A virtual engineering based approach to verify structural complexity of component-based automation systems in early design phase

Bugra Alkan[1], Robert Harrison[1]

WMG, University of Warwick, CV4 7AL, Coventry, West Midlands, UK

Abstract

Highly diverse factors including technological advancements, uncertain global market and mass personalisation are believed to be main causes of ever-growing complexity of manufacturing systems. Although complex systems may be needed to achieve global manufacturing requirements, complexity affects on various factors, such as: system development effort and cost, ease of re-configuration, level of skill required across the system life-cycle (e.g. design, operate and maintain). This article aims to develop a scientifically valid and industrially applicable complexity assessment approach to support early life-cycle phases of component-based automation systems against unwanted implications of structural system complexity. The presented approach defines component-based automation system as a constellation of basic components which can be represented in various design domains, such as: mechanical, electrical, pneumatic, control, etc. Accordingly, structural complexity is expressed as the combination of both inherent complexity of system entities and topological complexity resulting from the integration of elements of such constellations in a multi-layered network. The proposed approach is used to specify and implement a complexity assessment module which can be integrated into a series of virtual system design software solutions, in order to add complexity assessment as part of the design support and validation tools used by manufacturing engineers. Consequently, the proposed approach is integrated into the vueOne virtual engineering tool, wherein virtual automation system design data can be streamlined and used as input to the theoretical complexity model. In the developed tool, only mechanical and logical design domains are considered due to the limited data availability in early design phase. Inherent complexity of both mechanical and logical system entities and their interactions are expressed as a function of domain-specific design elements, and topological complexity is defined as the graph energy of the corresponding design connectivity matrix. Furthermore, the values of mathematical model parameters are determined based on an optimisation study, where subjective opinions of system/control engineers regarding the effort/difficulty associated with the development of thirty different component-based automation system designs are correlated with the corresponding complexity model outputs to minimise the prediction errors. The proposed approach is also demonstrated on a modular production system consisting of four sub-modules. The study shows that the approach can help designers/managers to better identify root causes of structural system complexity, and provides a systemic approach to compare alternate system designs during early system planning phase.

Keywords: Design verification, complexity, complex systems, complexity measure, manufacturing, automation systems, component-based design, virtual engineering

1. Introduction

1.1. Research background

Modern manufacturing systems are composed of components and sub-systems of various nature, including: machining and assembly units, smart material handling devices, complex control algorithms and interlinked safety units [1]. Due to the need for the system to be changeable and modular to accommodate changes in functionality, structural complexity increases as more components and interfaces are added to systems at both hardware and software layers [2]. Although complex systems are required to satisfy the need for flexibility and adaptability in manufacturing domain, complexity may bring fragility and unpredictability to manufacturing systems as a result of an improper complexity management [3, 4]. It is reported that, excessive complexity negatively impacts production quality [5], reliability [6], throughput [7] and production time [8], and disturbs the system's efficiency at design, operation, maintenance, and management levels [9].

According to Rechtin [10], "*the more complex a system, the more difficult it is to design, build and use, intuitively, the more difficult a task the more expensive it is, if not for any other reason that requiring access to select experts or lots of time to complete all the task involved*". Complex systems are often described by intricate connectivity patterns and topologies that may result in both lower productivity and higher failure rates during their development [11]. Assessing complexity and trying to reduce it as much as possible while maintaining functional requirements or performance targets is one way of nullifying mistakes [12]. According to Meyer and Lehnerd [13], "*Reducing complexity almost always reduces direct and indirect costs*". Similarly, McCabe [14] states that assessing complexity is one of the primary requirements in the design stage of an engineering system, which helps designers to better under-stand the cost and time required to realise it. Moreover, assessing the complexity of a design allows us to analyse whether it is comprehensible for humans [15]. Therefore, complexity of a manufacturing system design should be identified and assessed to remain competitive, profitable, and to better respond to the unstable market dynamics and increased product variety [16].

1.2. Research objectives and contributions

Complexity assessment allows manufacturing firms to detect stress points in their manufacturing systems, and to take most appropriate actions to handle it [2]. In recent years, proactive complexity assessment of manufacturing systems, conducted at early design stage has achieved a considerable amount of attention from academia, as it enables significant savings in terms of time and cost [17]. However, these measures include either paper-based assessment or face-to-face interviews and questionnaires for data collection making them costly and time consuming [3]. To bridge the gap between industry and academia, this article presents an automated complexity assessment framework which integrates theoretical complexity models with virtual system design tools, where various virtual design data can be streamlined as an input to the mathematical models.

The presented research defines an industrial automation system as an engineering network consisting of a number of connected components which are working and interacting with each other to realise a common manufacturing goal. Structural complexity is defined as a function of both the complexity of individual system elements and the effects of their connectivity pattern. The developed automated assessment framework is further integrated into vueOne virtual manufacturing tool, in which the structural complexity of component-based automation system (CBAS) architectures, composed of interacting logical and mechanical system design layers, can be verified at very early system architecting phase. Furthermore, values of the mathematical model parameters (component coefficients, k^M and k^L, and interface coefficients, c^M, c^L and c^Δ) are selected with a genetic algorithm (GA) based optimisation study, where the prediction errors between complexity model outputs and expert-defined complexity scores for thirty virtual CBAS designs with varying degree of structural complexity are minimised.

The developed approach brings forth two important aspects: *i*) a concurrent assessment of structural complexity during design phase so that those designs believed exceptionally complex can be identified, altered and enhanced, and *ii*) as opposed to the pen-and-paper based methods, the complexity assessment is integrated into a virtual system design and planning tool leading to ease of measurement.

1.3. Research scope

The scope of this research was carefully defined and focused on the assessment of structural complexity in CBASs at early system architecting phases. Due to the limited data availability in this phase, only mechanical and logical system architectures and their interactions are used in the implementations of the developed automated complexity assessment framework. This is reasonable as design information of manufacturing systems at the system architecting phase is often limited to machine elements, established components, and mechanisms in mechanical design and state transitions diagrams that represent sensors, actuators, and controller behaviours, in logical design [18]. However, there is a plan to integrate other design domains, such as: electrical, pneumatical etc., with the proposed approach as a future work. Please note that, in early system planning phases, complexity is often under-estimated due to the high level of abstraction [19]. As a result, when the system matures over time, the actual complexity of the system is exposed which could lead to exceptional situations that could results in exceeding project budgets, missing deadlines, etc [11]. The authors believe that the proposed complexity assessment framework investigating system architectures based on two fundamental domains, can provide valuable insights regarding actual complexity that may form in the later design stages; thereby allowing system designers to take proactive actions to control and manage system development projects in an effective way, and take necessary steps to minimise system re-design and alteration costs.

1.4. Paper structure

This article is structured as follows. Section 2 presents a review of complexity in the domain of manufacturing systems engineering and its application to practical evaluation of manufacturing systems. Section 3 introduces the theoretical basis of the research and its adoption to component-based automation systems. Section 4 presents the complexity assessment software module integrated into a virtual system design software solution. In Section 5, an optimisation study is performed to refine the complexity model parameters using a set of component-based automation system designs with varying degrees of structural complexity. Section 6 proposes a traffic-light based complexity classification scheme for virtual automation system designs. In Section 7, the proposed framework is tested on a case study. In Section 8, the results of the article are discussed and the future work is outlined.

2. Literature review

2.1. Types of manufacturing systems complexity

Complexity in manufacturing systems is a natural outcome of the evolution of manufacturing organisations to adapt today's highly uncertain market environment [3]. According to Schuh et al. [20], manufacturing complexity can be defined within two groups: *i*) internal and *ii*) external. Internal complexity primarily arises due to the increased product variety, whereas external complexity stems from market uncertainties, political and institutional regulations [16]. Internal complexity can be divided into three main categories: *i*) structural (static) complexity, *ii*) operational (dynamic) complexity and *iii*) organisational complexity [21]. Structural complexity is related to the architecture of a manufacturing system, which can be considered as a network that is made by a set of interacting resources and services. Operational complexity is steered by the manufacturing system's operational characteristics and is related to operational uncertainties [19]. Accordingly, a production system can be considered as complex, if its behaviours are hard to trace in an effective way [22]. Organisational complexity, on the other hand, is linked to the organisational structures, systems, processes and communication flows [3]. Previous research has resulted in several approaches to complexity measurement ranging from information theory to survey based approaches.

2.2. Previous studies on complexity assessment

In manufacturing, complexity is often associated with the uncertainty in describing the overall state of a production system or its components. In this context, the uncertainty is measured through Shannon's entropy (please see [23]). In the literature, a relatively high number of entropic definitions have been proposed for structural complexity [24, 25, 26, 27] and operational complexity [28, 29, 30, 31, 22, 32, 33, 34, 35, 36, 37, 38, 39]. Yet, entropic measures have been critiqued for their subjectivity in defining resource states [40], being tied to the level of detail [41], and costly assessment phases which is especially true for dynamic complexity investigations [42].

There are also other methods on the way to a quantitative definition of manufacturing systems complexity, such as chaos and non-linear dynamics theory [43, 44, 45, 46, 47]. These approaches provide a deep understanding for underlying causes of system behaviours, envisage the effect of operational system parameters on the key performance indicators (KPIs), and illustrate the sensitivity of the system [48]. Nevertheless, these approaches are incapable of capturing and analysing stochastic events such as machine breakdowns [49]; they are restricted to schematic analysis for the dynamic system behaviours [50] (excluding maximal Lyapunov exponents testing) and they often require huge data sets, making them costly. Moreover, these approaches are often sensitive to the fluctuations in external factors such as measurement errors and noise [49].

As opposed to above mentioned methods, which guarantee a quantifiable reflection of the system complexity, production system complexity can be assessed through heuristics-based methods attempting to provide an industrially readable picture of complexity based on the system's structure [51, 52, 53, 54, 55, 56, 57]. Heuristic methods are advantageous in that they are easily applied to real industrial systems and data collection is easy, allowing comparisons of design alternatives at early life cycle phases to detect potential stress points. Nevertheless, these approaches provide a limited insight on manufacturing system complexity and are incapable of analysing intricate structural patterns [2]. These metrics are also dependent on the industrial domain or specific application that they are designed and developed for, i.e. ad-hoc, thus, the applicability of heuristics approaches over different system and service applications is often restricted.

Complexity has a subjective nature as it depends on the specific context [58] and human spectator who perceives it [59]. Complexity arising from human-system interactions and manufacturing systems have been discussed in great detail, primarily, by the method of surveying [60, 28, 61, 62]. However, survey-based methods are only capable of understanding the perceived level of complexity and cannot be employed in situation where there is absence of a physical prototype. Most importantly, the results from surveys are significantly dependent on the opinion of interviewees, hence making the process subjective.

2.3. Research gaps

In summary, there are many ways to model and measure complexity which have their pros and cons. As an example, the methods derived from chaos and non-linear dynamics are used to analyse the complexity by means of the system's dynamic behaviours which need to be observed over a long interval of time, whereas, the heuristics-indices based methods estimate complexity solely based on the system's physical situation but with a low accuracy. The former is used to choose the most appropriate control policy to handle uncertain conditions, while the latter is chiefly employed to compare design alternatives at conceptual stages. It is concluded that the existing solutions to complexity management are still immature and typically target post-design phases of production system life-cycle, thus leading to costly and time consuming redesign phases. As a result, there is an increased need for tools and methods to pro-actively identify and minimise complexity during very early design stages. The research presented here, thus, aims to develop a proactive design support, where quantifiable data collected from virtual system design and process planning tools, can be streamlined and transformed into meaningful complexity values allowing designers to concurrently evaluate system designs to select the optimal design among various alternatives and to make modifications on existing systems.

3. Complexity modelling framework

3.1. Theoretical origin

In this research, structural complexity of industrial automation systems is assumed to be the result of the complexity of individual system entities and the effects of the system connectivity pattern. To formally define structural complexity, the presented research adopts the following model proposed by [63] as a base frame. This model is inspired by the relationships (please see [64]) defining the π electron energy in organic molecular orbitals, and defines structural complexity of any network-based engineering system as a function of *i*) the complexity of the individual components, *ii*) pair-wise interaction complexity between connected components, and *iii*) the effects of the resultant system topology. In this analogy, structural complexity C is formulated as below:

$$C = C_1 + C_2 C_3 \tag{1}$$

where, C_1, C_2 and C_3 represent component, interface and topological complexity, respectively. The following subsections detail the individual elements in the structural complexity model presented in this research.

3.1.1. Component complexity, C_1

The first term C_1 represents the sum of component complexities, and is defined as follows:

$$C_1 = \sum_{i=1}^{N} \alpha_i \tag{2}$$

where, α_i is the inherent complexity of a component i and N is total number of components in the system. Complexity of a component α_i represents the technical difficulty associated with the management of the component alone, not accounting the complexity of component's interfaces and the system's architectural information. In this research, the underlying complexity of a component is associated with the amount of information required to define/replicate the component. In other words, information is used as a representative for the relative effort required to use, operate, programme, control or interact with the component. In here, it is assumed that, inherent structures of system components are composed of a number of indecomposable elements. As an example, elements of a software component can be considered as any statement that describes computations to be performed by the component itself. In light of this assumption, α_i is defined in the form of an exponential function as follows:

$$\alpha_i = 5\left(1 - e^{-(\sum_{i=1}^{m} k_i n_i)}\right) \tag{3}$$

where, m is the total types of elements forming the component, n_i is the total number of elements belonging to a specific type i, and k_i is the exponential function parameter for the corresponding element type i ($k \in [0, 1]$). Please note that, an exponential function has been adopted in defining component complexity score as a result of two distinct reasons. The former is to scale complexity score between 0 and 5; thereby enabling a global range for all components. The latter involves a negative exponential function which is used to limit the complexity score to five, especially for those components, for which the development and management exceeds the human's limits of understanding. This is due to the fact that perceived complexity of an individual cannot be increased after reaching observer's limits of understanding [2].

3.1.2. Pair-wise interface complexity, C_2

The second term C_2 is the sum of pair-wise interaction complexities β_{ij}, and is formulated as follows,

$$C_2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_{ij} A_{ij} \tag{4}$$

where, A_{ij} defines the binary adjacency matrix visualising the connectivity between system components:

$$A_{ij} = \begin{cases} 1 & \text{if there is a connection between i and j} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Similarly, the term C_2 defines the technical/ergonomical effort associated with the development of pair-wise component interactions and involves knowledge about both nature of the connectivity and overall system topology. In this research, by adopting the approach proposed in [63], complexity of pair-wise component interfaces is characterised by two essential factors; *i*) complexity of the interfaced components and *ii*) the nature of the connectivity c_k:

$$\beta_{ij} = \left(\sum_{k=1}^{l} c_k\right) \frac{(\alpha_i + \alpha_j)}{2} \tag{6}$$

where, c_k is the interface coefficient defining the relative difficulty in establishing the interface type k (i.e. the nature of the connectivity), and l is the number of interfaces between components i and j. This representation describes interface complexity as a fraction of the connected component complexities, such that interface and component complexities are not dimensionally mismatched. It is also reasonable, as the interface complexities are expected to be much smaller than the component complexities in cyber-physical systems [19].

3.1.3. Topological complexity, C_3

The term C_3 captures the effects of overall system topology, and is formulated by the *graph energy* measure (please see [65]).

$$C_3 = \frac{E_A}{N} \tag{7}$$

In here, graph energy E_A is designated by the sum of singular values σ_i of the design connectivity matrix E_A of the system under consideration.

$$E_A = \sum_{i=1}^{N} \sigma_i \tag{8}$$

This metric outlines the nominal effective dimension entrenched within the connectivity pattern [66]. Note that, topological complexity, contrary to the first two terms, requires information regarding overall system architecture, and depicts a global effect which can be perceived during the system integration [63]. Therefore, the term $C_2 C_3$ can be referred as a general indicator of system integration effort.

According to [19], the values of graph energy can be used to categorise different architectural patterns. Accordingly, he defines the energy regimes for a system (S) with n number of components as: *i*) *hypo energetic*, *ii*) *transitional* and *iii*) *hyper energetic*. The hyper energetic regime, here, is considered by the graph energy which is greater than or equal to that of the fully connected system,

$$E(A) \geq 2(n-1) \tag{9}$$

The hypo energetic regime is defined as:

$$E(A) \leq n \tag{10}$$

The intermediate regime between these two where the energy is higher than that of the hypo energetic regime and smaller than the hyper energetic is labelled as translational regime. In the original source [19], these energy regimes is further translated into common architectural pattern categories as follows:

$$C_3 = \begin{cases} \geq 2(1-\frac{1}{N}) \approx 2 & \approx \text{distributed architecture} \\ 2 > \ldots \geq 1 & \approx \text{hierarchical architecture} \\ < 1 & \approx \text{centralised architecture} \end{cases} \quad (11)$$

As it is understood from the expression given above, the topological complexity increases from centralised towards more distributed architectures [19]. Please note that, topological complexity C_3 allows us to differentiate the system structures with similar component and interface complexities, and better prediction of the system integration effort [67].

3.1.4. *Overall complexity metric*

In summary, the presented analogy defines structural complexity of the system (A) as follows:

$$C = \sum_{i=1}^{N} \alpha_i + (\sum_{i=1}^{N}\sum_{j=1}^{N} \beta_{ij} A_{ij})(\frac{E_A}{N}) \quad (12)$$

The overall complexity metric can be seen as a comprehensive definition of structural system complexity, increasing the available choices in system architecting due to the inclusion of combinations of complex and simple topologies and components [19]. Moreover, it may provide a single complexity score which allows designers to improve/modify the system designs by comparing them with possible alternatives in a more practical way. **Figure 1** shows constituent elements of the presented complexity measure.

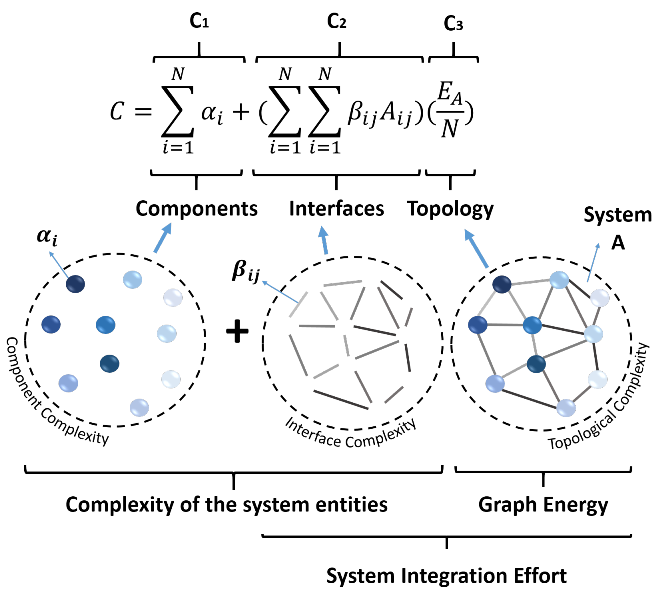

Figure 1: Elements of the complexity metric (Adapted from [66]).

3.1.5. *Reasons for the selection of the analogy*

In this research, the above-mentioned analogy is adopted as the fundamental basis due to a number of reasons. First, the approach is objective and mathematically rigorous and allows us to relate structural complexity to system development effort in a quantitative and explicit fashion. According to [19], the topological complexity metric is valid as it is compliant with Weyuker's criteria [68], which provides a set of properties of syntactic complexity measures. This indicates that the graph energy metric is computable and practical for large systems and independent of the level of abstraction used to represent the system under consideration [19]. This can be seen as an important advantage in comparing manufacturing system topologies modelled at different level of abstractions, which is valid for most manufacturing system development projects. Moreover, the mathematical model has been successfully used to assess structural complexity of various engineering systems, such as jet engines [19], and printing systems [11], etc. In addition to this, the approach is generic and systemic and can be adapted and customised for different engineering systems with network like topology. As an example, the same approach is adapted to assess assembly complexity of industrial products in [69, 17, 70]. The mathematical model is also well-aligned with the component-based design paradigm which is a widely used design methodology in manufacturing systems engineering. This design paradigm is the fundamental approach used in the engineering software called as the vueOne virtual manufacturing tool-set, within which the mathematical model for measuring complexity will be integrated in the next chapter.

3.2. *Adoption to component-based automation systems*

During early design phases of the manufacturing systems, the main aim is to identify the overall structure through the decomposition of its functions into sub-functions [71], and through finding the suitable components that can realise corresponding sub-functions [72]. In this context, overall architecture of the system includes not only geometric information, but also non-geometric phenomena such as control architecture and its relations to the overall system architecture [18]. According to the V-Model of system development, conceptual design phase is called as "*system architecting*", in which the system requirements are identified, and distributed into modules and further into components [18]. At the lowest layer of decomposition, all sub-functions should be realised by essential entities called as "*components*" [72]. According to Komoto and Tomiyama [18], "*components are called as machine elements, established components, and mechanisms in mechanical design, and fundamental building blocks i.e. state transitions diagrams that represent sensors, actuators, and controller behaviours, in control design* etc.".

The presented study defines a CBAS as a constellation of basic components which can be represented as a series of connected domains (**Figure 2**). These domains include: mechanical, pneumatic, electrical, logical, control, safety, etc., in which components can interact with other components within and between those domains. In here, a component is a basic unit of the system which at a finer level may compose of a set of indecomposable elements, and capable of functioning either as a single entity and/or integrated with other components to perform the

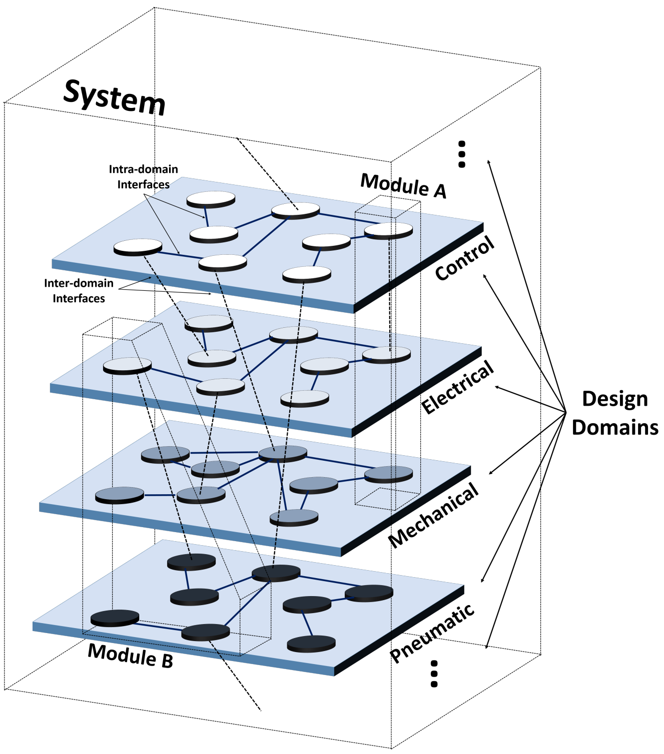

Figure 2: System-of-systems representation of component-based automation systems (cylinders represent system components).

required functions [73]. Components have standardised interfaces and explicit dependencies which can be deployed independently and are subjected to compositions to build an automation system [74]. The integration of basic components within a specific domain results in a specific architecture of the system, e.g. electrical system, etc. However, the integration of individual domains results in a final system architecture, where the system behaviours can be realised in a controlled and synchronised manner.

Having defined the reference model for CBAS architectures, Sinha and de Weck's [63] analogy can be adopted. Let's consider a system (S) composed of N number of sub-systems represented in different design domains. In here, the connectivity matrix of the resultant topology is defined as follows:

$$\Lambda = \begin{vmatrix} D_1 & \cdots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \cdots & D_N \end{vmatrix} = \begin{vmatrix} D_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D_N \end{vmatrix} + \begin{vmatrix} 0 & \cdots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \cdots & 0 \end{vmatrix} \tag{13}$$

where, Λ is the adjacency matrix of the resultant system-of-systems, D_k is the adjacency matrix of the domain k, and K_{ij} is the inter-domain connectivity matrix between domains i and j. Accordingly, the graph energy of the overall system architecture can be written as follows:

$$E_\Lambda = \sum_{i=1}^{N} E_i + \Delta \tag{14}$$

where, E_i is graph energy of i^{th} intra-domain connectivity and the term Δ represents the graph energy originated from the inter-domain connectivity. The resultant complexity of the system architecture C, thus, can be defined as follows.

$$C = \sum_{i=1}^{N} C^i + C_\Delta \tag{15}$$

where, C^i is overall structural complexity of i^{th} sub-system in isolation, and C_Δ represents topological complexity induced by the inter-domain connectivity. By re-writing the original metric, structural complexity of individual sub-systems can be written as follows.

$$C^i = C_1^i + C_2^i C_3^i \tag{16}$$

$$C^i = \sum_{j=1}^{m_i} \alpha_j^i + [\sum_{k=1}^{m_i} \sum_{l=1}^{m_i} \beta_{kl}^i] \frac{E_i}{\sum_{i=1}^{N} m_i} \tag{17}$$

where, α_j^i is the complexity of the j^{th} component represented in the i^{th} sub-system, β_{jk}^i is the complexity of the interface between components k and l in the i^{th} sub-system, m_i is the total number of components represented in the i^{th} sub-system, and E_i is the graph energy of connectivity matrix of i^{th} sub-system.

In the equation, structural complexity of the resulted architecture is also influenced by an additional element called as "*integrative complexity*" C_Δ, which is a direct result of two factors, i.e. complexity of inter-domain interfaces and graph energy of the inter-domain connectivity Δ [19]. The integrative complexity is defined as follows:

$$C_\Delta = \sum_{i=1}^{N} \sum_{j=1}^{N} [[[\sum_{k=1}^{m_i} \sum_{l=1}^{m_i} \beta_{kl}^i] \frac{E_j}{\sum_{i=1}^{N} m_i}] + \begin{matrix} [[\sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \beta_{kl}^{ij}] \frac{E_i + E_j}{\sum_{i=1}^{N} m_i}] + \\ [[\sum_{k=1}^{m_i} \sum_{l=1}^{m_i} \beta_{kl}^i] \frac{E_{ij}}{\sum_{i=1}^{N} m_i}] + \\ [[\sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \beta_{kl}^{ij}] \frac{E_{ij}}{\sum_{i=1}^{N} m_i}]] \end{matrix} \quad , i \neq j \tag{18}$$

where, β_{kl}^{ij} is the interface complexity between k^{th} component in i^{th} layer and l^{th} component in j^{th} layer.

4. An automated framework for complexity assessment

The use of theoretical complexity models can be time consuming and tedious, especially in large scale design projects, where a significant amount of data collection and analysis are required. Hence, there is a need for practical tools and methods that designers and managers can use concurrently with the design process, so that, conceptual designs can be improved, or compared with various design alternatives for a better design solution. This chapter presents a complexity-inclusive design support framework (**Figure 3**) which is achieved by the integration of theoretical model explained in the previous chapter, with a virtual system design and development software, namely: the vueOne virtual engineering (VE) tool.

4.1. vueOne virtual engineering tool

The vueOne VE tool is designed upon the "*component-based*" design paradigm, and is primarily used for the virtual commissioning of automation systems supported by integrated components which are dedicated to performing a set of specific functions [75]. In the vueOne, a component is defined as a reusable, reconfigurable building block of the automation system, providing a data integration mechanism for control behaviour, kinematics, geometries, and other data types defining a particular system resource [76]. Data which is encapsulated within a component can exist at a particular level of granularity [77], which is defined by the user.

Presently, vueOne tool set delivers functions such as: 3D modelling, process simulation and evaluation, auto-control code generation, but not complexity assessment functionality. The vueOne uses the standard Virtual Reality Modelling Language (VRML) format for 3D modelling, and common state-transition diagrams for module logic editing and visualisation. The vueOne tool set supports modelling of several types of components such as: sensors, actuators, digital human workers, robots and fixtures [74]. In the tool, a common component architecture is used to integrate component geometry, kinematic and control behaviours. Each component created by the tool has a unique ID which can be used in identification and debugging purposes. Component and systems can be stored in the library with any information associated with component parameters, and can be reused any number of times. The stored information can also be linked to component performance.

4.2. Complexity solver

In the proposed framework, virtual design data generated at the system modelling phase are streamlined into a Matlab plug-in called as "*complexity solver*". The complexity solver has four main modules: complexity engine, complexity optimiser, complexity database and graphical user interface.

The *complexity engine* is a software module, where structural complexity of virtual system designs can be assessed by means of the definition presented in the previous chapter. The complexity engine, in its current form, only reads outputs from of the vueOne tool written in the XML format. However, the solver can be integrated with other manufacturing system design tools/methods to enrich the virtual models. This is particularly true for detailed design stages, where various domains of system design e.g. electrical system complexity, communication system complexity, etc., are defined and assessed. As more data is fed to the solver, design complexity can be analysed at a higher resolution, as more component and interfaces from multiple domains can be modelled.

In its current form, the complexity engine is able to investigate structural complexity of virtual system designs within

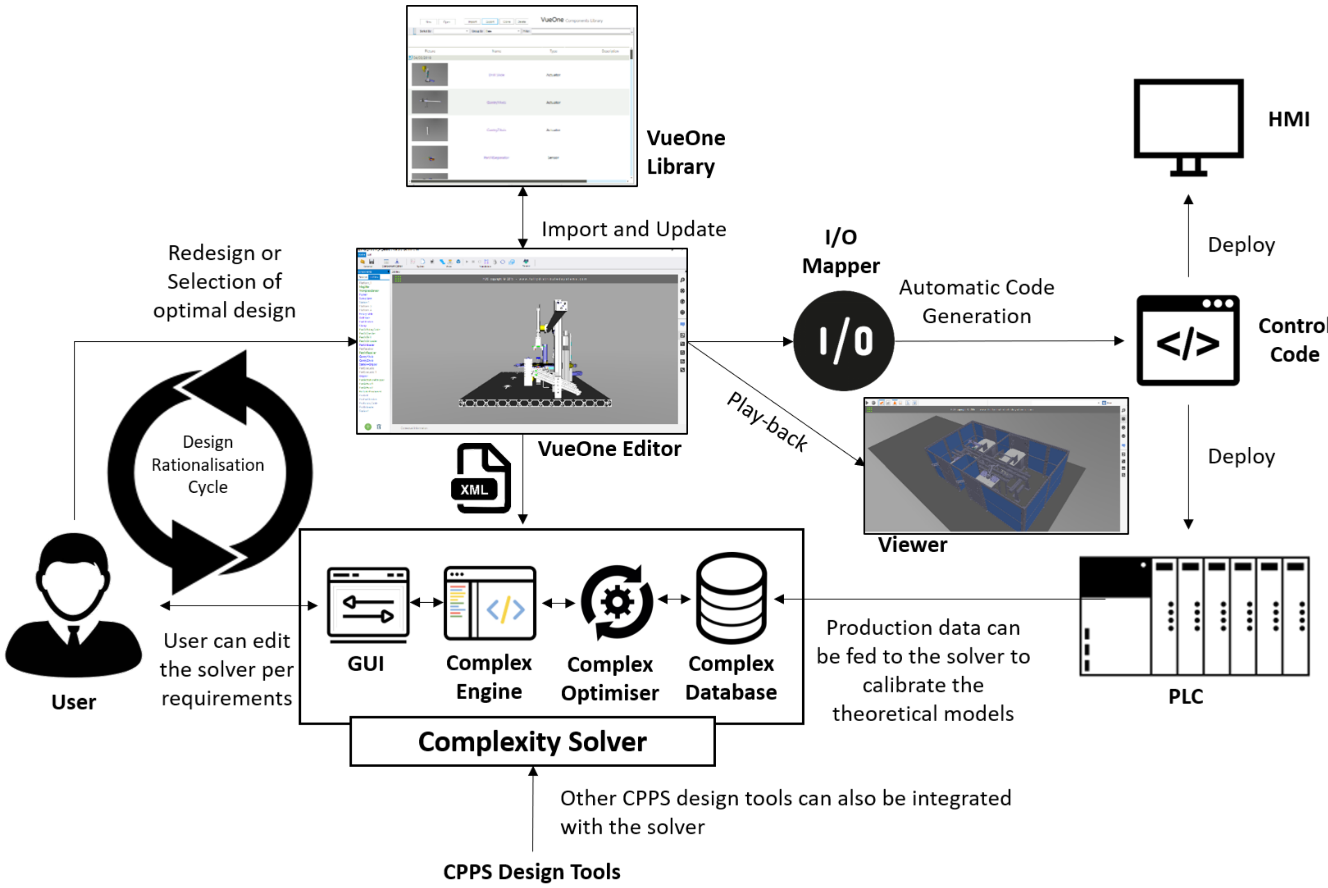

Figure 3: Complexity modelling and management framework integrated into the vueOne virtual engineering tool.

two inter-connected domains (i.e. mechanical and logical), as vueOne only allows modelling of kinematic relationships and process behaviours of components. This is reasonable as these domains are primarily important for the system architecting phase. A justification for the narrowing down of scope to two layers is provided in Section 1.3. In the vueOne tool, a component may be represented by a set of connected geometries and a finite-state machine. In the complexity solver, component geometries and finite-state machines are denoted as separate system components, as this assumption allows us to analyse structural complexity within a two-layered network, hence providing a better resolution (**Figure 4**). Mechanical design of an automation system is composed of a set of mechanical components connected to each other through a number of mechanical interfaces. In here, a mechanical component is the core constituent of the system, which has to be installed and commissioned as a part of the system development phase. These components include mechanisms required to transfer parts with the workstation, holding equipment e.g. jigs, fixtures and clamps, buffering and storage components and work holders e.g. grippers, etc. The logical system design (resembling the ISA-95 level 2a and above (see [78])), on the other hand, can consist of a number of logical components defining management and control policies needed to govern the states and behaviours of mechanical components. In the vueOne, component behaviours are outlined via state transition diagrams (STDs) conforming IEC 61131-3 standards [79], and so PLC code can be automatically generated and deployed to support a basic level of virtual commissioning. The STD within the vueOne has three types of states: *i*) initial state, *ii*) static state, and *iii*) dynamic state. Please note that, a vueOne component may have either one of those mechanical or logical components or both. As an example, a fixture can be only defined by mechanical components, whereas, a robot contains both mechanical and logical components in its design. Moreover, mechanical and logical components are assumed to be mapped by electrical dynamic interfaces. This interface type represents the data transfer between logical and mechanical domains, and can be counted as a directional relationship. Moreover, this relationship does not necessarily to be one-to-one mapping; i.e. the logical behaviour of a mechanical component can be defined using multiple logical components or multiple mechanical components can be controlled by a single logical component, *etc*. **Table 1** summarises the interface types that can be defined within the vueOne tool.

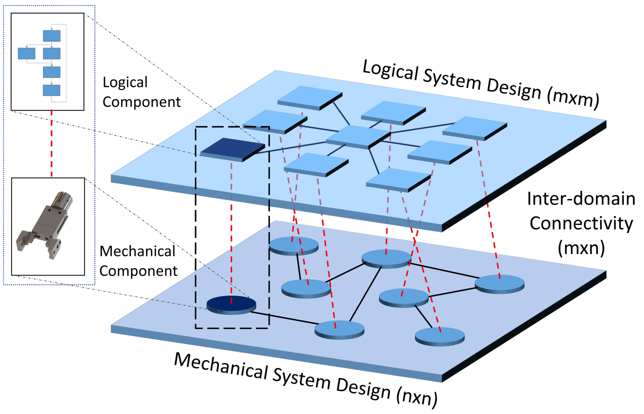

Figure 4: Virtual system design in the vueOne VE tool and module boundaries.

4.3. Estimation of complexity elements

Once the virtual design and the end-user specifications are defined and imported, the complexity engine reads the XML file, and then decomposes automation system into mechanical and logical components. Based on the interface information stored in the XML document, the engine builds the design structure matrix (DSM) of the resultant architecture. This information is used to estimate topological complexity of mechanical, logical and inter-domain system connectivities. The XML document also contains kinematic and control information. This data is used to estimate component complexities in the mechanical and logical domains. These values are then used to estimate domain specific design complexities as well as integrative complexity. Please note that, the above mentioned elements of complexity are different in nature, therefore, overall system complexity is considered as a multi-dimensional value, rather than a single number.

$$C^S = f(C^M, C^L, C^\Delta) \tag{19}$$

4.3.1. Estimation of mechanical component complexities

In the proposed approach, inherent complexity of mechanical components is associated with the information required to define their kinematics chain. It is assumed that mechanical components are standard off-the-shelf products, which are ready for the system integration and their inherent structures are hidden and composed of a number of indecomposable parts. Accordingly, complexity of a mechanical component α_i^M is defined as follows:

$$\alpha_i^M = 5(1 - e^{-(k^M N_i^M)}) \tag{20}$$

where, N_i^M is number of kinematics associated with the component i and k^M is an exponential function parameter ($k \in [0,1]$). **Figure 5** is a surface map defining the relationship between total number of associated kinematics and exponential function parameter k^M. This figure depicts that, for a constant N_i, component complexity will grow to positive 5, as k^M reaches a critical threshold. This threshold represents the point where an individual starts perceiving mechanical design of a component as complex, and varies depending on the individual's experience and ability to cope with complexity.

4.3.2. Estimation of logical component complexities

In a similar way, complexity of logical components is defined as the relative effort required to develop, maintain and comprehended the finite state machine in a software engineering perspective, and estimated as follows:

$$\alpha_i^L = 5(1 - e^{-(k^L N_i^L)}) \tag{21}$$

Table 1: VueOne compatible interface types.

Interface type	Domain	Description
Steady state structural	Mechanical	defines contacts between two components where they impose a steady state mechanical load on each other
Dynamic state structural	Mechanical	defines the fluctuating force or displacement between two components
Spatial connections	Mechanical	define a relationship between two components when adjacency and orientation are important between them
Part transfer	Mechanical	represents material transfer/exchange between two components.
Event exchanges	Logical	are required to verify that correct precedence relationships are obeyed throughout the operation
Safety interlocks	Logical	are used to help prevent a component from harming the operator or damaging itself by preventing one component from changing state due to the state of another component, and vice versa.
Electrical dynamics	Inter-domain	represent any type of interactions between logical and mechanical components

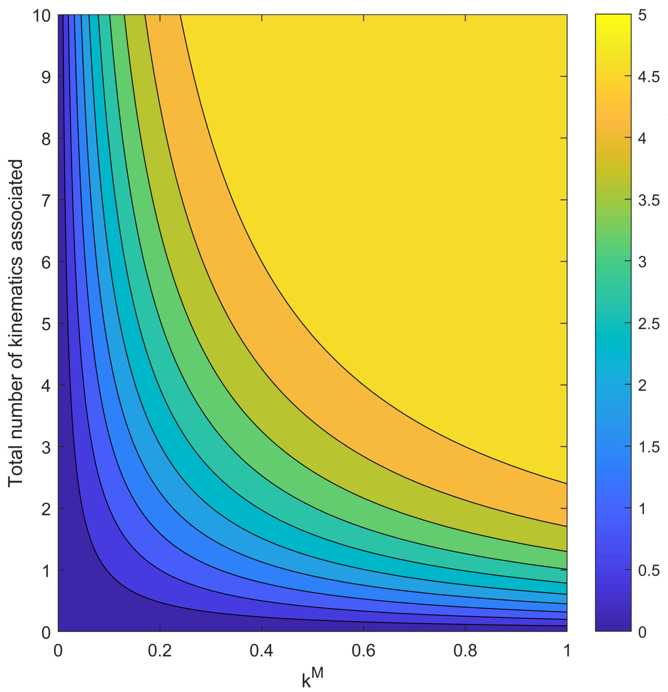

Figure 5: Surface plot of component complexity induced by its mechanical design

where, N_i^L is number of states exist in i^{th} logical component, and k^L is the exponential function parameter ($k^L \in [0, 1]$). Similarly, k^L is the exponential function parameter defining the individual's ability in comprehending (modifying or debugging) the finite-state machine.

4.4. Visualising structural complexity

Increasing number of components and numerous interactions with different kinds of flows exchanged between them increases overall system complexity considerably. An increased complexity may result in reducing system efficiency, therefore, should be reduced without compromising the functional requirements. The complexity solver is able to visualise the structural complexity of virtual system designs, and thereby allowing a general awareness about excessive complexity in an explicit fashion. A graphical user interface (GUI) is designed for this purpose (**Figure 6**). The GUI allows designer to import virtual design data written in an XML format, and visualises complexity results in both graphical and textual formats. Moreover, the developed GUI can be used in comparing alternate system designs in a quantitative fashion.

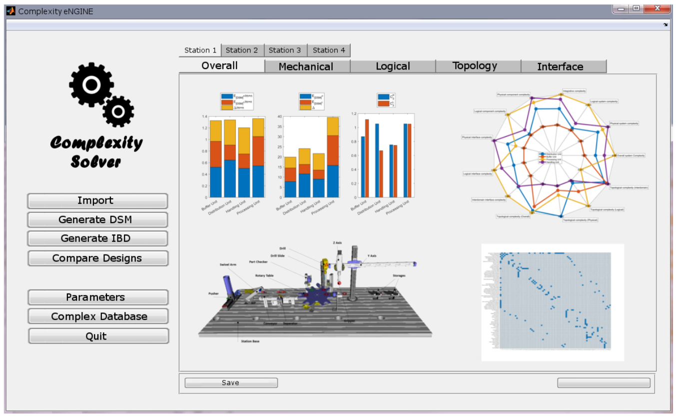

Figure 6: Complexity solver GUI.

5. Refining the values of the complexity model parameters

The model parameter values have to be specified before the proposed framework can be used as a practical design validation tool. Towards this aim, an optimiser module, where correct values for the complexity model parameters can be obtained through the use of historical production data and user defined inputs, has been added to the proposed complexity solver. **Figure 7** shows the data flow of the optimisation module and its interactions with the proposed complexity solver. The developed optimisation module simply relates system development effort to system complexity, and attains suitable complexity model parameter values through the minimisation of the sum of squared deviations between these two. The optimiser tool, currently, depends on subjective expert opinions to select the values of the complexity model parameters, however, if integrated and streamlined, production system data, i.e. mean time between failures, mean time to repair and deployment time, etc., filtered and processed through data analytics and machine learning techniques, can be used in predicting the complexity model parameter values with better accuracy.

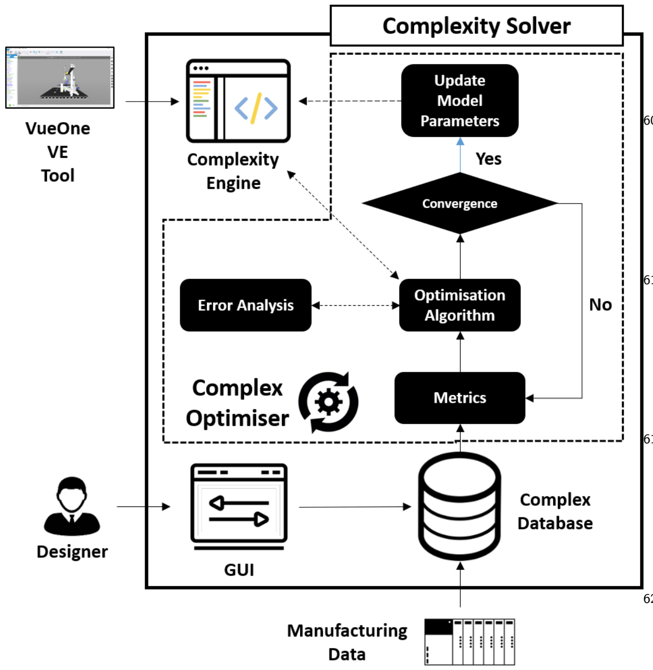

Figure 7: The data flow of the optimisation of model parameters.

5.1. Preliminary experiments

Preliminary experiments were conducted in an effort to validate the functionality of the proposed optimiser module with a number of virtual system designs with varying degrees of structural complexity. Since no accurate information is available regarding the *absolute complexity* of the production systems i.e. system development effort or time, it was decided that it would be best to ask a team of engineers to jointly nominate both complex and simple system designs. A two point scale (0 or 1) was used to represent subjective opinions of experts. The nine participants, who are currently serving as system engineers on a variety of large and high-profile research projects, were drawn from Automation Systems Group (ASG) at the University of Warwick. The participants are thoroughly experienced employees with at least five years of industrial experience. In the experiments, the participants were asked to subjectively assess structural complexity of thirty virtual automation system designs modelled in the vueOne VE tool-set. These stations perform operations in automotive power-train and electric battery assembly applications. In these experiments, overall complexity of an automation system was assumed to be a function of mechanical system complexity, logical system complexity and integrative system complexity. The importance of these complexity elements were assumed to be equal. **Table 2** shows the specifications of virtual automation system designs considered in the optimiser tool and the participants' subjective opinions.

5.2. Objective function

Objective function is designed as the sum of squared errors between actual and predicted complexity scores. In here, subjective expert opinions are used as an actual complexity indicator, as 1 and 0 represents highly complex and simple workstation designs, respectively. Mathematically, the objective function is expressed as follows:

$$Min: f_{(x)} = \sqrt{C_{error}} \tag{22}$$

Table 2: Virtual system designs used in the experiments and experts' opinions

Design	Number of Mechanical Components	Number of Logical Components	Number of Kinematics per Component	Number of States per Component	Number Mechanical Interfaces	Number Logical Interfaces	Number Inter-domain Interfaces	Subjective Opinions about Mechanical System Complexity S^M	Subjective Opinions about Logical System Complexity S^L	Subjective Opinions about System Integration S^Δ
1	6	4	1.667	4.250	8	8	3	LOW (0)	LOW (0)	LOW (0)
2	7	5	0.429	3.200	10	7	5	LOW (0)	LOW (0)	LOW (0)
3	10	8	0.800	4.625	17	19	7	LOW (0)	LOW (0)	LOW (0)
4	10	8	1.300	4.625	14	21	7	LOW (0)	LOW (0)	LOW (0)
5	14	10	0.429	3.300	19	25	8	LOW (0)	LOW (0)	LOW (0)
6	14	11	1.214	3.182	28	35	9	LOW (0)	LOW (0)	LOW (0)
7	18	7	1.111	3.429	40	15	6	LOW (0)	LOW (0)	LOW (0)
8	18	11	0.833	3.182	25	22	8	LOW (0)	LOW (0)	LOW (0)
9	20	10	0.500	4.546	25	18	7	LOW (0)	LOW (0)	LOW (0)
10	25	14	0.731	4.857	51	30	18	LOW (0)	LOW (0)	LOW (0)
11	25	16	1.000	5.118	40	35	13	LOW (0)	LOW (0)	LOW (0)
12	28	15	0.517	4.333	40	35	15	LOW (0)	LOW (0)	LOW (0)
13	29	14	1.379	3.800	55	45	11	HIGH (1)	HIGH (1)	LOW (0)
14	30	17	0.567	4.889	60	38	15	LOW (0)	LOW (0)	LOW (0)
15	33	20	0.697	4.250	67	48	18	LOW (0)	HIGH (1)	LOW (0)
16	35	9	0.629	7.111	50	21	9	LOW (0)	LOW (0)	LOW (0)
17	37	13	1.054	3.539	56	28	7	HIGH (1)	LOW (0)	LOW (0)
18	41	22	0.610	3.046	80	55	12	LOW (0)	LOW (0)	LOW (0)
19	45	30	0.756	5.067	109	80	17	HIGH (1)	HIGH (1)	LOW (0)
20	50	34	0.840	4.925	75	88	29	HIGH (1)	HIGH (1)	HIGH (1)
21	51	14	0.686	4.071	68	58	11	LOW (0)	LOW (0)	LOW (0)
22	55	30	0.691	5.200	140	70	21	HIGH (1)	HIGH (1)	LOW (0)
23	56	35	0.732	4.486	110	51	29	LOW (0)	HIGH (1)	LOW (0)
24	60	38	0.705	5.020	157	100	40	HIGH (1)	HIGH (1)	HIGH (1)
25	61	16	0.656	4.647	147	70	11	HIGH (1)	LOW (0)	LOW (0)
26	65	35	0.908	5.543	190	98	20	HIGH (1)	HIGH (1)	LOW (0)
27	69	39	0.710	4.255	210	102	45	HIGH (1)	HIGH (1)	HIGH (1)
28	70	39	0.614	4.256	200	158	40	HIGH (1)	HIGH (1)	HIGH (1)
29	71	45	0.718	4.550	210	120	57	HIGH (1)	HIGH (1)	HIGH (1)
30	72	19	0.764	5.158	122	82	10	HIGH (1)	LOW (0)	LOW (0)

where;

$$C_{error} = \sum_{i=1}^{30}[(100S_i^M - C_i^M)^2 + (100S_i^L - C_i^L)^2 + (100S_i^\Delta - C_i^\Delta)^2] \tag{23}$$

and, S_i^M, S_i^L and S_i^Δ are subjective ratings assigned to the mechanical design, logical design and integration complexities of i^{th} virtual system design.

5.3. Design variables

From the theoretical definition, it is clear that structural complexity is affected by exponential function parameters, i.e. k^M and k^L, and interface coefficients. To simplify the optimisation problem, interface coefficients are considered within three categories: *i*) mechanical interfaces c^M, *ii*) logical interfaces c^L, and *iii*) inter-domain interfaces c^Δ. Based on this assumption, the final single-objective optimisation model is developed for five design variables: k^M, k^L, c^M, c^L and c^Δ. All design variables are continuous and are subjected to the following design constraints.

$$g_1(x) = 0.9 \geq k^M > 0.1 \tag{24}$$

$$g_2(x) = 0.9 \geq k^L > 0.1 \tag{25}$$

$$g_3(x) = 0.25 \geq c^M \geq 0.05 \tag{26}$$

$$g_4(x) = 0.25 \geq c^L \geq 0.05 \tag{27}$$

$$g_5(x) = 0.25 \geq c^\Delta \geq 0.05 \tag{28}$$

5.4. Genetic algorithm

The proposed optimiser module uses a single-objective genetic algorithm (GA) method (please see [80]). GA is an evolutionary computation technique that belongs to the class of heuristic optimisation. It uses the natural selection mechanism and is particularly useful when the search space is large and not much is known about solving the problem. Genetic algorithms are used to solve a wide range of problems and they employ the processes of reproduction, selection, crossover and mutation. Each step of a genetic algorithm involves the production of offsprings from the current population. In this study, GA parameters are selected as follows: the population size is 80, mutation rate is 0.25 and cross over rate is 0.75 and the total number of iterations is 1000. Selection of parents is performed by means of Roulette wheel selection method in which multi-point crossover is employed. The optimal results obtained by the GA method are presented in **Table 3**.

Table 3: The optimisation results.

Variable	Value
k^M	0.44126954414429510
k^L	0.14880787695922565
c^M	0.11567665722511014
c^L	0.05311016162808964
c^Δ	0.08052666454755751

5.5. Logistic regression model

In this subsection, a statistical model relating the predicted scores achieved through the complexity engine and expert opinions is proposed. Since, there are only two (dependent) responses (HIGH or LOW), a linear regression is not appropriate for a statistical model. There exists, however, a model, called "Logistic Regression" or "Logit", that will calculate the probability that the resulting score of the system design indicates either HIGH or LOW complexity. Please note that, the value 1 is assigned to stations labelled as HIGH. Accordingly, the "LOGIT" model for all three complexity sources are found as follows:

$$P_{LOGIT}^{M}(High) = \frac{1}{1 + e^{9.9582 - 0.1755C^M}} \tag{29}$$

$$P_{LOGIT}^{L}(High) = \frac{1}{1 + e^{6.4281 - 0.1171C^S}} \tag{30}$$

$$P_{LOGIT}^{\Delta}(High) = \frac{1}{1 + e^{6.2827 - 0.1954C^\Delta}} \tag{31}$$

Table 4 contains the numerical results, as output by SPSS software. The model succeeds in classifying the cases 93.3% of the time correctly. **Figure 8** shows the binary fitted line plots of logistic regression models. Although the study was carried out using a limited sample size, the results show us that the optimiser module can be used to refine the theoretical model. It was also noticed, the mathematical model can be refined using a collection of historical data to ensure that complexity measurement and system characteristics can be properly correlated. The scores from the model could be used as independent variable for researching the impact of complexity on both direct and indirect costs, and on the subjective interpretation of complexity.

6. Traffic light system

One of the major goals in system development, is to verify structural design complexity so as to keep the dynamic and emergent complexity of the system under control. To provide a more industry-friendly complexity assessment practice, a traffic light based system is added to the solver, where system designs can be categorised based on three levels of structural complexity:

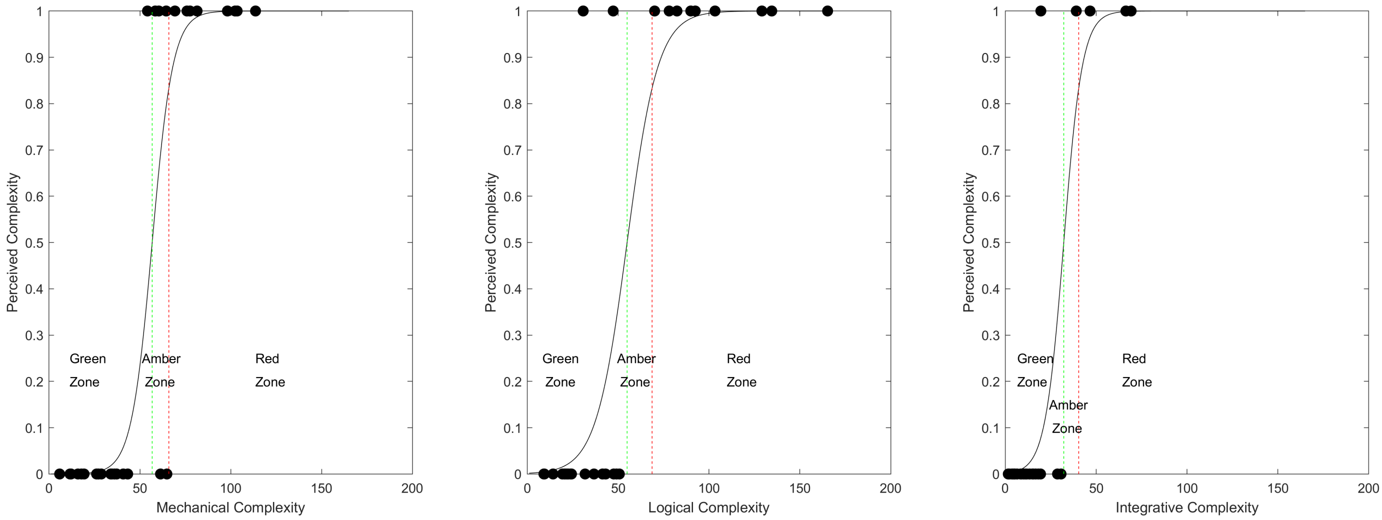

Figure 8: Binary fitted line plots for logistic regression models.

Table 4: Classification table.

	Observed	Predicted		% Correct
		High	Low	
Mechanical Complexity	High	11	1	91.7
	Low	2	16	88.9
Logical Complexity	High	9	2	81.8
	Low	0	19	100.0
Integrative Complexity	High	4	1	80.0
	Low	0	25	100.0
Overall				93.3

The cut value is 0.5.

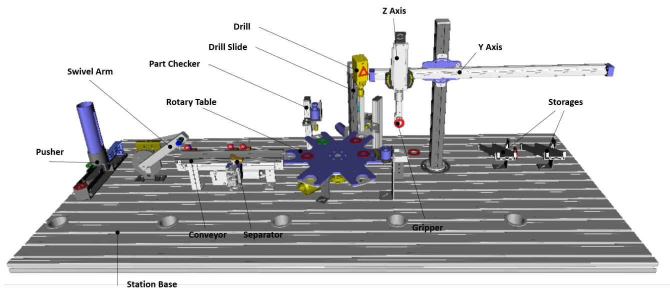

Figure 9: Virtual model of the Festo MPS.

- Green: ($P^k_{LOGIT}(High) < 0.5$) The complexity of the system design is manageable with sufficient level of personnel and expertise.
- Amber: ($0.75 > P^k_{LOGIT}(High) \geq 0.5$) The development and management of the workstation can be costly and time consuming, therefore allocation of the resources and expertise should be considered.
- Red: ($P^k_{LOGIT}(High) \geq 0.75$) The management of workstation complexity can be tedious with current personnel and expertise, therefore, workstation design should be reconsidered/simplified without promising the functional requirements, or its functions should be embodied into smaller workstations.

7. Case study

In this section, the proposed complexity assessment framework is tested using Festo Modular Production System (MPS) (**Figure 9**). Main operation of the Festo MPS is to move work pieces from one end to another by performing a number of sequential operations. The test rig contains four modules, i.e. distribution, buffer, processing and handling. The distribution module includes a pneumatic feeder and a converter, transports work pieces from the magazine stack to the buffer module [15]. Buffer module consists of a conveyor system and a separator actuator, which are used to transport and separate out work pieces. After passing the buffer module, work pieces are forwarded to the rotary table of the processing module, where a drilling operation is performed. At the end, a handling module removes parts from the processing station and sorts them according to their physical characteristics, e.g. shape and colour.

7.1. Component complexities

The Festo MPS is virtually modelled in the vueOne VE toolset, and its design XML is imported to the proposed Matlab plug-in. Process logic is defined by logically coupling component state machines and included in the design XML. In the logical design, process orchestrators are used to regulate a group of logical components. Please note that, mechanical and logical architecture of the test rig was mapped through electrical dynamics interfaces (directly between state machines and machine kinematics). **Table 5** shows the mechanical and logical component complexities for Festo MPS and its modules.

According to the results, the processing module has higher total component complexity in both the mechanical and logi-

Table 5: Total component complexities C_1 in the mechanical and logical domains.

	m	Mechanical Complexity	n	Logical Complexity
Distribution module	11	5.351710392	7	13.13348094
Buffer module	9	3.567806928	6	11.89743088
Processing module	15	8.91951732	14	26.78925825
Handling module	12	6.499151542	6	12.9555085
Overall system	48	24.33818618	33	64.77567857

cal domains, whereas, the buffer module has lowest complexity. This, in fact, is as expected since the processing module comprises relatively more functionality (i.e. feeding, drilling, checking and unloading), and is composed of comparatively high number of components ($m = 15, n = 14$). This is also in line with the hypothesis stating that functionality and complexity have a positive correlation. Accordingly, if a system has to perform a wide range of functionality or designed to support wide range of applications, it will likely have a complex structural composition.

7.2. Interface complexities

In the complexity engine, two components are considered connected, if at least one connection exists between them. By considering the three types of connections (i.e. mechanical, logical and inter-domain), a multi-domain matrix (MDM) was built. **Figure 10** shows the MDM of the test rig, where a marking in an off-diagonal cell indicates at least one interaction between two components.

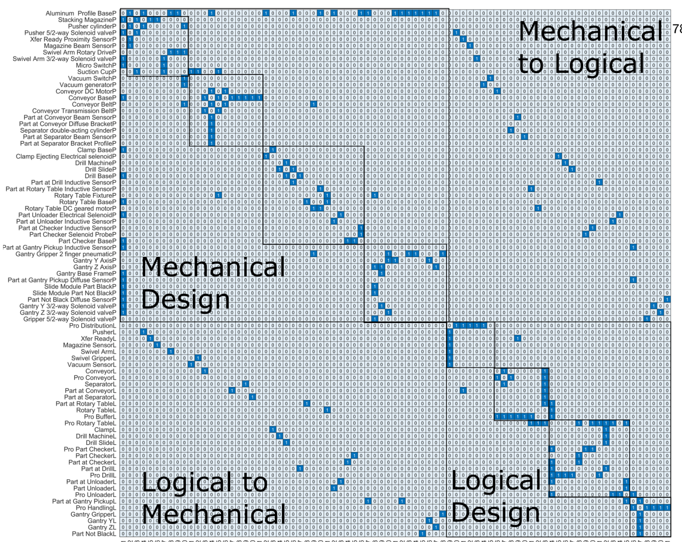

Figure 10: Multi-domain matrix representation of the Festo MPS.

Complexity of pair-wise component interfaces was calculated by the approach presented in Section 3.1.2. Interface complexity results of Festo MPS are given in **Table 6**. It is noted from the table that, the handling module has the highest interface complexity in the mechanical domain, whereas the buffer module has the lowest interface complexity. On the contrary, the processing module has a relatively higher interface complexity on the logical domain. This is again, a result of the number of functions that the module has to perform, i.e. more number of logical interlocks is required to control a wide range of applications in a synchronised manner. It is also noted that, the contribution of interface complexities are considerably small, indicating that structural complexity is predominantly affected by components' inherent complexities.

Table 6: Total interface complexity (Festo MPS).

	Mechanical	Logical	Inter-domain
Distribution module	0.5233	0.782169274	0.706093899
Buffer module	0.1195	0.307547026	0.357062599
Processing module	0.326875	1.708771381	1.002352501
Handling module	0.86925	1.350192349	0.689994316
Overall system	2.36055	3.438602915	2.755503314

7.3. Topological complexities

Based on the DSM analysis, complexity C_3 of the overall Festo MPS structure is found as 1.429 with a graph energy $E_{[MDM]}$ of 115.719 ($E^M_{[DSM]}$=53.004, $E^L_{[DSM]}$=33.490, and Δ=29.014). The contribution of mechanical, logical, and inter-domain topologies to the overall topological complexity is found to be 0.654, 0.413, 0.358, respectively. In a similar fashion, topological complexity of isolated mechanical C^M_3 and logical system architecture C^L_3, without considering the effects of inter-domain connectivity, are found as: 1.104 and 1.015, respectively. This points out a transitional regime between hierarchical and centralised structure patterns for stand-alone mechanical and logical system architectures. **Figure 11** compares the topological complexity of Festo MPS modules. According to

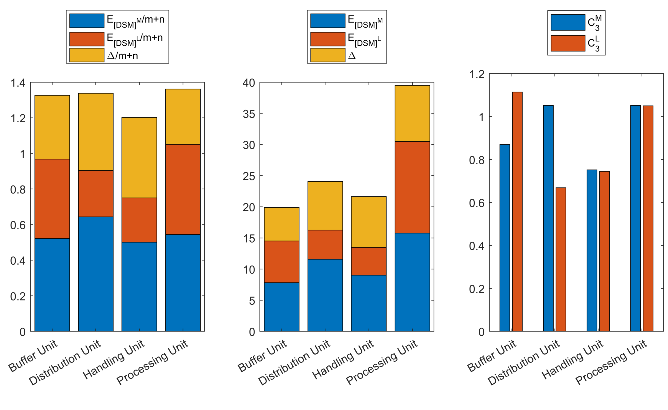

Figure 11: Comparison of Festo MPS modules: *left*: Overall topological complexity of modules, *middle*: graph energy results, *right*: topological complexity of isolated mechanical and logical architectures for each modules.

the results, overall topological complexity for all modules was found to be above one. This indicates a hierarchical connectivity pattern for all cases. Interestingly, topological complexity of logical architectures (considered in isolation) for distribution

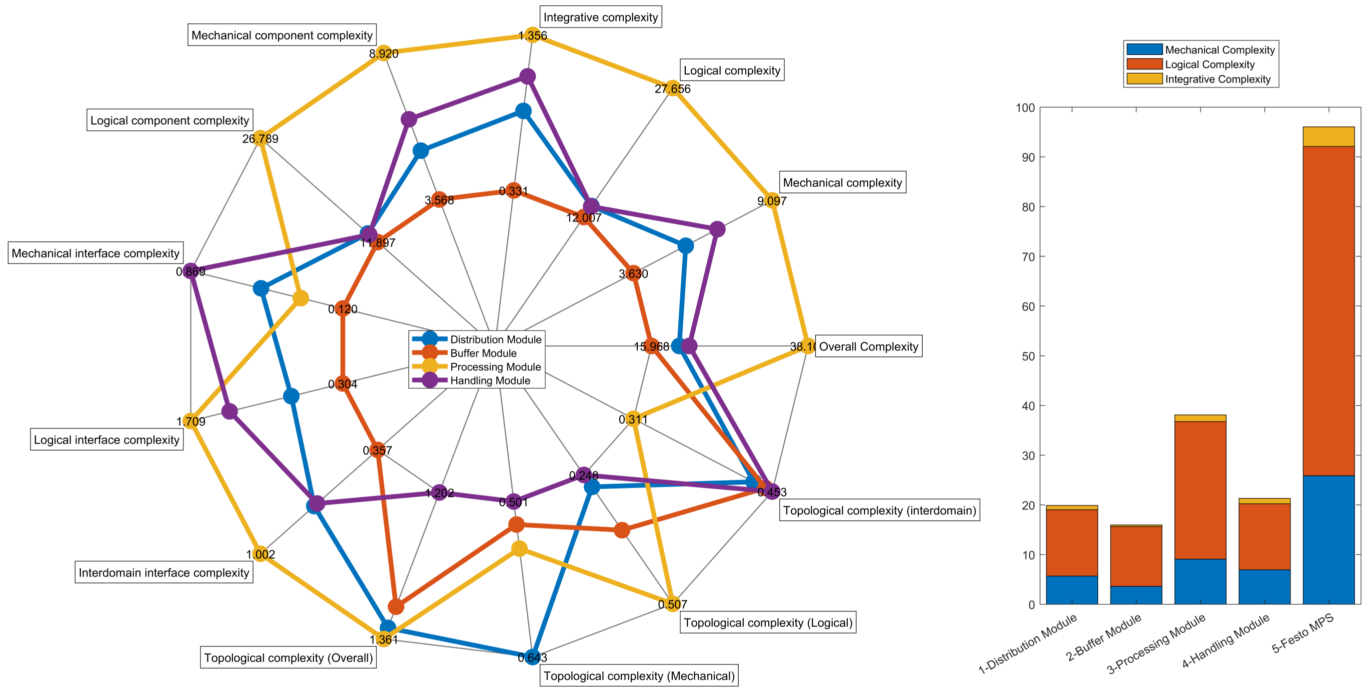

Figure 12: Complexity comparison of the Festo MPS modules

and handling modules are found to be below "one" indicating a centralised topology (**Figure 11** ***right***). This is reasonable as these subsystems are controlled by one process orchestrator, whereas, the number of logical components that are controlling the operation behaviours of buffer and processing subsystems are two and three, respectively.

7.4. Overall complexity

Overall complexity of Festo MPS is found as 96.028, of which the contribution of mechanical, logical and integrative system complexities are recorded as 25.882, 66.196 and 3.950, respectively. This shows that the vast majority of the system complexity stems from the complexity of the logical architecture. According to the results, overall mechanical and integrative system designs are considered in "*green zone*" indicating a design that can be easily managed through existing skill, expertise and personnel. However, logical system design is found in "*amber zone*". This indicates a tedious and tricky development stage, thereby requiring proper attention and resource allocation, i.e. people with expertise and training.

Figure 12 displays the overall difference between categories with multiple complexity elements. As it is expected, the processing module is found to have the most complex design (C = 38.109), whereas the buffer subsystem is found to be the simplest (C = 15.986). The complexity of the processing module is a result of the logical architecture rather than the mechanical system as seen from **Figure 12**. The approach, in addition to providing the overall system complexity value, is capable of indicating the source of complexity with a good degree of resolution. It is to be noted that, if the logical architecture has a high value it is to be expected that the programming of the process sequence and its logic will be complicated. On the other hand, a high value of mechanical system complexity represents the difficulty in integrating the associated components. The results were also presented to the engineers who were involved in the system build and based on the feedback it is understood that, the presented complexity values are in agreement with the numbers that the engineers intuitively proposed as the system complexity.

The developed approach's sensitivity on the level of abstraction is also studied (please see [17]). It has been found that, there is noticeable differences in the results of component C_1 and interface complexity C_2 scores for systems with coarse/finer level representations. This is due to the fact that the finer representation of the system has large number of components and interfaces, which are modelled at a deeper/finer level of system decomposition. However, as the vueOne provides a standard platform for developing automation systems, it is still possible to compare alternate system designs in a fair manner. On the other hand, as stated by [19], the topological complexity C_3 does not seem to be affected by the level of abstraction. This is attributed to the fact that the basic structure of the system remains the same beyond a level of decomposition that adequately describes and differentiates the system. Hence, the topological complexity metric can be considered as a better solution to compare systems with different levels of granularity.

8. Conclusion

In this paper, a virtual engineering based approach is proposed to assess the structural complexity of virtually designed component-based industrial automation systems. The benefits of the proposed approach are threefold, i) complexity assessment can be performed in the early design phase, where any change in design and its corresponding change in complexity can be assessed with minimal implications on cost and time ii) the complexity inclusive design support approach is automated and hence eliminates the laborious manual work associated with the existing approaches iii) complexity assessment performed encompasses different domains, thereby allowing the detection of the exact source of complexity with a great level of detail which enables identification of critical points or aspects which could help to reduce complexity at the design stage. Furthermore, to refine the proposed approach by optimising its model parameters, a work group, comprising of system, mechanical and control engineers, was established and their opinions and intuitive knowledge on the complexities of different systems was used to create a statistical correlation between complexity scores and expert opinions. Moreover, Festo didactic test rig was used to demonstrate and provide a first-hand evaluation of the proposed complexity assessment framework. The results showed that, the approach is mathematically rigorous, and assess structural complexity at a high resolution over a broad spectrum ranging from topological complexity to mechanical and logical system components.

Although the benefits of the model have been quantified with the help of the test case, the component and interface complexities are found to be sensitive to the selected level of system decomposition. In other words, if two different systems are decomposed at different levels of granularity (i.e. coarse and fine decompositions), it would not be possible to compare them. Therefore, it is necessary to perform the comparison by establishing a standard during the modelling, to ensure comparison is done across similar level of granularity. Please note that, the topological complexity metric can be considered as an alternative, as it has a weak sensitivity to the selected level of abstraction. Moreover, a collection of historical data is required to calibrate and validate the model and to ensure that complexity measurement and manufacturing systems engineering characteristics can be properly correlated. As part of the future work, the plan is to build a structured database of information, collected from real manufacturing system design and development projects. Suitable connectivity needed for the reported gap can be fulfilled through the Industry 4.0 viewpoint. Collecting real time life-cycle parameters (i.e. mean time between failures, mean time to repair, etc.) of system components during the operation phase in a structured manner, allows us to better predict component complexities stored in the virtual component library. It is also planned to integrate the developed software solution with other system design and development tools, e.g. electrical system design software such as EPLAN or Edraw, pneumatic system design software such as PnueDraw, etc., in order to establish a high fidelity assessment approach providing a better resolution of complexity. Finally, the proposed complexity assessment framework is planned to be used in conjunction with the automatic selection of feasible system configurations. As part of this, information such as required functionality, maximum cycle time, flexibility, scalability, etc., are envisioned to be input into vueOne VM tool. The use of a Product Life-Cycle Management (PLM) database in conjunction with the virtual system design tool, will allow the automatic generation of several alternate designs that meet the above-mentioned criteria, subsequently creating a design space of valid architectures. Consequently, a system optimiser will be developed to compare these designs with the support of information stored in a database, thereby providing an optimum solution that meets the requirements.

Acknowledgements

This work was partially supported by Turkey's Ministry of National Education's Post-Graduate Scholarship.

References

[1] B. A. Weiss, M. Sharp, A. Klinger, Developing a hierarchical decomposition methodology to increase manufacturing process and equipment health awareness, Journal of manufacturing systems 48 (2018) 96–107.

[2] W. Elmaraghy, H. Elmaraghy, T. Tomiyama, L. Monostori, Complexity in engineering design and manufacturing, CIRP Annals - Manufacturing Technology 61 (2) (2012) 793–814. doi:10.1016/j.cirp.2012.05.001.

[3] B. Alkan, D. A. Vera, M. Ahmad, B. Ahmad, R. Harrison, Complexity in manufacturing systems and its measures: a literature review, European Journal of Industrial Engineering 12 (1) (2018) 116–150.

[4] N. Asadi, M. Jackson, A. Fundin, Implications of realizing mix flexibility in assembly systems for product modularitya case study, Journal of manufacturing systems 52 (2019) 13–22.

[5] A. C. Falck, M. Rosenqvist, What are the obstacles and needs of proactive ergonomics measures at early product development stages? - An interview study in five Swedish companies, International Journal of Industrial Ergonomics 42 (5) (2012) 406–415. doi:10.1016/j.ergon.2012.05.002.

[6] G. Grote, A participatory approach to the complementary design of highly automated work systems, Human Factors in Organizational Design and ManagementIV (1994) 115–120.

[7] M. Perona, G. Miragliotta, Complexity management and supply chain performance assessment. A field study and a conceptual framework, International Journal of Production Economics 90 (2004) 103–115. doi:10.1016/S0925-5273(02)00482-6.

[8] R. J. Urbanic, W. H. W. ElMaraghy, Modeling of Manufacturing Process Complexity, Advances in Design (2006) 425–436.

[9] G. Schuh, T. Gartzen, J. Wagner, Complexity-oriented ramp-up of assembly systems, CIRP Journal of Manufacturing Science and Technology 10 (2015) 1–15. doi:10.1016/j.cirpj.2015.05.007.

[10] E. Rechtin, Systems architecting: Creating and building complex systems, Prentice Hall, 1991.

[11] K. Sinha, O. L. de Weck, Structural complexity quantification for engineered complex systems and implications on system architecture and design, matrix 1 (2013) 1.

[12] S. N. Samy, T. Algeddawy, H. Elmaraghy, A granularity model for balancing the structural complexity of manufacturing systems equipment and layout, Journal of Manufacturing Systems 36 (2015) 7–19.

[13] M. H. Meyer, A. P. Lehnerd, The power of product platforms, Simon and Schuster, 1997.

[14] T. McCabe, A Complexity Measure, IEEE Transactions on Software Engineering SE-2. doi:10.1109/TSE.1976.233837.

[15] B. Alkan, D. Vera, M. K. Chinnathai, R. Harrison, Assessing complexity of component-based control architectures used in modular automation systems, International Journal of Computer and Electrical Engineering 9 (1) (2017) 393–402.

[16] M. Götzfried, Managing Complexity Induced by Product Variety in Manufacturing Companies. Complexity Evaluation and Integration in Decision-Making, Ph.D. thesis, University of St. Gallen (2013). arXiv:arXiv:1011.1669v3, doi:10.1073/pnas.0703993104.

[17] B. Alkan, A complexity modelling approach to support early life-cycle phases of assembly automation systems, Ph.D. thesis, University of Warwick (2018).

[18] H. Komoto, T. Tomiyama, A framework for computer-aided conceptual design and its application to system architecting of mechatronics products, Computer-Aided Design 44 (10) (2012) 931–946.

[19] K. Sinha, Structural complexity and its implications for design of cyber-physical systems, Ph.D. thesis, Massachusetts Institute of Technology (2014).

[20] G. Schuh, U. Schwenk, Produktkomplexität managen–strategien, Methoden, Tools, München, Wien.

[21] U. Lindemann, M. Maurer, T. Braun, Structural Complexity Management: An Approach for the Field of Product Design, 2008.

[22] R. Vrabič, P. Butala, Computational mechanics approach to managing complexity in manufacturing systems, CIRP Annals - Manufacturing Technology 60 (1) (2011) 503–506. doi:10.1016/j.cirp.2011.03.050.

[23] C. E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Computing and Communications Review 5 (2001) 3. arXiv:9411012, doi:10.1145/584091.584093.

[24] A. V. Deshmukh, J. J. Talavage, M. M. Barash, Complexity in Manufacturing Systems Part 1 : Analysis of Static Complexity 1 Introduction, IIE Transactions 30 (1998) 645 – 655. doi:10.1023/A:1007542328011.

[25] J. Efstathiou, A. Calinescu, G. Blackburn, A web-based expert system to assess the complexity of manufacturing organizations, Robotics and Computer-Integrated Manufacturing 18 (3-4) (2002) 305–311. doi:10.1016/S0736-5845(02)00022-4.

[26] G. Frizelle, E. Woodcock, Measuring complexity as an aid to developing operational strategy, International Journal of Operations & Production Management 15 (5) (1995) 26–39. doi:10.1108/01443579510083640.

[27] Z. Zhang, Manufacturing complexity and its measurement based on entropy models, International Journal of Advanced Manufacturing Technology 62 (9-12) (2012) 867–873. doi:10.1007/s00170-011-3872-7.

[28] A. Calinescu, J. Efstathiou, J. Schirn, J. Bermejo, Applying and assessing two methods for measuring complexity in manufacturing, Journal of the Operational Research Society 49 (7) (1998) 723–733. doi:10.1057/palgrave.jors.2600554.

[29] G. Frizelle, Y. M. Suhov, An entropic measurement of queueing behaviour in a class of manufacturing operations, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 457 (2011) (2001) 1579–1601. doi:10.1098/rspa.2000.0731.

[30] Y. Wu, G. Frizelle, J. Efstathiou, A study on the cost of operational complexity in customer supplier systems, International Journal of Production Economics 106 (2007) 217–229. doi:10.1016/j.ijpe.2006.06.004.

[31] T. Zhang, J. Efstathiou, The complexity of mass customization systems under different inventory strategies, International Journal of Computer Integrated Manufacturing 19 (2006) 423–433. doi:10.1080/09511920500399011.

[32] G. Chryssolouris, K. Efthymiou, N. Papakostas, D. Mourtzis, A. Pagoropoulos, Flexibility and complexity: is it a trade-off?, International Journal of Production Research 51 (23-24) (2013) 6788–6802. doi:10.1080/00207543.2012.761362.

[33] K. Efthymiou, A. Pagoropoulos, N. Papakostas, D. Mourtzis, G. Chryssolouris, Manufacturing systems complexity : An assessment of manufacturing performance indicators unpredictability, CIRP Journal of Manufacturing Science and Technology 7 (2014) 324–334.

[34] D. Mourtzis, M. Doukas, F. Psarommatis, Design and operation of manufacturing networks for mass customization, CIRP Annals-Manufacturing Technology 62 (2013) 467–470.

[35] H. Sun, S. Fan, Car sequencing for mixed-model assembly lines with consideration of changeover complexity, Journal of manufacturing systems 46 (2018) 93–102.

[36] Å. Fast-Berglund, T. Fässberg, F. Hellman, A. Davidsson, J. Stahre, Relations between complexity, quality and cognitive automation in mixed-model assembly, Journal of manufacturing systems 32 (3) (2013) 449–455.

[37] H. Wang, H. Wang, S. J. Hu, Utilizing variant differentiation to mitigate manufacturing complexity in mixed-model assembly systems, Journal of Manufacturing Systems 32 (4) (2013) 731–740.

[38] H. Wang, X. Zhu, H. Wang, S. J. Hu, Z. Lin, G. Chen, Multi-objective optimization of product variety and manufacturing complexity in mixed-model assembly systems, Journal of Manufacturing Systems 30 (1) (2011) 16–27.

[39] X. Zhu, S. J. Hu, Y. Koren, N. Huang, A complexity model for sequence planning in mixed-model assembly lines, Journal of manufacturing systems 31 (2) (2012) 121–130.

[40] N. Papakostas, K. Efthymiou, D. Mourtzis, G. Chryssolouris, Modelling the complexity of manufacturing systems using nonlinear dynamics approaches, CIRP Annals - Manufacturing Technology 58 (1) (2009) 437–440. doi:10.1016/j.cirp.2009.03.032.

[41] S. Sivadasan, J. Efstathiou, a. Calinescu, L. H. Huatuco, Advances on measuring the operational complexity of supplier-customer systems, European Journal of Operational Research 171 (2006) 208–226. doi:10.1016/j.ejor.2004.08.032.

[42] S. N. Samy, H. ElMaraghy, A model for measuring products assembly complexity, International Journal of Computer Integrated Manufacturing 23 (11) (2010) 1015–1027. doi:10.1080/0951192X.2010.511652.

[43] G. Chryssolouris, N. Giannelos, N. Papakostas, D. Mourtzis, Chaos Theory in Production Scheduling, CIRP Annals - Manufacturing Technology 53 (2004) 381–383. doi:10.1016/S0007-8506(07)60721-5.

[44] R. Donner, B. Scholz-Reiter, U. Hinrichs, Nonlinear characterization of the performance of production and logistics networks, Journal of Manufacturing Systems 27 (2) (2008) 84–99. doi:10.1016/j.jmsy.2008.10.001.

[45] H. Wiendahl, H. Scheffczyk, Simulation based analysis of complex production systems with methods of nonlinear dynamics, CIRP Annals-Manufacturing Technology 48 (1) (1999) 357–360. doi:10.1016/S0924-0136(03)00177-8.

[46] P. Massotte, Behavioural analysis of a complex system, The International Journal of Advanced Manufacturing Technology 12 (1) (1996) 66–76.

[47] P. Massotte, Behavioural analysis of a complex system, The International Journal of Advanced Manufacturing Technology 12 (1996) 66–76. doi:10.1007/BF01178963.

[48] K. Efthymiou, D. Mourtzis, A. Pagoropoulos, N. Papakostas, G. Chryssolouris., Manufacturing systems complexity analysis methods review, International Journal of Computer Integrated Manufacturing.

[49] K. Efthymiou, On the assessment of manufacturing systems complexity, Ph.D. thesis, University of Patras (2013).

[50] K. Efthymiou, a. Pagoropoulos, N. Papakostas, D. Mourtzis, G. Chryssolouris, Manufacturing systems complexity review: Challenges and outlook, Procedia CIRP 3 (2012) 644–649. doi:10.1016/j.procir.2012.07.110.

[51] I. H. Garbie, A. Shikdar, Design for Manufacturing Systems Complexity: A Perspective Approach, ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, Volume 4 (2010) 751–762doi:10.1115/ESDA2010-25033.

[52] Y.-S. Kim, A system complexity approach for the integration of product development and production system design, Doctoral dissertation, Massachusetts Institute of Technology (1999).

[53] J. Sarkis, An empirical analysis of productivity and complexity for flexible manufacturing systems, International Journal of Production Economics 48 (1997) 39–48. doi:10.1016/S0925-5273(96)00025-4.

[54] K. Windt, T. Philipp, F. Böse, Complexity cube for the characterization of complex production systems, International Journal of Computer Integrated Manufacturing 21 (2) (2008) 195–200.

[55] S. N. Samy, H. ElMaraghy, A model for measuring complexity of automated and hybrid assembly systems, International Journal of Advanced Manufacturing Technology 62 (2012) 813–833. doi:10.1007/s00170-011-3844-y.

[56] H. ElMaraghy, T. AlGeddawy, S. N. Samy, V. Espinoza, A model for assessing the layout structural complexity of manufacturing systems, Journal of Manufacturing Systems 33 (1) (2014) 51–64.

[57] B. Alkan, D. Vera, M. Ahmad, B. Ahmad, R. Harrison, Design evaluation of automated manufacturing processes based on complexity of control logic, 26th CIRP Design Conference 50 (2016) 141–146. doi:10.1016/j.procir.2016.05.031.

[58] T. Lee, Complexity theory in axiomatic design, Ph.D. thesis, Massachusetts Institute of Technology (2003).

[59] M. Gell-Mann, The quark and the jaguar: adventures in the simple and the complex, 1997. doi:10.1119/1.18607.

[60] S. Mattsson, M. Tarrar, Å. Fast-Berglund, Perceived production complexity–understanding more than parts of a system, International Journal of Production Research 54 (20) (2016) 6008–6016.

[61] S. Mattsson, P. Gullander, A. Davidsson, Method for measuring production complexity, 28th International Manufacturing Conference.

[62] A. Falck, R. Örtengren, M. Rosenqvist, Relationship between complexity in manual assembly work, ergonomics and assembly quality, in: NES2012 Ergonomics for Sustainability and Growth, 2012.

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

[64] E. Hückel, Quantentheoretische beiträge zum problem der aromatischen und ungesättigten verbindungen. iii, Zeitschrift für Physik 76 (9-10) (1932) 628–648.

[65] V. Nikiforov, The energy of graphs and matrices, Journal of Mathematical Analysis and Applications 326 (2) (2007) 1472–1475.

[66] K. Sinha, N. R. Shougarian, O. L. de Weck, Complexity management for engineered systems using system value definition, in: Complex Systems Design & Management, Springer, 2017, pp. 155–170.

[67] K. Sinha, E. S. Suh, Pareto-optimization of complex system architecture for structural complexity and modularity, Research in Engineering Design 29 (1) (2018) 123–141.

[68] E. J. Weyuker, Evaluating software complexity measures, IEEE transactions on Software Engineering 14 (9) (1988) 1357–1365.

[69] B. Alkan, D. Vera, B. Ahmad, R. Harrison, A method to assess assembly complexity of industrial products in early design phase, IEEE Access 6 (2018) 989–999.

[70] B. Alkan, An experimental investigation on the relationship between perceived assembly complexity and product design complexity, International Journal on Interactive Design and Manufacturing (IJIDeM) (2019) 1–13.

[71] M. K. Chmarra, A. A. A. Cabrera, T. van Beek, V. D'Amelio, M. S. Erden, T. Tomiyama, Revisiting the divide and conquer strategy to deal with complexity in product design, in: Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on, IEEE, 2008, pp. 393–398.

[72] G. Pahl, W. Beitz, Engineering design: a systematic approach, Springer Science & Business Media, 2013.

[73] M. K. Chinnathai, B. Alkan, R. Harrison, Convertibility evaluation of automated assembly system designs for high variety production, Procedia CIRP 60 (2017) 74–79.

[74] B. Alkan, D. Vera, M. Ahmad, B. Ahmad, R. Harrison, Design evaluation of automated manufacturing processes based on complexity of control logic, Procedia CIRP 50 (2016) 141–146.

[75] B. Alkan, D. Vera, M. Ahmad, B. Ahmad, R. Harrison, A lightweight approach for human factor assessment in virtual assembly designs: an evaluation model for postural risk and metabolic workload, Procedia CIRP 44 (2016) 26–31.

[76] R. Harrison, D. Vera, B. Ahmad, Engineering methods and tools for cyber–physical automation systems, Proceedings of the IEEE 104 (5) (2016) 973–985.

[77] B. Ahmad, A component-based virtual engineering approach to PLC code generation for automation systems.

[78] I. E. Commission, et al., Iec 62264-1 enterprise-control system integration–part 1: Models and terminology, IEC, Genf.

[79] M. T. Karl-Heinz John, Building Blocks of IEC 61131-3, Tech. rep., Geneva (2003).

[80] L. Davis, Handbook of genetic algorithms.