human cognition on **system complexity** – Explore the link between system complexity and cognitive requirements it imposes on system developers. The system complexity is an inherent property of the system, while the perceptive complexity or complicatedness is an observer dependent property and varies across individuals. It appears that some individuals who can '*see through*' architectural patterns can handle complexity growth by *managing* their response to increased system complexity. This has a direct organizational implication for resource allocation strategies, given the knowledge of system complexity. This area requires collaboration with cognitive scientists to glean out fundamental insights.

Complexity Management – In relation to development of complex engineered systems, quantification of complexity is not the end goal, but containment of system development cost, schedule and risks are. While complexity is an intrinsic system property, its manifestation in terms of system observables like development cost is dependent on an observer/actor dependent property called complicatedness or perceptive complexity of the human actors who brings the system into reality. Study of human or organizational complicatedness function is paramount in devising efficient ways and means for better complexity management.

Complexity-inclusive system optimization framework - A complexity-inclusive system design and optimization framework (Fig. 8.1 below) is an important next step. Architectures are selected from a set of discrete, feasible set of architectures during the optimization process. Within the generic framework, the optimal

architectural topology computation involves network optimization problem with binary (0/1) design variables.

Simulated Annealing (SA) based optimization strategy [de Weck, 2010] with handling of multiple objectives [Jilla and Miller, 2004] provides an exhaustive optimization strategy, but is not computationally efficient. Fundamentally, the optimization framework described above requires three primary ingredients to be successful as a comprehensive system architecting and design system, and they are: (i) a rule-based system compositional feasibility engine; (ii) a library of system components with their compositional logic (i.e., interfaces) described; and (iii) a higher level *performance/feasibility guarantee* mechanism by leveraging the *correct-by-construction* principles [Pinto *et al.*, 2010], where the performance or feasibility at lower or more detailed level can be guaranteed by a higher level, more abstracted analysis.

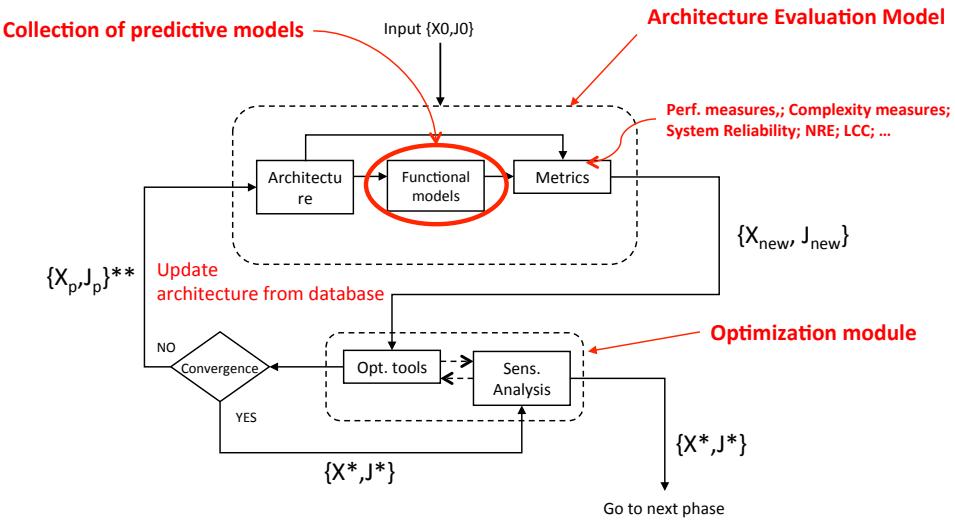


Fig. 8.1: A complexity-inclusive, system optimization framework.

This framework is likely to yield methods for network design under physical constraints, which can be applied in general to any networked system.

Agent-based System Architecting and Development - In system architecting, can we develop *agent* based modeling for developing complex engineered systems where system components serve as *agent*? The agents are embedded with

individual behavioral rules and interfaces between agents are defined by *interfacing rules*. It appears that this could become achievable in future with application of category theory [Giesa *et al.,* 2012] that provides the mathematical foundation. Architecting of networked complex systems in engineering can learn from similar applications in the area of biological systems, where system optimization played a critical role in architecting the system [Machado *et al*., 2011].

Extension to System-of-Systems – The system-of-systems can be represented as network of networks and structural complexity of such integrated systems can be computed using the same mathematical structure. The specific topic of importance is the techniques for estimating inter-system interfaces and bridge components that span connections across individual systems. What are the factors that drive the complexities of bridge components and the associated inter-system links?

Bibliography

Carlson J. M. and Doyle J. C., "Complexity and Robustness," PNAS, 2002.

Sussman, J. M., "Ideas on Complexity in Systems--Twenty Views", M.I.T., 2000.

Corning, P.A., "Complexity is Just a Word!" *Technological Forecasting and Social Change*, 58:1-4, 1998.

Conway, M. E., "How do Committees Invent?", *Datamation* **14** (5): 28–31, 1968.

Dan Sturtevant, "System Design and the Cost of Architectural Complexity", PhD Thesis, MIT, 2013.

http://www.darpa.mil/Our_Work/TTO/Programs/AVM/AVM_Design_Tools_(META).aspx, 2011.

Frey D.D., Palladino J., Sullivan J.P., and Atherton M., "Part Count and Design of Robust Systems," *Systems Engineering (INCOSE)* **10**(3), pp. 203- 221, 2007.

Lindemann, U.; Maurer, M.; Kreimeyer, M. (2005): Intelligent Strategies for Structuring Products. In: Clarkson, J.; Huhtala, M. (Eds.): Engineering Design – Theory and Practice. Cambridge, UK: Engineering Design Centre 2005, pp 106-115.

Lindemann, U, Maurer, M. and Braun, T. "Structural Complexity Management - An Approach for the Field of Product Design" – Springer, 2008.

MacCormack, A. D., Rusnak J, and Baldwin C.Y., "Exploring the Duality between Product and Organizational Architectures: A Test of the Mirroring Hypothesis." Harvard Business School Working Paper, No. 08–039, March 2008. (Revised October 2008, January 2011).

Malik, F., "Strategie des Managements komplexer Systeme". Bern: Haupt 2003.

Riedl, R., “Strukturen der Komplexität – Eine Morphologie des Erkennens und Erklärens”, Berlin: Springer 2000.

de Weck O., Roos D., Magee C., “Engineering Systems: Meeting Human Needs in a Complex Technological World”, MIT Press, 2011.

de Weck O.L., Ross A.M., Rhodes D.H., “Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)”, 3rd International Engineering Systems Symposium, Delft, 2012.

Crawley E., "Lecture notes for ESD.34 - System Architecture (2007), Massachusetts Institute of Technology," unpublished.

Baldwin, C.Y and Clark, K.B, “Design Rules”, MIT Press, 2000.

Browning, T. (2001): Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. IEEE Transactions on Engineering Management 48 (2001) 3, pp 292-306.

Jones, R., Hardin, P., and Irvine, A., “Simple Parametric Model for Estimating Development (RDT&E) Cost”, ISPA/SCEA Joint Conference, 2009.

Matti J Kinnunen, “Complexity Measures for System Architecture Models”, 2006 (MIT Thesis).

Sosa, M., “A Network Approach to Define Component Modularity”, Proceedings of the 7th International Dependency Structure Matrix (DSM) Conference, Seattle. Seattle, USA: Boeing 2005.

Sosa, M. E.; Eppinger, S. D.; Rowles, C. M., “Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions”, Journal of Mechanical Design 125 (2003) 2, pp 240-252.

Ulrich, K.; Eppinger, S., “Product Design and Development”, New York: McGraw-Hill, 1995.

Ulrich, H.; Probst, G., “Anleitung zum ganzheitlichen Denken und Handeln – Ein Brevier für Führungskräfte”, Bern: Paul Haupt, 2001.

Weber, C., “What Is “Complexity”?”, Proceedings of the 15th International Conference on Engineering Design (ICED 05), Melbourne. Melbourne: Institution of Engineers, 2005.

Weyuker, E., “Evaluating software complexity measures”, IEEE Transactions on Software Engineering, 1988, 14 (9), 1357-1365.

Whitney D. E., “Physical limits to modularity”, Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division, 2004.

Yassine, A.; Falkenburg, D.; Chelst, K., “Engineering Design Management: An Information Structure Approach”, International Journal of Production Research 37 (1999), pp 2957– 2975.

Rechtin, E. and Maier, M.W.,”The art of systems architecting*”*, CRC, 2002.

Simmons, W.; Koo, B.; and Crawley, E.,“Architecture generation for moon-mars exploraton using an executable meta-language”, volume AIAA-2005-6726, American Institute of Aeronautics and Astronautics, 2005.

Muller, G., “System Architecting*”*, ESI, 2005.

Meyer, M and Lehnerd, A., “The Power of Product Platforms”, Free Press, 1997.

Lankford, J., ”Measuring system and software architecture complexity”, Aerospace Conference, Proceedings, March 2003.

Suh, N. P., “Complexity: theory and applications”, Oxford University Press, 2005.

Hughes, T. P., “Rescuing Prometheus”, Pantheon Books, 1998.

McCabe T.J. and Butler C. W., “Design complexity measurement and testing”, Commun. ACM, 32(12): pp 1415-1425, 1989.

de Weck O.L., Roos D., and Magee C.L., “*Engineering systems: Meeting human needs in a complex technological world*”, The MIT Press, 2011.

Smith M.R., “*Military enterprise and technological change: Perspectives on the American experience*”, The MIT Press, 1985.

Fagen M.D., Joel A.E., and Schindler G., “*A history of engineering and science in the Bell system*”, New York: Bell Telephone Laboratories, 1975.

Crawley E., "Lecture notes for ESD.34 - System Architecture (2007), Massachusetts Institute of Technology," unpublished.

Baldwin C.Y. and Clark K.B., “*Design rules: The power of modularity, Volume 1”,* The MIT Press, 1999.

Sussman J.M., "The new transportation faculty: The evolution to engineering systems," 1999.

Leveson N., "A new accident model for engineering safer systems," *Safety Science,* vol. 42, pp. 237- 270, 2004.

Whitney D., Crawley E., de Weck O.L., Eppinger S., Magee C.L., Moses J., Seering W., Schindall J., and Wallace D., "The influence of architecture in engineering systems," *Engineering Systems Monograph,* 2004.

Mahadevan P., Krioukov D., Fomenkov M., Dimitropoulos X., and Vahdat A., "The Internet AS-level topology: Three data sources and one definitive metric," *ACM SIGCOMM Computer Communication Review,* vol. 36, pp. 17-26, 2006.

Tang, V. and Salminen, V., “Towards a Theory of Complicatedness: Framework for Complex Systems Analysis and Design”, 13th International Conference on Engineering Design, Glasgow, Scotland, August 2001.

Bashir, H.A. and Thomson, V., “Estimating design complexity”, Journal of Engineering Design Vol. 10 No 3, pp 247–257, 1999.

Hirschi N.W, and Frey D.D., “Cognition and complexity: An experiment on the effect of coupling in parameter design”, Research in Engineering Design 13, 2002, pp 123 – 131.

Ramasesh R.V. and Browning T.R., “Toward a theory of unknown unknowns in project management”, preprint, 2012.

Bernstein D. S., “Matrix Mathematics: Theory, Facts, and Formulas“, Princeton University Press, Second Edition, ISBN: 9781400833344, 2009.

Chung, F. and Lu, L., “complex graphs and networks”, CBMS volume 7, 2004.

Chung, F. “Spectral Graph Theory”, American Mathematical Society, 1st edition, 1997.

Cvetkovic´ D., Doob M., Sachs H., "*Spectra of Graphs – Theory and Application",* Academic, New York, 1980.

Fabrikant A., Koutsoupias E., Papadimitriou C., "Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet", *Automata, Languages and Programming*, Vol. 2380, January 2002.

Holtta, K., Suh, E.S. and de Weck, O. (2005). "Tradeoff Between Modularity and Performance for Engineered Systems and Products", ICED05, Melbourne, Australia.

Gutman, I., "The energy of a graph", Ber. Math.-Stat. Sekt. Forschungszent. Graz 103 (1978), 1–22.

Gutman, I; Soldatovic, T. and Vidovic, D., "The energy of a graph and its size dependence. A Monte Carlo approach", Chemical Physics Letters 297 (1998), 428–432.

Koolen, J. and Moulton, V., "Maximal energy graphs" Advances in Appld. Math., 26 (2001), 47–52.

Koolen, J., Moulton, V. and Gutman, I., "Improving the McClelland inequality for total p-electron energy" Chemical Physics Letters 320, pp 213–216, 2000.

Kharaghani H. and Tayfeh-Rezaie B., "On the energy of (0; 1)-matrices" Lin. Algebra Appl. 429, 2046-2051, 2008.

Horn and Johnson, "Topics in Matrix Analysis", Cambridge Press, 1994.

Klau G.W. and Weiskircher R., " Robustness and Resilience", chapter in "Network Analysis: Methodological Foundations", Ulrik Brandes and Thomas Erlebach (eds.), Lecture Notes in Computer Science, Vol. 3418, Springer, Berlin, 2005, pp.417-437.

Li X., Shi Y., Gutman I., "Graph Energy", Springer (2012), ISBN: 978-1-4614-4219-6.

Liu, D., Wang, H., and Van Mieghem, P.,"Spectral perturbation and reconstructability of complex networks", Physical Review E 81, 016101, 2010.

Najjar W. and Gaudiot J.L., "Network Resilience: **A** Measure of Network Fault Tolerance", IEEE transactions on computers, vol. 39. no. 2., pp 174-181, 1990.

Nikiforov V., "The energy of graphs and matrices", J. Math. Anal. Appl. 326 (2007), 1472–1475.

Newman, M. E. J., Barabasi, A. L., and Watts, D. J., "The Structure and Dynamics of Networks" Princeton University Press, Princeton (2003).

Pishkunov N., "Differential and Integral Calculus", CBS Publishers & Distributors, 1996, ISBN: 8123904924.

Wagner S., "Energy bounds for graphs with fixed cyclomatic number", MATCH Commun. Math. Comput. Chem. **68**, 661–674 (2012).

Liu, D., Wang, H., and Van Mieghem, P.,"Spectral perturbation and reconstructability of complex networks", Physical Review E 81, 016101, 2010.

Van Mieghem P., "Graph Spectra for Complex Networks", Cambridge University Press, ISBN 978-0-521-19458-7, 2011.

Kinsner W.,"System Complexity and Its Measures: How Complex Is Complex", Advances in Cognitive Informatics and Cognitive Computing Studies in Computational Intelligence, Vol 323, pp 265-295, 2010.

Candès E.J. and T. Tao. "The power of convex relaxation: Near-optimal matrix completion", *IEEE Trans. Inform. Theory*, 56(5), 2053-2080, 2009.

Candes E.J. and Recht B., "Exact Matrix Completion via Convex Optimization", Found. of Comput. Math., 2717-772, 2009.

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Kotz S., van Dorp J.R., "Beyond Beta: Other Continuous Families Of Distributions With Bounded Support And Applications", World Scientific, 2004, ISBN-10: 9812561153.

Babuscia A., "Statistical Risk Estimation for Communication System Design", PhD thesis, MIT, 2012.

Cooke R.M., "Experts in Uncertainty: Opinion and Subjective Probability in Science", Oxford University Press, 1991.

Goulet V., Jacques M. and Pigeon M., "Modeling without data using expert opinion", The R Journal, 1:31–36, 2009.

O'Hagan A., "Eliciting expert beliefs in substantial practical applications", Journal of the Royal Statistics Society, 47:21–35, 1998.

O'Hagan A., Buck C., Daneshkhah A., Eiser R. and Garthwaite P., "Uncertain Judgements: Eliciting Experts Probabilities", Lavoisier Libraire, 2006.

Bearden D.A., "Evolution of complexity and cost for Planetary Missions throughout the development lifecycle", IEEE Aerospace Conference, 2012.

Bearden D.A., "A complexity-based risk assessment of low-cost planetary missions: when is a mission too fast and too cheap? ", 4th IAA international conference on low-cost planetary missions, May 2-5, 2000.

Bearden D.A., "Complexity based cost estimating relationships for space systems", IEEE Aerospace Conference, Volume 6, 2004.

Greenberg M., "A Step-Wise Approach to Elicit Triangular Distributions", ICEAA Professional Development & Training Workshop June 18-21, 2013.

Young, D. C., and Young P.H., "A Generalized Probability Distribution for Cost/Schedule Uncertainty in Risk Assessment", Proceedings of the Society for Computer Simulation, 1995.

Clemen R.T and Winkler R.L., "Combining probability distributions from experts in risk analysis", Society for Risk Analysis, 19:187–203, 1999.

Martinez W.L. and Martinez A.R., "Computational Statistics Handbook with MATLAB", 2nd edition, CRC Press (2007), ISBN-10: 1584885661.

Mosteller F. and Tukey J.W., "Data Analysis and Regression", Addison-Wesley Publishing Company, Don Mills (1977).

DARPA Adaptive Vehicle Make (AVM),"http://www.darpa.mil/Our_Work/TTO/ Programs/Adaptive _Vehicle_Make__(AVM).aspx", 2011.

Ahn J., de Weck O.L., Steele M., "Credibility Assessment of Models and Simulations Based on NASA's Models and Simulation Standard Using the Delphi Method", Systems Engineering, 2013.

Eppinger S.D., Browning T.R., "Design Structure Matrix Methods and Applications", MIT Ptress, 2012, ISBN-10: 0262017520.

Steward D., "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management 28 (3), pp 71-74, 1981.

Sinha K., de Weck O., "Structural Complexity Metric for Engineered Complex Systems and its Application", *14th International DSM Conference*, 2012.

Denman J., Sinha K., de Weck O., "Technology Insertion in Turbofan Engine and assessment of Architectural Complexity", *13th International DSM Conference*, Cambridge, MA, September 14-15, 2011.
Sinha, K., & de Weck, O. (2009). Spectral and Topological Features of 'Real-World' Product Structures.

Proceedings of 11th International DSM Conference, 2009.

Sinha K., James, D. and de Weck O.L, "Interplay between Product Architecture and Organizational Structure", 14th International Dependency and Structure Modelling Conference, 2012.

Sinha K., de Weck O., "A network-based structural complexity metric for engineered complex systems", IEEE Systems Conference (SysCon), pp 426-430, 2013.

Sinha K., de Weck O., "Structural Complexity quantification for engineered complex systems and implications on system architecture and design", ASME International Design Engineering Technical Conference, 2013.

Sinha K., Omer H., de Weck O., "Structural Complexity: Quantification, Validation and its systemic implications for engineered complex systems", International Conference on Engineering Design, 2013.

Jiang X., Yao Y., Liu H., Guibas L., "Compressive Network Analysis", IEEE Transactions on Automatic Control. To appear. 2013.

Tilstra A.H., "Representing Product Architecture and Analyzing Evolvable Design Characteristics", PhD thesis, University of Texas, 2010.

Tilstra A.H., Backlund P., Seepersad C.C and Wood K., "Principles for Designing Products with Flexibility for Future Evolution", DETC2006-99583, 2013.

Suh E.S., Furst M.R., Mihalyov K.J. de Weck O.L., "Technology infusion for complex systems: A framework and case study", Systems Engineering, 13: 186–203, 2010.

Springer M.D., "The Algebra of Random Variables", John Wiley & Sons, 1979, ISBN-10: 0471014060.

Giesa T., Spivak D., Buehler M.J., "Category theory based solution for the building block replacement problem in materials design," *Advanced Engineering Materials*, 2012.

Machado D., Cost R.S., Rocha M., Ferreira E.C., Tidor B., Rocha I., "Modeling formalisms in systems biology", A.M.B. Express 45, 2011.

de Weck O.L., "Simulated Annealing: A Basic Introduction", Course Notes 16.888/ ESD 77, Massachusetts Institute of Technology, 2010.

Jilla C.D., Miller D.W., "A Multi-Objective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems", Journal of Spacecraft and Rockets, Vol. 41, No. 1, 2004.

Pinto A., Becz S. and Reeve H.M., "Correct-by-Construction Design of Aircraft Electric Power Systems", AIAA ATIO/ISSMO Conference, 2010.

Bashir, H.A. and Thomson, V., "Estimating design complexity", Journal of Engineering Design Vol. 10 No 3 (1999), pp 247–257.

Bashir, H.A. and Thomson, V., "Models for estimating design effort and time", Design Studies 22 (2001), pp 141–155.

Bettencourt L.M.A., Lobo J. and Strumsky D., "Invention in the city: Increasing returns to patenting as a scaling function of metropolitan size", Research Policy, Volume 36, Issue 1, 2007, pp 107–120.

Bettencourt L.M.A and West G.," A Unified Theory of Urban Living", *Nature* **467**, 2010, pp 912–913.

Conte S.D., Dunsmore H.E. and Shen V.Y., "Software Engineering Metrics and Models", The Benjamin /Cummings Pub- lishing Company, Menlo Park (1986).

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Hirschi N.W, and Frey D.D., "Cognition and complexity: An experiment on the effect of coupling in parameter design", Research in Engineering Design 13, 2002, pp 123 – 131.

Maier M.W. and Rechtin E., "The Art of Systems Architecting", CRC Press; 3rd edition (2009), ISBN-10: 1420079131.

Martinez W.L. and Martinez A.R., "Computational Statistics Handbook with MATLAB", 2nd edition, CRC Press (2007), ISBN-10: 1584885661.

Mosteller F. and Tukey J.W., "Data Analysis and Regression", Addison-Wesley Publishing Company, Don Mills (1977).

Norden P.V., "Manpower utilization patterns in research and development projects", Ph.D Dissertation, Columbia University, 1964.

Prentice Hall Molecular toolkit, "Prentice Hall Molecular Model Set for General and Organic Chemistry". 1997.

Putnam L.H., "General empirical solution to the macro software sizing and estimating problem", IEEE transactions on Software Engineering, Vol SE4, No 4 (1978), pp 345-361.

Sheard S.A. and Mostashari A., "A Complexity Typology for Systems Engineering", Systems Engineering, 2010.

Wertz J.R. and Larson W.J., "Reducing Space Mission Cost", Springer (1996), ISBN-10: 1881883051

Wood et al, "Interfaces and Product Architecture", ASME DETC, 2001.

Danilovic, M., & Browning, T. R. (2007), "Managing complex product development projects with design structure matrices and domain mapping matrices",International Journal of Project Management, 25(3), 300-314.

Eun Suk Suh, Michael R. Furst, Kenneth J. Mihalyov and Olivier L. de Weck "DSM Model Building Instructions" – working paper, 2009.

Suh E.S., Furst M.R., Mihalyov K.J. de Weck O.L., "Technology infusion for complex systems: A framework and case study", Systems Engineering, 13: 186–203, 2010.

Riegler, C., and Bichlmaier, C., "The Geared Turbofan Technology – Opportunities, Challenges, and Readiness Status," 1st CEAS European Air and Space Conference, 10-13 September 2007, Berlin, Germany.

Malzacher, F.J., Gier, J., and Lippl, F., "Aerodesign and Testing of an Aeromechanically Highly Loaded LP Turbine", Journal of Turbomachinery, 2006, Vol. 128, No. 4, pp 643-649.

Ulrich K. T., "The role of product architecture in the manufacturing firm", Research Policy, 24, 1995, pp. 419-441.

Garvey P.R., "Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective", CRC Press (2000), ISBN-10: 0824789660.

Tilstra A.H., "Representing Product Architecture and Analyzing Evolvable Design Characteristics", PhD Thesis, University of Texus, Austin, 2010.

Ranga V. Ramasesh and Tyson R. Browning, "TOWARD A THEORY OF UNKNOWN UNKNOWNS IN PROJECT MANAGEMENT", preprint, 2012.

Heimsch D., "http://www.scribd.com/doc/59779160/Boeing-Dream-Liner-Outsourcing-Study"

Cohan P.S., "http://www.dailyfinance.com/2011/01/21/boeing-dreamliner-delays-outsourcing-goes-too-far/"

Hiltzik M., "http://articles.latimes.com/2011/feb/15/business/la-fi-hiltzik-20110215"

Goldsmith S., Eggers W.D., "Governing by Network: The New Shape of the Public Sector", Brookings Institution Press, ISBN-10: 0815731299, 2004.

Weber J., "http://www.businessweek.com/bwdaily/dnflash/content/jan2009/db20090116_971202.htm"

Cohan P.S., "You Can't Order Change: Lessons from Jim McNerney's Turnaround at Boeing", Portfolio Hardcover, ISBN-10: 1591842395, 2008.

Hirschi N.W, and Frey D.D., "Cognition and complexity: An experiment on the effect of coupling in parameter design", Research in Engineering Design 13, pp 123 – 131, 2002.

Newman M.E.J., "Networks: An Introduction", Oxford University Press, ISBN-10: 0199206651, 2010.

Maier, M.W., "Architecting Principles for System of Systems". *Systems Engineering* **1** (4): 267–284, 1998.

Pasqual M.C., de Weck O.L.,"Multilayer Network Model for Analysis and Management of Change Propagation", *Research in Engineering Design*, Special Issue on Engineering Change, June 2012.

Bogdanov A., "Optimal Control of a Double Inverted Pendulum on a Cart", Technical Report CSE-04-006, 2004, Department of Computer Science & Electrical Engineering, OGI School of Science & Engineering, OHSU.

Botev Z. I., Grotowski J. F. and Kroese D. P., "Kernel Density Estimation via Diffusion", The Annals of Statistics, Vol. 38, No. 5, pp 2916 – 2957, 2010.

Cover T.M and Thomas J.A.: "Elements of Information Theory", Wiley Interscience; 2nd edition, 2006.

Jacek Marczyk (2008): "Complexity Management: New Perspective and Challenges for CAE in the 21st Century", www.oensys.com/articles

Kurzke J., "Advanced User-Friendly Gas Turbine Performance Calculations on a Personal Computer", ASME 95-GT-147, 1995.

Mattingly, J. D., "Elements of Gas Turbine Propulsion", ISBN 0-07-912196-9, McGraw Hill, 1998.

Raymer, D. P., "Aircraft Design: a Conceptual Approach, 3rd Edition". AIAA Education Series. New York, NY, 1999.

Recht et al., "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization", SIAM Rev. 52, pp. 471-501, 2010.

Riegler, C., and Bichlmaier, C., "The Geared Turbofan Technology – Opportunities, Challenges, and Readiness Status," 1st CEAS European Air and Space Conference, 10-13 September 2007, Berlin, Germany.

Roja N., "Numerical Analysis of the Dynamics of a Double Pendulum", project report, 2009.

Roskam J., "Airplane Design, part II", ISBN 978-1884885433, Design Analysis & Research, 1999.

Sheard S.A and Mostashari A., "A Complexity Typology for Systems Engineering", INCOSE International Symposium 2010.

Taylor J. R., "Classical Mechanics", ISBN 978-1891389221, 2005.

Willcox K., Allaire D., Deyst J., He C., and Sondecker G., "Stochastic Process Decision Methods for Complex-Cyber-Physical Systems", Final Report, DARPA META 2011.

Xin X., "Analysis of the Energy Based Swing-up Control for a Double Pendulum on a Cart", Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

Fan J., Liu H., "Statistical analysis of big data on pharmacogenomics", Advanced Drug Delivery Reviews 65 (2013) 987–1000.

Kolar M., Liu H., Xing E., "Graph Estimation From Multi-attribute Data", Preprint, arXiv:1210.7665, v2, 2013.

Feizi S., Marbach D., Médard M., and Kellis M., "Network deconvolution as a general method to distinguish direct dependencies in networks", Nature Biotechnology, July 2013.

Schlick C.M., Duckwitz S., Schneider S., "Project dynamics and emergent complexity", Computational and Mathematical Organization Theory, Springer, July 2012.

Kerschen G. and Golinval J.C., "Physical Interpretation of the Proper Orthogonal Modes using the Singular Value Decomposition", Journal of Sound and vibration 249(5), pp 849-865, 2002.

Wang H., Li Q., D'Agostino G., Havlin S., Stanley H.E. and Mieghem P.V., "Effect of the Interconnected Network Structure on the Epidemic Threshold", Physical Review E, Vol. 88, No. 2, 2013.

Jooyandeh M., Kiania D. and Mirzakhaha M., "Incidence Energy of a Graph", MATCH Commun. Math. Comput. Chem. 62, pp 561-572, 2009.

Chen M., "Mixing Time of Random Walks on Graphs", *Master of Science Thesis*, Department of Mathematics, University of York, 2004.

Butler, S., "Eigenvalues and Structures of Graphs", *PhD Thesis*, Department of Mathematics, University of California, San Diego, 2008.

Estrada E., Hatano N., Benzi M., "The physics of communicability in complex networks", Physics Report, Vol 514, Issue 3, pp 89‐119, 2012.

Monthus C. and Garel T., "Random elastic networks : strong disorder renormalization approach", *J. Phys. A: Math. Theor.*, Vol 44, Issue 8, 2011.

Zhao Y., "Why 787 Slips Were Inevitable?", Working Paper, http://zhao.rutgers.edu/787-paper-12-02-2013.pdf, 2012.

Xu, X. and Zhao Y., "Incentives and Coordination in Project-Driven Supply Chains", Working Paper, http://zhao.rutgers.edu/Xu-Zhao-11-23-2013.pdf, 2012.

MATLAB Release 2012a. Natick, Massachusetts: The MathWorks Inc., 2012.

Wolfram Research, Inc., Mathematica, Version 8.0, Champaign, IL, 2010.

Appendix

A: Origin of the functional form of Structural Complexity Metric

The earliest origin of graph energy can be traced back to the Huckel's molecular orbital theory (HMO theory) in 1940's, where it formed a part of the approximate solution of Schrodinger equation for a class of conjugate hydrocarbons. The graph energy relates to the energy of the Hamiltonian system representing the organic molecule [Li *et al.*, 2012].

The time-independent Schrodinger equation is a second-order partial differential equation of the form:

$$H\psi = \varepsilon\psi \qquad (A.1)$$

where ψ is the wave function of the system considered, H is the Hamiltonian operator of the system considered, and ε is the energy of the system considered [Li *et al.*, 2012]. When applied to a particular molecule, the Schrodinger equation enables one to establish their energies and also describe the behavior of the electrons. For this, one needs to solve eq. A.1, which is an eigen-system problem of the Hamiltonian operator. To have a feasible solution, one has to express ψ as a linear combination of a finite number of basis functions. Then in Eq. A.1 the Hamiltonian becomes a matrix, which is termed as the system Hamiltonian matrix. In fig. A.1 below, the chemical formula of biphenylene—a typical conjugated hydrocarbon is shown. It contains n = 12 carbon atoms over which the n = 12 π-electrons form waves.

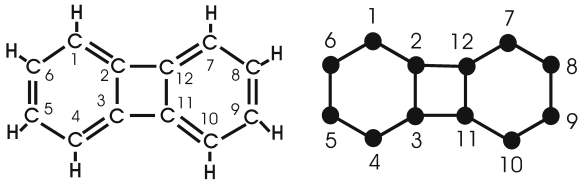


Figure A.1: Biphenylene is a typical conjugated hydrocarbon. Its carbon–atom skeleton is represented by the molecular graph *G*. The carbon atoms in the chemical formula H and the vertices of the graph G are labeled by 1, 2, … , 12.

In the HMO model, the wave functions of a conjugated hydrocarbon with *n* carbon atoms are expanded in an n-dimensional space of orthogonal basis functions, whereas the Hamiltonian matrix is a square matrix of order *n*, defined as:

$$[\mathbf{H}]_{ij} = \begin{cases} \alpha & \text{if } i = j \\ \beta & \text{if the atoms } i \text{ and } j \text{ are chemically bonded} \\ 0 & \text{if there is no chemical bond between the atoms } i \text{ and } j. \end{cases}$$

Assume that the parameters α and β are assumed to be constants, equal for all chemical atoms and chemical bonds respectively within this molecule. The system Hamiltonian matrix of biphenylene, based on HMO, can be written as:

$$\mathbf{H} = \alpha \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix} + \beta \begin{bmatrix} 0&1&0&0&0&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0&0&0&0&1 \\ 0&1&0&1&0&0&0&0&0&0&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0&1&0 \\ 0&0&1&0&0&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&1&0&0&0&1&0 \end{bmatrix}.$$

In the general case within the HMO model, one needs to solve the eigen system problem of an approximate Hamiltonian matrix of the form:

$$H = \alpha I_n + \beta A(G) \qquad (A.2)$$

where α and β are certain constants, $\mathbf{I}_n$ is the identity matrix of order *n*, and $\mathbf{A}(G)$ is the adjacency matrix of the graph G on *n* vertices corresponding to the carbon–atom skeleton of the underlying molecule.

From eq. A.2, the absolute energy levels ε_i of the π electrons are related to the singular values σ_i of the graph G (determined by the singular values of the binary adjacency matrix, A) by the simple relation:

$$\left|\varepsilon_i\right| = \alpha + \beta\sigma_i \qquad (A.3)$$

Using the HMO approximation, the form of total energy of all π electrons can be expressed as:

$$\varepsilon_\pi = \sum_{i=1}^{n} h_i \left|\varepsilon_i\right| \qquad (A.4)$$

where h_i acts as weights associated to each energy level and is constrained by the following relation amongst the weights:

$$h_1 + h_2 + \ldots + h_n = n \qquad (A.5)$$

since the number of π electrons in the molecules is n. Using Eq. A.5 and Eq. A.3 in Eq. A.4, we can write the total π electron energy content as:

$$\varepsilon_\pi = \sum_{i=1}^{n} h_i \left|\varepsilon_i\right| = n\alpha + \beta \sum_{i=1}^{n} h_i \sigma_i \qquad (A.6)$$

Using Cauchy-Schwarz inequality, we can write,

$$\sum_{i=1}^{n} h_i \sigma_i \le \left(\sum_{i=1}^{n} h_i \right) \left(\sum_{i=1}^{n} \sigma_i \right) \qquad (A.7)$$

Using inequality A.7 in A.6, we can write,

$$\varepsilon_\pi = n\alpha + \beta \sum_{i=1}^{n} h_i \sigma_i \le n\alpha + \beta \underbrace{\left(\sum_{i=1}^{n} h_i \right)}_{n} \underbrace{\left(\sum_{i=1}^{n} \sigma_i \right)}_{E(A)} \qquad (A.8)$$

where the sum of singular values of the binary adjacency matrix, $E(A) = \sum_{i=1}^{n} \sigma_i$ is defined as the *graph energy* or the *matrix energy* or the *nuclear norm*.

Using A.5 in A.8, we can write,

$$\varepsilon_\pi \le n\alpha + n^2 \beta \left(\frac{E(A)}{n} \right) \qquad (A.9)$$

Now let us look at the RHS of inequality A.9 carefully. We can think of the first term, $n\alpha$ as the sum of self-energy associated to each atom. Notice that the number of entries or the non-zero entries in the atomic interaction part of the Hamiltonian matrix scales as n^2 where n is the number carbon atoms in the molecule. This is the upper bound of the number of possible interactions.

Hence the $n^2\beta$ term can be thought of as the upper bound of the sum of interaction energy, where β is a representative interaction strength between atoms. The actual sum of interaction energy can be expressed as (#interactions)*(strength of interaction) and this quantity is bounded by $n^2\beta$. The remaining part of expression A.9 is related to the arrangement of the interactions, that is, the topology of interactions as manifested by the adjacency matrix. This term differentiates between connectivity patterns while other parts of expression A.9 remain identical, as shown below.

$E(A_1) < E(A_2)$

Figure A.2: Two architectures having the same number of nodes and connections but are differentiated based on their internal structure with $E(A_1)$ = 4.9 and $E(A_2)$ = 6.83.

We can therefore introduce a notion of *configuration energy*, Ξ expressed as,

$$\Xi := n\hat{\alpha} + m\hat{\beta}\left(\frac{E(A)}{n}\right) \qquad (A.10)$$

The above *configuration energy* expresses the innate ability of the interacting system to respond to the environment and a higher value indicates increasing difficulty to manage the system.

Any engineering system can be represented as networks where its components represent the nodes and pair-wise connections between components

represents the edges or interactions. The system architecture can be thought of as a connected network of system components.

The notional configuration energy described above can be thought of as the complexity associated to the system structure. The system behavior is not considered explicitly here but how the system components are connected amongst themselves. We term this quantity as the Structural Complexity of the system where $\alpha's$ stand for component complexity while $\beta's$ stand for interface complexity.

Assuming $\hat{\alpha}$ to be the average component complexity and $\hat{\beta}$ to be the average interface complexity, we can express the structural complexity, C as,

$$C = n\hat{\alpha} + m\hat{\beta}\left(\frac{E(A)}{n}\right) \qquad (A.11)$$

where *m* is the number of pair-wise interfaces in the system. The generic form of the Structural Complexity matric is written as,

$$\begin{aligned} C &= C_1 + C_2 C_3 \\ &= \sum_{i=1}^{n} \alpha_i + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}\right)\left(\frac{E(A)}{n}\right) \end{aligned} \qquad (A.12)$$

We can further express A.12 as,

$$C = \sum_{i=1}^{n} \alpha_i + \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}\right)\gamma E(A) \qquad (A.13)$$

where we have used $\gamma = \frac{1}{n}$ as the scaling factor. There are several other justifications for choosing this scaling factor and are discussed next.

B: Linking binary adjacency matrix with weighted adjacency matrix

The Structural Complexity metric, defined by expression A.12, effectively decouples the complexities originating from pair-wise interfaces (C_2) and those due only to the specific topological arrangement (i.e., existence/non-existence of connection) amongst system components (C_3). The local characteristics of particular

pair-wise interfaces are encoded within the C_2 term while the topological implications of the underlying connectivity structure amongst components, a global characteristic, is captured in term C_3.

An interesting aspect to investigate is the direct use of the sum of singular values of the weighted adjacency matrix, instead of the binary adjacency matrix and interface complexities in a decoupled form. Using the weighted adjacency matrix removes the decoupled feature in expression A.12 and brings in a combined metric instead of C_2 and C_3. In what follows, we will show the relationship between expression A.12 and one using the weighted adjacency matrix and analytically demonstrates that the decoupled version is a conservative estimate over the coupled version using the weighted adjacency matrix.

Let us define two square matrices $A, B \in \mathbb{R}^n$ that are non-negative (i.e., $[a_{ij}]$ and $[b_{ij}] \geq 0$). They define the binary and weighted adjacency matrices respectively. We can define these matrices as follows,

$$\left.\begin{aligned} a_{ij} &= \begin{cases} 1 \ \textit{iff i and j are connected} \\ 0 \ \textit{otherwise} \end{cases} \\ b_{ij} &= \begin{cases} > 0 \ \textit{iff} \ \ a_{ij} = 1 \\ 0 \ \textit{otherwise} \end{cases} \end{aligned}\right\} \qquad (B.1)$$

From the above definition we can see that the weighted adjacency matrix, B can also be the *Hadamard* product of matrices A and B. This is nothing but the *element-wise* multiplication of two matrices of same size, which is also known as the *Schur* product [Horn and Johnson, 1994]. We represent the *Hadamard* product as $A \circ B = [a_{ij} b_{ij}] \ \ \forall i, j$. From B.1, we can write,

$$B = A \circ B \qquad (B.2)$$

Using the *submultiplicative* property of matrix norms of *Hadamard* products [Horn and Johnson, 1994; Bernstein, 2009], we can write,

$$\sum_{i=1}^{n}\sigma_i(A\circ B)\le\sigma_1(B)\sum_{i=1}^{n}\sigma_i(A) \qquad (B.3)$$

where $\sigma_i(.)$ represents the i[th] singular value with $\sigma_1(.)$ being the largest singular value. Using the definition of nuclear norm, $E(.)=\sum_{i=1}^{n}\sigma_i(.)$, the above inequality can be written as,

$$E(A\circ B)\le\sigma_1(B)E(A)$$
$$\Rightarrow E(A)\ge\frac{E(A\circ B)}{\sigma_1(B)} \qquad (B.4)$$

Using B.2 in B.4, we observe,

$$E(A)\ge\frac{E(B)}{\sigma_1(B)} \qquad (B.5)$$

Now, by definition, the sum of interface complexities is,

$$C_2=\sum_{i=1}^{n}\sum_{j=1}^{n}b_{ij}=\|B\|_1 \qquad (B.6)$$

where $\|B\|_1$ is the Holder norm [Bernstein, 2009], defined as the sum of the absolute values of entries of the matrix. Hence, we have,

$$C_2C_3=\|B\|_1\left(\frac{E(A)}{n}\right)\ge\left(\frac{\|B\|_1}{n\sigma_1(B)}\right)E(B) \qquad (B.7)$$

Let us define a quantity, κ as,

$$\kappa=\frac{\|B\|_1}{n\sigma_1(B)} \qquad (B.8)$$

Now, the weighted adjacency matrix B is a non-negative matrix since all its elements are positive or zero. Therefore, by the Perron Frobenius theorem [Van Mieghem 2011; Bernstein 2009], we have the following inequality involving the largest singular value for any vector *y*,

$$\sigma_1(B)\ge\frac{y^TBy}{y^Ty} \qquad (B.9)$$

Equality is satisfied if *y* is the eigenvector corresponding to the largest eigenvalue of B matrix. Since it is valid for any vector, we can assume *y* to be the vector of ones, i.e., $y = u = 1$. From B.9, we derive,

$$\sigma_1(B) \geq \frac{u^T B u}{u^T u} = \frac{\| B \|_1}{n}$$

$$\Rightarrow \frac{\| B \|_1}{n\sigma_1(B)} \leq 1 \qquad (B.10)$$

Combining B.7 and B.10, we have,

$$\| B \|_1 \left(\frac{E(A)}{n} \right) \geq \kappa E(B) \qquad (B.11)$$

where $\kappa \leq 1$. Observe from inequality B.10 that $\kappa \to 1$ as the structure becomes more distributed, but depends on the magnitude of the entries in *B* matrix. This dependence can be shown by analyzing the degrees of the *B* matrix. We can express κ in terms of the weighted degrees using a variant of Peron Frobenius theorem. The largest singular value of matric B^2 is the square of the largest singular value of matrix *B*. Since B^2 is also a non-negative matrix, we can use Perron Frobenius theorem that yields,

$$\left[\sigma_1(B) \right]^2 \geq \frac{u^T B^2 u}{u^T u} \qquad (B.12)$$

where *u* is the vector of all ones. Now, $u^T u = n$ and now we have to express the numerator on the RHS in B.12.

$$u^T B^2 u = u^T B B u = (B^T u)^T (Bu) \qquad (B.13)$$

Since $u = 1_n$, we have $Bu = d_{out}$ where d_{out} is a vector of row-sums of matrix *B*. Also $B^T u = d_{in}$ where d_{in} is a vector of column-sums of matrix *B*. Hence, we can express B.13 as,

$$\begin{aligned} u^T B^2 u &= (B^T u)^T (Bu) \\ &= d_{in}^T d_{out} \\ &= \sum_{k=1}^{n} d_{in,k} d_{out,k} \qquad (B.14) \end{aligned}$$

where $d_{out,k}$ stands for the weighted out degree of node *k*. From elementary statistics, we have,

$$COV(d_{in},d_{out}) = \frac{\sum_{k=1}^{n}(d_{in,k}-\bar{d}_{in})(d_{out,k}-\bar{d}_{out})}{n}$$

$$= \frac{1}{n}\sum_{k=1}^{n} d_{in,k}d_{out,k} - \left(\bar{d}_{in}\right)\left(\bar{d}_{out}\right)$$

$$\Rightarrow \sum_{k=1}^{n} d_{in,k}d_{out,k} = n\left[\left(\bar{d}_{in}\right)\left(\bar{d}_{out}\right) + COV(d_{in},d_{out})\right]$$

$$= n\bar{d}^2\left[1+\frac{COV(d_{in},d_{out})}{\bar{d}^2}\right] \qquad (B.15)$$

where $\bar{d} = \bar{d}_{in} = \bar{d}_{out} = \frac{\| B \|_1}{n}$. Combining B.14 and B.15 into B.12, we get,

$$\frac{u^T B^2 u}{u^T u} = \bar{d}^2\left[1+\frac{COV(d_{in},d_{out})}{\bar{d}^2}\right]$$

$$= \left(\frac{\| B \|_1}{n}\right)^2\left[1+\frac{COV(d_{in},d_{out})}{(\| B \|_1/n)^2}\right] \qquad (B.16)$$

Applying B.16 in B.12, we get,

$$\sigma_1(B) \geq \left(\frac{\| B \|_1}{n}\right)\sqrt{1+p} \quad where,\ p = \frac{COV(d_{in},d_{out})}{\left(\| B \|_1/n\right)^2}$$

$$\Rightarrow \frac{\| B \|_1}{n\sigma_1(B)} \leq \frac{1}{\sqrt{1+p}} \qquad (B.17)$$

Using definition B.8, we have,

$$\kappa \leq \frac{1}{\sqrt{1+p}} \qquad (B.18)$$

The quantity, *p* is therefore related to the degree of association between the row-sum and column-sum vectors of weighted connectivity matrix *B*. A positive covariance leads to a positive value of *p*, which leads to a reduced value of κ. As $p \to 0$, we have $\kappa \to 1$ as the covariance between the row-sum and column-sum vectors *B* matrix vanishes. Please note that the value of *p* will be negative is there exist a high degree of dissociation between d_{in} and d_{out}. In such a scenario, we could

have $\kappa > 1$. In case of symmetric adjacency matrix, we have $B^T = B$ and that implies $d_{in} = d_{out} = d$ and we obtain, $p = \frac{VAR(d)}{\left(\| B \|_1 / n\right)^2}$. Note that *p* is a non-negative quantity by definition. The value of *p* depends on the degree of diversity in the weighted degree distribution with even distribution of weighted degrees leading to $\kappa \to 1$.

From inequalities above, we can conclude that the product $C_2 C_3$ and $E(B)$ are of same order. Simulation studies indicate that the LHS in B.11 is greater than $E(B)$, but depends on the weights in the adjacency matrix, i.e., the complexities of pair-wise interfaces. Hence, we conclude that the adopted decoupled version of structural complexity metric and one using the weighted adjacency matrix are of the same order and that the decoupled version is a conservative estimate over the weighted adjacency matrix version. The relation B.11 shows another justification for using the scaling factor of $\gamma = \frac{1}{n}$. Use of this scaling factor also aids classification of network structures into different regimes (i.e., hyper or hypo energetic regimes that maps to distributed and hierarchical topological forms).

The adopted decoupled version of the structural complexity metric also helps distinguish sources of structural complexity originating from topological aspects vis-à-vis those originating from individual pair-wise interfaces and component complexities. The component complexities, α is attached to individual compositional elements of the system and therefore, local to that particular element alone. The second term involves complexities due to pair-wise interfaces, β while the third term reflects the effects of the underlying connectivity structure. This term is defined as topological complexity and signifies the challenges of system integration. Higher topological complexity will likely lengthen system integration efforts significantly and it is a global property that is not visible locally.

C: Analogies to simple physical systems

The elastic network model has been used as a simple yet powerful tool for normal mode analysis of biological and other advanced material system synthesis. A

2D elastic network is represented as a set of random masses connected by random springs (see fig. C.1). Such representation has been used to relate abstract complex networks with a physical oscillator model [Estrada *et al.*, 2012].

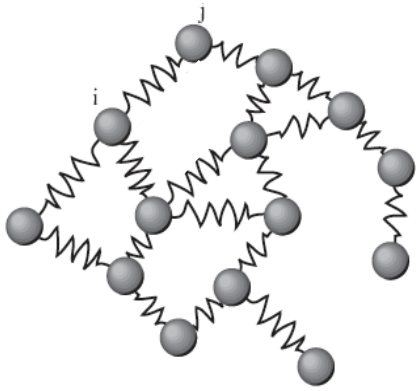


Fig. C.1: Random elastic network with masses connected by elastic springs. Since all interfaces are bidirectional, the adjacency matrix is symmetric.

Let us consider every node has mass *m* and every link as a spring with the spring constant *k* connecting two neighboring masses. Also assume that there is no damping and no external forces are applied to the system. Hence we have a 2D elastic network, where sites are indexed by *i*. With each site is associated a random mass m, and with each link (i, j) is associated a random spring constant *k*. The scalar model is defined by the following harmonic Hamiltonian for the scalar displacements $u_i(t)$. The harmonic Hamiltonian of the elastic network system can be written as [Monthus and Garel, 2011],

$$H = \frac{1}{2}\sum_i m\dot{u}_i^2 + \frac{1}{2}\sum_{(i,j)} kA_{ij}(u_j - u_i)^2 \qquad (C.1)$$

Here the adjacency matrix, A is symmetric since the springs are bidirectional and hence its singular values equal its absolute eigenvalues.

If we look at the second term in eq. C.1, it is the potential energy due to the assemblage of elastic springs and is linked to the network adjacency matrix. By considering the i^{th} mass as the anchoring point and setting $u_i = 0$, we can write the potential or the spring energy term as a quadratic form and the magnitude of total spring energy can be written as,

$$E_k = \sum_i |U_i| = \frac{1}{2} k \sum_i |\lambda_i u_i^T u_i| = \frac{1}{2} k \sum_i |\lambda_i| = \frac{1}{2} k \underbrace{\sum_i \sigma_i}_{E(A)} = \frac{1}{2} k E(A) \qquad (C.2)$$

Hence, from eq. C.2, we observe that the graph energy can be interpreted as the magnitude of energy content of a connected assemblage of elastic springs.

D: Graph Energy and Proper Orthogonal Decomposition (POD)

Proper orthogonal decomposition (POD) is a procedure for finding the dominant structures, expressed using proper orthogonal modes (POM), in an ensemble of spatially or temporally distributed data. It has been applied to estimate the dimensionality of a system, to build reduced order models, and to the identification of non-linear systems. As defined and developed in this thesis, the graph energy is same as the *matrix energy,* defined as the sum of singular values or the nuclear norm of the matrix, applied to graphs/networks.

On the other hand, Proper orthogonal decomposition (POD) is a procedure for extracting a basis for a modal decomposition from an ensemble of signals/data. It possesses a very appealing property of optimality in posteriori data/signal reconstruction. Here we will look at similarities between the process of POD and graph energy and argue that graph energy plays the equivalent role of proper orthogonal values (POV), which measures the relative energy of the system contained in the associated POM [Kerschen and Golinval, 2002], and is therefore, related to posterior topology reconstruction in the context of graphs/networks.

Proper Orthogonal Decomposition procedure is applied on a $(m \times n)$ data matrix, X where *m* stands for the number of system responses variables (i.e., degrees of freedom) and *n* stands for the number of data samples. Then the POMs are the eigenvectors of the covariance matrix, $C = (1/n) XX^T$ and the corresponding eigenvalues are the proper orthogonal values (POVs). The POVs measures the relative energy of the system contained in the associated POM [Kerschen and Golinval, 2002].

Let us look into details on how to compute the POMs and POVs. For any real $(m \times n)$ data matrix X, there exists a real factorization, called the Singular Value Decomposition (SVD):

$$X = U\Sigma V^T \qquad (D.1)$$

where U is an $(m \times m)$ orthonormal matrix, whose columns form the left singular vectors. Σ is an $(m \times n)$ matrix with diagonal entries containing the singular values $\sigma_i \geq 0$. V is an $(n \times n)$ orthonormal matrix, whose columns form the right singular vectors. The left singular vector U is expressed as the eigenvector of the matrix XX^T as shown below:

$$\begin{aligned} XX^T &= U\Sigma \underbrace{V^T V}_{I} \Sigma^T U^T \\ &= U\Sigma^2 U^T \qquad (D.2) \end{aligned}$$

Hence, the singular values of X are found to be the square roots of the eigenvalues of XX^T . The left singular vectors of X is the eigenvector of XX^T and the POMs, defined as the eigenvectors of the covariance matrix $C = (1/n)XX^T$, are the left singular vectors of X. The POVs, defined as the eigenvalues of XX^T , are the square of the singular values of X. Therefore, $\sigma_i^2(X)$ represents a measure of energy associated to the *i*th *mode* in the data and has the explicit meaning of vibration energy associated to the *i*th mode in structural dynamics problems. The sum of the POVs, $\sum_{i=1}^{m} \sigma_i^2(X)$ can be interpreted as the total energy associated to the system or data/signal.

POMs could be used to reconstruct a signal using a minimum number of modes. The POMs are related to the vibration eigen-modes in the area of structural dynamics and POD is an alternative way of modal analysis for extracting the mode shapes of a structure [Kerschen and Golinval, 2002].

If we look at the data matrix X, the matrix XX^T is a $(m \times m)$ square matrix representing the relationship embedded in the data/signal, where *m* represents the system responses or the degrees of freedom of the system.

This is in essence the exact replica of what the adjacency matrix A is in the system connectivity domain. While describing the connectivity among the system components, adjacency matrix describes the relationship among the system components (i.e., degrees of freedom in system connectivity).

In the same light, the graph energy of the adjacency matrix A (i.e., sum of singular values or the nuclear norm of the adjacency matrix) can be interpreted as total energy associated to the systems connection topology. As opposed to the physical meaning of the singular vectors in case of structural dynamics, we do not know of any general physical meaning associated to the singular vector of the adjacency matrix.

One can also look the relation linking the adjacency matrix A and the unsigned incidence matrix, B: $A+\Delta = BB^T$ where Δ is the diagonal degree matrix where each diagonal element is the degree associated to corresponding elements of the system.

We can express BB^T as the adjacency matrix A by setting the diagonal elements of BB^T to zero. In summary, what POD and POV are for data reconstruction, the graph energy is for reconstruction of the connection topology.

E: Graph Energy and Network Reconstructability

The reconstructability index θ of a network can be regarded as a spectral metric of the graph that expresses how many dimensions of the *n*-dimensional space are needed to *exactly* represent or reconstruct the graph [Liu *et al.* 2010, Van Mieghem 2011].

A larger reconstructability index reflects a "simpler" graph that only needs a few dimensions to describe. In other words, the higher the reconstructability index, the easier it is to reconstruct the graph accurately from reduced information. Please observe that *matrix energy* or *graph energy* express the sum of weights associated with the building block matrices required to represent or reconstruct the adjacency matrix *A*. This naturally leads us to the *graph reconstructability* viewpoint [Liu *et al.*

2010, Van Mieghem 2011] and the ability to easily reconstruct system structure can be viewed as the dual of topological complexity.

There is a close analytical link between Proper Orthogonal Decomposition (POD) and graph energy (see appendix D for details). It can be shown that the graph energy has a parallel in the Proper Orthogonal Values (POV) and the procedure of network reconstructability is similar in spirit to the POD methodology for data reconstruction.

In [Liu *et al.* 2010], the eigenvalue decomposition was used as the basis for spectral perturbation analysis and was restricted to undirected graphs only whose adjacency matrix is symmetric. Here, we follow a similar procedure but use singular value decomposition as the basis and therefore the procedure works for both undirected and directed graphs (see appendices E and F for detailed analysis). We can express the adjacency matrix using the singular value decomposition as:

$$A = U\Sigma V^T = \sum_{k=1}^{n} \sigma_k \underbrace{u_k v_k^T}_{E_k} = \sum_{k=1}^{n} \sigma_k E_k \qquad (E.1)$$

The adjacency matrix can, therefore, be considered as a weighted sum of simple building block matrices E_k of unit matrix energy where the associated weights are the corresponding singular values. In terms of individual elements a_{ij} of matrix A, we can write:

$$a_{ij} = \sum_{k=1}^{n} \sigma_k [E_k]_{ij} = \sum_{k=1}^{n-\theta} \sigma_k [E_k]_{ij} + \sum_{k=n-\theta+1}^{n} \sigma_k [E_k]_{ij} \qquad (E.2)$$

We define θ as the *reconstructability index*. It is the maximum number of singular values that can be removed from the spectrum of the graph without affecting our ability to reconstruct it using the retained information. The minimum number of singular values and associated singular vectors to be retained for exact reconstruction is $(n-\theta)$.

The fundamental idea here is to remove smaller singular values from the spectrum of a graph and check whether the graph can still be exactly reconstructed by exploiting the zero-one nature of the elements of the adjacency matrix. The

spectral perturbation here considered consists of consecutively removing more singular values from the spectrum, starting with the smallest singular value, until we can no longer exactly reconstruct the adjacency matrix A. Notice that, for exact reconstruction, we have to exactly replicate each element of the adjacency matrix. This requirement of *exact* reconstruction is in contrast to, for example, image compression, where some image information might be lost.

Now, after removing the smallest θ singular values, the adjacency matrix reduces to:

$$\tilde{A}_{(\theta)} = \sum_{k=1}^{n-\theta} \sigma_k E_k \qquad (E.3)$$

The elements of this matrix can be expressed as:

$$\tilde{a}_{ij} = \sum_{k=1}^{n-\theta} \sigma_k \underbrace{\left(E_k\right)_{ij}}_{\leq 1} \qquad (E.4)$$

Notice that any individual entry in set of E_k matrices cannot be greater than one. Now the individual elements of the original adjacency matrix and its reduced version are related as:

$$a_{ij} = \tilde{a}_{ij} + \sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij} \qquad (E.5)$$

The element-wise perturbation in the adjacency matrix is thus:

$$Elementwise\ perturbation = \left|a_{ij} - \tilde{a}_{ij}\right| = \left|\sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij}\right| \qquad (E.6)$$

Clearly, when $\theta = 0$, we have $\tilde{A}_{(0)} = A$. For any other $\theta > 0$, we have $\tilde{A}_{(\theta)} \neq A$ and also $\tilde{A}_{(\theta)}$ is no longer a zero-one binary matrix. The removal of a part of the singular value spectrum impacts the distribution of entries around 1 and 0 elements of the adjacency matrix A. Let us take the mid-point of this range, 0.5 as the pivot. We truncate the elements of matrix $\tilde{A}_{(\theta)}$ using the *Heavyside*'s step function, where $x = h\left(\tilde{a}_{ij} - \frac{1}{2}\right)$

$$h(x)=\begin{cases} 1 & if\ x>0 \\ \frac{1}{2} & if\ x=0 \\ 0 & if\ x<0 \end{cases} \qquad (E.7)$$

A graphical illustration of $h(x)$ is shown in fig. E.1 below.

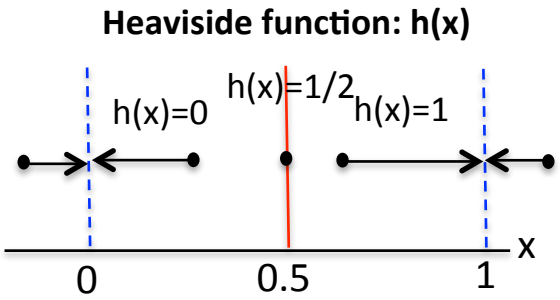


Fig. E.1: Graphical representation of Heavyside's function in one variable.

Pivoting around 0.5, the existing values of $\tilde{a}_{ij}$ are transformed into $h\left(\tilde{a}_{ij}-\frac{1}{2}\right)$ as depicted below:

$$\tilde{a}_{ij}\begin{cases} >\frac{1}{2} \Rightarrow x=\tilde{a}_{ij}-\frac{1}{2}>0 \Rightarrow h\left(\tilde{a}_{ij}-\frac{1}{2}\right)=1 \\ <\frac{1}{2} \Rightarrow x=\tilde{a}_{ij}-\frac{1}{2}<0 \Rightarrow h\left(\tilde{a}_{ij}-\frac{1}{2}\right)=0 \\ =\frac{1}{2} \Rightarrow x=\tilde{a}_{ij}-\frac{1}{2}=0 \Rightarrow h\left(\tilde{a}_{ij}-\frac{1}{2}\right)=\frac{1}{2} \end{cases} \qquad (E.8)$$

The new transformed matrix is $A_{(\theta)}=H\left[\tilde{A}_{(\theta)}-\frac{1}{2}\right]$ with elements given by $a_{(\theta)ij}=h\left(\tilde{a}_{ij}-\frac{1}{2}\right)$. It is a zero-one matrix, with the possible exception of elements with value 0.5. One can notice that such elements with exact value 0.5 leads to reconstruction error. For exact reconstruction, we should have $a_{ij}=h\left(\tilde{a}_{ij}-\frac{1}{2}\right)$.

Now, a_{ij} can be either 0 or 1 for the adjacency matrix. Therefore, we get the following conditions for exact reconstruction such that, $A = A_{(\theta)}$ where $A_{(\theta)} = H\left[\tilde{A}_{(\theta)} - \frac{1}{2}\right]$ with $a_{(\theta)ij} = h\left(\tilde{a}_{ij} - \frac{1}{2}\right)$:

$$If\ a_{ij} = \begin{cases} 1\ \ then\ \tilde{a}_{ij} > \frac{1}{2} \Rightarrow \sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij} < \frac{1}{2} \\ 0\ \ then\ \tilde{a}_{ij} < \frac{1}{2} \Rightarrow \sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij} > -\frac{1}{2} \end{cases} \qquad (E.9)$$

Combining these two requirements, we obtain the following element-wise perturbation for exact reconstruction,

$$\left|a_{ij} - \tilde{a}_{ij}\right| < \frac{1}{2} \ \Rightarrow \left|\sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij}\right| < \frac{1}{2} \qquad (E.10)$$

Notice that this condition has to be satisfied for each element of the matrix. Hence, the problem reduces to maximizing θ such that the following inequality is satisfied:

$$\left|\sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij}\right| < \frac{1}{2} \qquad (E.11)$$

In other words, there exist a $\theta = \theta_*$ such that,

$$A_{(\theta)} = A \ \ for\ \theta \le \theta_*$$

$$A_{(\theta)} \ne A \ \ for\ \theta > \theta_*$$

Therefore, θ is the maximum number of singular values that can be removed from the original spectrum of the graph without affecting our ability to reconstruct it using the retained information. The minimum number of singular values (with associated singular vectors) to be retained for exact reconstruction is $(n-\theta)$.

Finding an analytical solution is difficult due to existence of the matrix elements $\left(E_k\right)_{ij}$. Since $\left|\left(E_k\right)_{ij}\right| \le 1$, we have the following inequality relation,

$$\left| \sum_{k=n-\theta+1}^{n} \sigma_k \left(E_k\right)_{ij} \right| \leq \sum_{k=n-\theta+1}^{n} \left|\sigma_k\right| \underbrace{\left|\left(E_k\right)_{ij}\right|}_{\leq 1} \leq \sum_{k=n-\theta+1}^{n} \sigma_k \qquad (E.12)$$

Combining this inequality with our basic matrix element level bound, we can write the following bound for exact reconstruction:

$$\left(\sum_{k=n-\theta+1}^{n} \sigma_k \right) < \frac{1}{2} \qquad (E.13)$$

In general, this bound turns out to be very conservative and use of simulation is the only worthwhile option for computing reconstructability of graphs.

Representative results from simulation studies are shown below (see fig. E.2):

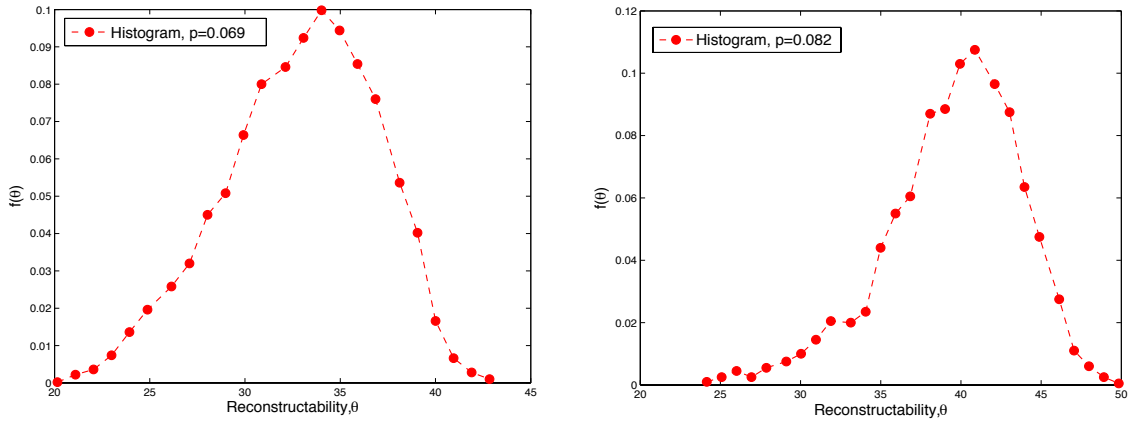


Fig. E.2: Probability density function of reconstructability index, *f(θ)* for Erdos-Renyi random graphs with 73 and 84 nodes respectively with different connection density *p*. For each *p*, 5000 samples were used.

The probability density function for the reconstructability index tends to show *left-skewed* behavior – there is a finite possibility of a small reconstructability index value for a random graph with given number of nodes and link density. In other words, there is always some graph structure that embodies small reconstructability (also large topological complexity according to P.7 in chapter 3).

We can normalize the reconstructability index by the number of nodes in the graph and define the reconstructability coefficient, $\hat{\theta} = \frac{\theta}{n}$. The typical probability

density function for the reconstructability coefficient is a scaled version of that for the reconstructability index (see fig. E.2) and is shown in fig. E.3 below.

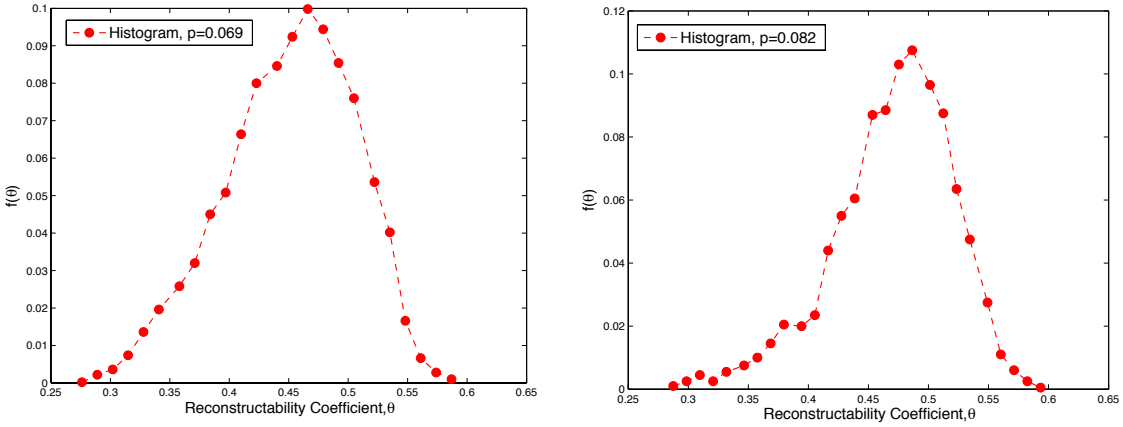


Fig. E.3: Probability density function of reconstructability coefficient, $f(\hat{\theta})$ for Erdos-Renyi graphs with 73 and 84 nodes respectively with different connection density *p*.

Also, notice that the reconstructability coefficient has the most likely value of around 0.6 or 60% in both cases. This means that for ER random graphs, on the average, we need to keep about 60% of singular values and associated singular vectors for exact reconstruction of the graph. Now, using simulation, let us look at the relationship between topological complexity and graph reconstructability and possible *dualism* between the two.

As we observed, the *matrix energy* or *graph energy* expresses the sum of weights associated with the building block matrices required to represent or reconstruct the adjacency matrix A. This naturally leads us to the *graph reconstructability* viewpoint discussed above. We can view the ability to easily construct system structure as the dual of topological complexity. Simulation studies indicate such dualism between graph energy and graph *reconstructability*. Minimum topological complexity implies maximum reconstructability (see fig. E.4 below). The behavior of graph reconstructability with increasing link density is essentially the inverse of the behavior observed earlier for graph energy. To explore this further, we looked at their respective behavior on Fabrikant model [Fabrikant *et al.,* 2002]. Upon normalizing both on a [0, |1|] scale, this dualism becomes quite prominent.

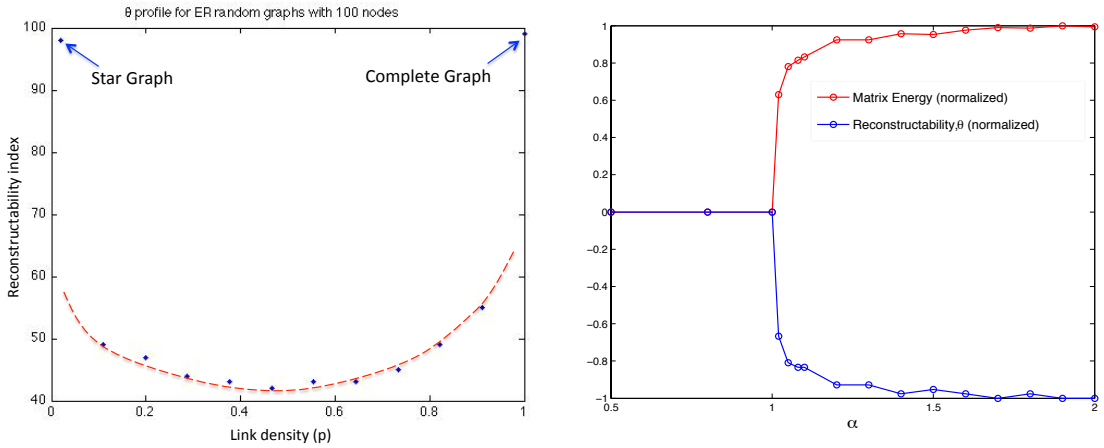


Fig. E.4: Observed Dualism between topological complexity metric (i.e., matrix energy) and reconstructability index: (a) variation of reconstructability with link density for Erdos-Renyi random graphs of size, n = 100 nodes; and (b) on Fabrikant networks [Fabrikant *et al.,* 2002] with varying α. The two quantities are normalized in [0,1] (for matrix energy) and [- 1,0] (for reconstructability index) respectively for visualization.

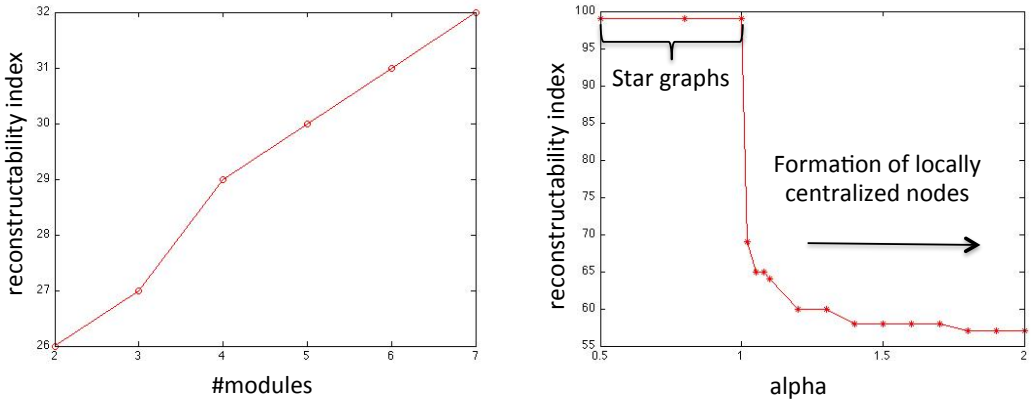


Fig. E.5: Variation of reconstructability for (a) random modular graphs with 100 nodes with 5% inter-module links – reconstructability increases with number of modules; and (b) Random trees with n = 100 nodes – reconstructability increases with parameter α [Fabrikant *et al.,* 2002].

Similar behavior was also when witnessed when applied to random modular graphs and random trees (see fig. E.5). Graph reconstructability increases as we increase the number of modules. When applied to the same setting, the graph energy reduces as we increase the number of modules. In case of Fabrikant's model (i.e., random trees), formation of locally central structures in lieu of global centralization drives the network reconstructability down while increasing the

topological complexity. Theoretical development linking graph energy and graph reconstructability from its deck is still elusive and beyond the scope of this thesis, but simulation studies indicate a strong *dualism*. A probabilistic analysis linking the two and proof of such dualism properties remains an interesting area for future exploration.

The variation of graph energy and topological complexity metric captures the changes in connectivity structure for Fabrikant trees with given number of nodes, starting with star networks, and morphing into tree's with multiple hubs (closer to pref. attachment type). This structural transition is accomplished by varying parameter, α.

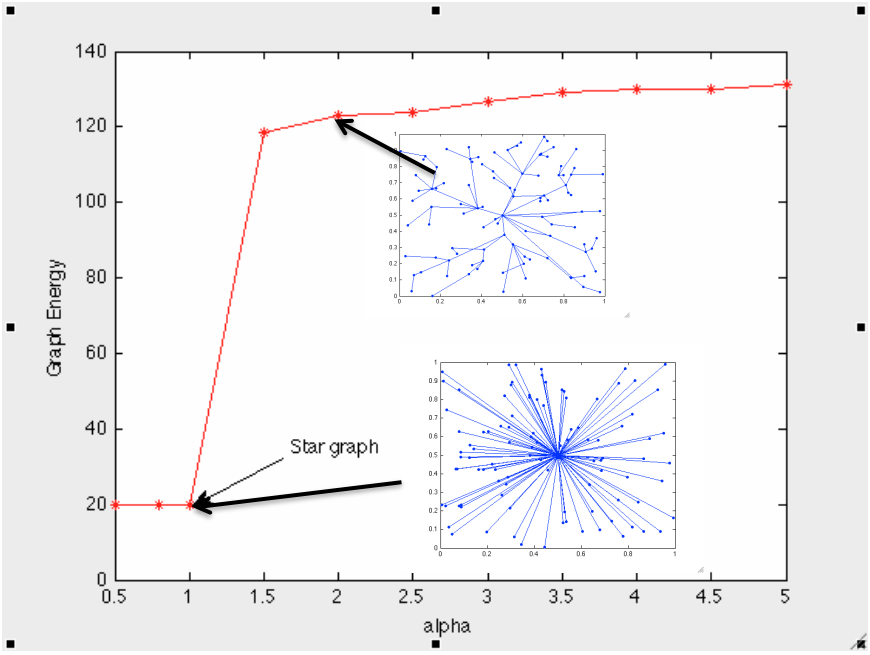


Fig. E.6: Variation of graph energy for Fabrikant trees with 100 nodes and 99 edges with increasing parameter, α.

If α is less than a particular constant, the resulting network is easily seen to be a star. If α is larget than a certain constant but grows slower than $\sqrt{n}$, where n is the final number of points, then the resulting graph is almost certainly, obey a power law type degree distribution with multiple local hubs as shown in fig. E.6.

F: Symmetric vis-à-vis asymmetric adjacency matrices - impact on Structural Complexity

From chapter 3, the Structural Complexity is given by the expression

$$C = C_1 + C_2 C_3 = \underbrace{\sum_{i=1}^{n} \alpha_i}_{C_1} + \underbrace{\left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{i,j} A_{i,j} \right)}_{C_2} \underbrace{\frac{E(A)}{n}}_{C_3} \qquad (F.1)$$

where C_1 stands for the sum of component complexities, C_2 stands for the sum of interface complexities, and C_3 stands for topological complexity and concerns only the adjacency or connectivity matrix of the system.

From eq. F.1, we observe that the component and interface complexities in the Structural Complexity metric are not directly impacted by the presence or absence of symmetry in the binary adjacency matrix (assuming that the interface complexities for bi-directional interface remains the same as for directed interfaces). The only term in eq. F.1 that is impacted by the presence of asymmetry in the adjacency matrix is the topological complexity, C_3.

In any engineered systems all of the element-to-element connections are neither completely undirected (i.e., bi-directional) or completely directed (i.e., unidirectional). Hence, the random graph that is representative of real engineered systems is a mixed Erdos-Renyi (ER) random graph. The mixed ER graph is a modification to the undirected ER graph in which a fraction of the links is directed. While studying the impact of directness of links in the adjacency matrix, we have to analyze for two different situations: (i) the number of links between the undirected and mixed versions of the network is fixed; (ii) the number of links in the mixed graph is larger than that of the undirected graph while keeping the number of non-zero entries in the adjacency matrix remains constant.

In the first scenario, the mixed graph has the same number of links as in the undirected graph. Now each undirected links gives rise to two interfaces since each link is bi-directional. Hence the number of non-zero entries in the adjacency matrix

for purely directed graph is half of the purely undirected graph. The total number of links = #undirected links + #directed links. Let us define, ratio = (#directed links)/(total number of links). Table F.1 shows the averaged value of the worst-case difference, defined as $\{E(A) - E(A_{mixed}) / E(A)\}$, for different values of the ratio.

Table F.1: The number of non-zero entries in mixed ER graph varies, depending on the value of ratio. The worst-case difference between purely undirected vis-à-vis purely directed graphs reached as high as 37.43 %. Worst case difference, averaged over 100 instances.

Ratio	E(A)	$E(A_{mixed})$	Average worst case difference (%)	(m_{dir}, m_{undir})	#non-zeros in adjacency matrix
0.1	168.14	155.45	7.54	(20, 180)	380
0.2	168.2	151.57	9.88	(40, 160)	360
0.3	168.27	144.46	14.15	(60, 140)	340
0.4	168.24	141.5	15.89	(80, 120)	320
0.5	168.18	137.18	18.43	(100, 100)	300
0.6	168.09	132.02	21.46	(120, 80)	280
0.7	168.26	125.43	25.45	(140, 60)	260
0.8	168.34	118.58	29.56	(160, 40)	240
0.9	168.15	113.2	32.68	(180, 20)	220
1.0	168.28	105.3	37.43	(200, 0)	200

As the value of ratio increases from 10% to 100%, the average worst-case difference increases from about 7% to as high as 37%. This is expected as the having less number of interfaces (i.e., number of non-zero entries in the adjacency matrix) usually results in reduced topological complexity.

In fig. F.1 below, the average graph energy is plotted against the variation in fraction of undirected edges in the mixed ER graph. The graph energy was plotted after normalizing with respect to the graph energy of purely undirected ER graph. Variation in density (with increasing number of links) does not impact this profile. We observe that, even for mixed graphs with as high as 33% directed links, the graph energy of is more than 90% of the corresponding pure, undirected ER graph

with the same number of nodes and links. In most engineered systems, the number of directed links as a fraction of the total number of links is well within this value. If the number of directed links is within 20% of the total number of links, the error in graph energy (by using the graph energy of the pure undirected ER graph) is within 5%.

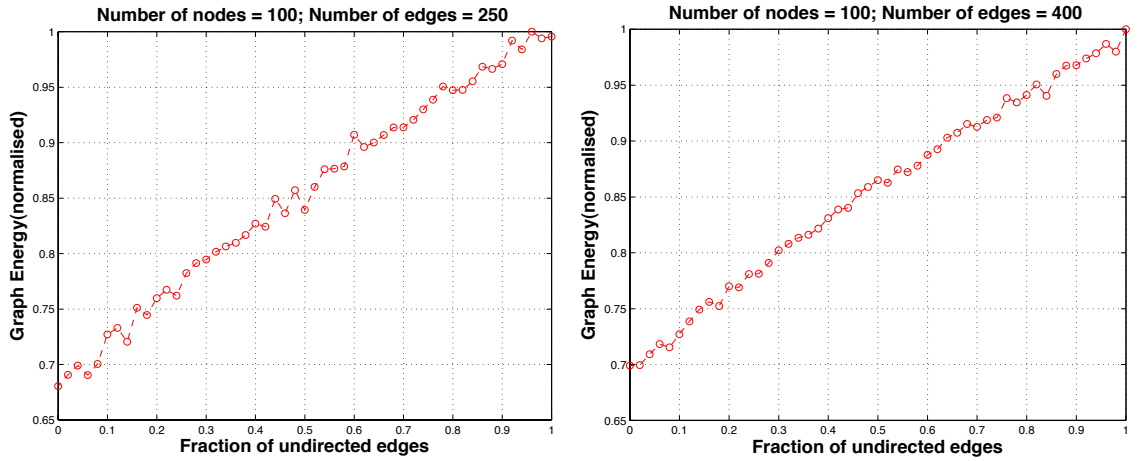


Fig. F.1: Variation of graph energy (normalized to 1) for mixed ER graphs without link adjustment for (i) 100 nodes and 250 links; and (ii) 100 nodes with 400 links.

Simulation results show that the same behavior holds other, more esoteric random graphs with specified degree distribution. Also, graph energy of the symmetrized version of the mixed ER graph (i.e., $A_{symm} = unitize\left[\frac{A_{mixed} + A_{mixed}^{T}}{2}\right]$ where $\left(a_{ij}\right)_{symm} = 1 \; iff \; \left(a_{ij}\right)_{mixed} + \left(a_{ji}\right)_{mixed} > 0$) lies within 4% of the graph energy of the corresponding undirected ER graph.

Under the second scenario, the number of non-zero entries in the adjacency matrix, A remains constant. In order to satisfy this constraint, let m_{eq} be the total number of links in the adjacency matrix of the mixed ER graph, A_{mixed}. Hence the number of directed links is $ratio \times m_{eq}$ and the number of undirected links is $(1 - ratio) \times m_{eq}$. Let m be the number of links in pure ER graph (i.e., undirected). The total number of links in the mixed adjacency matrix can be computed as show below,

$$nz_{ER} = nz_{dir} + nz_{undir}$$
$$2m = ratio \times m_{eq} + 2m_{eq}(1 - ratio)$$
$$\therefore m_{eq} = m\left(\frac{2}{2 - ratio}\right) \qquad (F.2)$$

Following the same procedure as earlier, table F.2 shows the averaged value of the worst-case difference, defined as $\{E(A) - E(A_{mixed}) / E(A)\}$, for different values of the ratio.

Table F.2: Worst case difference, averaged over 100 instances. The number of non-zero entries in the adjacency matrix is set to 200 for all cases. The worst-case difference is nearly constant at 4% for all case, indicating invariance to the value of *ratio* under this scenario.

Ratio	E(A)	$E(A_{mixed})$	Average worst case difference (%)	(m_{dir}, m_{undir})	Equivalent number of links (m_{eqv})	#non-zeros in adjacency matrix
0.1	168.14	161.1	4.187	(22, 189)	211	400
0.2	168.2	161	4.28	(44, 178)	222	400
0.3	168.27	160.88	4.39	(70, 165)	235	400
0.4	168.24	160.85	4.39	(100, 150)	250	400
0.5	168.18	160.78	4.40	(134, 133)	267	400
0.6	168.09	160.73	4.38	(172, 114)	286	400
0.7	168.26	160.68	4.50	(214, 93)	311	400
0.8	168.34	160.74	4.51	(266, 67)	333	400
0.9	168.15	160.61	4.484	(326, 37)	363	400
1.0	168.28	160.92	4.38	(400, 0)	400	400

As the value of ratio increases from 10% to 100%, the average worst-case difference remains fairly constant at around 4%. This is expected as we always have the same number of interfaces (i.e., number of non-zero entries in the adjacency matrix) resulting in nearly the same level of topological complexity.

G: Graph Energy bounds for general asymmetric adjacency matrices

We have shown important analytical bounds for graph energy in chapter 3 and assumed undirected graph, resulting in symmetric adjacency matrix. In case of symmetric matrix, the singular values were equal to the absolute eigen values and the singular vectors were directly related to signed eigen vectors.

This helped us leverage some well-established mathematical properties related to the eigenvalues and establish bounds analytically. Here we will look at the extension of such bounds for mixed graphs where the links are a mix of directed and undirected ones.

Here, we focus on the bounds for graph energy for generalized asymmetric adjacency matrices and show that, $\sqrt{2sm} \le E \le sm$, where m is the number of edges/links in the simple graph.

In case of a mixed graph with both, directed and undirected links, the sum of all the elements of the adjacency matrix is,

$$\| A \|_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left| a_{ij} \right| = sm \qquad (G.1)$$

where $\| A \|_1$ is the Holder norm [Bernstein, 2009], defined as the sum of the absolute values of entries of the matrix and $s \in [1,2]$. Here $s = 1$ for purely directed ER graph (i.e., all links are unidirectional) and $s = 2$ for purely undirected ER graph (i.e., all links are bidirectional). Using the Frobenius norm, $\left\| A \right\|_F$ [Bernstein, 2009], we have:

$$\left\| A \right\|_F = \sum_{i=1}^{n} \sigma_i^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2 = sm \qquad (G.2)$$

Now, we can express the squares of the sum of the singular values as,

$$\left(\sum_{i=1}^{n} \sigma_i \right)^2 = \sum_{i=1}^{n} \sigma_i^2 + 2 \sum_{1 \le i \le j \le n} \sigma_i \sigma_j$$
$$= sm + 2 \sum_{1 \le i \le j \le n} \sigma_i \sigma_j \qquad (G.3)$$

In eq. G.3, for the second term on the right hand side, we are not aware of any closed form analytical bounds. We do have a lower bound of $2m$ if A is symmetric (i.e., undirected graphs). Using extensive simulation (see fig. G.1 and G.2), we write the following lower bound,

$$2\sum_{1\le i\le j\le n}\sigma_i\sigma_j \ge sm \qquad (G.4)$$

Fig. G.1 and G.2 shows existence of the above bound for mixed ER graphs and such bounds were found for other types of random graph (e.g., with specified degree distributions). Combing the bound in G.4 with relation G.3, we obtain the following lower bound for the graph energy of general mixed graphs with asymmetric adjacency matrices,

$$\left(\sum_{i=1}^{n}\sigma_i\right)^2 \ge 2sm$$

$$\therefore E \ge \sqrt{2sm} \qquad (G.5)$$

Now, let us turn our attention to the upper bound. Using the Cauchy-Schwarz inequality for arbitrary real-valued numbers a_i, b_i with i=1,2, . . . ,*N*, we have

$$\left(\sum_{i=1}^{N}a_ib_i\right)^2 \le \left(\sum_{i=1}^{N}a_i^2\right)\left(\sum_{i=1}^{N}b_i^2\right)$$

If we choose N = n, $a_i = \sigma_i$ and b_i = 1, we get,

$$\left(\sum_{i=1}^{n}\sigma_i\right)^2 \le \underbrace{\left(\sum_{i=1}^{n}\sigma_i^2\right)}_{=sm} n$$

$$E^2 \le smn \qquad (G.6)$$

In order to have at least a *connected* graph, the following inequality should hold,

$$sm \ge n \qquad (G.7)$$

Utilizing G.7 in G.6, we can write,

$$E^2 \le s^2m^2$$

$$\therefore E \le sm \qquad (G.8)$$

Combing relations G.5 and G.8, we arrive at the following bound for the graph energy of general, asymmetric adjacency matrices with both, directed and undirected edges,

$$\sqrt{2sm} \le E \le sm \qquad (G.9)$$

Please note that for undirected graph, we have $s = 2$ and we get back the established bounds, $2\sqrt{m} \le E \le 2m$. For purely directed graphs, we have $s = 1$ and the bounds are given by, $\sqrt{2m} \le E \le m$.

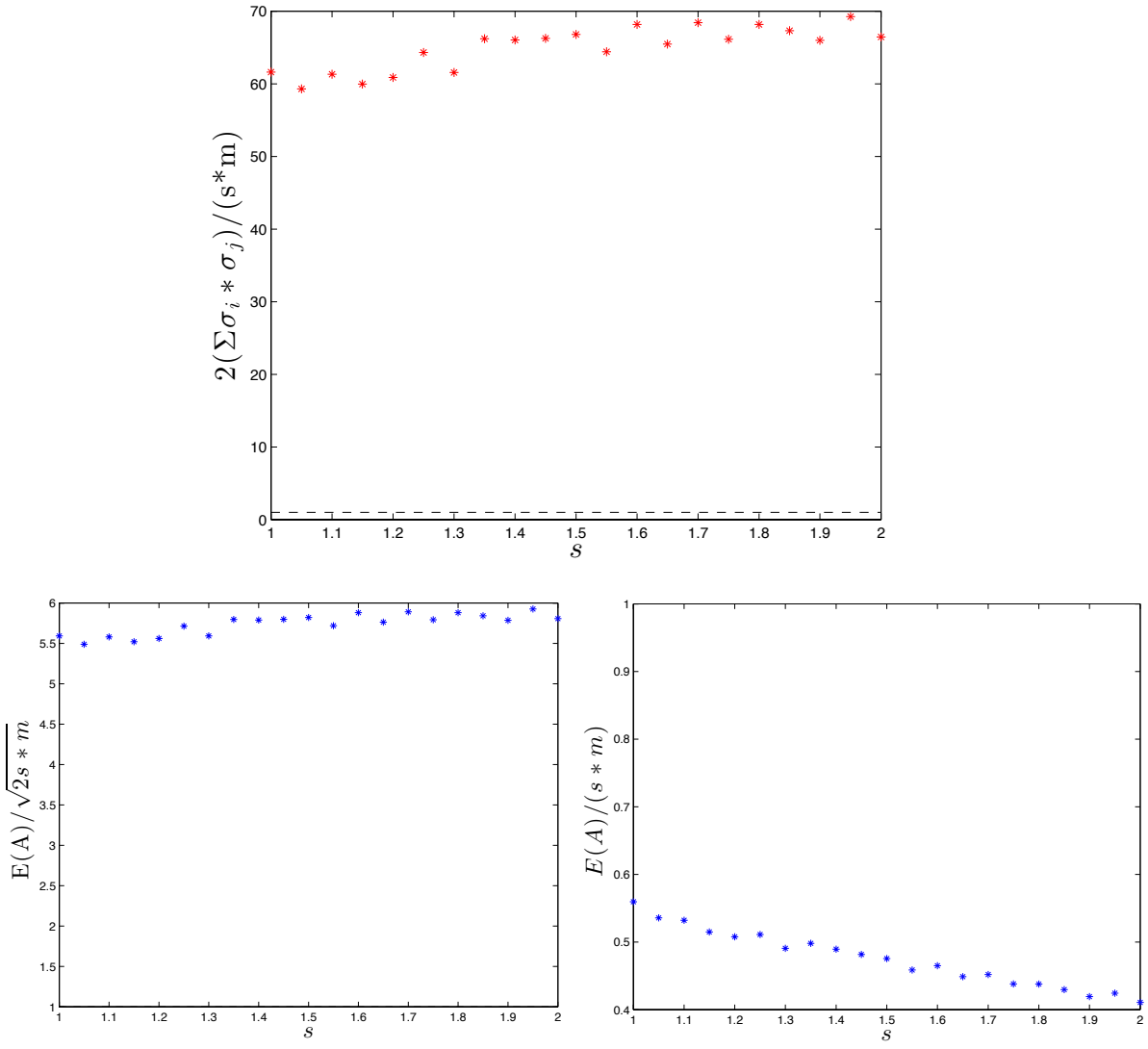


Fig. G.1: Simulation results supporting (i) the bound in F.4 for mixed ER graphs with n = 100 nodes and m = 200 links; (ii) the lower and (iii) the upper bound of graph energy, E(A) for asymmetric adjacency matrices, modeling mixed ER graphs.

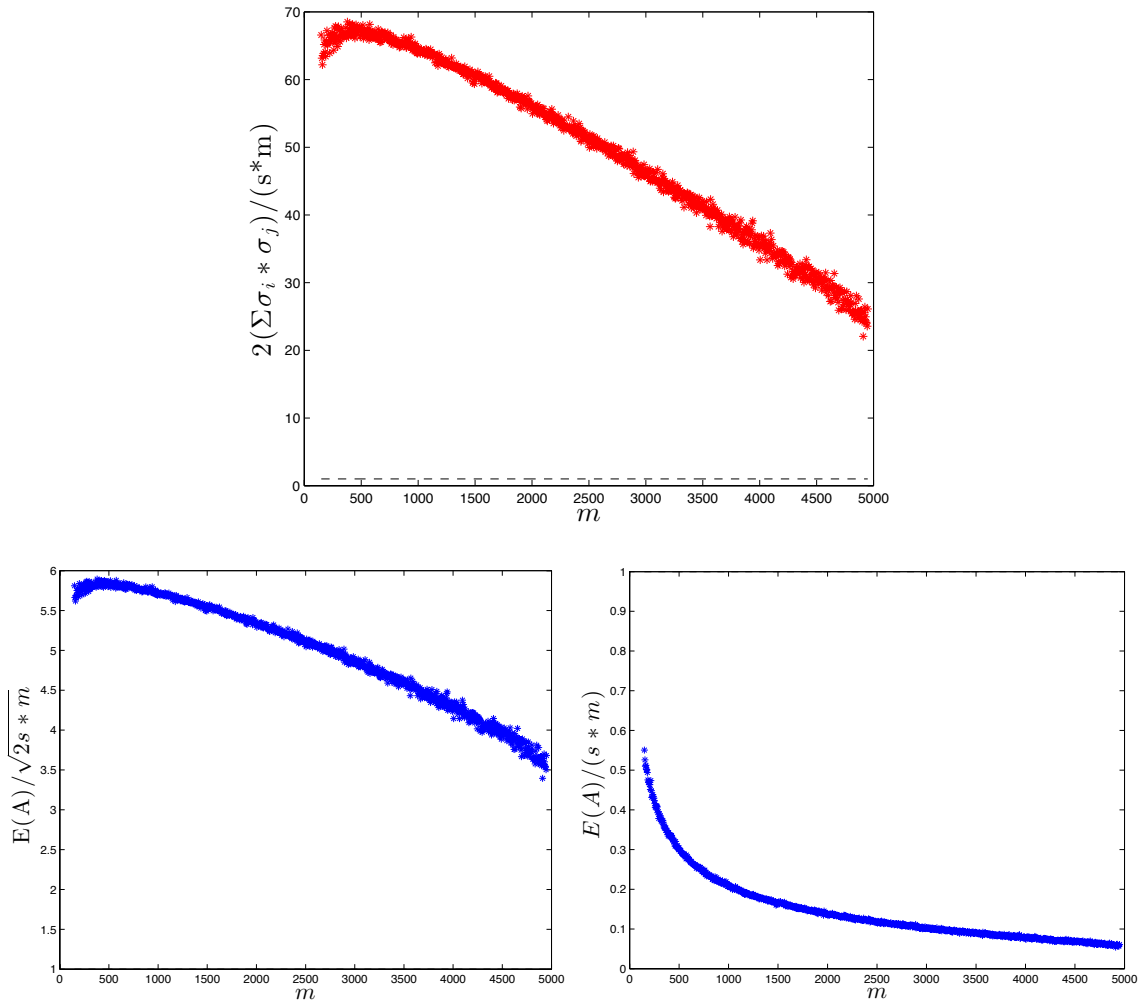


Fig. G.2: Simulation results supporting (i) the bound in F.4 for mixed ER graphs with n = 100 nodes, s = 1.5 with varying number of links, *m*; (ii) the lower and (iii) the upper bound of graph energy, E(A) for asymmetric adjacency matrices with n = 100 nodes, s = 1.5 with varying number of links, *m*.

In general this bound was not found to be *tight* for any kind of generic random graphs.

Now, using the Cauchy-Schwarz inequality again with N = n-1, $a_i = \sigma_{i+1}$ and b_i = 1, we obtain the following for mixed graphs with asymmetric adjacency matrices,

$$(E-\sigma_1)^2 \le (n-1)(sm-\sigma_1^2)$$
$$\therefore E \le \sigma_1 + \sqrt{(n-1)(sm-\sigma_1^2)} \qquad (G.10)$$

Using the earlier relations: $\sum_{i=1}^{n}\sigma^2{}_i = sm$ and $\sigma_1 \ge \frac{sm}{n}$, the above relationship implies,

$$E \le \frac{sm}{n} + \sqrt{(n-1)[sm - \left(\frac{sm}{n}\right)^2]} \qquad (G.11)$$

The limiting form of the above relation can be expressed in the following form for fixed *n,* (e.g., given matrix size):

$$f(m) = sm/n + \sqrt{(n-1)[sm - (sm/n)^2]} \qquad (G.12)$$

Let us maximize the function $f(m)$ above where n is fixed. Applying the *Kuhn-Tucker* optimality criteria [Pishkunov, 1976], we should have:

$$\frac{df}{dm} = 0$$

$$\Rightarrow \frac{s}{n}\left\{1 + \frac{(n-1)(n^2 - 2sm)}{2\sqrt{(n-1)sm(n^2 - sm)}}\right\} = 0$$

On algebraic simplification, we get,

$$(n-1)(n^2 - 2sm)^2 = 4sm(n^2 - sm)$$

$$\Rightarrow (2sm - n^2)^2 = n^3$$

$$\therefore m = \frac{n^2 + n^{3/2}}{2s} = \frac{n^2\left(1 + \frac{1}{\sqrt{n}}\right)}{2s} \qquad (G.13)$$

Using the above result, we compute the corresponding value limiting graph energy for the general case of asymmetric adjacency matrices with both, directed and undirected links,

$$f = \frac{n\left(1+\frac{1}{\sqrt{n}}\right)}{2} + \sqrt{(n-1)\left[\frac{n^2\left(1+\frac{1}{\sqrt{n}}\right)}{2} - \frac{n^2\left(1+\frac{1}{\sqrt{n}}\right)^2}{4}\right]}$$

$$= \frac{n\left(1+\frac{1}{\sqrt{n}}\right)}{2} + (n-1)\frac{\sqrt{n}}{2}$$

$$= \frac{n\left(1+\sqrt{n}\right)}{2}$$

Therefore, for the maximal limiting value of graph energy in this case, we get:

$$m^* = \frac{n^2\left(1+\frac{1}{\sqrt{n}}\right)}{2s} \approx O(n^2) \qquad (G.14)$$

$$f_{\max} = \frac{n(1+\sqrt{n})}{2} \approx O(n^{3/2}) \qquad (G.15)$$

Hence, using G.11 and G.15, we conclude that,

$$E_{\max} \le \frac{n\left(1+\sqrt{n}\right)}{2} \qquad (G.16)$$

Therefore, the maximal graph energy is bounded by $n^{3/2}$,

$$E_{\max} \le n^{3/2} \qquad (G.17)$$

Hence, the maximal graph energy bound is not depended on the adjacency matrix being symmetric (i.e., undirected graphs).

The results in this appendix extend the results derived analytically in chapter 3 in the special case of undirected graphs, to general asymmetric adjacency matrices, representing graphs with both, directed and undirected links.

H: *P* point and its role in System Architecture

The *P* point relates to a certain level of density of a simple graph. The graph energy regime for graphs with a given number of nodes, can be divided into: (i) *hyperenergetic* and (ii) *hypoenergetic*. The *hyperenergetic* regime is defined by graph energy greater than or equal to that of the fully connected graph:

$$E(A) \geq 2(n-1) \qquad (H.1)$$

There does exist an intermediate or transition regime between these two where the energy is higher than that of the *hypoenergetic* regime but is smaller than the *hyperenergetic* one [Li *et al.*, 2012].

Hence, in terms of topological complexity metric, the regimes are defined as:

$$C_3 = \frac{E(A)}{n} = \begin{cases} \geq 2\left(1-\frac{1}{n}\right) \approx 2 & \text{- hyperenergetic} \\ < 1 & \text{- hypoenergetic} \end{cases} \qquad (H.2)$$

The intermediate or transitional energetic regime is the interval: $1 \leq C_3 < 2$. The *P* point is the minimum level of density of the graph at which the equality in H.1 holds and beyond this density level, almost all realizations of graphs with the given number of nodes are hyperenergetic.

Based on extensive simulation studies on ER random graphs with 100 vertices and averaged over 1,000 instances at each network density level, the variation of the critical density and corresponding critical average degree for varying the graph size revealed that the critical density reduces with graph size, but the critical average degree tends to remain nearly constant around 5.7. Similar behavior was also observed for scale-free random graphs that yield a marginally higher critical average degree of 6.4. We can say, for practical purposes, that the *P* point is characterized by an average degree of 6. The average degree of *non-hyperenergetic* networks is less than this critical value of average degree of network of given number of vertices.

In chapter 3, it was shown that the *P* point indicates transition to hyperenergetic region in terms of graph energy. Interestingly, *P* point also coincides the random graph becoming rank-sufficient (i.e., full-rank) on average. We define a metric termed *Rank Sufficiency Factor* (RSF) as the normalized matrix rank,

$$RSF = \frac{r}{n}$$

where r stands for the rank of the adjacency matrix. Fig. H.1 shows the relationship between the topological complexity, C_3 and RSF. Notice that, on average, the RSF saturates as we enter into the hyperenergetic regime with $C_2 \approx 2$.

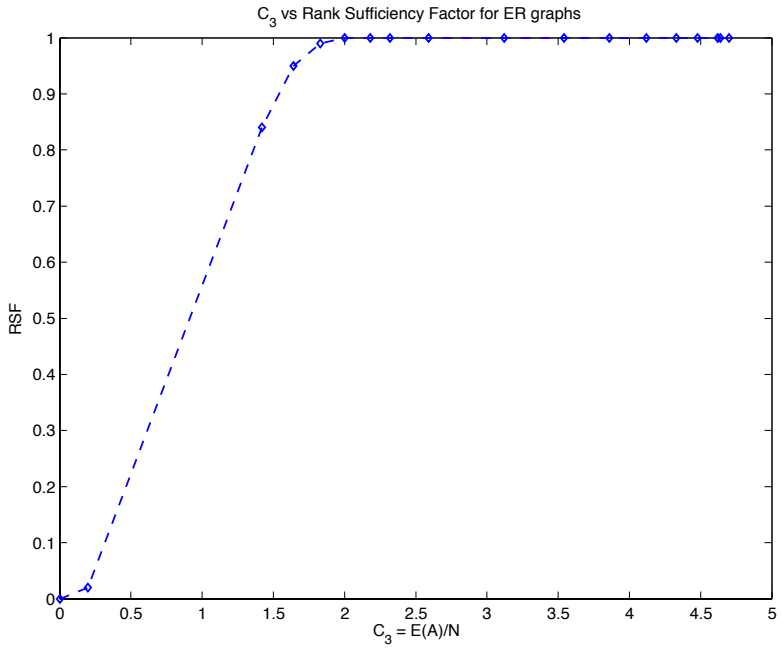


Fig. H.1: Variation of topological complexity, C_3 and Rank Sufficiency Factor (RSF) for ER graphs with n = 100 nodes and varying density, averaged over 1000 realizations at each density. The adjacency matrix becomes rank-sufficient (i.e., full-rank) around $C_3 \approx 2$, and this is defined as the *P* point. At this point, the graph energy, *E(A)* becomes equal to that of the fully connected graph.

The *P* point shows very interesting features that relates to interesting characteristics of the graph. It appears that nearness to rank-sufficiency of the network has important bearing on other network metrics as well. Simulations indicate approaching saturation in terms of improvement in other network metrics

like maximum diameter, average path length (see fig. H.2) and other network communicability indices.

It is interesting to note that the *P* point seem to indicate a kind of transition region for network diameter beyond which it settles down to a near constant value with increasing connection density. A lower diameter indicates that information can pass between any two nodes with smaller hop distance. A somewhat similar measure of how quickly information can flow within a network is given by the *mixing time* of a network [Chen, 2004; Butler, 2008]. The *P* point is again in the *knee* region of mixing time vs. density plots (see fig. H.2). Hence beyond the critical density the mixing time remains almost constant but increases exponentially if the density is reduced. So the *P* point acts like a transition point for both network diameter and mixing time.

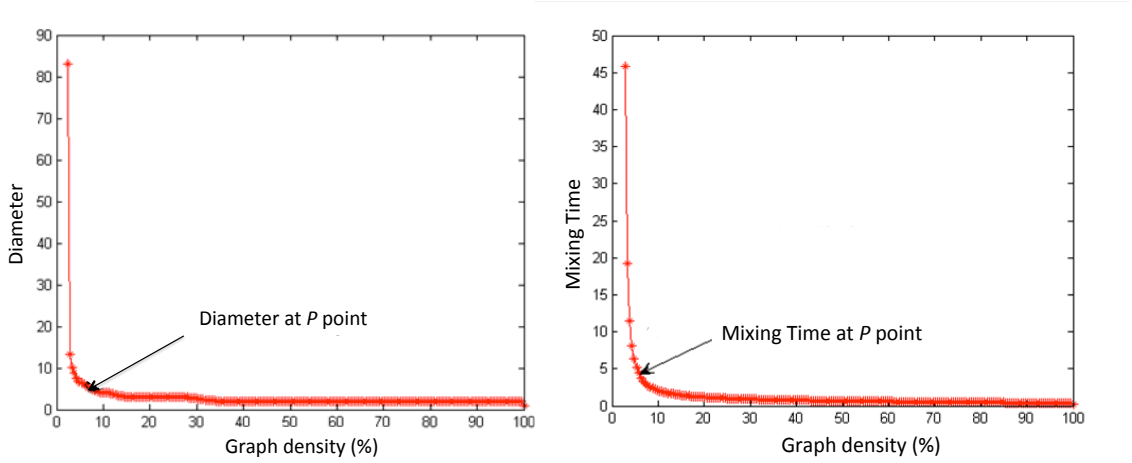


Fig. H.2: Saturation of various network metrics beyond the *P* point for ER graphs: (a) network diameter; and (ii) mixing time. The P point appears to classify a *knee* region, beyond which we do not gain from enhancement in other network metrics, although topological and structural complexity keeps increasing.

Another interesting observation was made in an analytical study by [Valente et al., 2004] regarding the resilience of general random networks against both, targeted and random attack on nodes. They defined two metrics to measure the resilience against nodal failures as defined below:

f_a: fraction of targeted nodes before the giant component vanishes.

f_r: fraction of randomly deleted nodes before giant component vanishes.

and studied the (f_a vs. f_r) envelope for general random graphs. The envelopes are shown in fig. G.3 (b) and it appears that the outward growth of the envelope saturates beyond the average degree, <k> = 6 (see fig. H.3). This again coincides with *P* point on the graph density plot.

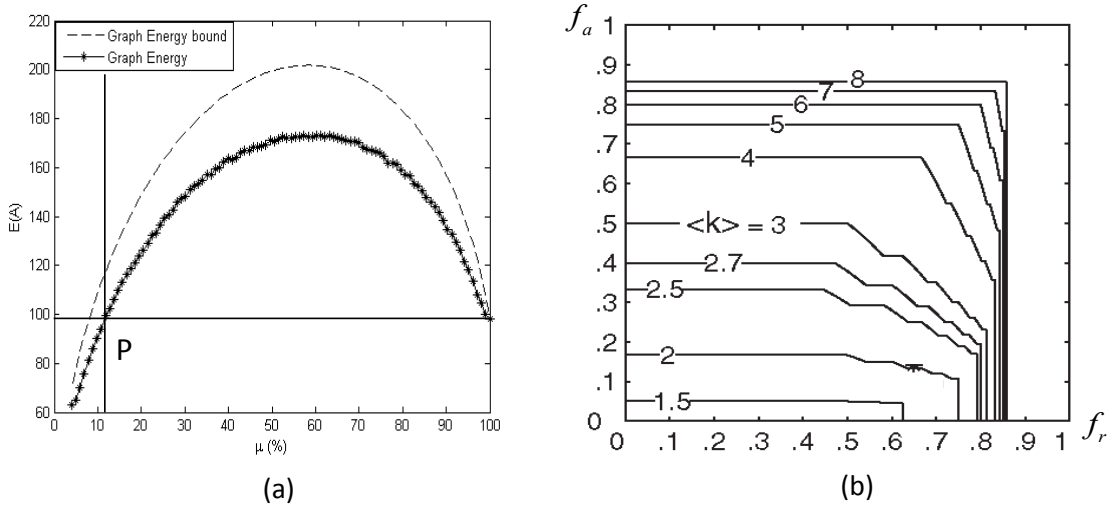


Fig. H.3: Characterization of (a) the *P* point on graph energy vs. graph density plot, and (b) the network resilience contour (f_a vs. f_r) for general random graphs [Valente *et al.*, 2004] – beyond the average degree <k> = 6, the is minimal outward growth of the network resilience envelope.

Graph Energy and Network Probabilistic resilience: Intuitively, a complex network is robust if it keeps its basic functionality even under failure of some of its components (e.g., nodes). The study of robustness in networks is important because a thorough understanding of the behavior of certain classes of networks under failures and attacks may help to protect important systems that serve the society [Klau G.W. and Weiskircher R, 2005]. This section (i) describes robustness statistics that explicitly consider the failure probabilities of system/network components and are thus more appropriate to describe untargeted component failure, and (ii) link that robustness statistic to the topological complexity of the system/network. The probabilistic resilience of graph/network was proposed in [Najjar and Gaudiot, 1990] where the probability of disconnection through random node/vertex failures was examined for a class of regular networks. They define the disconnection probability of a network G as:

$$P(G,i) = Pr\left[G \text{ disconnected exactly after } i^{th} \text{nodal failure}\right]$$

The concept of disconnection probability enables us to define probabilistic resilience. Intuitively, a resilient graph/network should sustain a large number of nodal failures until it becomes disconnected.

Let G be a network with *n* vertices/nodes. The probabilistic resilience is the largest number of vertex/nodal failures such that G is still connected with probability (1 – p):

$$PR(G,p) = \max\left\{S \mid \left[\sum_{i=1}^{S} P(G,i)\right] \leq p\right\} \qquad (H.3)$$

The normalized probabilistic resilience (NPR) relates PR(G,p) to the size of G:

$$NPR(G,p) = \frac{PR(G,p)}{n} \qquad (H.4)$$

Hence, normalized probabilistic resilience (NPR) represents the fraction of failed nodes that the graph can sustain without disconnection. Notice that disconnection here means that the remaining nodes do not form a connected graph. There is no concept of largest connected component as a measure of graph resilience. Analyzing P(G,i) for regular graphs shows that the probabilistic resilience PR(G,p) grows with the size of G. The normalized probabilistic resilience NPR(G,p) for such graphs, however, decreases with the size if the degree of the network remains constant. It is quite difficult to compute the probabilistic resilience for other complicated families of graphs and P(G,i) can only be estimated. Due to its current analytical intractability, simulation and probable heuristics are most likely approaches for empirical evaluations. The probabilistic resilience can be used to describe system disintegration under random component failure.

Applying the above procedure on a real-world aircraft engine Design Structure Matrix (DSM) with 73 components [Sinha and de Weck, 2012] (see chapter 5 for details), the distribution of probabilistic density for p = 0.1 is plotted in fig. H.4 (a) below and the associated probability of disconnection is shown in fig. H.4 (b).

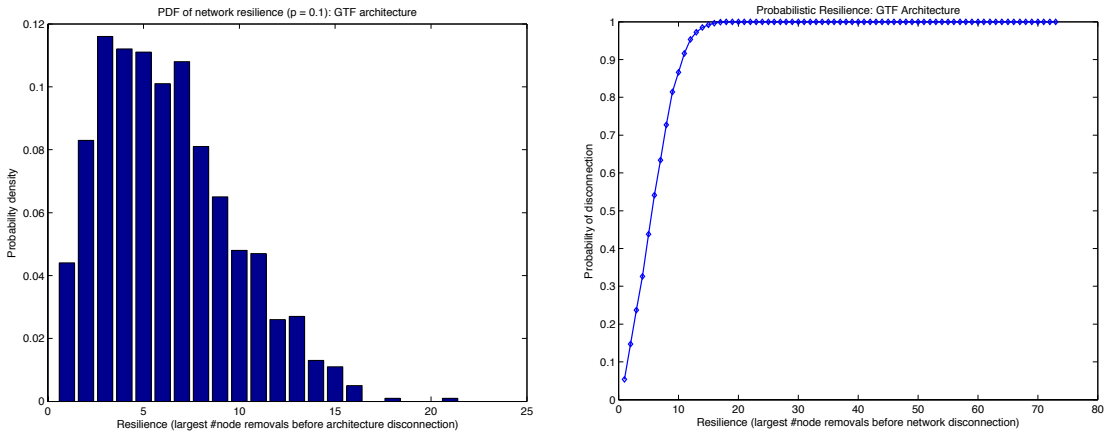


Fig. H.4: For a representative gas turbine engine architecture: (a) PDF of probabilistic resilience for *p* = 0.1, and (b) probability of disconnection as an integral of PDF.

Let us now turn our attention to effects that topological complexity might have on probabilistic resilience of the underlying graph/network. We consider *Erdos-Renyi* random graphs with 100 nodes and vary its connection density to generate graphs of varying topological complexity metric, C_3 = E(A)/n. Here E(A) stands for the graph energy of the underlying network and *n* is the number of nodes in the graph. Each point on fig. H.5 below signifies the fraction of nodal failures that can be tolerated by a network at different topological complexity levels. At each link density levels, the valued of NPR and topological complexities were averaged over 1000 instances of Erdos-Renyi random graphs with 100 nodes. At a given topological complexity level, NPR is higher if we allow a higher *p* (i.e., probability of being disconnected). For a given probability of disconnection (*p*), the NPR is negligible at lower topological complexity levels. Beyond a topological complexity level of around 2, the NPR starts to build up and keeps increasing steadily to a point (i.e., C_3 = 3.5) beyond which the graph almost impossible to disintegrate.

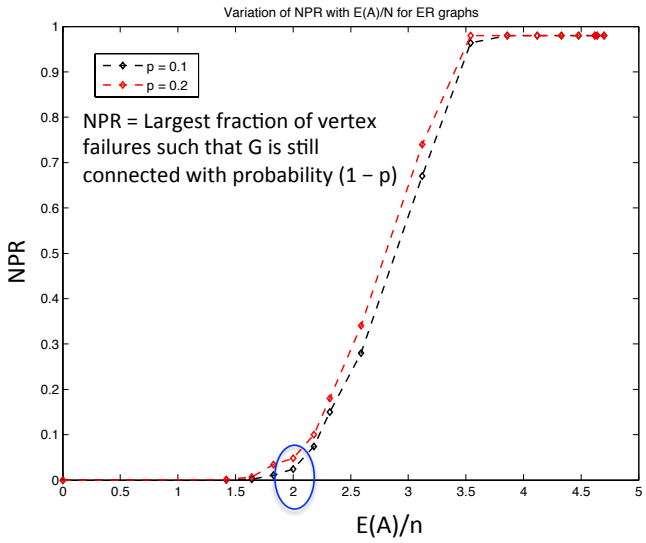


Fig. H.5: Relationship between normalized probabilistic resilience (NPR) and topological complexity $C_3 = E(A)/n$, where *n* is the number of nodes in the graph

Before reaching the topological complexity level of 2 (i.e., $E(A)/n < 2$), it takes failure of only 5% of nodes on the average to bring down or disconnect the graph. For a specified *p* value and permissible level of resilience, there is a target level of topological complexity that is essential. It appears that networked systems require a minimum complexity level to guard against network disintegration. But does the *P* point (i.e., $C_3 = E(A)/n \approx 2$) provide adequate insurance against network disintegration? From extensive simulations, it appears that topological complexity at or beyond the *P* point may provide a reasonable level of structural robustness against disintegration. On a more philosophical note, does the *P* point notionally represent the *edge of chaos* in system architecture parlance, beyond which our usual reductionist design strategies disintegrate?

These are interesting analytical observations regarding the *P* point and its significance in network design. In an empirical study by [Whitney *et al.,* 1999] using a large and diverse set of engineered products and systems showed that the average number of connections to any component (i.e., average degree) was about 6 (see fig. H.4). This is a very interesting empirical finding and backs up the analytical predictions. It appears that the *P* point might suggest an important system

architecting guideline. If the system architecture is breaching this point, it is likely that it needs to be re-visited.

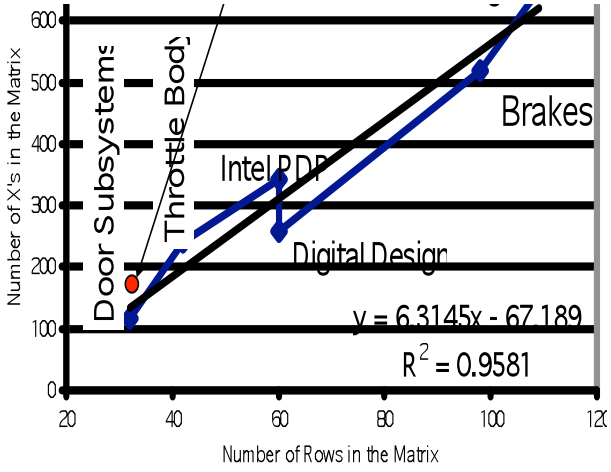


Fig. H.4: The average number of marks in engineered products and systems studied were found to be about 6 [Whitney *et al.,* 1999].

Please note that this is primarily a simulation based finding at this point and requires more theoretical and empirical work in future.

I: System-of-Systems as network of networks

Most real-world networks are not isolated. In order to function fully, they are interconnected with other networks, and this influences the dynamic processes occurring over this network-of-networks. For example, power grids are almost always coupled with communication networks. Power stations need communication nodes for control and communication nodes need power stations for electricity.

Let us consider the simpler case of two interconnected systems, represented as networks A_1 and A_2 with n and m system components respectively. These two systems are then connected with the bipartite system interconnection matrix, K. Hence the individual binary system adjacency matrices A_1 and A_2 are $(n \times n)$ and $(m \times m)$ respectively. The system interconnection matrix or the Domain Mapping Matrix (DMM), K is a binary rectangular $(n \times m)$ matrix.

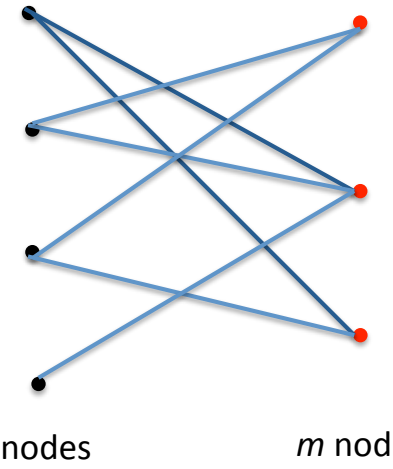


Fig. I.1: K matrix is a binary $(n \times m)$ rectangular matrix and represents a bipartite network as that connects two other networks.

The overall system-of-systems adjacency matrix Λ is a $[(n+m)\times(n+m)]$ binary square matrix written as,

$$\Lambda = \begin{bmatrix} A_1 & K \\ K^T & A_2 \end{bmatrix}$$
$$= \underbrace{\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} 0 & K \\ K^T & 0 \end{bmatrix}}_{B}$$
$$= A + B$$

Note that all the constituent sub-matrices (A_1, K, A_2) are not independent. Given matrices A_1 and K, the A_2 is obtained by unitizing the product $K^T A_1 K$. Utilizing the properties of graph energy (see chapter 3), we can write the following graph energy bounds for the system-of-systems (SoS):

$$E(A) \le E(\Lambda) \le E(A) + E(B)$$
$$\Rightarrow E(A_1) + E(A_2) \le E(\Lambda) \le E(A_1) + E(A_2) + \underbrace{E(B)}_{2E(K)}$$
$$\therefore E(A_1) + E(A_2) \le E(\Lambda) \le E(A_1) + E(A_2) + 2E(K) \qquad (I.1)$$

The graph energy of the SoS can be written as $E(\Lambda) = E(A) + \Delta = E(A) + pE(B)$ where $p \in [0,1]$. The actual value and bounds on graph energy of SoS are sketched below.

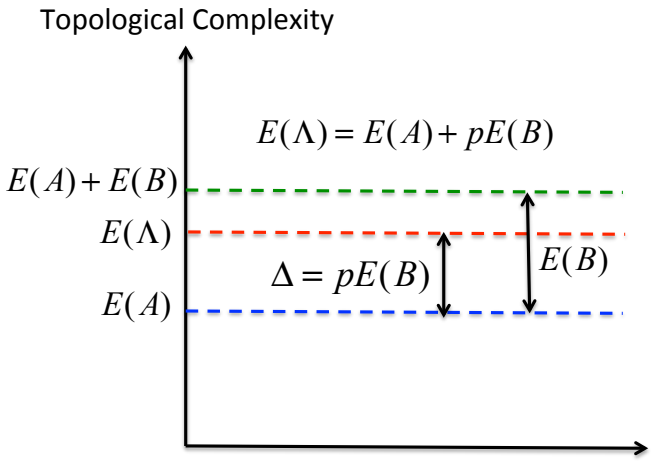


Fig. I.2: SoS graph energy and its bounds based on the graph energies of the constituent systems and their bipartite interconnection matrix with $p \in [0,1]$.

The inequality (I.1) reiterates that the whole (i.e., SoS) is more complex than just the sum of individual systems. The upper bound includes the topological complexity of the bipartite, interconnection network, in addition to the additive complexity due to the individuals systems. There has not been much academic work on investigating spectra of interconnected networks, except some recent work [Wang *et al.*, 2013] that explored the relationship between epidemics spreading and the spectral radius of the interconnected networks. It developed the spectral radius of matrix Λ in terms of matrices A and B. We are not aware of any literature that looked at the graph energy of system of systems connectivity matrix Λ in terms of matrices of constituent subsystems and the bipartite domain mapping matrices. We have explored the evolution of the graph energy of the system of systems by varying the size of second constituent system A_2, while keeping the size of the other constituent A_1. We used a known network with n = 73 nodes as A_1, randomly generated the mapping matrix K by varying m. The other constituent system has a related connectivity matrix, A_2 given by unitizing the matrix $K^T A_1 K$. The simulation results, averaged over 100 realizations of K and A_2 for different values of m, are

shown below. If m<<n, then the topological complexity of the SoS is smaller. There is a discrete jump across n=m, that is, when both the systems have one to one mapping between their components and the bipartite network interconnection matrix becomes, $B = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}_{(2n \times 2n)}$. This map to the *balanced condition* advocated by the *Axiomatic Design* philosophy [Suh, 2000].

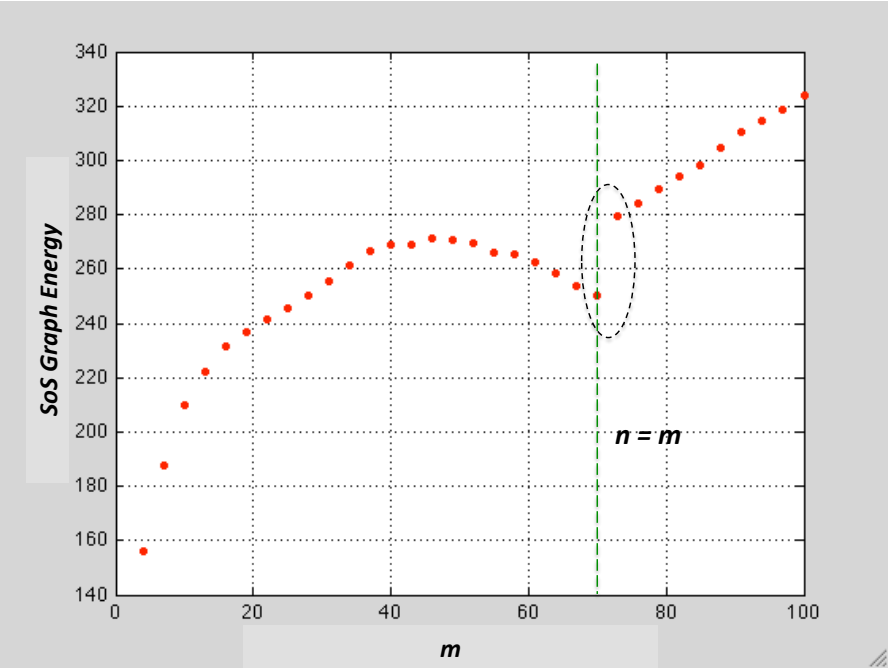


Fig. I.3: The variation in SoS graph energy, averaged over 100 realizations for each value of *m*, with increasing size of the second individual system. The size of the first system if fixed at *n* = 73 nodes. The SoS graph energy is small for m<<n, then increases steadily, but eventually decreases as $m \to n$. There is a discrete jump in SoS graph energy across *n* = *m* line (i.e., the SoS topological complexity suddenly increases as *m* > *n*).

Let us consider the two networks represent (i) the physical architecture of the system (with *n* nodes) and (ii) control architecture for that physical system (with *m* nodes) and the *K* represent the association between the system components and their associated control elements. If m<<n, then the control system could be largely centralized while still accomplishing all the system requirements. The topological complexity of such an overall system is smaller. If, on the other hand, m>n indicates that each physical component on average, requires more than one

control component for it to operate and the resulting nested complexity of the overall system gets significantly higher. The *balanced condition* yields m = n, indicating one-to-one mapping between the physical and control elements. This is a much simpler situation topologically and also from an engineering perspective and the one advocated by the *Axiomatic Design* literature, if that is otherwise feasible from the economic or functional perspectives.

J: Computation of Structural Complexity for Satellites and drills

We compute the normalized development cost and normalized structural complexity for a family of three electrical drills [Wood et al. 2001] and a set of three complex satellite systems [Wertz and Larson, 1996, Larson and Wertz, 1999, DARPA Report 2011]. The development costs and system architectural information were taken from the existing literature [Wood et al. 2001, DARPA Report 2011].

The development costs and structural complexities are normalized with respect to the respective minimum values within that product category. For example, the development costs and structural complexities of each satellite are each divided by the respective minimum values among the set of satellite programs considered. In this case, Orsted satellite had the minimum structural complexity and minimum development cost. Hence Orsted is mapped to point (1,1) on the normalized complexity vs. development cost plot (see fig. J.2), and similarly the development costs and structural complexities for the family of electric drills, the normalization was with respect to the Skill Twist drill.

In all cases, we assumed component complexities to have triangular distribution with most likely estimated being the point estimates.

For all electric drill examples, the number of components and interfaces are listed in table J.1. The topological complexity was fixed (since the existence of interfaces were fixed) and is listed in table J.1.

Table J.1: The number of system components and the interfaces in the structural graphs of the three electric drills considered. The last column lists the topological complexity for the electric drills.

Product	#Components	#Interfaces	C_3
Skill Twist	57	91	0.9
DeWalt Drill	56	134	0.95
B&D Drill	68	144	0.98

We assumed symmetric triangular distribution with most optimistic and pessimistic estimates being ± 5 percent from the most probable value (see fig. J.1). Hence, we have b/α_m as 1.05, a/α_m as 0.95 and $(a+b)=2\alpha_m$. For the electric drill examples, we use $\alpha_m=1$ for all components with b=1.05 and a=0.95. We sample the component complexities from the triangular distribution defined by the three parameters and computed the structural complexity for each sample. The procedure outlined in chapter 3 was followed for computing the distribution of structural complexities in all cases. The chosen interface factors for different interface types are listed in table J.3. Within this group of products, the structural complexities are normalized as described before.

For the satellite systems, the choice was primarily driven by availability of the relevant information about the structural graphs. Structural graphs of these satellite systems were extracted from Wertz and Larson [Wertz and Larson, 1996]. Four different types of interactions have been considered while creating the structural graph namely, matter, energy, force and information and their interface factors can be found from table L.3. For this problem, we have chosen to analyze the system in the component level of abstraction since all the chosen satellites look almost identical at subsystem level of abstraction.

For the satellite examples, the number of components and interfaces are listed in table J.2 below with their respective topological complexity (since the existence of interfaces were fixed).

Table J.2: The number of system components and the interfaces in the structural graphs of the three satellites considered. The last column lists the topological complexity for three satellite systems.

Satellite	#Components	#Interfaces	C_3
Orsted	41	58	1.08
HETE	68	88	1.14
Clementine	86	156	1.42

Table J.3: Interface factor used for different connection types for al drill examples. The interface factors were all scaled up by a factor of 2 in the satellite examples.

Connection type	Interface factor, $1/c^{(k)}$
Mechanical connection	0.05
Information/Control	0.15
Fluid flow	0.10
Energy	0.10

In all satellite examples, the inequality $(b-\alpha_m)>(\alpha_m-a)$ or $(a+b)>2\alpha_m$ was ensured while choosing the most optimistic and pessimistic values randomly. The most optimistic values *a* and the most pessimistic values *b* were set based on the ranking of point estimates of component complexities α_m. We also assumed b/α_m as 2 and a/α_m as 0.85 for the most complex component and the values were varied based on the relative ranking of the component in terms of its complexity. For the satellite examples, we assume the most likely component complexity, $\alpha_m=1$ for all components with b=2 and a=0.85.

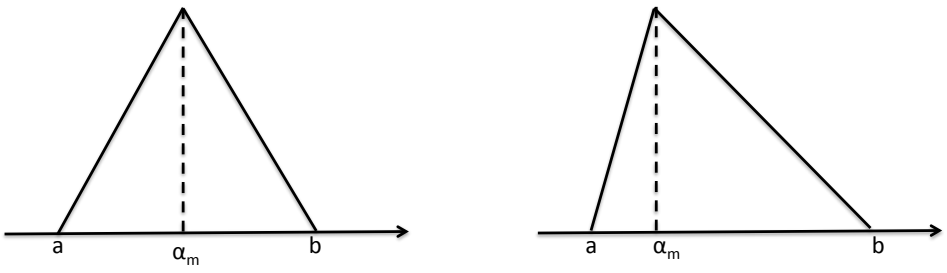


Fig. J.1: (a) Symmetric triangular pdf of component complexities used in electric drill example; (b) right-skewed triangular pdf of component complexities [Garvey 2000] in case satellites.

We compute the mean structural complexity for each system and compute the corresponding normalized mean structural complexities following the

procedure described before. Fig. J.2 shows the mean normalized structural complexity and normalized development cost for the group of electric drills and satellites respectively.

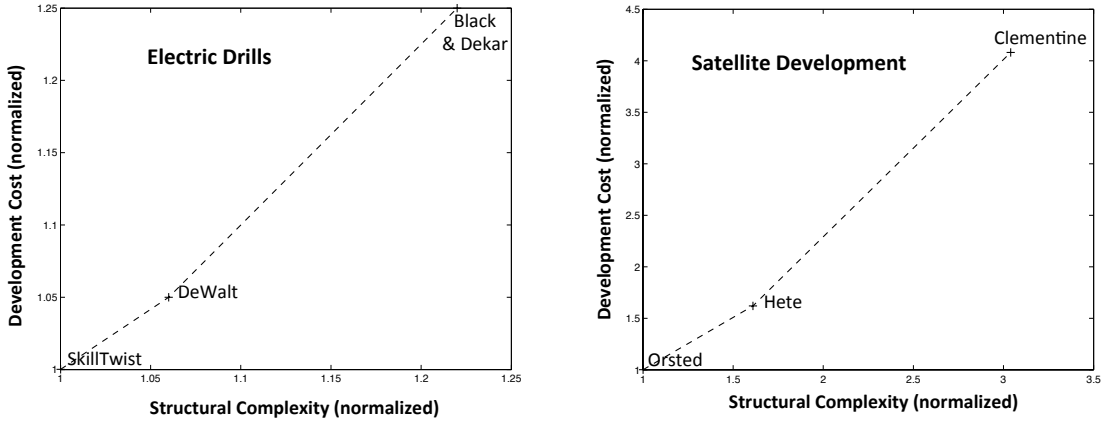


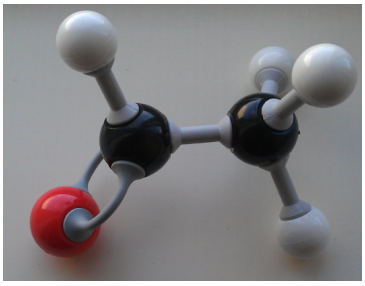
Fig. J.2: Normalized development cost and normalized structural complexity for a class of electric drills and satellites respectively.

The fig. J.2 hints at the super-linear relationship between the structural complexity and development cost, where development costs increases super-linearly with increasing structural complexity.

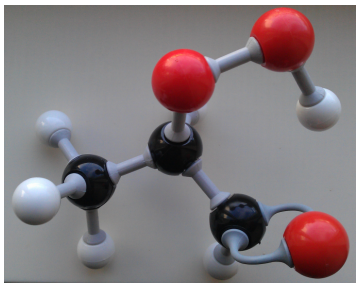
K: Experimental data from ball and stick model building exercise

We perform the ball and stick model building experiment using the molecular modeling kit from Prentice Hall [Prentice Hall Molecular toolkit, 1997], for constructing structure of organic molecules. The atoms are the components, and the bonds between them are the interfaces. Test subjects were required to correctly assemble structures given this molecular kit and a 2D picture of the structure to be built. The order of molecules was randomized for each test subject and for each molecule one would start with the entire, fully dissembled kit. Notice that this is a natural experimental setting and the idea is to mimic the real-world assembly process with the sequence in which different subjects were given the molecular structures was randomized to contain any significant learning effects.

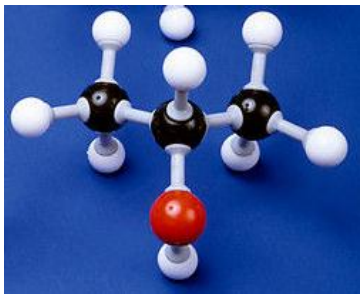
We picked a set of 12 ball and stick structures to be built by the subjects (see fig. K.1 below). They were chosen such that they spanned a reasonable spectrum of structural complexities while the expected build time is not too high. This was done keeping in view of the availability of subjects for successfully conducting the experiments.



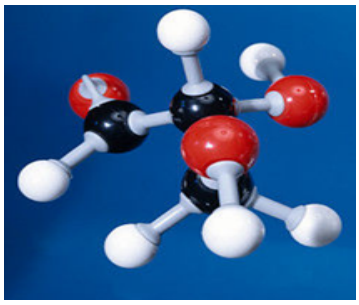
ID: 2



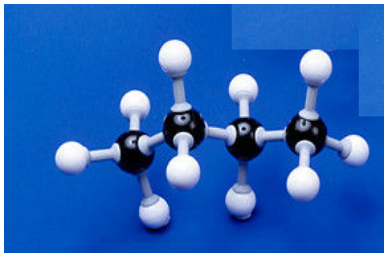
ID: 3



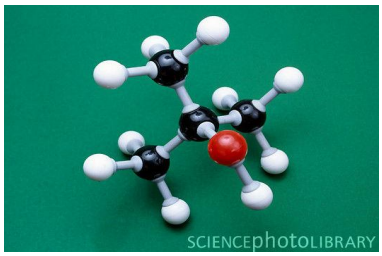
ID: 4



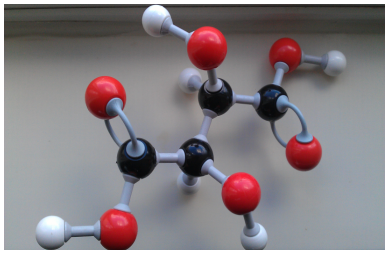
ID: 5



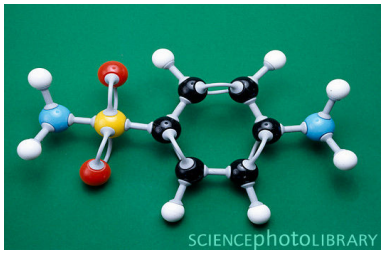
ID: 6



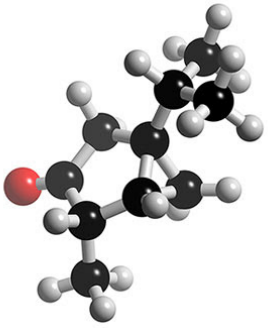
ID: 7



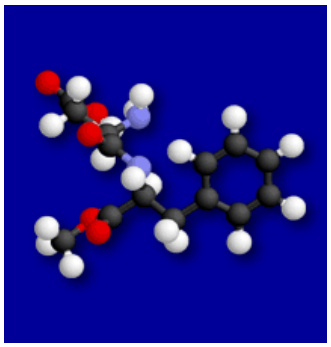
ID: 8



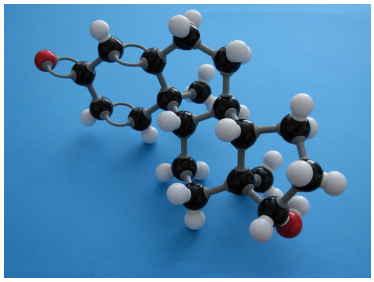
ID: 9



ID: 10



ID: 11



ID: 12

Fig. K.1: The 12 molecular structures build using a molecular kit as part of the experiment with details on number of atoms and bonds depicted in table K.1 below.

In all cases, we assumed $\alpha = 0.1$ for all atoms, $\beta = 0.1$ for all links and $\gamma = 1/n$ where *n* is the number of atoms in a given molecule (see the table K.1 below). Each molecular bond is treated as a bi-directional edge and the number of interfaces, m is twice the number of molecular bonds. Also notice that the curved double bonds are treated as a single interface between the atoms. This is because all atoms are used as is and there is no perceptible difference observed in using different bond types (i.e., curved vs. straight bonds). Please note that the number of edges is computed assuming each physical link is bi-directional. In table K.1, please note that for the most of the models, the ratio of the number of edges to the number of atoms (i.e., average degree), m/n can be expressed as $2(1-1/n)$ for most of the models and has a value close to 2. If you look at just the average degree, the molecular structures are closer to simple chains and binary trees, but their internal topological structure could be more complicated (i.e., *intricate*) in cases, leading to higher topological complexity (for example, see molecule no. 9).

Table K.1: Details of the set of 12 ball and stick structures of varying structural complexities used for the experiments.

Molecule No.	n	m	C1	C2	C3= E(A)/n	C2*C3	SC = C1 + C2*C3
1	3	4	0.3	0.4	0.94	0.38	0.68
2	7	12	0.7	1.2	1.13	1.35	2.05
3	12	22	1.2	2.2	1.13	2.48	3.68
4	12	22	1.2	2.2	1.00	2.20	3.40
5	12	22	1.2	2.2	1.27	2.80	4.00
6	14	26	1.4	2.6	0.96	2.50	3.90
7	15	28	1.5	2.8	0.97	2.70	4.20
8	16	30	1.6	3	1.40	4.21	5.81
9	19	38	1.9	3.8	1.58	6.00	7.90
10	27	56	2.7	5.6	1.08	6.05	8.75
11	39	80	3.9	8	1.12	8.96	12.86
12	46	100	4.6	10	1.19	11.92	16.52

With simple ball and stick model building experimental setup, it is easier to contain and isolate other exogenous, *confounding* factors [Mosteller and Tukey, 1977]. We track the total build time for each structure as the observable representing system development effort. Any incorrect assembly involves rework and leads to increasing total assembly time.

The experimental setup is described below:

- For the experiment we choose 12 different structures to assemble. They were all based on real molecules but that had no importance for our experiment. Photographs were taken of the assembled structures from angles such that the topology of the molecules was visible.

- The experimental subject received an initial briefing including the explanation of what will be expected from him. Before before started, they were shown the molecular kit to familiarize themselves with the components and the two different kinds of bonds, rigid and flexible.

- The subjects were given the completely unassembled kit and were shown a picture of a molecule. Our test subjects consisted of 17 student volunteers with largely similar backgrounds to keep the sample as homogeneous as

possible. To contain any influence of the *learning effect*, the order of molecules was randomized.

- The volunteers were asked to assemble these structures as quickly as possible and without error. The total build time, $T_{total} = T_{cognition} + T_{construction} + T_{rework}$ was recorded. We focused only on the total build time T_{total} and not its individual constituents. When completed, the structure will be unassembled and then the next picture was shown to the subject under study.

This experimental setup helps isolate the effect of structural complexity on the system development cost/effort since components of dynamic and organizational complexities are not present here. This helps us capture the effect of structural complexity on development cost/effort by using a simple, single variable parametric model. In order to smooth out the individual differences, we consider the averaged build time for the group of 17 subjects for each ball and stick structure.

From table K.2 below, notice that for almost all cases (except molecule ID 6), we have mean build time > median build time. This is indicative of the long-tailed distribution for the build time. This phenomenon is shown graphically in fig. K.2 below.

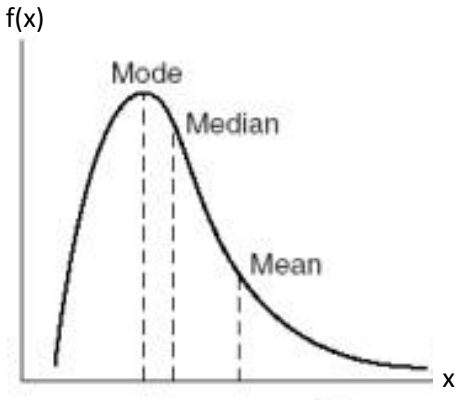


Fig. K.2: Typical long-tailed probability distribution with finite instances of very high values of the variable under consideration.

Table K.2: The Structural Complexity and the build time statistics for 17 test subjects who build the set of 12 ball and stick structures.

Molecule No.	Structural Complexity	Max. build time (sec.)	Min. build time (sec.)	Mean build time (sec.)	Median build time (sec.)	Build time Standard Deviation (sec.)
1	0.68	18	8	12	11	2.97
2	2.05	98	34	54	52	17.93
3	3.68	216	77	113	95	41.55
4	3.4	205	47	96	89	40.83
5	4.0	276	63	126	109	53.16
6	3.9	135	66	100	102	36.41
7	4.2	187	78	117	103	42.61
8	5.81	387	86	181	161	76.47
9	7.9	543	159	300	275	121.17
10	8.75	775	215	427	381	178.74
11	12.86	1140	324	605	552	279.28
12	16.52	1496	350	932	922	356.88

This observation indicates that there were test subjects who took much longer to correctly build the ball and stick models, thereby driving the mean build time higher. Regression-based parametric model development resulted in power-law type model (i.e., $Y = aX^b$) as the best functional form linking build time to the structural complexity meric.

The ball and stick experiment is a very simple system building exercise with system performance aspects being taken out of the equation. This exercise is about building structural forms with given blue print. The ball and stick experiment is a simplified experimental setup without additional constraints that appear in large engineered system development efforts. Below we list the primary result from this experiment and subsequent claim

Primary Result: The average build time, Y varies super-linearly with structural complexity, X. The best parametric relationship is found to be, where {a=14.68, b=1.4775} represent the model parameters for the experiments conducted using 17 subjects, building 12 ball and stick models. We assumed component complexity, $\alpha = 0.1$ for all component (i.e., each atom) and interface complexity, $\beta = 0.1$ for all interfaces (i.e., bonds).

Claim: We claim generalizability of the functional form found above and that the super-linear form dependence with exponent b>1 will hold for large, real-world engineered systems.

Discussion: We have seen that the development effort (i.e., average build time used as a proxy for development effort) increases super-linearly with structural complexity. An important issue to consider is whether the structural complexity metric has some inherent bias towards super-linearity or that the super-linearity is primarily due to inherent human cognitive mechanisms. Let us look into the possibility of inherent bias in the structural complexity metric in the light of ball and stick experiment. We used the experimental data and apply the jackknifing technique that incorporates cross-validation of the model in all cases. We analyzed the individual attributes that constitute the structural complexity metric and in all cases the power-law functional form yield the best model.

The graph energy is found to vary *almost* linearly with the number of atoms with the exponent, b=0.999 suggesting that the relationship is almost linear for the set of ball and stick models considered during the experimentation. With C_2C_3 as the dependent variable (using component and interface complexities as 0.1), we find the exponent, b=1.123 indicateing a mildly super-linear behavior. The structural complexity, C is also shows an almost linear or mildly super-linear variation with the number of atoms (see table below).

Table K.3: Parametric relationships between number of atoms (n), structural complexity metric and their components for the ball and stick experiment.

Y	X	Parametric Model	Parameters {a,b}	Model Quality
Graph Energy	n	$Y=aX^b$	{1.143, 0.999}	[R^2=0.96; MMRE=0.12; PRED(0.25)=0.9167]
C_2C_3	n	$Y=aX^b$	{0.15, 1.123}	[R^2=0.96; MMRE=0.13; PRED(0.25)=0.9167]
C	n	$Y=aX^b$	{0.248, 1.08}	[R^2=0.98; MMRE=0.09; PRED(0.25)=1.0]

Now let us compare the functional relationships between structural complexity metric and the number of atoms and also between structural complexity metric and mean build time, as recorded from the ball and stick experiments:

Table K.4: Parametric relationships between (i) structural complexity metric and the number of atoms (n); (ii) mean build time and structural complexity metric for the ball and stick experiment.

Y	X	Parametric Model	Parameters {a,b}
C	n	$Y=aX^b$	{0.248, 1.08}
Mean build time (sec.), t	C	$Y=aX^b$	{14.68, 1.4775}

The degree of super-linearity in build time-structural complexity relationship is much stronger with about 40% higher value of the exponent (see table K.4 above). The architecture of organic molecules are locally centralized around carbon atoms and an order analysis shows the structural complexity metric is of the order of number of atoms: $C = O(n)$ (see chapter 3), and the relationship above corroborates the theoretical prediction. It would be interesting to observe the development effort for systems with distributed architecture where $C = O(n^{2.5})$. For such systems, we predict significant system integration efforts and the development effort will likely vary as, $t = O(n^{\phi})$ where $\phi \gg 2.5$. Such findings will further strengthen the argument for human cognitive limitation that leads to usually observed super-linear relationship between development effort and structural complexity for engineered complex system development.

Results indicate that the super-linearity observed in the build time – structural complexity relationship is not due to inherent bias in the structural complexity metric. The reason for this super-linear behavior is likely to be embedded in the human cognitive capability and this aspect remains to be explored in future by combining with cognitive scientists.

Existence of sub-linear behavior (i.e., b < 1) implies that enhancement in cognitive capability outstrips growth in structural complexity. In theory, it is possible to come up with component and interface complexity numbers that will show linear (or even sub-linear) behavior. The situations under which we might have b<1 is highly unlikely in practice and might be due to some fundamental limitation of human cognition (as commented by cognitive psychologists). Future

developments on cognitive models in the context of engineered complex system development efforts might throw light on this observed super-linear behavior and if there exists conditions under which the cognitive ability outstrips the level of increased complexity, leading to sub-linear model with exponent, b<1.

L: Complexity budget and complexity-performance trade-space

As described in chapter 6, the system performance measure (note that it is usually a composite measure, incorporating different aspects of system performance), P and the Non-Recurring Engineering cost/effort (NRE) are modeled as,

$$P = P_{\max}\left(\frac{kC^n}{1+kC^n}\right) \qquad (L.1)$$

$$NRE = aC^m \qquad (L.2)$$

where C stands for complexity, P_{max} stands for limiting performance achievable and (n,m,k,a) are the parameters that are usually determined from historical data or through expert opinion.

The value function for the system architecture, V is defined as,

$$V = \frac{P}{NRE} = \underbrace{P_{\max}\left(\frac{k}{a}\right)}_{S}\left[\frac{C^{(n-m)}}{1+kC^n}\right]$$

$$= S\left[\frac{C^{(n-m)}}{1+kC^n}\right] \qquad (L.3)$$

Hence the value is basically in terms of performance per unit system development cost/effort. In eq. L.3, please note that S is a constant and since complexity, C is a number greater than one, the value function, V decreases monotonically if $n \le m$. The condition $n > m$ yields interesting insights and leads to an optimal value function for the system architecture due to the trade-off between the two rate parameters (n,m). We assume that the limiting performance P_{max} is achievable before NRE becomes unbounded and $n > m$. In this case, for value optimality, we must have,

$$\frac{dV}{dC} = 0$$

$$\Rightarrow \frac{n-m}{C_*} = \frac{knC_*^{\,n-1}}{1+kC_*^n}$$

$$\Rightarrow C_*^n = \frac{n-m}{km}$$

$$\therefore C_*^n = \frac{\left(\frac{n}{m}\right)-1}{k} \qquad (L.4)$$

To check if this is a maximal solution, we compute the second derivative. After tedious algebraic simplifications, using the symbolic mathematics package *Mathematica*™, we arrive at the following expression for the second derivative at $C = C_*$,

$$\left.\frac{d^2V}{dC^2}\right|_{C_*} = \frac{-Sm^2(n-m)}{n} C_*^{n-m-2} \qquad (L.5)$$

For $n > m$, we have $\left.\frac{d^2V}{dC^2}\right|_{C_*} < 0$ and V is maximized at $C = C_*$. The corresponding expression for V_{max} is given by,

$$V_{\max} = \frac{SC_*^{n-m}}{1+kC_*^n}$$

$$= \left(\frac{m}{n}\right) SC_*^{n-m}$$

$$= \left(\frac{m}{n}\right) S\left[\frac{\left(\frac{n}{m}\right)-1}{k}\right]^{\left(1-\frac{m}{n}\right)} \qquad (L.6)$$

Notice that the above quantities defined in L.4 and L.6 exists iff $n > m$.

Otherwise the value function, V is monotonically decreasing with respect to complexity (see fig. L.1).

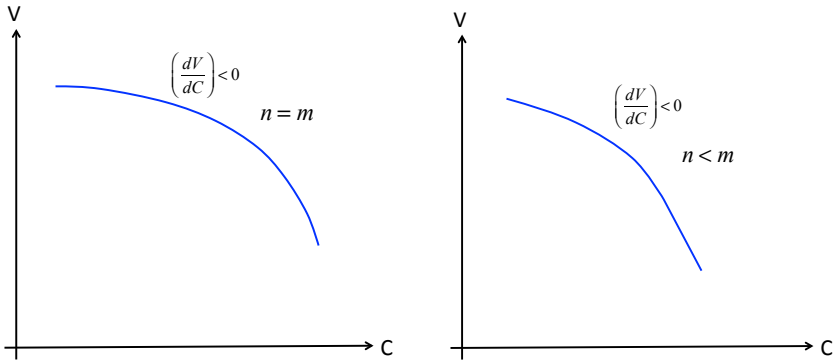


Fig. L.1: Value function, V is monotonically decreasing with respect to complexity, C if $n \leq m$.

Hence, a higher value of n/m ratio and a lower k leads to a more prominent optimal value, V_{max}. The role of lower value of k leads to a shift in the performance-complexity profile and for a given exponent n, requires greater complexity to reach a given target performance as shown in fig. L.2.

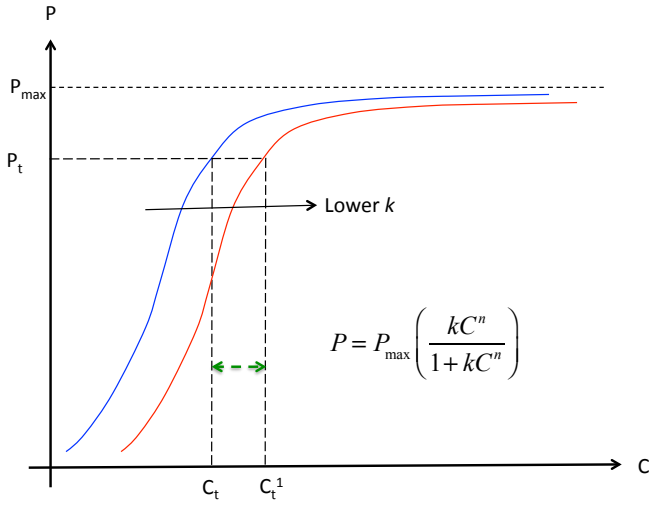


Fig. L.2: Effect of lower k value is to shift the performance-complexity curve to the right. The same level of performance target now requires higher complexity to be *spent*.

A larger value of n signifies quicker saturation of performance at a lower level of complexity. In other words, higher complexity does not buy us increased performance, but rather hurts due to increased NRE or cost/effort. A higher n/m ratio can be thought of as rates of performance improvement and NRE penalty respectively, with increase in complexity. Since performance improvement is

bounded while NRE is not, the value function improves initially for high n/m ratio and then starts to drop as performance improvement saturates.

Now, let us look at the optimal value for different n/m ratios, assuming $n > m$. Let us define,

$$r = \frac{n}{m}$$

We can write the optimal value from L.6 as,

$$V_{\max} = \frac{S}{r}\left(\frac{r-1}{k}\right)^{1-\frac{1}{r}}$$

For two different values of the ratio (r_1, r_2), we can express the ratio of the corresponding optimal values as,

$$\begin{aligned}\frac{V_{\max,2}}{V_{\max,1}} &= \left(\frac{r_1}{r_2}\right)\left(\frac{r_2-1}{k}\right)^{1-\frac{1}{r_2}}\left(\frac{k}{r_1-1}\right)^{1-\frac{1}{r_1}} \\ &= \frac{\left(1-\frac{1}{r_2}\right)}{\left(1-\frac{1}{r_1}\right)}\left(\frac{1}{k}\right)^{\left(\frac{1}{r_1}-\frac{1}{r_2}\right)}\frac{\left(r_1-1\right)^{\frac{1}{r_1}}}{\left(r_2-1\right)^{\frac{1}{r_2}}} \qquad (L.7)\end{aligned}$$

This expression is rather cumbersome to generalize easily and the ratio of optimal value function depends on the specific values of the triple (r_1, r_2, k). In general, a larger n/m ratio and smaller $k < 1$ leads to a larger $V_{\max}$ value.

M: Inferring adjacency matrix from system behavioral data

A significant challenge to inferring networks is that, both direct and indirect dependencies arise together [Feizi et al., 2013]. While recognizing direct relationships between variables connected in a network from observed data alone we have to overcome the problem of correlation-based networks containing numerous indirect relationships.

To segregate direct dependencies from an observed correlation matrix containing both direct and indirect effects, [Feizi *et al*., 2013] proposed a general method based on network de-convolution. This method filters out the combined effect of all indirect paths of arbitrary length exploiting matrix decomposition. Given an observed matrix of correlations that might contain both direct and indirect/longer range dependencies, the process of network de-convolution recovers the original adjacency matrix that gave rise to the observed correlation matrix.

Basic premise of this method is that all transitive/longer range dependencies can be computed by summing this adjacency matrix and all its powers, which convolves all direct and indirect paths at all lengths. For a weighted network where edge weights represent the correlation strength (or mutual information) relating two elements in the network, this method seeks to recognize the fraction of the weight of each edge attributable to direct versus indirect contributions. Here, we describe the basic methodology and algorithmic implementation in the context of inferring the adjacency matrix from behavioral data. For details on robustness of the method to noise and applicability to general asymmetric matrices, please refer to [Feizi *et al.,* 2013].

Suppose G_{obs} represents the $(n \times n)$ matrix of observed dependencies: an appropriately scaled matrix linking variables dependencies (nodes in the network) with self-loops being excluded by setting its diagonal components to zero. This appropriate scaling refers to a linear scaling of the input correlation matrix, based on the largest absolute eigenvalue (or singular values for symmetric matrices) of the un-scaled correlation matrix. The observed dependency matrix G_{obs} captures both direct and indirect effects: $G_{obs} = G + G_{indir}$, where G stands for the direct dependency (i.e., length 1) matrix and $G_{indir} = G^2 + G^3 + \ldots + G^k + \ldots G^{n-1}$ with G^k referring to indirect dependencies of length k .

Assuming the size of the system to be large enough (i.e., $n \to \infty$), we can approximate the sum by the following infinite series:

$$G_{obs} = G + G^2 + \ldots + G^k + \ldots$$
$$= G(I - G)^{-1} \qquad (M.1)$$

Note that, the observed dependency matrix is linearly scaled so that the largest absolute eigenvalue of *G* is smaller than 1 (i.e., $|\lambda_k(G)| < 1 \ \forall k$) and eq. M.1 holds. Therefore, the effects of indirect dependencies decrease exponentially with the length of indirect paths. From relation M.1, we can express the direct (i.e., length 1) dependency in terms of the linear scaled observed dependency matrix G_{obs} :

$$G_{obs} = G(I - G)^{-1}$$
$$\therefore G = G_{obs}(I - G)$$
$$\Rightarrow G = G_{obs}(I + G_{obs})^{-1} \qquad (M.2)$$

Since the correlation matrices are symmetric by definition, the observed dependency matrix, being a scaled version of the correlation matrix, have real eigenvalues and singular values given by the magnitude of the eigenvalues. It can be decomposed to its eigenvalues and eigenvectors as $G_{obs} = U\Sigma_{obs}U^T$ where *U* and Σ_{obs} represents the matrix of eigenvectors and a diagonal matrix of eigenvalues of matrix G_{obs} respectively.

Based on the eq. M.2, the direct dependency matrix can be written as $G = U\Sigma U^T$ where the individual eigenvalues are related by:

$$\lambda_k(G) = \frac{\lambda_k(G_{obs})}{1 + \lambda_k(G_{obs})} \quad \forall 1 \le k \le n \qquad (M.3)$$

Note that eq. M.3 is only valid if $|\lambda_k(G)| < 1 \ \forall k$.

This condition can be satisfied by linearly scaling the un-scaled observed correlation matrix, $\hat{G}$ using a linear scaling factor α such that, $G_{obs} = \alpha\hat{G}$. The eigenvalues of G_{obs} and $\hat{G}$ are related by $\lambda_k(G_{obs}) = \alpha\lambda_k(\hat{G}) \ \forall k$. Using this in M.3, the eigenvalues of G and $\hat{G}$ are related as:

$$\lambda_k(G) = \frac{\lambda_k(\hat{G})}{1/\alpha + \lambda_k(\hat{G})} \quad \forall 1 \le k \le n \qquad (M.4)$$

We have to choose the scale factor $\alpha > 0$ such that the condition $|\lambda_k(G)| < 1 \; \forall k$ is satisfied. Since the correlation matrix is symmetric, its singular values σ_k are related to its eigenvalues as $\sigma_k = |\lambda_k| \; \forall k$. Let us define the largest absolute eigenvalue (i.e., the largest singular value) of the direct dependency matrix G as β. Since the diagonal elements of the correlation matrix $\hat{G}$ are set to zero, its smallest eigenvalue is negative and we have $\sigma_n(\hat{G}) = -\lambda_n(\hat{G})$. The same is true for direct dependency matrix G. For the largest absolute eigenvalue, β to be less than 1, we should have:

$$\frac{\sigma_1(\hat{G})}{1/\alpha + \sigma_1(\hat{G})} \le \beta < 1 \Rightarrow \frac{1}{\alpha} \ge \left(\frac{1-\beta}{\beta}\right)\sigma_1(\hat{G}) \qquad (M.5)$$

and,

$$\frac{\sigma_n(\hat{G})}{1/\alpha - \sigma_n(\hat{G})} \le \beta < 1 \Rightarrow \frac{1}{\alpha} \ge \left(\frac{1+\beta}{\beta}\right)\sigma_n(\hat{G}) \qquad (M.6)$$

Combining eq. M.5 and eq. M.6, we have

$$\frac{1}{\alpha} \ge \max\left[\left(\frac{1-\beta}{\beta}\right)\sigma_1(\hat{G});\; \left(\frac{1+\beta}{\beta}\right)\sigma_n(\hat{G})\right] \qquad (M.7)$$

We use a scaling factor $\hat{\alpha}$ given as:

$$\frac{1}{\hat{\alpha}} = \max\left[\left(\frac{1-\beta}{\beta}\right)\sigma_1(\hat{G}); \left(\frac{1+\beta}{\beta}\right)\sigma_n(\hat{G})\right] \qquad (M.8)$$

and the eigenvalues of the direct dependency matrix are given as:

$$\lambda_k(G) = \frac{\lambda_k(\hat{G})}{1/\hat{\alpha} + \lambda_k(\hat{G})} \quad \forall 1 \le k \le n \qquad (M.9)$$

The direct dependency matrix is given by:

$$G = U \begin{pmatrix} \lambda_1(G) & . & . & 0 \\ . & . & & . \\ . & & . & . \\ 0 & . & . & \lambda_n(G) \end{pmatrix} U^T \qquad (M.10)$$

If β is small, higher order interactions play insignificant roles in observed dependencies since they decay proportionally to β^k where *k* is the order of indirect interaction. Choosing β close to one (i.e., considering higher order indirect interactions) leads to higher performances in all de-convolution applications where we assume that significant indirect, higher order interactions effects do manifest in our observations. Based on our simulations, we recommend $\beta \geq 0.8$ for most applications in complex dynamical systems.

The binary adjacency matrix, A is computed by unitization operation:

$$a(i,j) = \begin{cases} 1 \; if \; g(i,j) > \varepsilon \\ 0 \quad otherwise \end{cases}$$

where we use $\varepsilon = 10^{-3}$ as the threshold parameter. All computations performed in chapter 7 followed the basic procedure of network de-convolution applied to the observed correlation matrix and used the parameter values suggested herein.