SYM-AM-17-067

# Proceedings
## of the
# Fourteenth Annual Acquisition Research Symposium

## Wednesday Sessions
## Volume I

**Acquisition Research:**
**Creating Synergy for Informed Change**

**April 26–27, 2017**

**Published March 31, 2017**



Prepared for the Naval Postgraduate School, Monterey, CA 93943.

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School

# A Systems Complexity-Based Assessment of Risk in Acquisition and Development Programs

**Antonio Pugliese—**is a PhD candidate in systems engineering at Stevens Institute of Technology in Hoboken, NJ. He received his BS and MS in aerospace engineering and a postgraduate master in systems engineering from the University of Naples Federico II in Italy. His doctoral research is on structural complexity metrics for cyberphysical systems.

**Roshanak Rose Nilchiani—**is an Associate Professor of systems engineering at the Stevens Institute of Technology. She received her BS in mechanical engineering from Sharif University of Technology, MS in engineering mechanics from University of Nebraska-Lincoln, and a PhD in aerospace systems from Massachusetts Institute of Technology. Her research focuses on computational modeling of complexity and systemalities for space systems and other engineering systems, including the relationship between system complexity, uncertainty, emergence, and risk. The other track of her current research focuses on quantifying, measuring, and embedding resilience and sustainability in large-scale critical infrastructure systems.

## Abstract

Development and acquisition efforts of cyberphysical systems can often encounter cost or schedule overruns due to the complexity of the system. It has been shown that a certain amount of system complexity is related to the system functionalities (effective complexity), whereas excessive complexity is related to unnecessary intricacies in the design (apparent complexity). While the former is necessary, the latter can be removed through precise local redesign. One of the major challenges of systems engineering today is the development of tools, quantitative measures, and models for the identification of apparent complexity within the system.

This research has the goal of evaluating and measuring the structural complexity of the engineered system, and does it through the analysis of its graph representation. The concepts of graph energy and other spectral invariant quantities allow for the definition of an innovative complexity metric. This metric can be applied knowing the design of the system, to understand which areas are more in need of redesign so that the apparent complexity can be reduced.

## Introduction

Complexity is one of the hallmarks of all engineered systems, specially a prominent feature of defense acquisition programs. Complex engineered systems are continuously exposed to various types of uncertainties, risks, and failures in their life cycle. The causes of failures and risks are either a known observed phenomenon and perhaps overlooked in the development phase, or it is a new type of failure. The former case can lead to improvements in engineering design and management and systems engineering processes of a complex system. The latter case, instead, can potentially provide useful information that can be obtained only through unfolding of these types of failures and events. Complex engineered systems design effort resides partially in the domain of known risks and uncertainties. This domain, also known as the domain of complicated systems, is characterized by known unknowns which can be addressed with time and effort, through theoretical and experimental research. This means that systems in this domain can express large epistemological emergence, given to the lack of knowledge, but a low ontological one, meaning that the knowledge can be obtained.

However, the engineered systems with high levels of ontological emergence, meaning that the system under study is so far from the current level of knowledge, show low levels of predictability in behavior. Complex engineered systems involve humans with

certain levels of autonomy interacting with the engineered systems. Predicting the behavior of a complex system, characterized emergent phenomena is very challenging. To reduce various risks in design and operation of an acquisition program, systems engineers have attempted to operate within the domain of knowable risks as much as possible, through mitigation and exploitation techniques such as trade space exploration, modular designs, open architectures, redundancy, verification, and testing. However, a comprehensive, applied, and formal way to measure and capture various dimensions of complexity and risks in the life cycle of a complex engineered system or an acquisition program is lacking. In past years, the systems engineering research community has introduced some measures of complexity for engineering design; however, domain dependence and limitation in universality of use of these measures has seriously limited their use in the decision-making process. In this paper, we introduce a constructed measure of complexity that has carefully tried to address as well as to avoid many shortcomings of existing quantitative measures of complexity.

This paper begins with a literature review on state of the art complexity and emergence. The literature review also covers some existing measures of complexity and their merits as well as shortcomings and limitations. The paper continues to introduce the concepts of spectral theory of systems complexity and explains matrix energy and directed edges in our suggested complexity measure. The paper concludes by analyzing the results and sets the stage for the future work of various case studies, quantitatively connecting emergence to spectral complexity measures of an engineered system or an acquisition program.

## Literature Review

### *What Is Complexity?*

#### *Three Types of Problems*

The first hint to the role of complexity in science and engineering design has been given by Weaver (1948). He described three distinct types of problems: problems of simplicity, problems of disorganized complexity, and problems of organized complexity. Problems of simplicity are the problems with a low number of variables that have been tackled in the nineteenth century. An example is the classical Newtonian mechanics, where the motion of a body can be described with differential equations in three dimensions. In these problems, the behavior of the system is predicted by integrating equations that describe the behavior of its components. Problems of disorganized complexity are the ones with a very large number of variables that have been tackled in the twentieth century. The most immediate example is the motion of gas particles, or as an analogy, the motion of a million balls rolling on a billiard table. The statistical methods developed are applicable when particles behave in an unorganized way and their interaction is limited to the time they touch each other—which is very short. In these problems, it has been possible to describe the behavior of the system without looking at its components or the interaction among them. Problems of organized complexity are the ones that are to be tackled in the twenty-first century, and the ones that see many variables showing the feature of organization. These problems have variables that are closely interrelated and influence each other dynamically. This high level of interaction that gives rise to organization is the reason these problems cannot be solved easily. Weaver (1948) described them as solvable with the help of powerful calculators, but today's technology is not yet able yet to solve the most complex of these problems. These are the problems that nowadays we define as "complex." Predicting the behavior of a system with many interconnected parts changing their behavior in line with

the state of other components is a problem of organized complexity, and the system itself is referred to as a complex system.

### *The Point of View of the Observer*

On his blog, Rouse (2016) wrote about the absolute or relative nature of complexity. To illustrate his stand on the matter, he considers two uses for a Boeing 747: as a paperweight and as an airliner. This thought exercise allows us to understand that the 747 as a paperweight is not very complex. It does a perfect job, given its large mass, but carries no complexity in its operation. On the other side, the airliner function exposes all the operational difficulties of flying and maintaining an airplane. From this example, he concludes that complexity should be defined in terms of a relationship between the entity and an observer. Thus, complexity is relative to the point of view of the observer.

Wade and Heydari (2014) categorized complexity definition into three major groups, according to the point of view of the observer. When the observer is external to the system and can only interact with it as a black box, then the type of complexity that can be measured is called behavioral complexity, since it looks at the overall behavior of the system. When the observer has access to the internal structure of the system, such as blueprints and source code for engineered systems, or scientific knowledge for natural systems, then the structural complexity of the system is the one being measured. If the process of constructing the entity is under observation, then the constructive complexity is to be measured, which is the complexity of the building process. This definition relates complexity to the difficulty of determining the output of the system.

Fischi built a framework for the measurement of dynamic complexity entirely based on the role of the observer (Fischi, Nilchiani, & Wade, 2015). The definition of complexity used in this framework is based on the system being observed, the capabilities of the observer, and the behavior that the observer is trying to predict.

### *Complexity and Emergence*

Often complex systems have behaviors that cannot be immediately explained, and for this reason complexity is associated with the concept of emergence. As defined by Checkland (1981), emergence is “the principle that entities exhibit properties which are meaningful only when attributed to the whole, not to its parts.” In other words, an emergent phenomenon is a phenomenon at the macro-level that was not hard-coded at the micro-level (Page, 1999), and which can be described independently from the underlying phenomena that caused it (Abbott, 2006).

Both natural and engineered systems are capable of expressing emergence. One example of emergence in natural system is wetness. Water molecules can be arranged in three different phases (i.e., solid, liquid, and gas), but only one of them expresses a certain type of behavior—that is, high adherence to surfaces. This behavior is due to the intermolecular hydrogen bonds that affect the surface tension of water drops. These bonds are also active in the solid and liquid phase, but in those cases, they are either too strong or too weak to generate wetness. As we will see for many systems, some properties emerge only when conditions are just right. In engineered systems, the system requirements and software specifications are supposed to be written in such a way that they are independent from their implementation. For this reason, the functions and properties they describe are emergent (Abbott, 2006).

These definitions of emergence often do not differentiate on whether the emergent property is expected or unexpected, and this is obvious, since not every system has a designer that is putting together components to generate the system, and therefore

sometimes there is no one to expect the property. Natural systems, which are created through evolution, do not justify the classification of emergence into expected and unexpected. In engineered systems, this is of course not true. The system engineer is responsible for the identification of the properties of the system in relation to its environment. Not only the operational environment, but also assembly, integration, testing, and disposal environment. In the design process, it is customary to differentiate between the attributes of the system that are wanted, and therefore expected, and the ones that are unexpected, which can be beneficial or adverse.

*Two Types of Emergence*

The works of Chalmers (2008), Bedau (1997), and Kauffman (2007) identify differences between two types of emergence: epistemological and ontological.

Kauffman proposes two approaches to the nature of emergence. The reductionist approach sees emergence as epistemological, meaning that the knowledge about the systems is not yet adequate to describe the emergent phenomenon, but it can improve and explain it in future. This is the case of wetness, where knowledge about molecules and intermolecular interactions can explain the emergent phenomenon. On the other hand, there is the ontological emergence approach, which says that "not only we don't know that will happen, [but] we don't even know what can happen," meaning that there is a gap to fill not only about the outcome of an experiment (or process), but also about all the possible outcomes (Kauffman, 2007). Ontological emergence is given by the enormous amount of states the system could evolve into. The evolution of the swimming bladder in fish is an example of ontological emergence (Longo, Montévil, & Kauffman, 2012). An organ that gives neutral buoyancy in the water column as its main function also enables the evolution of some kinds of worms and bacteria that will live in it. Ontological (or radical) emergence is given by the enormous amount of states the system could evolve into. In these cases, we not only are not able to predict which state will happen, but not even what are the possible states.

Chalmers (2008) provides definitions for two different types of emergence, weak and strong, based on the capabilities of the observer. At the lower level, there is weak emergence, which includes any property possessed by the whole and not its parts. A chair is an example of weak emergence since the property of allowing someone to sit is present in the whole but not in its parts. At the upper level, there is strong emergence with the example of consciousness. In the case of weak emergence, the emergent phenomenon is just unexpected, while in the case of strong emergence, it is completely non-deducible. This of course depends on the capabilities of the observer in linking the phenomena at the two levels. Chalmers, being a philosopher and cognitive scientist, implicitly assumes that the observer has the knowledge and capabilities of a human being. An example that he provides to illustrate the difference between weak and strong emergence is the high-level patterns in cellular automata. These patterns are unexpected but deducible just by looking at the low-level rules of the automaton, making them weakly emergent and not strongly emergent. The only example that Chalmers provides of strong emergence is consciousness, and he goes along to state that there is no other such phenomenon other than the ones in which the strong emergence derives "wholly from a dependence on the strongly emergent phenomena of consciousness." Thus, the way of differentiating between a system with weak or strong emergence is to look for conscious elements within the system.

***Complexity and Complication***

The idea that complexity also depends on the tools and knowledge available to the observer is common to many researchers. Crawley, Cameron, and Selva (2015) use the

concepts of essential and apparent complexity to make a distinction between complexity and complication. Being engineers, they only consider designed systems. Essential complexity comes from functionality and represents the minimum amount of complexity required for the desired functionalities to emerge. Apparent complexity, on the other hand, represents the unnecessary intricacies that a designed system can have. These are the architectural features that are not required from the functionality, which make the design complicated and hard to understand. The role of the system architect is to minimize the apparent complexity without affecting the essential one.

Evolved systems do not have functions, but they create advantages to their stakeholders. The presence of a heart in the cardiovascular system of many animals is only explained by the advantage it creates in the distribution of resources within the organism. The heart does various things, such as pumping blood and making the characteristic heart sound. The categorization of these behaviors into functions and side effects has no meaning in evolved systems, and therefore the concepts of essential and apparent complexity lose their validity (Longo, Montévil, & Kauffman, 2012). On the other side, the work of Chaisson suggests that evolution keeps the apparent complexity as low as possible by selecting the unfit organisms. This implies that the complexity of an evolved organism, which has thrived for a substantial amount of time in its environment, is close to its essential complexity (Chaisson, 2014). This means that evolved systems are complex but not complicated.

Gell-Mann (1995), being a physicist, has a more holistic definition of effective complexity and logical depth (apparent complexity). Effective complexity is the length of a concise description of the regularities of an entity. This quantity should not be confused with logical depth. Mandelbrot's set has high logical depth, since being a fractal, a simple rule is applied infinite times in a recursive fashion, but a low amount of effective complexity, since the formula used to describe it is relatively short. This is in general true for all fractals.

In the decision-making field, the Cynefin framework has been proposed by Snowden (2005) to help identify the best approach to solving a specific problem.This sense-making model can be used to understand (from the data available) the characteristics of the problem at hand and which strategy will lead to a solution. As shown in Figure 1, the framework identifies five domains of knowledge: simple and complicated, which are ordered; complex and chaotic, which are unordered; and disorder.

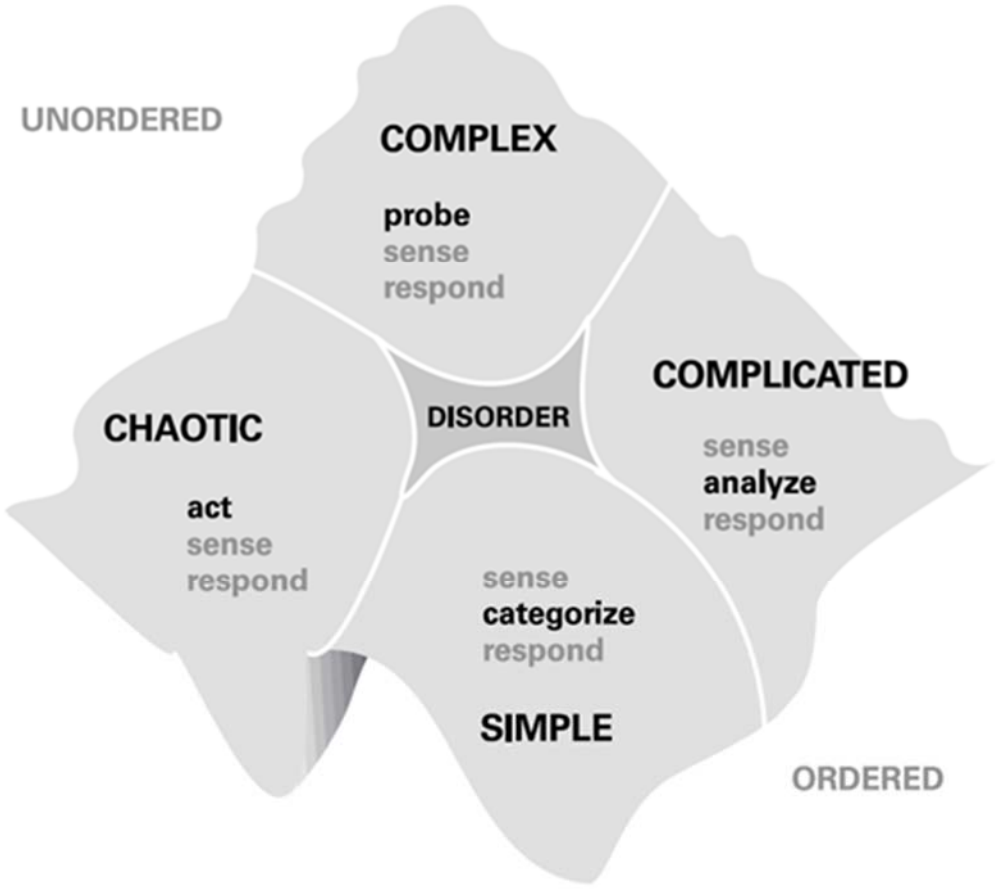


**Figure 1. Representation of the Cynefin Framework**
(Snowden & Boone, 2007)

In the simple domain, systems decisions can be taken unanimously with all the parties, due to the shared understanding of the matter and to the clear relationships between cause and effects. The simple domain is the domain of best practice, where once a solution to a problem has been found, it is applicable unless there is a domain shift. In the complicated domain, there still is a relationship between cause and effect, but not all the parties are able to discern it. The answer to the problem often is not best practice, and some relatively deep analysis is necessary to find a proper solution. This definition is common to the definition of complication as arising from intricacies, where the problem is made difficult because of the way it is formulated. In the complex domain, there might not be a right answer at all, and the relationships between cause and effect can be identified only in retrospect. This is due to ontological emergence, which can create higher logical structures in which feedback loops are hidden. In the chaotic domain, no patterns are discernible, and no relationships can be identified. This is the domain of emergency, where the immediate goal is not to find a solution to a problem, but to bring the system back to an ordered state, from which a solution can be found (Snowden & Boone, 2007).

Wade and Heydari (2014) provide a more technical description of the Cynefin framework. The simple and chaotic domains have both low complication, the former with low complexity, and the latter with high one. The complicated and complex domains have both high complication, the former with low complexity, and the latter with high one. Systems with

high complication can be analyzed with a reductionist approach such as decomposition, while systems with high complexity cannot. This point of view on reductionism is shared by Bedau, according to whom weak (epistemological) emergence can be analyzed using reductionist techniques (Bedau, 1997). The possibility of applying a reductionist approach to the systems exhibiting only weak emergence allows to connect these two types of emergence with the definition of complicated and complex system. Complicated systems are the ones which exhibit weak emergence, which can be analyzed using reductionist techniques, and in which the emergent phenomena are unexpected but still predictable. Complex systems are the ones in which strong emergence comes in place, where the reductionist approach does not work, and where high-level behaviors are not predictable.

#### *Structural, Dynamic, and Socio-Political Complexity*

In engineered systems, complexity can be divided into six types (Sheard & Mostashari, 2010).

- Structural complexity
  - Size, or number of elements in the system, number of types of elements, instances of a certain type.
  - Connectivity, number of connections, types of connections.
  - Topology, architectural patterns, local and global patterns.
- Dynamic complexity
  - Short term, at the time scale of the system operations, behavior of the system while executing its functions.
  - Long term, at the lifetime scale, evolutionary process of the system, retirement or mission extension.
- Socio-political complexity, anything having to do with humans, cognitive limitations, social phenomena.

### *How to Measure Complexity?*

With a better understanding of complexity, we can now look at how this quantity can be measured.

#### *Cyclomatic Complexity*

McCabe (1976) provided a complexity metric for software systems. This metric looks at the graph representation of the program, and it is defined as

$$v(G) = e - n + p$$

where *e* is the number of edges, *n* the number of vertices, and p the number of connected components in the graph. This metric is called the *cyclomatic number*. It can be demonstrated that in a strongly connected graph, the cyclomatic number is equal to the maximum number of linearly independent circuits (McCabe, 1976).

#### *Free Energy Density Rate*

Chaisson (2004) proposed a metric for the evaluation of complexity based on the amount of energy of the entity under study. More precisely, energy rate density, which is "the amount of energy available for work while passing through a system per unit time and per unit mass" (Chaisson, 2015). This metric is a boundary metric, since it considers the input and output of the system without looking at its internal structure. It has been derived through the generalization of various metrics used in various fields, such as propellant to mass ratio for engineering, or metabolic rate for biology. The metric has been evaluated for

multiple entities such as galaxies, stars, planets, plants, animals, societies, and technological systems, showing a rising trend in complexity (Chaisson, 2014).

Per this metric, a system with a large intake of energy per second (i.e., power) and a low mass will be a very complex one. From the engineering point of view, this can also represent a very inefficient system. The success of Chaisson's metric is because the systems under study are mostly evolved systems or are designed with efficiency in mind. Therefore, the applicability of this metric assumes that the system has been designed, or shaped by evolution, in such a way that there is no waste of energy, or useless mass.

***Propagation Cost and Clustered Cost***

MacCormack presented two types of metrics for the evaluation of the complexity of software systems (MacCormack, Rusnak, & Baldwin, 2006). The directed dependency between files in the source code is the function call. The propagation cost is the average of the visibility of modifications to dependent files, while the clustered cost considers the importance of the node scaling the relative cost accordingly.

***Spectral Structural Complexity Metric***

Sinha presented a structural complexity metric based on the design structural matrix (DSM) of the system (Sinha & de Weck, 2012). The metric is defined as

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

where *n* is the number of components in the system, m the number of interfaces, *A* the DSM, $\alpha_i$ the complexity of each component, $\beta_{ij} = f_{ij}\alpha_i\alpha_j$ the complexity of each interface, $\gamma = 1/n$ a normalization factor, and *E(A)* the matrix energy of the DSM. $C_1$ is the complexity contribution of the components, $C_2$ is the contribution of the interfaces, and $C_3$ is the contribution of the topology. The application of the metric sees the evaluation of $\alpha_i$ through expert judgment, and assumes $f_{ij} = 1$ for lack of information (Sinha & de Weck and Olivier, 2013).

*Graph Energy*

The metric is inspired from the Hückel Molecular Orbital (HMO) Theory, which evaluates the energy of $\pi$-bonds in conjugated hydrocarbon molecules as a solution of the time-independent Schrödinger equation

$$H\psi = E\psi$$

where *H* is the Hamiltonian matrix, and *E* the energy corresponding to the molecular orbital. This equation is an eigenvalue problem of the Hamiltonian. In 1978, Gutman defined the energy of a graph, as

$$E = \sum_{i=1}^{n} |\lambda_i|$$

where $\lambda_i$ are the eigenvalues of the adjacency matrix representing the carbon substructure of the molecule (Gutman & Shao, 2011).

Instead of the eigenvalues, the approach introduced by Nikiforov (2007) and embraced by Sihna evaluates the graph energy using the singular values of the matrix. This modification extends the applicability to directed graphs where the adjacency matrix is not

symmetric, while for undirected ones where the adjacency matrix is symmetric matrix, this new approach is coincident with the original eigenvalues one.

The HMO theory is applied to structures of carbon atoms, which are homogeneous. Its application to systems of heterogeneous components, this metric does not consider the role of components with different levels of complexity.

## Methodology

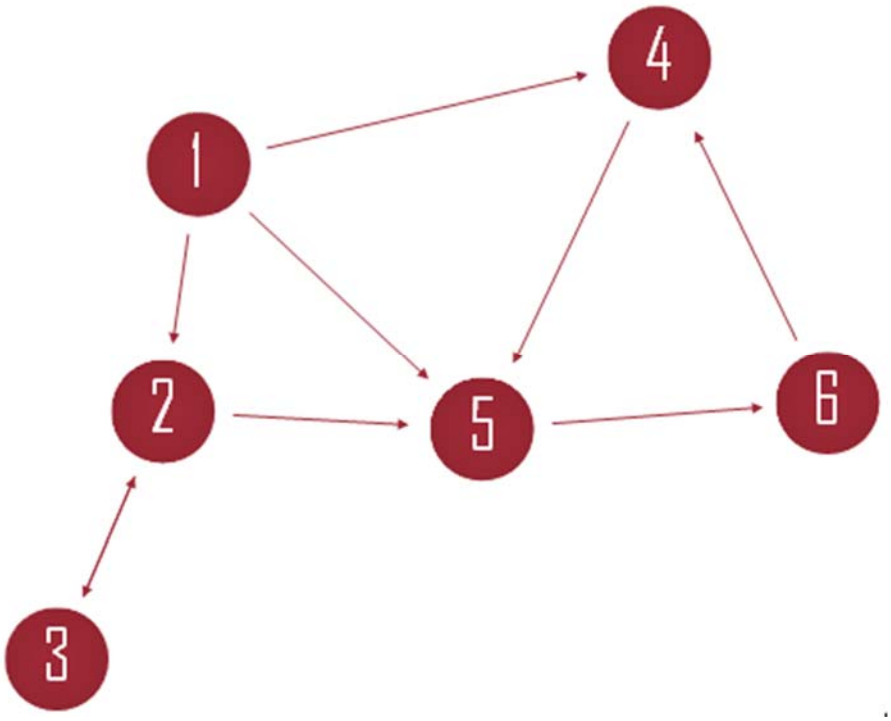


**Figure 2. Graph Representation of a System**

The goal of this research is to measure the structural complexity of engineered systems. The system of interest can vary from a piece of software controlling a reaction wheel to an attitude control system, a satellite, and up to a whole network of satellites. Let's consider the system represented in Figure 2. This graph is a general representation of any engineered system, in which the components are represented by the vertices and the interfaces by the edges. The generality of this approach allows one to evaluate the complexity of the more disparate engineered systems if they can be represented as a graph. The complexity metric proposed by Sinha needs the following data to be available: the complexity of each component $\alpha_i$, the complexity of each interface $\beta_{ij}$, and the adjacency matrix *A*. Here we describe two limitations of Sinha's approach, which will be overcome by the newly developed metric.

***Component Swap Test***

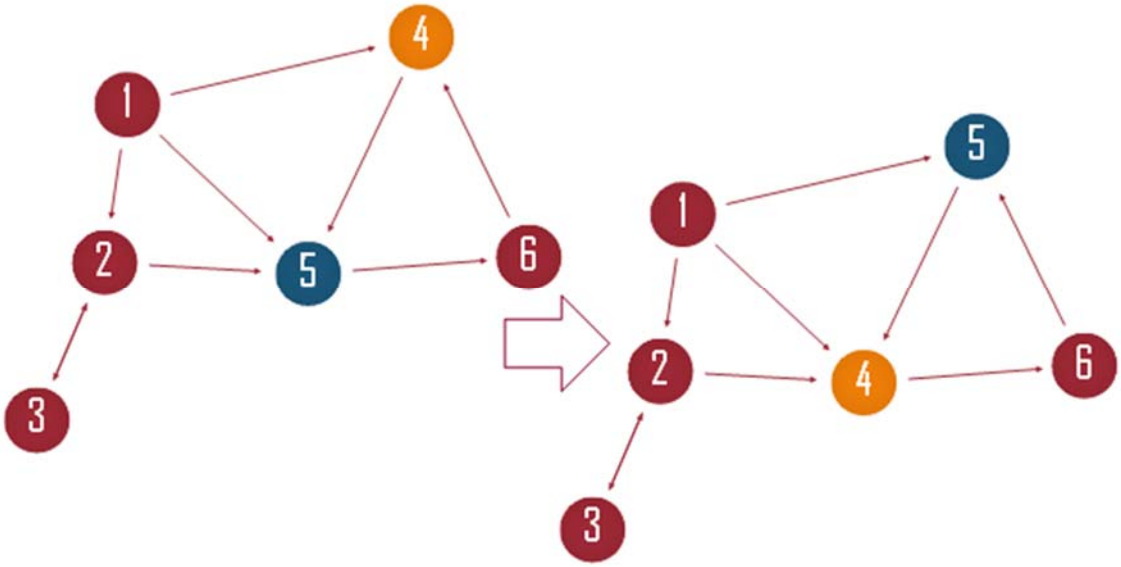


**Figure 3. Swapping of Nodes Within a Graph**

The component complexity $\alpha_i$represents the complexity of the irreducible components at a certain hierarchical level within the system representation. In the graph representation of the system, it can be represented as the weight of a looping edge over each vertex. Let's consider two vertices, $u$ and $v$, and their weights, $w(u,u) = \alpha_u$ and $w(v,v) = \alpha_v$. Swapping the weights to have $w(u,u) = \alpha_v$and $w(v,v) = \alpha_u$ should generally reflect a change in the value of the structural complexity metric.

As an example, consider two separate temperature control systems within a building. One takes care of a conference room and the other one of a biotech laboratory. These two systems are going to have in general very different essential complexities, as it can be seen from their required level of performance (e.g., accuracy, responsiveness). The complexity of the whole building would generally be affected in case these two systems are swapped. A good complexity metric should be able to verify the component swap criterion.

The complexity metric developed by Sinha is not able to distinguish between the two systems. This is because the contribution of the components $C_1$ is evaluated using a sum of the component complexities, which is commutative.

***Interface Swap Test***

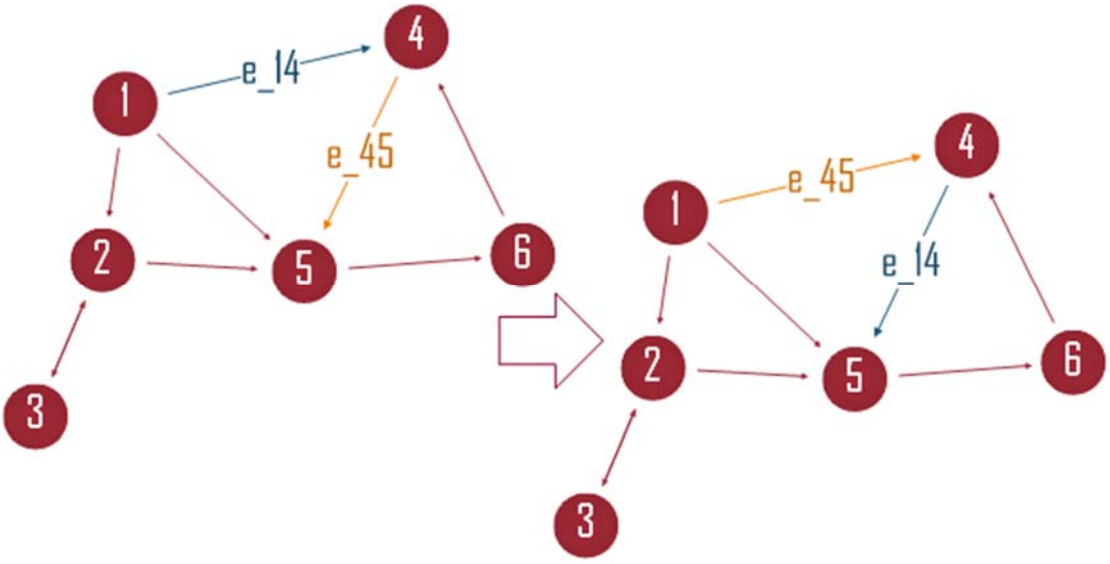


**Figure 4. Swapping of Edges in a Graph**

The interface complexity $\beta_i$ represents the complexity of the interconnection between two components at a certain hierarchical level within the system representation. In the graph representation of the system, it can be represented as the weight of the edge between two vertices. Let's consider two edges having weights $w_1(u_1, v_1) = \beta_1$ and $w_2(u_2, v_2) = \beta_2$. Swapping the weights to have $w_1(u_1, v_1) = \beta_2$ and $w_2(u_2, v_2) = \beta_1$ should generally reflect a change in the value of the structural complexity metric.

In this case as well, the metric developed by Sinha does not reflect a change following this swap because of the commutative property of the sum.

***Approach for the Development of a New Metric***

***Requirements for a Structural Complexity Metric***

In this research, we are developing a spectral structural complexity metric that can overcome the limitations in other structural complexity measures while maintaining all the good features of the existing ones. The new metric shall be able to

1. Measure the complexity of a system with directed interfaces, in which the adjacency matrix is asymmetric.
2. Measure the complexity of a system with multiple parallel edges, in which two components can be connected via more than one edge.
3. Measure the complexity of a system with respect to its size, meaning that the complexity metric should be normalized with respect to the extension of the system.
4. Pass the component swap test.
5. Pass the interface swap test.

***Directed Edges***

Any engineered system can be represented through a graph in which the components are vertices and the interfaces are edges. In general, interfaces have a direction, such as for broadcasting communication systems in which one components

transmits data to many receivers. Directionality can create asymmetry in the representation of the graph, in case the adjacency matrix is used, with subsequent complex eigenvalues. This approach will use the Laplacian matrix, which is Hermitian, and since we are going to use real values for the matrix, it will be symmetric. Therefore, the use of the Laplacian matrix allows us to have real eigenvalues—more precisely non-negative ones—which can be used for the definition of the metric.

#### *Multiple Parallel Edges*

Engineered systems can also have multiple interfaces between components. In a graph representation, this means that the edge $(u, v)$ can have multiple instances, namely, $(u, v)_1$ , $(u, v)_2$ , … , $(u, v)_k$. For example, the interfaces between two components in a cyberphysical system can be thermal, mechanical, electromagnetic, or logical (i.e., in software). In our approach, multiple interfaces are simply bundled together and considered as one. The same approach has also been used already by Sinha, even if without explicit mention (Sinha & de Weck, 2012).

#### *Size Normalization*

Since the size of the system influences its complexity, we want to adopt Chaisson's approach and normalize the metric with the size of the system. This can be done by normalizing the graph metric with the number of vertices, or by using normalized matrices such as the normalized Laplacian, which is normalized with the degree of the nodes.

#### *Weighted Edges*

The role of the graph in this application is to carry the information about complexity of components and interfaces. For this reason, the edges of the graph need to be weighted according to their complexity. The complexity of the components is represented through weights on self-looping edges.

In the following section, the theory behind the development of a new metric is presented, and a running example is used to illustrate the $\alpha_u = 1$ for all the vertices, and $\beta_{uv} = 1$ for all the edges. While the theory considers the more general case and is not based upon this assumption, its use in the illustrative example allows the reader to more easily focus on the topological contribution to the system complexity.

## Spectral Theory of Systems Complexity

### *Spectral Graph Theory*

Spectral Graph Theory is the study of graphs through the eigenvalues of their matrix representation. The set of eigenvalues is known as the spectrum. The elements of spectral graph theory here reported were published by Chung (1997) and Spielman (2007). Let's consider a graph with n vertices and m edges. If u and v are two vertices in the graph, and they are connected by an edge, we say that they are adjacent. An edge that connects a vertex to itself is called a loop. Graphs that contain no loops are called simple graphs. Edges can be associated to a direction. Directed graphs have edges with an associated direction, meaning that the edges $(u, v)$ and $(v, u)$ are two distinct entities. For undirected graphs, those are two representations of the same entity.

Edges in a graph can also be weighted, meaning that we can define a function

$$w(u, v) : V \times V \rightarrow \mathbb{R} .$$

where

$$w(u, v) \geq 0$$

and, in the case of undirected graphs,

$$w(u,v) = w(v,u)$$

The degree of a vertex is defined as the number of incoming edges connected to it, and in the case of weighted edges

$$d_v = \sum_u w(u,v)$$

In this section, we introduce various matrix representations of graphs that will be useful in the creation of a spectral complexity metric. To do this, we will consider the graph represented in Figure 2 as a running example.

***Adjacency Matrix***

The adjacency matrix is defined as

$$A(u,v) = \begin{cases} 1 & if\ u\ and\ v\ are\ adjacent, \\ 0 & otherwise. \end{cases}$$

The adjacency matrix is symmetric in the case of undirected graphs. For directed graphs, the symmetry holds only if edges appear in pairs. In the case of weighted edges, the adjacency matrix is defined as

$$A(u,v) = \begin{cases} w(u,v) & if\ u\ and\ v\ are\ adjacent, \\ 0 & otherwise. \end{cases}$$

The eigenvalues of the adjacency matrix are labeled in increasing order and represented as

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

and the following is true

$$\sum_{i=1}^{n} \lambda_i = 0$$

$$\sum_{i=1}^{n} \lambda_i^2 = 2m$$

In our example, the adjacency matrix will have the following values in case we consider the edges of the graph as directed or undirected

$$A_{dir} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad A_{undir} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

***Laplacian Matrix***

The Laplacian matrix is defined as $L(u,v) = D(u,v) - A(u,v)$, where $D(u,v)$ is the diagonal matrix of the vertex degrees. This definition is equivalent to

$$L(u,v) = \begin{cases} d_v & if\ u = v, \\ -1 & if\ u\ and\ v\ are\ adjacent, \\ 0 & otherwise. \end{cases}$$

If edges are weighted, its definition is given by

$$L(u,v) = \begin{cases} d_v - w(u,v) & if\ u = v, \\ -w(u,v) & if\ u\ and\ v\ are\ adjacent, \\ 0 & otherwise. \end{cases}$$

In the case of directed graphs (Chung, 2005), the Laplacian matrix is defined as

$$L(u,v) = \Phi - \frac{\Phi P + P^*\Phi}{2}$$

where $\Phi$ is the diagonal matrix of the flow of a vertex $\phi(v)$ and $P$ is the transition probability matrix. For a weighted directed graph (Butler, 2007),

$$P(u,v) = \frac{w(u,v)}{d_{out}(u)}$$

The Laplacian matrix is always symmetric, both in the case of directed and undirected graphs. The eigenvalues of the Laplacian matrix are usually labeled in a decreasing order, and are represented as

$$0 = \mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$$

and the following is true:

$$\sum_{i=1}^{n} \mu_i = 2m, \sum_{i=1}^{n} \mu_i^2 = 2m + \sum_{i=1}^{n} d_i^2$$

In our example, the directed and undirected Laplacian matrices assume the following values

$$L_{dir} = \begin{bmatrix} 2.97 & -0.26 & -0.017 & -0.16 & -0.18 & -0.064 \\ -0.26 & 2.97 & -1.20 & -0.041 & -0.32 & -0.034 \\ -0.017 & -1.20 & 0.99 & -0.027 & -0.032 & -0.022 \\ -0.16 & -0.041 & -0.027 & 2.97 & -1.64 & -1.19 \\ -0.18 & -0.32 & -0.032 & -1.64 & 3.97 & -1.38 \\ -0.64 & -0.034 & -0.022 & -1.19 & -1.38 & 1.98 \end{bmatrix}$$

and

$$L_{undir} = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

***Normalized Laplacian Matrix***

The normalized Laplacian matrix for undirected graphs is defined as $\mathcal{L} = D^{-1/2}LD^{-1/2}$ which is equivalent to

$$\mathcal{L}(u,v) = \begin{cases} 1 & if\ u = v, \\ -\dfrac{1}{\sqrt{d_u d_v}} & if\ u\ and\ v\ are\ adjacent, \\ 0 & otherwise. \end{cases}$$

For weighted graphs, the weighted normalized Laplacian matrix is

$$\mathcal{L}(u,v) = \begin{cases} 1 - \dfrac{w(u,v)}{d_u} & if\ u = v, \\ -\dfrac{w(u,v)}{\sqrt{d_u d_v}} & if\ u\ and\ v\ are\ adjacent, \\ 0 & otherwise. \end{cases}$$

In the case of directed graphs (Chung, 2005), the normalized Laplacian matrix is defined as

$$\mathcal{L}(u,v) = I - \frac{\Phi^{1/2}P\Phi^{-1/2} + \Phi^{-1/2}P^{*}\Phi^{1/2}}{2}$$

where *I* is the identity matrix, $\Phi$ is the diagonal matrix of the flow of a vertex $\phi(v)$, and *P* is the transition probability matrix. For a weighted directed graph (Butler, 2007),

$$P(u,v) = \frac{w(u,v)}{d_{out}(u)}$$

The normalized Laplacian matrix is always symmetric, and its eigenvalues are represented as

$$0 = \nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$$

In our running example, the values of this matrix for the directed and undirected case are

$$\mathcal{L}_{dir} = \begin{bmatrix} 0.99 & -0.088 & -0.0096 & -0.052 & -0.052 & -0.026 \\ -0.088 & 0.99 & -0.69 & -0.014 & -0.092 & -0.014 \\ -0.0096 & -0.69 & 0.99 & -0.016 & -0.016 & -0.016 \\ -0.052 & -0.014 & -0.016 & 0.99 & -0.47 & -0.49 \\ -0.052 & -0.092 & -0.016 & -0.47 & 0.99 & -0.49 \\ -0.026 & -0.014 & -0.016 & -0.49 & -0.49 & 0.99 \end{bmatrix}$$

and

$$\mathcal{L}_{undir} = \begin{bmatrix} 1 & -0.33 & 0 & -0.33 & -0.29 & 0 \\ -0.33 & 1 & -0.58 & 0 & -0.29 & 0 \\ 0 & -0.58 & 1 & 0 & 0 & 0 \\ -0.33 & 0 & 0 & 1 & -0.29 & -0.41 \\ -0.29 & -0.29 & 0 & -0.28 & 1 & -0.35 \\ 0 & 0 & 0 & -0.41 & -0.35 & 1 \end{bmatrix}$$

### *Matrix Energy*

#### *Graph Energy*

Graph energy has been defined by Gutman in 1978 (Gutman, 2001; Gutman & Shao, 2011) as

$$E_A(G) = \sum_{i=1}^{n} |\lambda_i|$$

and it has the following properties

1. $E(G) \geq 0$, where equality is attained only for $m = 0$, meaning that the graph has no edges, and all the vertices are disconnected;
2. the energy of two disconnected graph components $G_1$and $G_2$ is $E(G) = E(G_1) + E(G_2)$
3. if one component is $G_1$ and all the other components are isolated vertices, then $E(G) = E(G_1)$.

#### *Laplacian Graph Energy*

Gutman also defined the Laplacian energy of a graph (Gutman & Zhou, 2006) as

$$E_L(G) = \sum_{i=1}^{n} |\gamma_i| = \sum_{i=1}^{n} \left|\mu_i - \frac{2m}{n}\right|$$

where $\gamma_i$ are the auxiliary Laplacian eigenvalues defined as

$$\gamma_i = \mu_i - \frac{2m}{n}$$

#### *Generalized Matrix Energy*

A generalization of all these definitions can be given considering a general matrix (Cavers, Fallat, & Kirkland, 2010)

$$E_M(G) = \sum_{i=1}^{n} \left|\lambda_i(M) - \frac{tr(M)}{n}\right|$$

where $tr(M)$ is the trace of the matrix *M*. Thanks to this generalization it is possible to define the normalized Laplacian energy of a graph

$$E_{\mathcal{L}}(G) = \sum_{i=1}^{n} \left|\nu_i - \frac{tr(\mathcal{L})}{n}\right| = \sum_{i=1}^{n} |\nu_i - 1|$$

## Spectral Structural Complexity Metrics

The advancements in spectral graph theory presented in the previous section allow us to define a series of complexity metrics based on the spectrum of a certain representation of the system. Let's start with defining the weight function as

$$w(u,v) = \begin{cases} \alpha_u & if\ u = v \\ \beta_{u,v} & otherwise \end{cases}$$

where $\alpha_u$ represents the complexity of the component $u$, and $\beta_{u,v}$ the complexity of the interface between components $u$ and $v$. This function allows us to use the definitions for the weighted adjacency, Laplacian, and normalized Laplacian matrices, for both the case of

directed and undirected graphs. The structural complexity evaluated using the adjacency matrix is defined as

$$C_A = \frac{E_A(G)}{n}$$

where the adjacency matrix considers the weights of the edges, and n is the number of vertices of the graph. In the case of unweighted edges, this metric is equivalent to the $C_3$ component of the one defined by Sinha (Sinha & de Weck, 2012). The adjacency matrix is historically the most used in systems engineering (as DSM) and in spectral graph theory. In recent years, there has been a shift in spectral graph theory, given by the interesting properties of the Laplacian eigenvalues. The second smallest eigenvalue is particularly interesting, since it represents the connectivity of the graph. Also, the multiplicity of zero in the Laplacian spectrum represents the number of connected components, used in the metric proposed by McCabe. For these reasons, we are defining the structural complexity evaluated using the Laplacian matrix as

$$C_L = \frac{E_L(G)}{n}$$

where the Laplacian matrix considers the weights of the edges, and *n* is the number of vertices of the graph. This type of normalization has an alternative, which is to normalize using the degree matrix of the system. This alternative approach brings to the normalized Laplacian matrix. The structural complexity evaluated using the normalized Laplacian matrix is defined as

$$C_{\mathcal{L}} = E_{\mathcal{L}}(G)$$

where the normalized Laplacian matrix is defined considering the weights of the edges.

In Figure 5 we report the values of these metrics for the directed and undirected case of our running example.

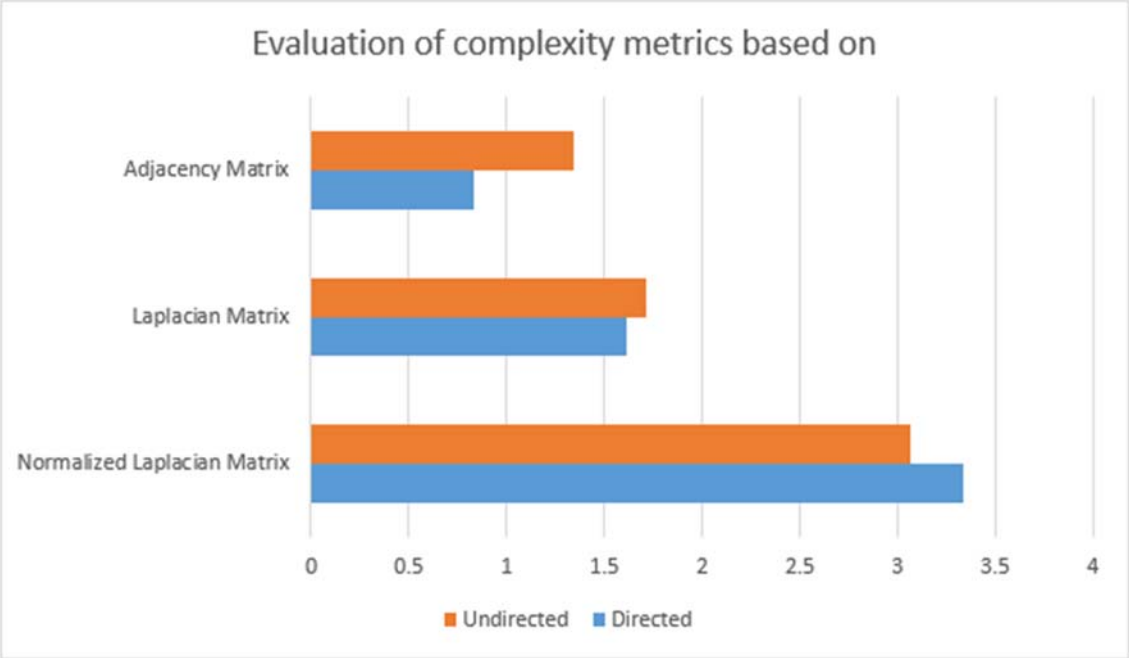


**Figure 5. Evaluation of Complexity Metrics Based on the Matrix Energy of the Adjacency Matrix, Laplacian Matrix, and Normalized Laplacian Matrix, for the Directed and Undirected Graph Represented in Figure 2**

## Conclusion and Future Work

In this paper, we presented an alternative to existing structural complexity metrics. The purpose of the metrics is to measure the structural complexity of the system, considering the contributions of the size, the connectivity, and the topology of the system. At this stage of the research, the focus has been shifted on the topological contribution, with the plan of addressing size and connectivity in a later stage.

These metrics will subsequently be applied to real world systems, with the goal of verifying their applicability and understanding their differences in terms of features and limitations. The results will then be compared to the ones from other complexity metrics, with the goal of validating the new metrics and clarifying their possible shortcomings.

In the context of the systems engineering practice, these metrics represent a continuation of the widespread effort to introduce quantitative tools to increase objectivity of measurements. The spectral approach developed by Sinha is the starting point of possibly a series of research efforts that will gradually introduce new metrics trying to patch limitations in the existing ones. The long-term expectation is for practitioners to converge on the use of a low number of metrics that will be applied depending on the specific case.

## References

Abbott, R. (2006). Emergence explained: Abstractions: Getting epiphenomena to do real work. *Complexity, 12*, 13–26.

Bedau, M. A. (1997). Weak emergence. *Noûs, 31,* 375–399.

Butler, S. (2007). Interlacing for weighted graphs using the normalized Laplacian. Electronic *Journal of Linear Algebra, 16*, 87.

Cavers, M., Fallat, S., & Kirkland, S. (2010). On the normalized Laplacian energy and general Randić index R-1 of graphs. *Linear Algebra and Its Applications, 433,* 172–190.

Chaisson, E. J. (2004). Complexity: An energetics agenda. *Complexity, 9,* 14–21.

Chaisson, E. J. (2011). Energy rate density as a complexity metric and evolutionary driver. *Complexity, 16*, 27–40.

Chaisson, E. J. (2014). The natural science underlying big history. *Scientific World Journal*, *2014*.

Chaisson, E. J. (2015). Energy flows in low-entropy complex systems. *Entropy, 17,* 8007–8018.

Chalmers, D. J. (2008). Strong and weak emergence. In *The re-emergence of emergence.* Oxford University Press.

Checkland, P. (1981). *Systems thinking, systems practice*.

Chung, F. (2005). Laplacians and the Cheeger inequality for directed graphs. Annals of *Combinatorics, 9*, 1–19.

Chung, F. R. (1997). *Spectral graph theory* (Vol. 92). American Mathematical Society.

Crawley, E., Cameron, B., & Selva, D. (2015). *System architecture: Strategy and product development for complex systems.* Pearson.

Fischi, J., Nilchiani, R., & Wade, J. (2015). Dynamic complexity measures for use in complexity-based system design. *IEEE Systems Journal*.

Gell-Mann, M. (1995). What is complexity? Remarks on simplicity and complexity by the Nobel Prize-winning author of *The Quark and the Jaguar. Complexity, 1*, 16–19.

Gutman, I. (2001). The energy of a graph: Old and new results. In *Algebraic combinatorics and applications* (pp. 196–211). Springer.

Gutman, I., & Shao, J.-Y. (2011). The energy change of weighted graphs. *Linear Algebra and Its Applications, 435*, 2425–2431.

Gutman, I., & Zhou, B. (2006). Laplacian energy of a graph. *Linear Algebra and Its Applications, 414,* 29–37.

Kauffman, S. (2007). Beyond reductionism: Reinventing the sacred. *Zygon, 42*, 903–914.

Longo, G., Montévil, M., & Kauffman, S. (2012). No entailing laws, but enablement in the evolution of the biosphere. *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation* (pp. 1379–1392).

MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2006). Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science, 52,* 1015–1030.

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, 308–320.

McCabe, T. J., & Butler, C. W. (1989). Design complexity measurement and testing. *Communications of the ACM, 32,* 1415–1425.

Nikiforov, V. (2007). The energy of graphs and matrices. *Journal of Mathematical Analysis and Applications, 326,* 1472–1475.

Page, S. E. (1999). Computational models from A to Z. *Complexity, 5,* 35–41.

Rouse, W. B. (2016). *Complexity: Absolute or relative?*

Sheard, S. A., & Mostashari, A. (2010). A complexity typology for systems engineering. *Twentieth Annual International Symposium of the International Council on Systems Engineering.*

Sinha, K., & de Weck, O. (2012). Structural complexity metric for engineered complex systems and its application. Gain Competitive Advantage by Managing Complexity: *Proceedings of the 14th International DSM Conference Kyoto, Japan* (pp. 181–194).

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

Snowden, D. (2005). Multi-ontology sense making: A new simplicity in decision making. *Journal of Innovation in Health Informatics, 13*, 45–53.

Snowden, D. J., & Boone, M. E. (2007). A leaders' framework for decision making. *Harvard Business Review, 85*, 68.

Spielman, D. A. (2007). Spectral graph theory and its applications. *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)* (pp. 29–38).

Wade, J., & Heydari, B. (2014). Complexity: Definition and reduction techniques. *Proceedings of the Poster Workshop at the 2014 Complex Systems Design & Management International Conference* (pp. 213–226).

Weaver, W. (1948). Science and complexity. *American Scientist, 36*, 536–544.


**Acquisition Research Program**
**Graduate School of Business & Public Policy**
**Naval Postgraduate School**
**555 Dyer Road, Ingersoll Hall**
**Monterey, CA 93943**

www.acquisitionresearch.net