System Design of Mechatronic Products Models and Methods to Utilize Mass Customization

Habib, Tufail

Publication date:
2014

Document Version
Early version, also known as pre-print

Department of Mechanical and Manufacturing Engineering
Aalborg University, Denmark

System Design of Mechatronic Products

Models and Methods to Utilize Mass Customization

Ph.D. Thesis

by

Tufail Habib

Department of Mechanical and Manufacturing Engineering
Fibigerstraede 16, DK-9220 Aalborg East, Denmark
E-mail: habib@m-tech.aau.dk

ISBN: 87-91200-72-5

Thesis title: System Design of Mechatronic Products-Models and Methods to Utilize Mass Customization

PhD student: Tufail Habib

Supervisors: Associate Professor Kaj A. Joergensen, Associate Professor Kjeld Nielsen

List of Papers used in the thesis:

Paper 1
Innovations in Mechatronic Products and Mass Customization. Habib, Tufail ; Nielsen, Kjeld ; Jørgensen, Kaj Asbjørn. Proceedings of MCPC 2011 international conference on Mass Customization and open innovation (16^{th}-19^{th} Nov, 2011) Berkeley, San Francisco, USA.

Paper 2
Comparative Analysis of Design Concepts of Mechatronics Systems with a CAD Tool for System Architecting. Habib, Tufail; KOMOTO, Hitoshi. Published in Mechatronics Journal 2014. DOI: 10.1016/j.mechatronics.2014.03.003.

Paper 3
Design Models in the Development of Mechatronic Systems: Virtual Prototyping and Mechatronic module development process. Habib, Tufail; Thomas Ditlev; Nielsen Kjeld; Jørgensen, K Asbjørn. Submitted to International Journal of Design Engineering, 2014 (under review).

Paper 4
Multidisciplinary Product Decomposition and Analysis Based on Design Structure Matrix (DSM) Modelling. Habib, Tufail. Proceedings of the 7^{th} world conference on Mass Customization, Personalization and Co-Creation (February 4^{th} - 7^{th}, 2014) Aalborg, Denmark.

Paper 5
Modelling and Investigation of Electromechanical Valve Train Actuator at Simulated Pressure Conditions. Habib, Tufail. Proceedings of the Nord design 2012 conference (August 22^{nd}-24^{th}, 2012), Aalborg, Denmark. Pp.208-216.

This thesis has been submitted for assessment in partial fulfilment of a PhD degree. The thesis is based on the submitted or published scientific papers included in the thesis. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Preface

This PhD thesis documents the outcome of research carried out at the Department of Mechanical and Manufacturing Engineering at Aalborg University Denmark. The PhD project was partly sponsored by Aalborg University, Denmark and partly by the UET Peshawar, Pakistan. The project was initiated in June 2010 and ended in June 2014. During the PhD programme, I have been cooperating and in contact with a number of researchers and practitioners who have made the completion of this work possible. Therefore, I would like to take the opportunity to thank the following people for their contributions at different levels.

First of all I would like to thank my supervisors Associate Professors Kaj A. Jørgensen and Associate Professor Kjeld Nielsen for their time and especially for their encouragement and confidence in my research. I am grateful for many inspiring, constructive and insightful discussions. Both have given me invaluable insights while writing scientific articles and support during thesis writing, with valuable discussions and feedback in the form of comments and suggestions.

Furthermore, I would like to thank staff and colleagues at the department of Mechanical and Manufacturing Engineering, especially the members of the Mass Customization Research Group. I would like to express my sincere thanks to Associate Professor Thomas Ditlev Brunø for his support, valuable discussions and professional inputs on various topics. I wish to thank Associate Professors Stig B. Taps and Shaoping Bai for their encouragement and support during the project. Further, I would like to thank my fellow PhD students for valuable input and fruitful discussions of the topic.

My heartiest thanks to Aalborg University, Department of Mechanical and Manufacturing Engineering and to the Otto Mønsted Foundation for supporting my external research stay. I would like to express my gratitude to Hitoshi Komoto, research scientist at the National Institute of Advanced Industrial Science and Technology (AIST) Tsukuba, Japan, who enthusiastically and altruistically helped me in my research during my external research stay in Japan. Many fruitful discussions with him helped expand my horizon towards system design using system architecting in SA-CAD. I would like to thank Gunner Erixon for his inspiring lecture about modularity and in depth discussion about the research topic.

I wish to thank all the companies and their representatives who willingly shared their knowledge, experience and development material for me to conduct the research. Thanks to Modular Management AB, Asetek A/S and FEV GmbH in this respect.

Finally, I would like to express my grateful thanks to my family and friends for supporting me at all times during the project and always giving me encouragement and motivation.

Tufail Habib

Aalborg, June 2014

Sammendrag (Danish)

Denne Phd afhandling retter sig mod emner vedrørende systemdesign og udvikling af mekatroniske produkter såsom (1) de-komponering af systemer, håndtering af grænseflader og identifikation af systemegenskaber og (2) håndtering af fleksibilitet ved udvikling af moduler og identifikation af grænseflader på tværs af domæner for modulære produktarkitekturer.

Mass Customization (MC) er anerkendt som en succesfuld strategi i design og udvikling af produkter ”skræddersyet” til specifikke kundebehov. Globale konkurrenceforhold kræver nye produkter med nye funktionaliteter, som gældende for mekatroniske produkter. Disse produkter bliver mere og mere betydningsfulde som produkttype og nye udviklingstrin har resulteret i drastiske ændringer i design og udvikling af sådanne produkter. Anvendelse af fagområdet mekatronik er baseret på funktionel og rumlig integration af subsystemer med forskellige ingeniørmæssige discipliner, der repræsenterer vigtige midler til succesfuld udvikling af innovative produkter. De innovative potentialer ved mekatronik er accelereret ved de ændringer ved kundebehov, hvilket implicerer, at mass customization spiller en afgørende rolle.

Innovative kapabiliteter, som er drevet af kundetilpasning, bliver brugt af virksomheder og deres produkter med funktioner, der er velegnet til kundekrav og–behov. Kundetilpasning af mekatroniske produkter har mange fordele som f.eks. at software har givet produkter forbedrede kapabiliteter. For eksempel kan styring af et transportbånd programmeres på forskellige måder: start og stop på forskellige tidspunkter samt femdrift med en bestemt afstand. Yderligere kan software bruges til opnåelse af design fleksibilitet. Software kan udvikles med et antal parametre, som kan tildeles forskellige værdier og derved udnyttes til kundetilpasning. Flere karakteristika kan integreres i en enkelt enhed med mekatronisk teknologi. En printer kan ikke kun printe men måske også faxe og foretage scanning.

Med denne udvikling bliver mekatroniske systemer imidlertid stadig mere komplekse i størrelse og i form af den multi-disciplinære karakter. Det er i dette arbejde, at system design og udvikling spiller en central rolle i udfordringerne i særlig grad for at støtte håndtering af kompleksitet, krav til systemspecifikation og integration af ingeniørmæssige domæner for at opnå ønskede resultater. Nogle af fordelene ved at anvende en sådan systemtilgang er at identificere grænseflader mellem domæner, at øge effektivitet eller at genanvendelse af design og hierarkisk de-komponering af systemer fra funktioner til strukturer af sub-systemer, osv. For at støtte denne system design er den velkendte V-model bredt anvendt som procesmodel til at specificere aktiviteter ved udviklingsprocesserne.

Ud over system design er mass customization begreber som modularisering i denne afhandling anvendt ved mekatroniske systemer for at identificere og udforske potentialer ved modulære produktarkitekturer. Under udvikling af løsningsrum og funktionelle forbedringer kan man forøge forretningskapabilitet ved at anvende disse metoder og modeller. Afhandlingen bidrager med udvikling af modelleringsmetoder ved begrebsmæssig design, der er uafhængig af specifikke discipliner. Denne modellering støtter ikke bare det at etablere relationer mellem funktion og form men adressere også konsistens i systembeskrivelser på forskellige hierarkiske niveauer. Yderligere ændrer det på de eksisterende metoder vedrørende system de-komponering, håndtering af grænseflader og kompleksitet ved at udvikle moduler, der som case eksempler benytter konsumprodukter og industrielle produkter.

Summary

This Phd thesis addresses issues in the system design and development of mechatronic products such as (1) system decomposition, interface management and identification of system properties and (2) complexity management by developing modules and identifying interfaces across domains for modular product architectures.

Mass Customization (MC) has been recognized as a successful strategy in the design and development of products tailored to specific customer needs. Global competition demands new products with added functionalities, as in the case of mechatronic products. These products are becoming more and more important as a product type and new inventions have resulted in drastic changes in the design and development of mechatronic products. The application of the research area mechatronics comes from the functional and spatial integration of subsystems with various engineering disciplines that represents an important means of successfully creating innovative products. The innovation potentials of mechatronics are accelerated by the changing demands of the customers, which imply that mass customization has an important role.

Innovative capabilities that are driven by customization are used by companies and their products with functions that correspond to customer requirements and needs. Customization of mechatronic products has many advantages, e.g. software has given products improved capabilities. For example, controlling a conveyer belt can be programmed in different ways, such as starting, stopping at different intervals and advancing a certain distance. In addition to that, software can be used to gain design flexibility. Software can be designed with a number of parameters, which can be assigned different values and thereby be used for customization. Multiple features can be integrated into a single device with mechatronic technologies. A printer may not only be able to print but may also enable faxing and scanning.

However, with all this developments, mechatronic systems are increasingly becoming complex in terms of their size and multi-disciplinary character. In this work it is recognized, that system design and development play a crucial role in the challenges, especially to support complexity management, conceptual design, and integration of engineering domains to attain desired results. Some of the benefits of using such a systems approach include identifying interfaces between domains, increasing effectiveness or re-using of design and hierarchical system decomposition from functions into sub system structures etc. To support such system design, the well known V-model is used as a process model to specify the activities during the development processes.

In this thesis, along with system design, mass customization concepts such as modularity are applied to mechatronic systems to identify and explore the potentials of modular product architectures. With the application of these concepts, business capability can be achieved by solution space development and functional improvements. The thesis contributes by developing modelling schemes at conceptual design that are independent of specific disciplines. This modelling not only helps to establish relationships between function and form but also addresses consistency of system descriptions at different hierarchy levels. This work contributes also by performing system decomposition, the identification of interface relationships and structure analysis for complexity management in multi-domain products. Furthermore, this thesis contributes with new knowledge by applying the mass customization concepts to the development of models and methods using consumer and industrial products as case examples.

Table of Contents

PART I - EXTENDED SUMMARY

PART II- PAPERS

System Design of Mechatronic Products
Models and Methods to Utilize Mass Customization

Part I

Extended Summary

1. INTRODUCTION

This chapter presents the background for the scientific work, as well as gaps and the initial problem identification made prior to final decision about research paradigm, methods, problems and research questions. This chapter has mainly been organized in two sections with system design and development as a first section, and product modularity and mass customization as the other. A short description of the contribution and organization of the thesis concludes the chapter.

The global economy is driven by rapid innovation, competition to introduce innovative products, shortened development and product life cycles, and rising customer demands in terms of the performance, quality and cost of future products. Product innovations make a considerable contribution, and innovation is one of the drivers to be competitive in the business. Mechatronics represents an important means of successfully creating innovative products [VDI 2206, 2004].The most widely used definition of mechatronics is formulated by Tomizuka [2000]: “Mechatronics is the synergetic integration of physical systems with information technology and complex decision making in the design, manufacture and operation of industrial products and processes.”

The application of mechatronics due to the functional and spatial integration of subsystems with various engineering disciplines results in innovative systems such as industrial robots, hybrid vehicles, modern computer numerical control machines, medical instruments, communication, and satellite systems. Along with the benefits of having several engineering disciplines involved in the design activity, the complexity of these systems has increased owing to this integration because no common language has yet been established for describing such integrated product models. Such a language is crucial to enable designers and engineers to transfer design information among the domains derived from various engineering disciplines. Similarly, special attention must be paid to dependencies in the system design of the product and using process models by specifying the activities to be performed during the development process. The V-model is widely used as a process model to specify the activities during the development process.

The V-model (based on VDI 2206, 2004) shown in Fig.1 describes the generic procedure for the development of mechatronic systems. The V-model development process mainly consists of system design, domain specific design and system integration.

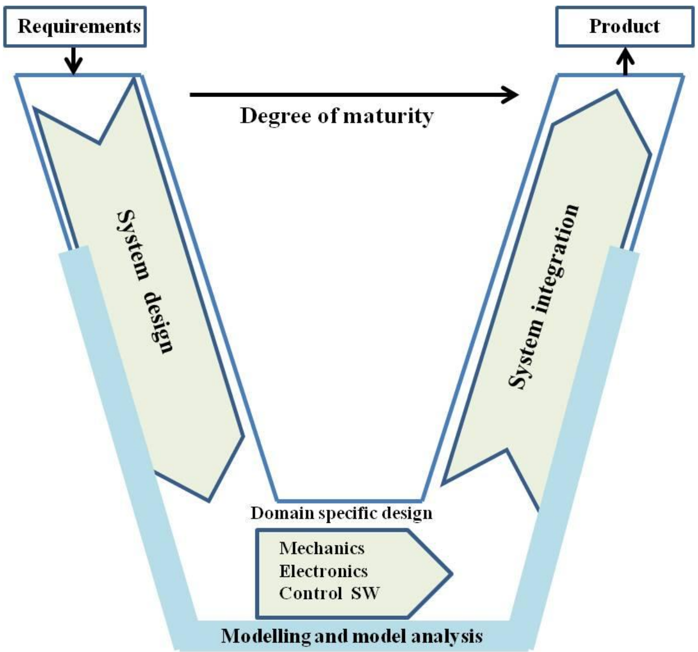


Fig. 1. V-model as an overall approach for the development of mechatronic systems [adapted from VDI 2206, 2004].

The V-model consists of the following steps:

- In system design, the specifications derived from user needs are transformed into component level specifications by defining subsystems and components. This is the conceptual design phase, where the purpose is to define concepts and solutions which describe the main functional and structural characteristics of the product. Modelling and model analysis are also performed in the preliminary design phase.
- In detailed design, domain specific components are developed further on the basis of established domain-specific development methodologies. Furthermore, domain-specific development tools are used for the modelling, analysis and evaluation of product properties.
- In system integration, all the functions, components and subsystems are combined and then verified in an iterative process in order to conform to the requirements and system specifications.

In the V-model, degree of maturity of a product is shown by an arrow, (see Fig.1), however the development process is an iterative one, and upgrading of the product is required according to the market demands.

1.1. System Design and Development

Originating from the area of system theory, a *system* is regarded as a mental construct and thus an abstraction to describe and model a specific area of interest. With more specific reference to technical systems, a system is concrete and dynamic and consists of elements. The relations that exist between the elements define the structure of the system. A system can be part of a larger system and can be decomposed into subsystems [Pahl & Beitz 1996; Andreasen 2005].

As modern mechatronic systems are increasingly becoming complex in terms of their size and multi-disciplinary character, system level design and development play a crucial role, especially in supporting complexity management, conceptual design, and the integration of domains to attain desired results. Some of the benefits of using systems approaches are that they

- Identify interfaces between domains
- Co-ordinate hardware and software and
- Increase the effectiveness or re-use of the design.

1.1.1. System Architecting

System architecting (part of the system design) of multidisciplinary systems defines subsystems and their interfaces through hierarchical system decomposition. According to the V-model of product development from a system engineering perspective [VDI 2206, 2004; Dieterle, 2005], conceptual design is also referred to as system architecting and is considered part of the system decomposition phase. Conceptual design as described by various authors [Pahl and Beitz, 1996; Chmarra, 2008; Hehenberger, 2009] plays an important role in the design of systems because at this phase of design, the product's overall functions, important sub-functions and their interactions are determined. In the conceptual design, principle solutions are determined, along with a structure of realizable modules and their interactions and interfaces, to achieve the successful design of systems. In addition, it is also necessary to perform these tasks with computational support.

One of the issues in system architecting is how to represent mechatronic systems in one model that is independent of all domains. Burr [1990] states that function modelling can be used across the mechanical, electronic and software disciplines and that this enables methods based on function modelling to be used across mechatronic domains. Another issue in the early stage of mechatronic product design is how to effectively derive the parameters of a product and its subsystems on the basis of abstract descriptions of products, such as function requirements. However, to carry out the design tasks, the development of the functional model of the system should also occur in parallel with consideration of the real physical environment or decomposition from function to form at multiple levels. Furthermore, to support the system decomposition and configuration tasks (such as system architecting, interaction and interrelations among design domains) in mechatronic systems, the use of system design support tools is necessary.

The major challenge facing the researchers studying conceptual design is to develop modelling schemes supporting the initial design processes. However, the modelling schemes should be independent of specific disciplines. One of the problems while modelling at early design stages is not only to establish relationships between function and form based on the design concepts of a product and its subsystems, but also how to address the consistency of system descriptions at different levels of hierarchy.

Furthermore, the development of mechatronic products needs integration among domains, but special attention also has to be paid to dependencies in the product and between the design activities. It is necessary for the system designers to perform system modelling by not only establishing appropriate component level specifications but also by being able to transfer system knowledge to communicate with the domain experts. It is crucial to address the challenge of cooperation and communication among design engineers in different domains.

Therefore, it has been recognized that in order to effectively utilize system design tasks in complex mechatronic systems, methods and tools for systematic support are crucial. To address these requirements, a new method is proposed (one of the contributions of this research project) for supporting the tasks of system decomposition, interface management and the identification of system properties. The system architecting tasks must be supported by a computational tool to perform the above tasks.

1.1.2. Mechatronic Module Development

Another important aspect in mechatronic systems design is the integration of domains and process models that support the development process. Systems formed by sequential designs will not have the best match and compatibility between the subsystems and components to get better results. For an optimal design, it is necessary to address tasks such as system integration, subsystem interactions and verification of the overall design [Tomizuka, 2000; De Silva, 2007; Isermann, 2008]. Previous research has shown that modelling and simulation are among the essential steps required to perform the above mentioned tasks before the detailed design. They decrease product development time, and ultimately they reduce costs. At the system design stage of development, modelling and simulations are performed without real-time requirements to obtain design specifications, dynamic requirements and performance measures.

One problem in the design of mechatronic system is the integration of domains to obtain the desired response. In a typical mechatronic system, the subsystems and components are interconnected by power domain and information domain. This process of integration must be represented in a systematic way from the integration of domains to simulations of the system response. This modelling must support the mechatronic module development process.

In order to address the above mentioned issues in this thesis, mechatronic module development is illustrated with two examples. The first example involves the modelling and simulation of an antenna pedestal drive system. The second example involves investigations of the physical prototype of an actuator.

1.2. Product Modularity and Mass Customization

In the last two decades, product modularity and product platforms have been advanced as effective strategies to achieve the purpose of mass customization (MC) and to offset some of the challenges faced by businesses due to frequent and rapid changes. When a product or process is modularized, the elements of its design are split up and assigned to modules according to some architecture or plan. Pine [1993] has argued that for businesses to cope with challenges and future uncertainty, mass customization is a solution which has evolved into a flexible, fast delivery and cost-effective production and marketing strategy in the competition of a global market. The best way of achieving mass customization is by creating modular components that can be configured into a wide variety of end products and services. In order to successfully achieve the modularization of products and processes, various approaches and methods are used.

Various modularization methods such as modular function deployment (MFD), design structure matrix (DSM) and modular product development (MPD) are widely used in academia and in industry, and they have a range of applications to product and organization domains. All these methods use three main steps: system decomposition, module development and evaluation, to introduce modules into products. However, they differ with respect to the purpose of modularization.

Modular function deployment as proposed by Ericsson [1999] is a well established methodology for product modularization. In this method, modularity drivers are mapped against technical solutions and their reasons for being modules. The modularization purpose is to form modules on the basis of strategic aspects. Another widely accepted method for decomposing a product into modules is the design structure matrix method (DSM). This method was introduced by Steward [1981] to manage the parameter dependencies in the design of a complex system. According to Eppinger and Browning [2012], the DSM is a relational matrix that forms a framework for documenting and evaluating interface architecture. The design structure matrix and domain mapping matrix are useful to support system decomposition, module identification and the modelling of relationships between elements from different mechatronic domains, but they do not support the modelling of relationships from function to form at multiple levels and do not support the reasons for developing modules in relation to product life cycle issues. The modular product development method proposed by Pahl and Beitz [1996] is based on the function structure of a product. A function structure is a functional decomposition block diagram of all the product's functions and of the material, energy, and information flows between them. Stone *et al*. [2000] developed a function structure heuristic (FSH) method, based on the Pahl and Beitz [1996] function structure approach, where modules from a single product's function structure are introduced by finding the dominant flow, branching flows, or conversion-transmission function pairs. However, the function structure heuristic method is limited to functional decomposition.

After the assessment of various modularization methods, some gaps and limitations have been identified as these methods are not sufficiently addressing: (1) hierarchical system decomposition at multiple levels; (2) the identification of variants in the decomposition phase from function to form; (3) lateral relationships between components and assemblies/elements at a certain level of decomposition.

From this assessment, it has been recognized that modular function deployment is a more comprehensive method to form modules because (1) it addresses product life cycle issues; (2) it enables the transformation of market requirements into product specifications and facilitates product decomposition from function to form; and (3) it supports module evaluation and the identification of interface relations. However, there are certain limitations in this method as well, and it needs modification in certain ways as mentioned above so that it becomes more feasible to apply to mechatronic products. To indicate its strength, the modified method is applied to an industrial product.

1.3. Research Requirements, Contribution and Organization of the Thesis

In the design and development of mechatronic systems, the most commonly reported challenges and problems are related to dependencies in the product concept and between the design activities. In addition to that more specific problem is how to represent a mechatronic system independently of disciplines. Furthermore, requirements such as development of the functional model of the system should occur in parallel with consideration of the real physical environment and the transfer of design knowledge from system designers to domain experts is addressed in this project. Along with these challenges and problems, some gaps have been identified in this research project, including the utilization and application of mass customization concepts (modularity, platforms, interfaces etc). Since modularization methods are not adequately addressed and applied in the design and development of mechatronic systems.

This study recognizes that to successfully implement the concepts of mass customization in the design and development of mechatronic products, it is necessary to address the above mentioned issues by developing new methods, modifying the existing ones and using tools to support: (1) the tasks of system decomposition, interface management and identification of system properties and the ability to redesign and develop product architectures; (2) mechatronic module development with system models to address tasks such as system integration, subsystem interactions, the verification of design and the transfer of knowledge among design engineers in different domains ; and (3) complexity management by developing modules and identifying interface across domains.

During this research project, various software tools have been used to facilitate modelling, design and simulation tasks. For instance, the SA-CAD tool [Komoto, 2010] was used to support configuration tasks in system architecting.The software tool 20-Sim [Controllab products, 2010] was used to develop a Bond Graph model of the system. 20-Sim is a graphical modelling and simulation tool which is used for generating and processing dynamic systems. It supports system-level design, simulation, automatic code generation, and the testing and verification of embedded systems. In addition, Matlab [MathWorks] was used to analyze the data obtained from an electromechanical actuator under simulated pressure conditions. Finally, the

Cambridge advanced modeller (CAM) tool [Wyne et al, 2010] was used to develop the system structure matrix. This tool also supports clustering, partitioning, and structure analysis tasks.

This thesis will make a number of contributions to the field: (1) new knowledge regarding the design and development of multidisciplinary products; (2) a new method with functional modelling to support the development of product architecture for the next generation of products; (3) complexity management by performing system decomposition, identifying interface relationships and structure analysis in multi-domain products; (4) system models in mechatronic module development; and (5) the application and modification of existing modularization methods to develop modules based on product lifecycle issues in multidisciplinary products.

This thesis is organized in two parts; the first part is the main thesis, including an introduction, a state-of-the-art chapter, a chapter with objectives, hypothesis, research questions and research delimitations, a chapter with scientific approach and methods, two chapters with results and contributions, and finally conclusion and future perspectives. The second part includes the two journal papers and three conference papers which have been selected for the thesis.

2. State-of-the-art

This chapter presents the state-of-the-art in system design and development, product modularity and mass customization.

2.1. System Design and Development

Modern mechatronic systems are increasingly becoming complex in terms of their size and multi-disciplinary character. Due to multi-domain activity, the complexity of the design tasks increases along with the product related advantages. The challenges are related to the way a product concept can be described and how information linked to the product concept can be shared in engineering disciplines. Further, the issue of dependencies between the product and the design activities must be addressed to reduce integration problems and ultimately costs [Tomiyama, 2007; Torry-Smith, 2013].

System level design plays an important role, especially in supporting conceptual design, integration of domains and complexity management. In the literature, various approaches are used to address these issues.

System architecting is part of the system design of multi-disciplinary systems, and it defines subsystems and their interfaces through hierarchical system decomposition. According to the V-model of product development from a systems engineering perspective [VDI 2206, 2004; Dieterle, 2005], conceptual design is also referred to as system architecting and is considered part of the system decomposition phase. It is described as the process of translating requirements into system requirements and transforming system-level specifications into subsystems and components. Conceptual design is described by multiple authors [Forsberg, 1992; Pahl and Beitz, 1996; Isermann, 2005; Desilva, 2005; Chmarra, 2008; Hehenberger, 2009] as playing an important role in the design of systems because at this phase of design the product's overall functions, important sub-functions and their interactions are determined. In addition, the main functions of conceptual design are to generate and evaluate broad solutions, given the specification, which provides a suitable starting point for preliminary design and detail design [Rehman, 2011]. During conceptual design, the principle solutions are determined, along with a structure of realizable modules and the interactions and interfaces needed to achieve the successful design of systems. In performing these tasks, designers have to deal with complexity derived from the interactions and constraints among the subsystems in multi-domain systems [Tomiyama, 2007; ElMaraghy, 2012].

Design theories are fundamental and important contributions to systems design research. A basic assumption of the theory of domains [Hansen and Andreasen 2002] as well as other design theories based on systems theory such as the theory of technical systems [Hubka and Eder, 1988] is that the structure of a system is determined by its characteristics, whereas the behaviour of the system is how it reacts to stimuli as well as how its *properties* are perceived by humans. In other words, *behaviour* is what 'the system does' and *structure* is what 'it is'. The theory of domains states that an artefact may be seen in three different domains such as transformation, organ (as a functional carrier) and part domain. Here, the term 'domain' refers to a specific viewpoint and not to an engineering discipline such as mechanics, electronics or software [Hansen and Andreasen 2002].

In the literature, various researchers have proposed approaches and methods for the embodiment of function structures in forms (i.e. physical structure) and for developing product architecture. For example, the axiomatic design presented by Suh [1990] aimed to systematically develop complex systems. In this method, design is considered as the mapping process between the functional requirements (FRs) in the function domain and the design parameters (DPs) in the physical domain. The result of the process is a functional decomposition of the design and the physical realization of the system. According to Suh's axioms, the best design is one that is functionally uncoupled and that has the minimum information content. Pahl and Beitz [1999] describe how, in the development of modular systems, the physical structure comprising the assemblies and components used as building blocks and the relationships among these assemblies and components must be reflected in the function structure. Baldwin et al. [2006] focus on critical modules in the design of complex systems. Jiao [2000] describes product family architecture (PFA) by applying the functional–behavioural–structural view to the modelling of mechanical products. A well-developed PFA can provide a generic architecture to capture and utilize the commonalities within which each new product instantiates and facilitates future designs in a common product line structure.

Stone et al. [2000, 2008] propose a heuristic method for identifying the functional modules of a product. In this method, energy and signal flows are analyzed and heuristic rules are developed to identify functional modules for the development of modular product architecture. Pimmler [1994], Van Wie [2002] and Ulrich [2008] use block diagrams with functional aspects to model system architectures. Further, Albers et al. [2011] apply the contact and channel approach to mechatronic products with the support of a software tool to help designers understand and communicate the complex dependencies between function and physical structure and to generate system architecture through the function and part database. Borches [2010] develops an A3 architecture overview which provide a systems overview related to functional and physical aspects, and focuses on generating architecture knowledge about complex systems. Bonnema [2011] proposes an architecting approach based on function key drivers to model relationships with the aim of providing insight to different stakeholders in design.

Some of the approaches proposed in the literature on product architecture are applicable only to mechanical systems [Pimmler, 1994; Jiao, 2000; Ulrich and Eppinger, 2008] and not to mechatronic systems. Other methods such as the design structure matrix and domain mapping matrix support system decomposition and the modelling of relationships between elements from different mechatronic domains, but they do not support the modelling of relationships from function to form at multiple levels in a single model. Methods such as PFA [Jiao, 2000] cannot be implemented in software tools for dealing with the complexity derived from the interactions and constraints in multidisciplinary mechatronic systems. Commercial tools such as Modelica [V-3.2, 2010] and Simulink do not support hierarchical system decomposition, especially functional decomposition before the physical realization of the product. Methods which use functional modelling support part of the design activities, but such methods do not provide sufficient information on issues such as how system elements contribute to system properties. Nevertheless, some established approaches can be applied to mechatronic products to deal with complex interactions between function and form [Stone, 2000, 2008; Van Wie, 2002; Huang, 2003; Albers 2011].

A review of the state of the art indicates useful approaches and methods to perform the system architecting tasks or the embodiment of functions into forms, but on the other hand it also indicates gaps in relation to mechatronic systems development.

2.1.1. Mechatronic Module Development

In the design of a mechatronic system, it is possible to design the mechanical equipment before any of the control system design has been initiated. A clear drawback of this sequential approach is the lack of compatibility between the subsystems, which results in additional efforts and costs to meet the specifications of the overall system [Hehenberger, 2010]. Systems formed by sequential design do not attain the level of match and compatibility between the subsystems and components that is needed to get better results. For an optimal design, it is necessary to address tasks such as system integration, subsystem interaction and the verification of the overall design [Tomizuka, 2000; De Silva, 2005; Boucher 2008; Craig, 2009; Hehenberger, 2009, 2013]. Because of the many varieties of designs, diverse components must be modelled using general modelling principles. In the system design phase of development, software-in-the-loop simulation is used so that components and control algorithms are simulated on a computer without real-time requirements to obtain design specifications, dynamic requirements and performance measures (Isermann, 2008). Modelling and simulation are among the essential steps in addressing the above mentioned tasks before the detailed design stage.

In the development of complex mechatronic systems, there is a need to look into the interdependencies among subsystems and the designers that develop them [Cabrera et al, 2010]. For the design of a computer controlled system, it is crucial that the dynamics of systems that exchange power and energy in various forms be thoroughly understood, and methods for modelling, ways of analyzing systems and techniques to simulate the response of the systems must be developed. One of the main and most challenging steps in the design and analysis of a mechatronic system is to generate a computer based model [Granda Jose, 2002; Karnopp, 2006]. Multi domain systems can be modelled using a common notation such as Bond Graph [Gawthrop, 2007; Behbahani, 2007], which is important in the design of mechatronic systems.

In addition to modelling and simulation at the system design phase, it is useful to use the concept of modularity to support the design process of mechatronic systems.

2.2. Product Modularity and Mass Customization

Mass customization (MC) has evolved into a flexible, fast delivery and cost effective production and marketing strategy in global market competition, just as described by Davis [1989] and Pine [1993]. Industries with high volume and a demand for customizable products have adopted the strategy over the last decade [Welcher and Piller, 2012; Nielsen, 2014]. According to Pine [1993], the best way of achieving mass customization is by creating modular components that can be configured into a wide variety of end products and services. Interchangeable parts innovation is the term used by Pine to describe substitution using modular, interchangeable parts across products and services. Meyer and Lehnerd [1997] argue that companies should plan and manage on the basis of product platforms, that is, the combination of subsystems and interfaces that constitutes a common product structure for a series of derivative products. In MC strategy, process flexibility is one of the important elements. Two important components of process flexibility are the use of modular product design combined with delayed differentiation of a product and the use of a flexible manufacturing system [Berman, 2002].

Mass customization is enabled through modular product architectures, from which a wide variety of products can be configured and assembled [Mikkola, 2007]. Modularity can occur in products (product architecture designs), processes (manufacturing processes) and logistics (supply chain configuration) as described by Frederickson [2005].The standardized interfaces of components of modular product architectures mean that mass customization and related manufacturing strategies can be realized [Mikkola, 2007; Steffen, 2013]. Two of the central principles of MC are that product ranges should be developed on the basis of modules and that configuration systems should be used to support the tasks involved in the customer-oriented business processes related to the specification of customer-specific products [Hvam, 2008]. Even a small number of modules will produce a large portfolio of final products. The modules should enable the best response to customer needs. These modules must be developed by integrating the information about anticipated customer requirements at the product design phase [C. da Cunha, 2010].

The development of modularity at various development stages is the result of a search for potential common technical solutions. The early stage modularization process provides more freedom to define architectural content and to enable the function–component mapping relationship [Liu Zhou (2010]. Function based module definitions can explore conceptual product architecture and gain an early insight into common and unique functionality [Stone and Wood 2000; Stone et al. 2000; Dahmus et al. 2001]. Physical modularization generates the modular product architecture by re-arranging the physical elements into modules and is adopted for product or platform redesign [Martin and Ishii 2002; Hsiao and Liu 2005]. Parametric modularity considers the product structure as essentially fixed, and the product characteristics are varied only within the boundaries of the individual elements or parameters. This kind of approach provides the least freedom to change the product structure and only pursues a certain commonality at the detailed module and assembly design stage [Simpson et al, 2001].

Modularization generally follows the three steps: (1) decomposition into elements; (2) identification of the relations between the elements; and (3) clustering the elements into modules [Pimmler and Eppinger, 1994]. Decomposition into elements corresponds to describing the product in terms of functionalities so that the functions are associated with the physical elements that realize them. The functional decomposition of a product is usually developed in the conceptual design phase. The decomposition corresponds to building the functional model of the product that will guide the physical implementation. There are various approaches to the functional modelling of systems, as discussed by Erden *et al*. [2008].The various modularization methods use the above steps to form modules in products.

Several functional modelling methods are used in the development of mechatronic systems. Many researchers [Burr, 1990; Suh, 2001; Negal, 2008; Tomiyama, 2009; Albers, 2011; Hehenberger, 2010] use functional approaches. All these methods use functional thinking to represent mechatronic systems in models. Some methods use hierarchical decomposition models, while others attempt to help designers to identify the complex dependencies between function and form. Van-Beek [2010] develops a modularization method based on the functional model of a system to derive component relations in mechatronic systems. In this method, the design structure matrix (DSM) is constructed on the basis of the functional model of the system. Similarly, other approaches such as the design structure matrix and domain mapping matrix (DMM) used by Kreimeyer [2008] and Danilovic [2007] are examples of modelling relationships between functions, components, physical structure and resources during the mechatronic product development process. The main aim of these approaches is to manage multiple relations during design.

A gap analysis of the state-of-the-art of mechatronic product development methods based on functional modelling indicates that present contributions identify decomposition models, modelling relationships between function and form, and modelling relationships between elements from different mechatronic domains. However, there is no indication in these methods of how the functional modelling and modelling relationships can be used to identify modules and to generate variants.

3. OBJECTIVES, HYPOTHESIS AND RESEARCH QUESTIONS

This chapter is a collection of objectives, hypothesis and research questions prepared for the thesis. The initial problem and gaps identified in the previous chapters have been transformed into hypotheses, and each hypothesis is supported by several research questions. Furthermore, research delimitations are presented at the end of the chapter.

3.1. Objectives

The objectives of this thesis in relation to system design, product modularity and mass customization are to:

- Contribute new knowledge by applying the concepts of mass customization to the development of mechatronic systems
- Contribute a new method for the conceptual design of mechatronic products to develop system architectures
- Develop a modelling scheme to represent a mechatronic system independent of any discipline
- Contribute by performing system decomposition, the identification of interface relationships and structure analysis for complexity management in multi-domain products
- Contribute system models to the development of mechatronic modules
- Contribute by applying and modifying an existing modularization method to identify modules.

3.2. Hypothesis and Research Questions

Hypothesis 1

It is possible in the system design of mechatronic systems to develop system architectures for next generation products.

Research questions

The following research questions are used to clarify the Hypothesis 1:

a) How to define system decomposition and interfaces among subsystems?
b) How to define the physical and logical configurations of subsystems and components that realize the desired functions and behaviours and evaluate the system performance, i.e. functionality?
c) How to effectively derive the parameters of a product and its subsystems on the basis of the abstract descriptions of products such as function requirements?
d) What is the optimal module structure to minimize life cycle costs and maximize common parts in product family/modular interactions to attain high reliability?
e) How to redesign and develop multiple system architectures?

Hypothesis 2

It is possible to develop mechatronic system models using modelling and simulations.

Research questions

The following research questions are used to clarify the Hypothesis 2:

a) How can the simultaneous design and integration of domains be performed?
b) How can state equations be developed and simulations performed from the bond graph model?
c) How to verify the scaled down actuator and optimize the design parameters to the desired requirements?
d) How can dependencies between product and process models be developed?

Hypothesis 3

It is possible to manage complexity in mechatronic products by developing modules and identifying interfaces across domains.

Research questions

The following research questions are used to clarify the Hypothesis 3:

a) How can hierarchical system decomposition be performed at multiple levels?
b) How can consistency of the system specifications be maintained at different levels of decomposition?
c) How to identify modules in relation to product life cycle issues?

3.3. Research Delimitations

- In the context of mechatronic systems, the research does not focus on contributing to theories and models in general or specifically to software and electronic issues. As part of the system, however, software itself is discussed in relation to customization.
- Detailed design is not discussed in relation to mechanic, electronic and software issues.
- Quantitative cost benefit analyses are not performed.
- Modularity in the manufacturing process, in logistics and in relation to organizational issues is not discussed.

4. RESEARCH METHODS

This chapter presents the research design, which is composed of the scientific paradigms, strategies of inquiries and methods applied in this thesis. In addition to research design, this chapter includes the methodological research procedure and the thesis structure.

4.1. Research Design

Creswell [2009] presents research design as a plan or proposal to conduct research that involves the intersection of philosophy, strategies of inquiry and specific methods. Ghauri and Grønhaug [2005] describe that research design is an overall plan for relating the conceptual research problem to relevant and predictable empirical research. According to Mouton [1996], the research design serves to 'plan, structure and execute' the research to maximize the validity of findings. Yin [2009] explains that it is the action plan from empirical insights to conclusions. The research design contains the philosophical worldview and ideas of the researcher that influence the practice of research in explaining and understanding reality and hence in generating knowledge.

4.1.1. Philosophical Worldview or Scientific Paradigms

Guba [1990] uses the term worldview to denote 'a basic set of beliefs that guide action'. Others have called worldviews as paradigms [Lincoln and Guba, 2000; Mertens, 1998], epistemologies and ontology [Crotty, 1998] or broadly conceived research methodologies [Arbnor and Bjerke, 2009; Neuman, 2000].

Creswell [2009] argues that a worldview is the general orientation to the world and the nature of research that a researcher holds. According to Arbnor and Bjerke [2009], all people have certain "ultimate presumptions" about the world, the surrounding environment and their role in that environment. This is relevant since these presumptions will influence the way people address certain problems and their use of techniques. A commonly used term for a set of ultimate presumptions is a paradigm [Kuhn, 1962]. A paradigm is a set of presumptions, values and ideals, typically within a certain scientific area [Arbnor and Bjerke, 2009].

Several authors have proposed different classifications of paradigms. Some authors promote only two views, i.e. a positivistic view and a hermeneutic view [Gummesson 2000; Coughlan and Coghlan 2002], while some argue for three simultaneous paradigms [Arbnor & Bjerke, 2009]. Others promote four classes, such as positivism, post positivism, critical theory and constructivism [Guba & Lincoln 1994] or post positivism, constructivism, advocacy/participatory and pragmatism [Creswell, 2009].

As a result of the different paradigms within the social science, Arbnor and Bjerke [2009] classify the social science paradigms into six different categories, also known as categories of knowledge, where these categories could be placed on a scale between *objectivist-rationalist* and *subjectivist-relativist* paradigms as two extremes. In Table1, each category is illustrated with ultimate reality presumptions, ambitions for creating knowledge, common metaphors, pictures, descriptions and some techniques for creating knowledge.

It is argued that most of the social science paradigms can be placed somewhere on this scale between the two extremes. The individual categories are not discussed in detail here, but Arbnor and Bjerke point out two patterns of higher and lower numbers:

According to the classification in table 1, the more we approach lower numbers, the more

- Reality is considered to be objective and rational
- Relations to philosophy decrease
- Knowledge as explanation is seen as a lodestar
- General and empirical results are sought

At the other extreme, the more we approach higher numbers, the more

- Reality is considered subjective and relative
- Relations to philosophy increase
- Knowledge as understanding is seen as a lodestar
- Results are sought that are specific and concrete but eidetic

Table 1 Classification of social science paradigms, adapted from Arbnor and Bjerke [2009].

← *Objectivist-Rationalistic Explaining reality*						*Subjectivist-Relativistic Understanding reality* →
	1	2	3	4	5	6
Ultimate reality presumptions	Reality as concrete and conformable to law from a structure independent of the observer	Reality as a concrete determining process	Reality as mutually dependent fields of information	Reality as a world of symbolic discourse	Reality as a social construction	Reality as a manifestation of human intetentionally
Ambitions for creating knowledge	To reconstruct external reality-the empirically general one	To explain entireties in their regularities and breaks	To reconstruct contexts in terms of information	To understand patterns of social interaction in terms of symbolic discourse	To understand how social reality is constructed maintained and defined	To develop eidetical insight instead of an empirical one
Some common metaphors, pictures and descriptions	Machine, Mathematics, Logic	Organism, natural selection	Cybernetics, network of information	Role playing, theatre, culture	Language games, typifications, network of meaning	Intentionality, transcendence
Some techniques for creating knowledge	Surveys; operational definitions	Historical analysis	Contextual analysis	Symbolic analysis	Hermeneutic diagnosis	Variations of free imagination; to bracket (epoche) appearances

In general, the lower numbers are closely related to the natural sciences or technical engineering sciences, where reality is seen as objective and rational. On the other hand, the higher numbers are related to social sciences, where reality is seen as subjective.

4.1.2. Methodological Approaches

On the basis of the paradigmatic categories in Table1, Arbnor and Bjerke [2009] introduce three methodological approaches that make assumptions about reality as illustrated in Fig.2. The three approaches are:

- The analytical approach
- The systems approach
- The actors approach

According to Arbnor and Bjerke [2009], the main assumption that the analytical approach makes about reality has a summative character; that *is the whole is the sum of its parts.* This means that once the researcher knows the parts of the whole, the parts can be added together to get the total picture. One of the characteristics of this approach is that knowledge created during the research process is independent of individual subjective experiences. This approach assumes that knowledge advances by formal logic that is represented by judgements, and these judgements consist of assumptions that can only be verified or falsified.

On the other hand, the assumption behind the systems approach is that reality is arranged in such a way that the *whole differs from the sum of its parts*. This means, that not only the parts but their relations are essential, and this is the feature that distinguishes it from the analytical approach. These relations between parts lead to positive effects which are known as synergy. Knowledge developed through the systems approach depends on systems, i.e. the behaviour of the parts follows system principles. In brief, this approach explains or understands parts through the characteristics of the whole [Arbnor and Bjerke, 2009].

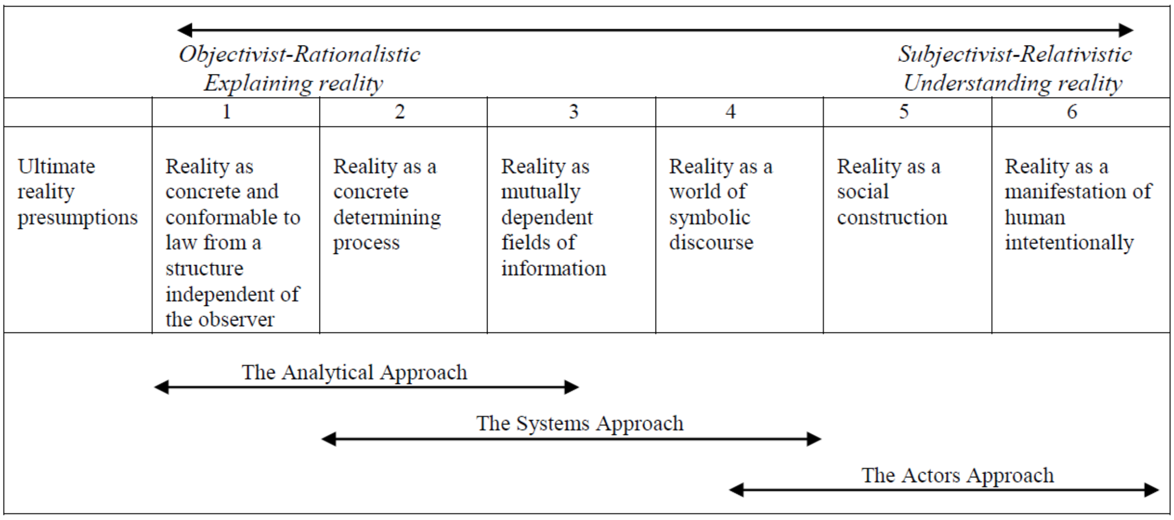


Fig. 2. Relation between paradigmatic categories of figure 1 and methodological approaches, adapted from Arbnor and Bjerke [2009].

In the actors approach, the whole is understood by the characteristics of its parts. Reality is considered as a social construction; wholes and parts are ambiguous and are continuously reinterpreted. In the actors approach, even though phenomenological principles are used that how social reality is constructed, but knowledge development is therefore dependent on actors [Arbnor and Bjerke, 2009].

Arbnor and Bjerke [2009] argue that the analytical approach and the systems approach are relevant to explanatory knowledge, while the actors approach is associated with understanding knowledge. The explanatics assumes that the world is so complex that explanatory knowledge as a science must devote itself to simplification and reduction. In this thesis, mechatronic systems are seen in this perspective, and this is further elaborated in section 4.1.4.

4.1.3. Critical Rationalism

Critical rationalism as envisioned by Popper is used in this research thesis to improve existing models in general and existing methods in particular. It is rationalist because it makes use of deductive reasoning. Popper uses deduction as a method to move from theory to hypothesis [Popper 1974:51], unlike classical rationalists, who considered deduction a non- empirical way of acquiring knowledge. It is called critical due to the persistent testing of hypotheses, which is why it is termed *critical rationalism.*

Poppers philosophy originates with his doubts about the concepts of verification and induction in positivism. Induction is unable to give certain knowledge and has thus been considered a problem and has been debated since the eighteenth century in the philosophy of science [Popper 1974: 42-46]. In relation to induction Popper argues that from a logical point of view, previous observation can never reveal what future studies will show. In order to solve the problem of induction, Popper [1959] suggested that scientists should give up the principle of sufficient reason and instead use the principle of critical testing. Logic can be used not only as an instrument of proof, but also as an instrument of criticism. In proofs, truth is transmitted from the premises to the consequence; in criticism, falsity is re-transmitted from a consequence to at least one of the premises [Popper 1992, p. 75].

For the discussion of scientific theories, it is important that a theory is false if it has a false consequence. Theories can be falsified by singular statements about observable events and by test statements. It is important to understand that not only isolated hypotheses like ‘all swans are white’ can be falsified in this way, but also complicated theoretical systems consisting of many hypotheses and also including auxiliary hypotheses [Andersson 1994].As Schroeder-Heister points out, in Popper’s philosophy of critical rationalism “Theoretical progress is made by successive critique and revision of existing theories, which is governed by the idea of objective truth.” [Schroeder-Heister, 2004].

Falsification holds Popper's answer to the question of what characterizes scientific methods. He argues that the reason for making hypotheses is that they can be tested directly in experiments, unlike theory itself (see Fig.3).

In this research project, critical rationalism is applied implicitly and explicitly. Theories are deduced into research questions, and tested in experiments by applying various methods and approaches to produce the following scenarios:

- The results are inconsistent with the theory, in which case the theory is considered falsified,
- The experiment yields the exact results predicted by the hypotheses and research questions, in which case the theory is considered valid until new and better ways of testing are discovered.

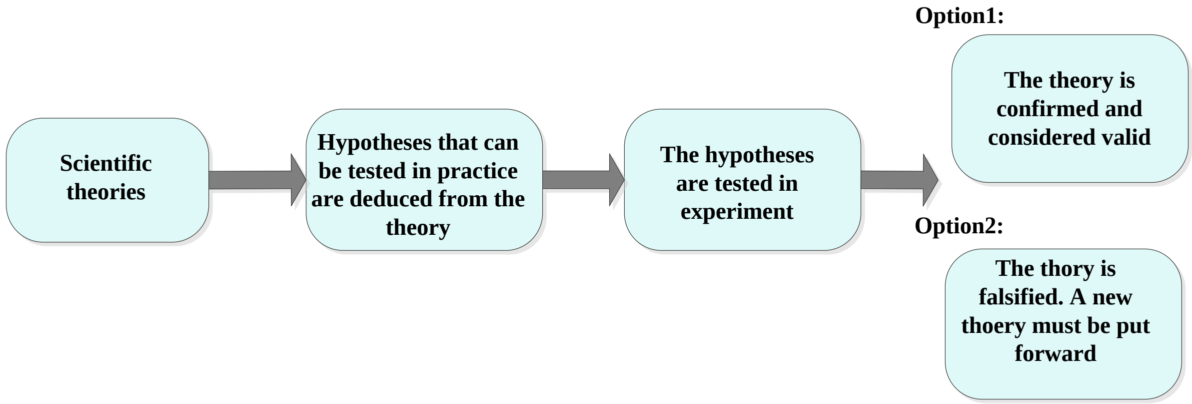


Fig.3.The scientific method based on Poppers philosophy of critical rationalism.

From this it follows that scientific progress is made through the continuous testing of theories, but one can never be sure that the truth about the world has been uncovered. However, when the theory is falsified and revised, the new theory tends to be closer to truth than the falsified theory [Schroeder-Heister, 2004]. In terms of the paradigmatic categorization presented by Arbnor and Bjerke [1997] in Table 1, critical rationalism is close to the lower numbers relevant to the objectivist-rationalistic paradigm.

4.1.4. The Underlying Paradigm Based on Methodological Approaches

The selection of the paradigm is based on the research questions and the nature of the field of investigation. Addressing the basic question of why the systems approach has been chosen as the dominant research approach involves comparing it with the analytical approach in relation to the field of investigation. However, this research also involves using the analytical approach and the actors approach as well.

The field of investigation in this project is mechatronic systems, where complexity is usually increased due to the intended interaction between physical systems and information technologies and the resulting increase in design details. Moreover, the lack of methodology and lack of a common language also increase complexity in mechatronics. In addition, the other challenges in mechatronic systems design comprise the difficulty of modelling and controlling multiple relations in product concept, insufficient transfer of information between domains and lack of detailed information of the system faced by engineers. Furthermore, the concept of modularization is characterized by various definitions and understandings in the literature that are not fully utilized or explored in the context of mechatronic system development.

All these requirements while dealing with mechatronic systems can be best addressed by adopting the systems approach. Simple cause and effect relations may not be sufficient to investigate complex mechatronic systems while using the concept of modularization. As argued by Arbnor and Bjerke [2009], the advantages of the systems approach are realized when a researcher faces a complex problem where it might be dangerous, both in terms of the understanding of the problem and the usefulness of the developed solutions, to simplify things by trying to reduce it to something it is not (p.432).

The reality perceived in the systems approach is different from that of the analytical approach, where it is assumed that reality consists of components that are often mutually dependent on each other. This entails *synergistic effects*, which means that not only the contents of the individual parts but also the way they are put together provides information [Arbnor and Bjerke, 2009]. These synergies are fully utilized in mechanic, electronic and software domains while designing mechatronic systems. The functional and spatial integration in mechatronics is implemented to achieve a better whole according to the essence of systems theory.

Knowledge developed using the systems approach cannot be called 'general' in the same absolute way as knowledge developed using the analytical approach. The researcher seeks the relations not only to understand but also to explain systems and their positive and negative results [Arbnor and Bjerke, 2009]. The systems approach can be composed of multiple fields that are also relevant to the multidisciplinary nature of mechatronic systems. However, the analytical approach is also used to some extent to apply established theories from the natural sciences to explain reality. In addition, the actors approach is applicable to the effort to understand reality in a social context because customer feedback is necessary to utilize and apply the concepts of mass customization.

There are some considerations that must be addressed after the selection of the systems approach. Arbnor and Bjerke [2009] define five levels of objectives within the systems approach:

- Determine the type of system
- Describe the system
- Determine the relations of the system
- Forecast
- Guide.

In the context of this project, the first two levels of objectives were explained in the previous chapters, while the remaining levels are part of the research papers and conclusions.

4.1.5. Methodological Research Procedure

On the basis of the principles of general system theory, Joergensen [2000] developed a methodological research procedure for research and development projects. As argued in section 4.1.4, this project uses the systems approach according to the definition from Arbnor and Bjerke [1997]. Hence a methodological procedure which is based on systems theory seems to be an appropriate approach. The methodological procedure is based on the two fundamental system concepts analysis and synthesis. Joergensen [2000] defines these two concepts in the following way:

- Analysis (of an existing system) is: (1) to investigate properties of the system; and 2) to divide the system into system components and a system structure.
- Synthesis (of a new system) is: (1) to create the system by relating existing components to each other by a structure; and (2) to add properties to the system.

According to this research method, analysis and synthesis are elementary operations which can be combined and sequenced in various ways. If analysis and synthesis are performed after each other, two basic activities are created:

- An approach consisting of an analysis operation followed by a synthesis operation is termed *problem solving.*
- An approach in the opposite order is termed *design.*

In research and development projects, *design* and *problem solving* activities are often mixed together. A possible structure is to have *problem solving* as the outer activity and to place the design activity as the internal structure of the synthesis operation, as suggested by Jørgensen [2000] and illustrated in Fig.4.

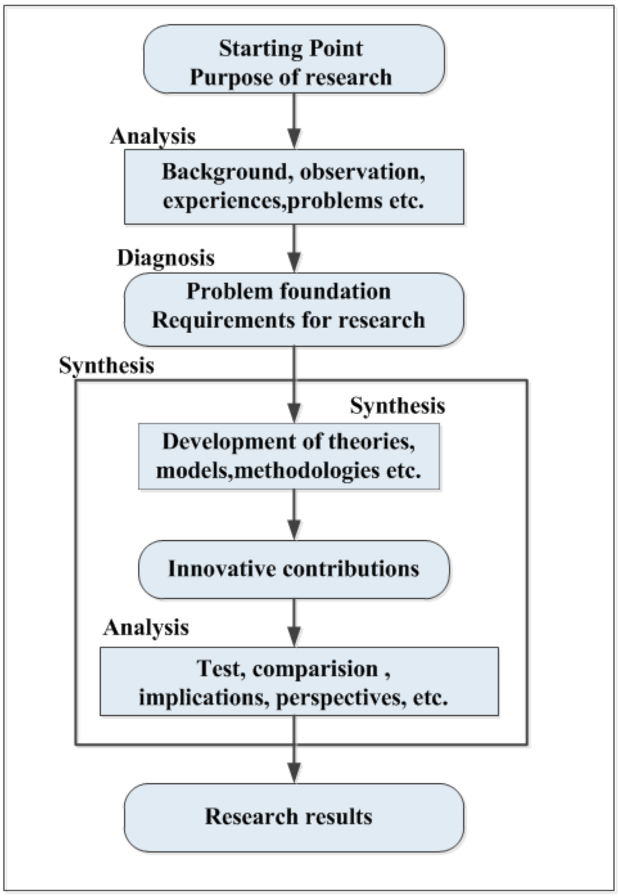


Fig.4. Methodological research procedure based on systems theory [Jørgensen, 2000].

The paradigms and the research procedure are used implicitly and explicitly in this thesis. The methodological research procedure is used to structure the research process. In applying this procedure, the outer layer is used to identify problems in mechatronic design and to utilize the concept of mass customization. This involves analysing the capabilities of the existing theories and approaches of mass customization that can also be applicable to this kind of system. The identification and formulation of problems in the outer layer are followed by the innovative part of the design activity, which includes the testing, implications and perspectives of the theories and related methods.

4.1.6. Research Methods and Strategies of Inquiries

According to Creswell [2009], research designs (based on world views, strategies of inquiry and methods) tend to use quantitative, qualitative or mixed methods. A researcher can decide upon these research methods according to the characteristics of the project:

Quantitative methods: This type of method makes use of questionnaires, surveys and experiments as strategies of inquiry to gather numerical data and use this it in a statistical analysis. It measures variables and expresses the relationship between them on a sample of subjects, where the focus is mainly on cause and effect and, to a large extent, on testing a theory. Procedures are developed before the study begins and a hypothesis is formulated to test a theory; in other words, it is deductive in nature. From the paradigmatic point of view, a single reality is perceived and can be measured by an instrument, where the researcher ideally plays the role of objective observer.

Qualitative methods: This type of research presents data as a descriptive narration expressed in words and attempts to understand phenomena in their natural setting. Qualitative research is exploratory, especially when used to explore and conduct research about a new study or topic. Various strategies of enquiry are pursued in this method, including case studies and narrative research. Its assumptions about the world include multiple realities. The results are detailed context based generalizations about the topic.

Mixed Methods: In these approaches, both qualitative and quantitative methods are combined on the assumption that the solution to the problem can be best handled by using diverse types of data. Sequential, concurrent and transformative procedures are usually used in mixed methods.

This thesis predominantly uses qualitative methods, but it also adopts quantitative methods to a more limited extent. One of the subprojects in this investigation is relevant to sequential mixed methods. In the literature review from the previous chapter, it is evident that the concepts of mass customization and modularization have not been sufficiently utilized and explored in the context of mechatronic systems. Consequently, qualitative and mixed methods are used in this project, since one aspect of qualitative research is exploratory in a broad context. Furthermore, these methods can help to identify phenomena and influence and generate new theories that are needed in order to fully utilize the concepts of mass customization in mechatronic systems. Taken from the qualitative realm, case study research is employed as the dominant strategy of inquiry in this project, where the details are explained in the next section.

4.2. Case Study Research

Yin [2003] defines a case study as an empirical inquiry that investigates a contemporary phenomenon within its real life context, especially when the boundaries between phenomenon and context are not clearly defined. This statement implies that an important strength of case studies is the ability to undertake an investigation into a phenomenon in its context, without the need to replicate the phenomenon in a laboratory or experimental setting in order to understand the phenomenon. Gillham [2000a, p.1] defines a case study as an investigation to answer specific research questions which seeks a range of evidence from the case settings.

Case studies are useful in providing answers to ‘how and why questions’ and they can be used in this way for exploratory, descriptive or explanatory research. Case studies have often been viewed as a useful approach to explore new research areas or an inadequate existing theory or knowledge about a topic. According to Eisenhardt [1989], case studies are particularly well suited to new research areas or research areas for which existing theory seems insufficient. This type of work is highly complementary to incremental theory building from normal science research. The former is useful in the early stages of a research project or when a fresh perspective is needed, while the latter is useful in the later stages of knowledge development.

4.2.1. Case Study Design

A case study design can be categorised along two dimensions: the number of case studies contributing to the design and the number of units in each case study (Fig.5).

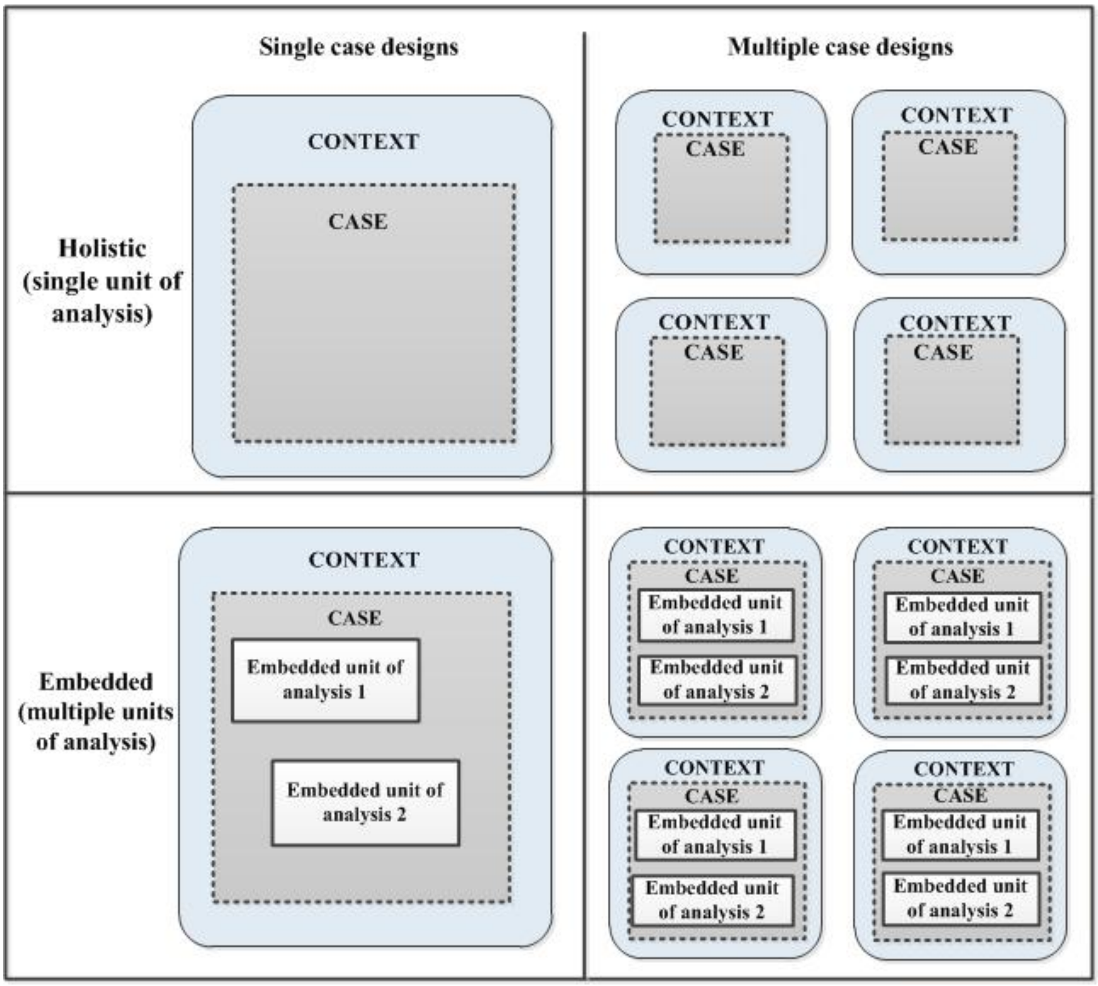


Fig.5. Types of case studies design based on Yin [2009].

Multiple case designs are preferred in this project due to replication logic: if two or more cases are shown to support the same theory, replication can be claimed. In analytic generalization, each case is viewed as an experiment, and the greater the number of case studies that conform to replication, the greater the thoroughness with which a theory can be established. Case study design can involve holistic or embedded studies. In this project, holistic design has been used in multiple case designs.

4.2.2. Generalization, Validity and Reliability

Generalization of the case study is important so that it can contribute to theory. In this project, the method of generalization is not statistical generalization, but analytical generalization in which a previously developed theory is used as a template and is applied to cases to develop knowledge from the empirical results of the case study. The following tests were performed to establish the *quality* of the empirical research:

<u>Construct validity:</u> this is achieved by linking data collection measures to research questions and by establishing measures for the concepts in question and their application to the cases using experiments.

<u>Internal validity:</u> this is not relevant to this project since internal validity is used for causal studies, where cause and effects are examined. Basically, this approach is more relevant to quantitative case studies.

<u>External validity:</u> in this project, external validity is established by identifying the domain to which study findings are generalised, i.e. the application of mass customization theories and concepts to the design and development of mechatronic systems.

<u>Reliability:</u> it is very important to ensure that the operations of research study such as data collection can be repeated with the same results, and therefore they must be properly documented. In this project, the

documentation and reporting have been undertaken by writing papers and articles for conferences and in journals.

4.2.3. Data Collection and Analysis

An important aspect of data analysis in qualitative case study is the search for meaning through direct interpretation of what is observed by the researcher as well as what is experienced and reported by the subjects. For the case studies in this thesis, multiple sources of evidence are collected, such as documents, direct observations and physical artefacts. All these sources are used to establish the same fact or finding by a process known as triangulation. Mostly primary data sources are used to undertake the cases studies. Each source has its strengths and its weaknesses, and the richness of case study derives mainly from these multiple perspectives derived from various sources of evidence.

4.3. Structure of the Thesis

An overall overview of the project is illustrated in Fig.6 and includes the philosophical assumptions, methodological procedure, relevant chapters, case studies and contributions. The inner synthesis part in the methodological procedure is covered by the results and contributions in chapter 5 and chapter 6. The inner analysis part in the methodological procedure is covered by the testing of a modularization method in chapter 6.

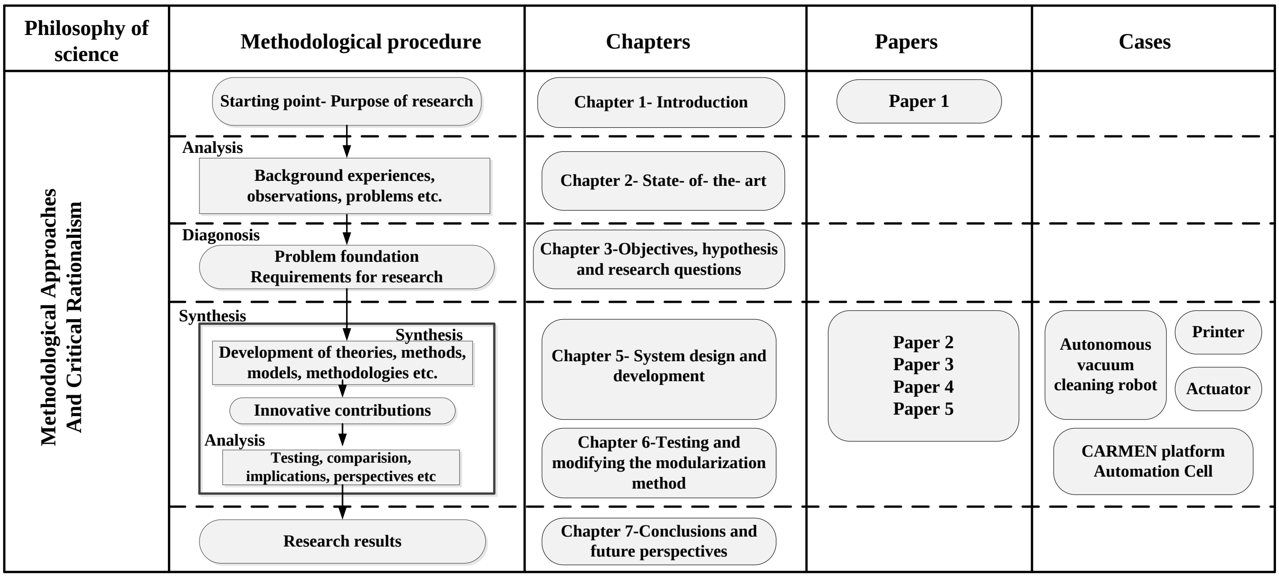


Fig.6. Overview of how the methodological procedure, thesis structure (chapters), scientific papers, and the empirical cases are used and related to each other.

5. SYSTEM DESIGN AND DEVELOPMENT

This chapter presents the results and contributions regarding system design of mechatronic systems in relation to mass customization. The chapter is divided into three sections. In the first section, a method is proposed using system architecting to support the development of product architecture for the next generation products. In the second section, multidisciplinary product decomposition and structure analysis is presented by employing design structure matrix modelling. In the last section, mechatronic module development is presented by two studies: first, modelling and simulations using Bond Graph approach and second, modelling and investigations of a physical prototype of an actuator.

In this thesis, the V-model (refer to Introduction) is being used as an overall approach for system design and development of mechatronic systems. In the first study, a method is proposed to support the process of translating requirements into system requirements and transforming system-level specifications into subsystems and components. In this process, system architecting is effectively applied to derive the parameters of a product and its subsystems based on the abstract descriptions of products such as function requirements. In the next study, multidisciplinary product decomposition and structure analysis is performed by employing design structure matrix modelling. This modelling is used for module identification and to assess the degree of modularity in multidisciplinary products. Finally, modelling and simulations of electromechanical system is conducted to address the tasks system integration, subsystem interactions, and verification of the design. System models are developed to assist the mechatronic module development process. Furthermore, modelling and investigations of a physical prototype of an actuator is presented.

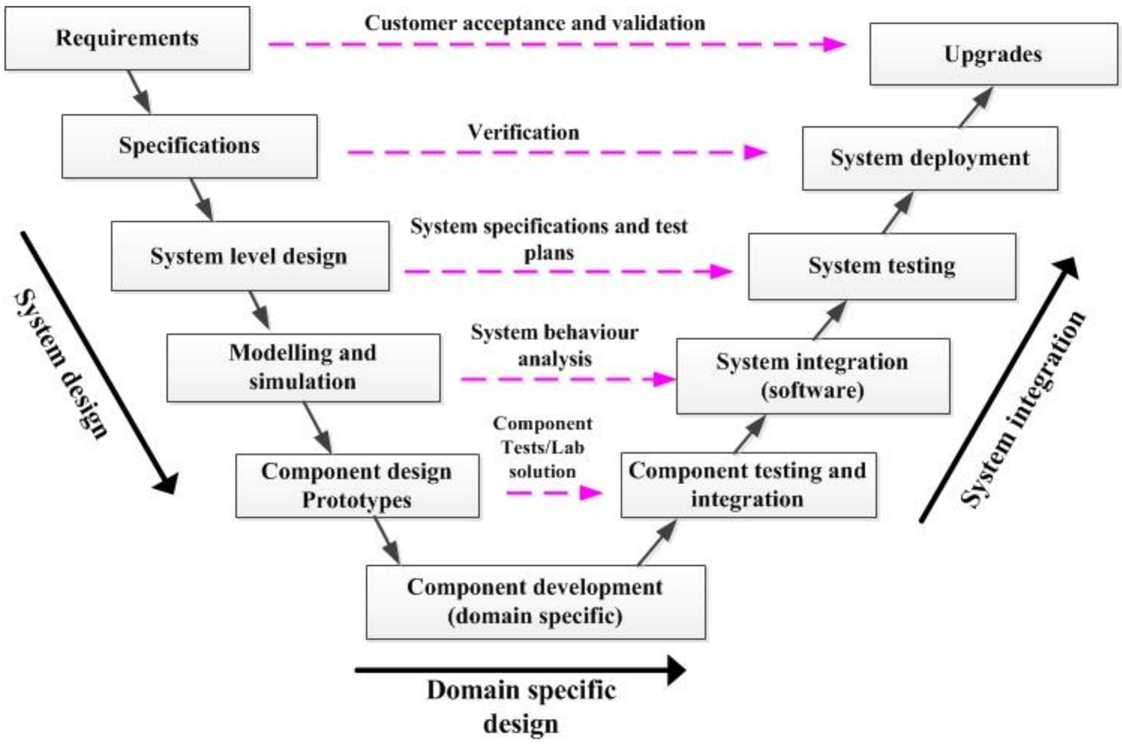
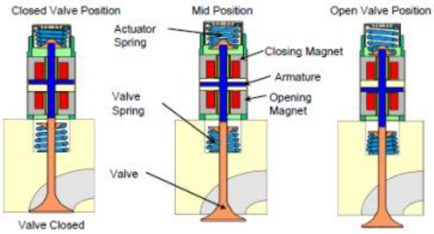
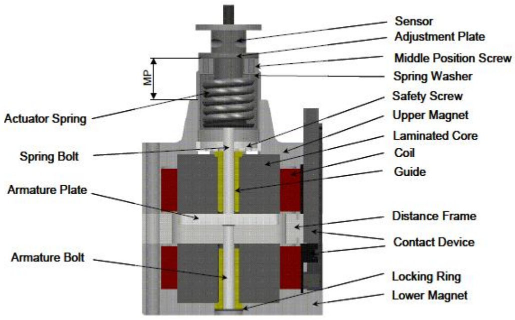
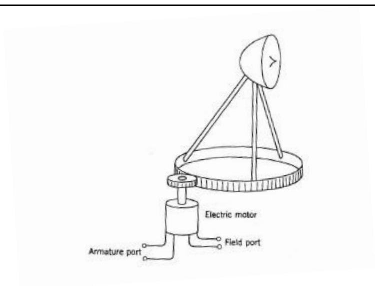


Fig.7. V-model is used as an overall approach for system design and development of mechatronic systems (adapted from VDI 2206, 2004).

A brief introduction of the cases and examples used in this thesis are presented in the following table.

Table. 2. Overview of products used for analysis and verification of results.

<table>
<tr><th>Cases and examples used in this project</th><th>Picture of Product</th><th>Short description of product</th></tr>
<tr><td>Autonomous Vacuum cleaning Robot (VC robot)</td><td> </td><td>An autonomous robot is being used as vacuum cleaner at living places. It automatically guides itself around for cleaning surfaces.</td></tr>
<tr><td>Electromechanical actuator</td><td>


</td><td>Main part of the system is an electromechanical actuator, which operates as a free oscillation system with electromagnets holding the valve for opening and closing positions in engines.</td></tr>
<tr><td>Laser Jet Printer</td><td></td><td>Laser Jet printer used in this project as an example for system decomposition and structure analysis.</td></tr>
<tr><td>Radar antenna pedestal drive system</td><td>
</td><td>Basic model of the drive system consist of a DC motor, gears and pedestal used as an example for modeling and simulations.</td></tr>
</table>

At system design the specifications derived from user needs are transformed into component level specifications by defining subsystems and components. This is the conceptual design, where the purpose is to define concepts and solutions, which describe the main functional and structural characteristics of the product (Fig.8).

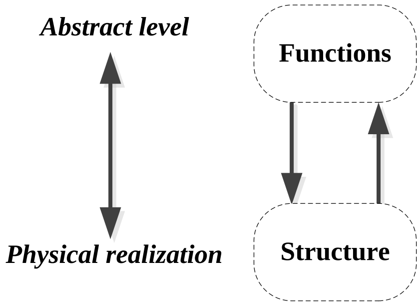


Fig.8. Function and structure representation of products.

5.1. System Architecting Using Function-Behaviour-State Modelling

One of the issues in the early stage of mechatronic product design (i.e., system architecting as defined in 1.1.1) is to effectively derive the parameters of a product and its subsystems based on the abstract descriptions of products such as function requirements. To support system architecting, parametric representation (parameter network) using the definition of conceptual relations to identify the physical structure is crucial. The major challenge of the researchers studying conceptual design is to develop modelling schemes supporting the process to derive them. However, the modelling schemes should be independent of specific disciplines. It is because of that designer (system architects) have to communicate with engineers of specific disciplines (mechanical and control engineers).

In this study the Function-Behaviour-State (FBS) modelling scheme has been used as a fundamental solution to deal with the process of system architecting, because the FBS modelling scheme supports conceptual design independent of specific disciplines.FBS modelling is based on three main concepts: function, behaviour and state (Fig.9). It defines a function as ‘a description of behaviour abstracted by humans through recognition of the behaviour in order to utilize it’. It represents a function as an association of two concepts: symbol of human intention represented in the form of *to do something* and behaviour that can exhibit the function. Behaviour is defined as a transition of states over time. In this modelling, a state is represented by entities, their attributes and their structure, which also represents a physical realization of the system [Umeda et al, 1996]. The FBS modelling scheme employs physical phenomena as symbolic concepts defining (conceptual) relations among the parameters of a product.

The FBS modelling is a domain independent modelling scheme that can be used to represent mechatronic systems in one model. In this modelling, function is a concept at abstract level that is independent from any domains and can be applicable to both hardware and software, and as well as to purely mechanical and electronically controlled sensor actuator systems. In this representation, behavioural description of the product is necessary to gather information about its structure and combine it with the information regarding the involved phenomenon, with the purpose of determining how the product works and the relations between its parameters.

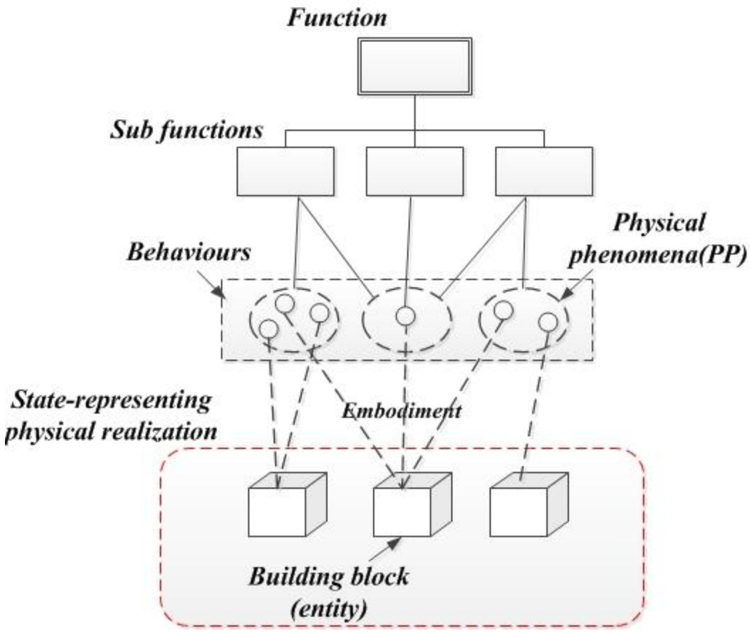


Fig.9. Hierarchical decomposition of system into functions, behaviours and states using FBS modelling.

FBS modelling represents mapping among function, behaviour and product structure, which is important for the design methodology to facilitate redesign and develop next generation products. FBS modelling also supports the embodiment of functions into physical structure.

Hierarchical decomposition of the system, which is based on the FBS modelling (see Fig.9), has been used as a representation scheme for product models in the proposed method (see section 5.1.1).

The advantages of the FBS modelling are:

- It associates the functional descriptions with the states representing physical structure via a behavioural level in between. In this way developing the functional model of the system goes in parallel with consideration of the real physical environment.
- FBS modelling facilitates the decisions at the lower level i.e. at entities that should be traced to higher level abstract concepts, such as functions.
- This modelling can be implemented in the computational tool that supports system architecting tasks in complex mechatronic systems.

In the following section, a method is proposed and explained by comparative analysis of autonomous vacuum cleaning (VC) robots for the development of product architecture. Autonomous VC robots are modelled from the functional, behavioural, and structural perspectives, and used for the development of a design guideline for the product architecture for the next generation from these perspectives.

5.1.1. A Method to Develop Product Architectures

In this study, a method (Fig. 10) is proposed to support the development of product architecture for the next generation by (a) effectively utilizing the design knowledge of the current generation, and by (b) computationally supporting the development process. To demonstrate and validate the method, a product modelling scheme and computational modelling environment are crucial. Especially, the modelling scheme should support domain independent modelling description and a computational modelling environment should be compatible with the modelling scheme. As an example of the available combinations of modelling schemes and computational environments, the study employs Function-Behaviour-State (FBS) modelling and a CAD system for system architecting (SA-CAD).

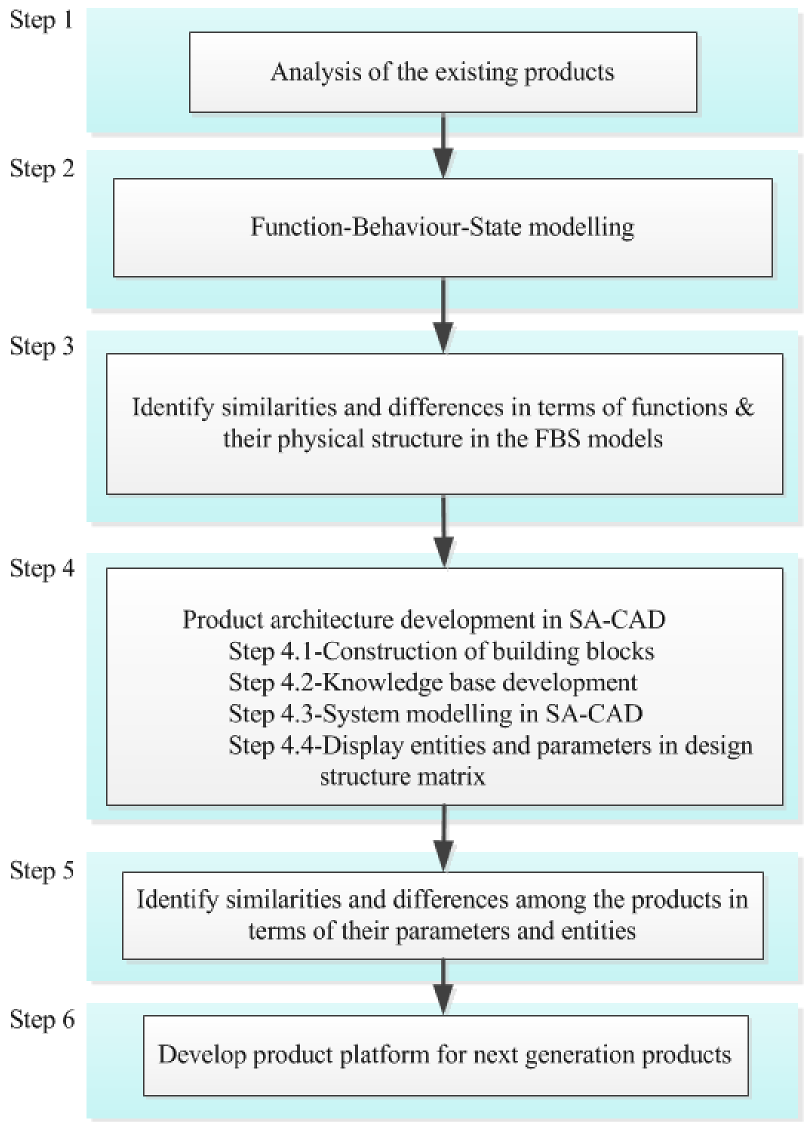


Fig.10. Schematic description of steps in the proposed method. A method for the development of product architectures for next generation products.

This method explains the suitability of the scheme and computational support to the execution of the study in comparison with the related work. This method describes the application of the proposed method in the development of product architecture of autonomous vacuum cleaning robots.

The proposed method supports the development of product platforms and product families based on the analysis of mechatronic systems of the same kind by FBS modelling. Fig.10 shows a brief schematic overview of the proposed method.

The designer first clarifies the functions of existing products of the same kind in the market and then analyses how these functions are physically realised. The designer subsequently analyses the differences among the products in terms of their functions and physical realization. The FBS models of the products are developed to perform the comparison. Using these FBS models, the designer can explicitly identify the similarities and differences in the products in terms of the functions and their decomposition. Furthermore, a mapping from the functions to their realizations in terms of physical phenomena (PP) and entities (i.e.

physical object to describe the abstract concrete relationship among concepts) is identified. The conceptual relations between the physical phenomena and the entities given in these FBS models provide designers with a mapping from the conceptual relations to the customer requirements and also with parameters used for design of the products and their dependency. The designer extracts parts of these mappings and constructs building blocks at different (functional, behavioural and structural) levels. With these building blocks, the same products are modelled by hierarchical physical decompositions and compared in terms of the dependency among the parameters of the products. Finally, the designer identifies the similarities and differences among the products in terms of their parameters and dependencies, and this information is subsequently used for the development of a product platform for the next generation.

Step 1. Analysis of Existing Products

Initially, the functions of existing products of the same kind are observed. Then how these functions are physically realized is identified. Then the differences among the existing products in terms of the functions and their physical realization are analysed. For this analysis, multiple sources of evidence have been used such as manuals, direct observations, videos and physical analysis.
In the initial analysis of VC robots (Table 3), it has been observed, that all these system have three main functions that are to collect dust and debris, to generate motion and to move itself (as these are autonomous systems). However, these systems differ on how to collect dust while in motion. Some of the systems use side cleaning and throw dust under the robot and lift above by rotation to be vacuumed inside, while others use rotation to direct dust particles with side cleaning. Further analysis reveal that all systems use vacuum for lifting dust particles but differ in the type of vacuum (bypass or direct injection). All of them use dust filters but there is a difference in the size of the filters.

Table.3. Analysis of multiple vacuum cleaning robots (product) to identify the differences in their functions and their physical structure (see paper 1 for details).

Step 1	**Analysis of multiple vacuum cleaning robots**	
a.	Functional analysis of all systems	In the functional analysis it has been observed that how the different systems detect, collect and store the dust and particles while navigating autonomously. The three main functions identified are related to *collect dust, generate motion and to move itself.*
b.	Identification of functional differences	From functional analysis differences in these systems are identified in sub functions such as *to navigate itself* and *to collect dust and particles* etc.
c.	Physical analysis of all systems	Then their physical structures are observed with criteria such as robot shape, size, number and type of sensors and actuators, way of collecting dust (i.e. using single or counter rotating brushes etc) and way of lifting dust and particles.
d.	Differences in physical level of all robot systems	Differences in physical structure are identified in all systems on the basis of above criteria.

All systems are autonomous to perform its intended function of cleaning surfaces. Unlike a normal cleaner, these robots move itself around room with two large threaded wheels, each one independently driven by a separate electric motor. The wheels can turn in opposite directions, which means it can spin and clean almost any space it can drive into. All systems use spot mode i.e. to stay at dirty spots, but differ how they clean and type of sensing dust particles. All system follows walls and sensing obstacles in their way and also able to sense heights. They are able to perform self charging, when the battery is low, and able to moves back into its docking station and recharges itself for next time.

Further analysis reveal that there are some special functions in some systems i.e. various onboard sensors and able to anti tangle while stuck in rugs and cords. They are also able to navigate from one room to other room.

Step 2. FBS Modelling

After the analysis of the existing products of same kind, the next step in the proposed method is the development of FBS models of multiple products. To perform system decomposition and compare the systems, these models are developed with the following system architecting tasks,

- The customer requirements are translated into system level specifications i.e. abstract as well as parameter-level descriptions to model a product.
- The main function is decomposed into sub-functions until no further functions can be realized
- The system models are developed by embodying the functions into behaviours and physical structures, i.e. identifying the physical phenomena and their relation to the entities of the system.

First, customer requirements are translated into system-level specifications. Then, the main functions are identified and decomposed into sub-functions until no further functions can be realized. After the functional decomposition, in the second phase, develops the system by embodying the functions into behaviours and structures. At the behaviour level, the physical phenomena and the related parameters are identified. At the lower level of decomposition, all sub-functions must be realized by structural elements, called entities or building blocks. These entities are related to each other by parameters. The parameter network represents the architecture of a system. This process is used to decompose the system specifications into subsystems and components until all the functions of the system have been realized (see Fig.9).

Using the system architecting tasks, Function-Behaviour-State models of models of the VC robots are developed. For instance, in the FBS model, as shown in Fig.11, functions are linked to physical phenomena (*PP*) at the behaviour level and further linked to relevant entities at the state level. For instance, to fulfil the customer requirement such as cleaning floors independently, the main function is decomposed into sub functions until no further functions can be realized. For example function to navigate itself is decomposed into various sub functions. One of the sub functions, to avoid obstacles is further decomposed into two sub functions such as: to sense obstacles and to take turns. The robot sense obstacles by proximity and by touching. In the next step at the behaviour level, physical phenomena are identified, for instance to touch any object the designer identify collision as physical phenomenon .At the state level, this PP is further linked to the relevant entities. In this example the physical phenomenon i.e. collision is related to object and bumpers as entities. Similarly PP collision sensing is linked to mechanical switches (2 touch sensors), object and microcontroller as entities. Function-Behaviour-State models of the remaining systems are developed in the same manner.

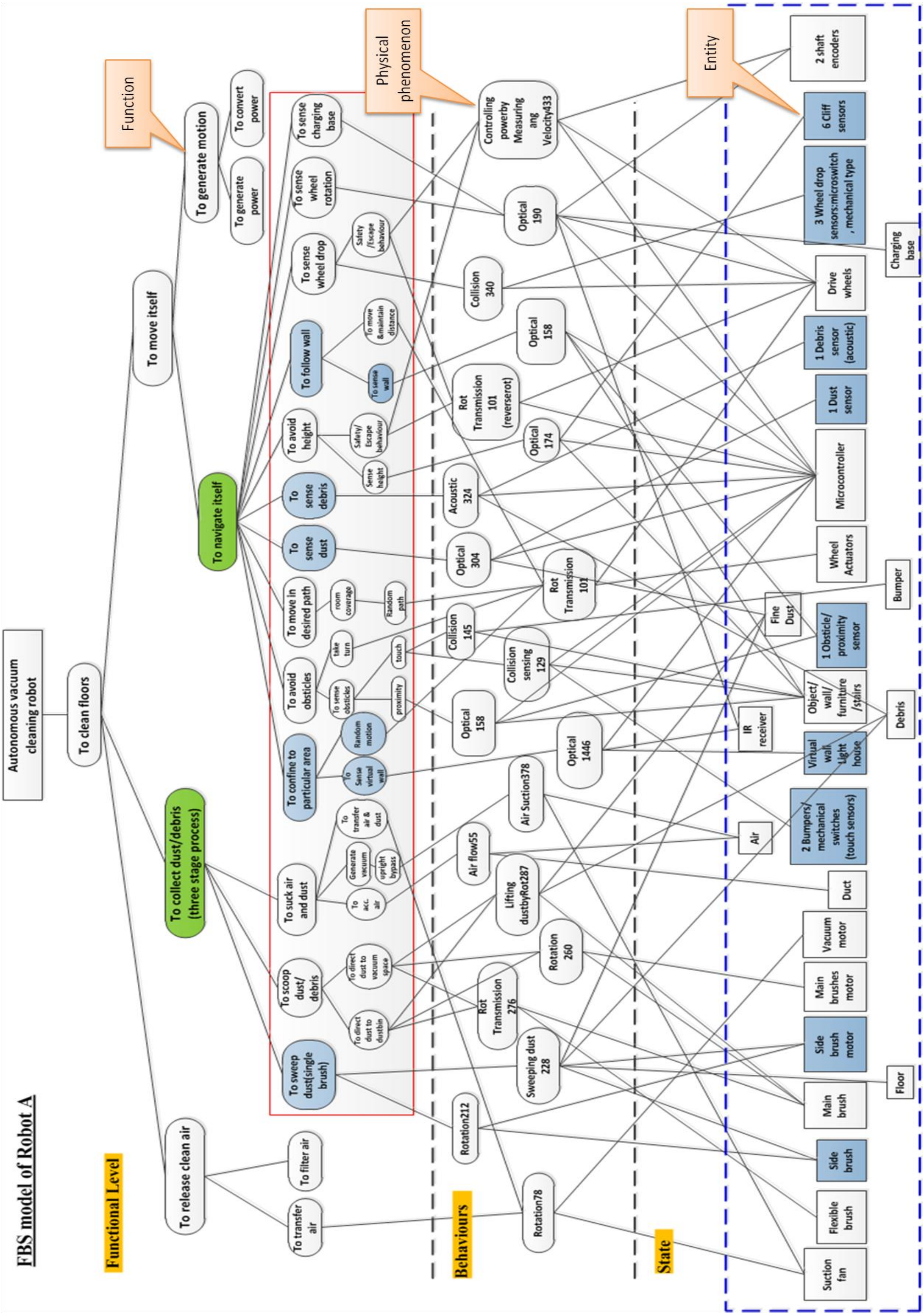


Fig. 11. Function-Behaviour-State model of robot system A. Coloured boxes represents the differences in functions and entities in the FBS models. In the FBS model, customer requirements are transformed into system level specifications and further into entities of the system.

Step 3. Identify Similarities and Differences in Function-Behaviour-State Models

In this step, FBS models of the multiple products of same kind can be analysed to identify the difference in the respective systems. These models can explicitly be used to identify the similarities and differences in terms of functions and their physical decomposition as shown in Fig. 12. The differences in the models help to clarify how these systems can be redesigned.

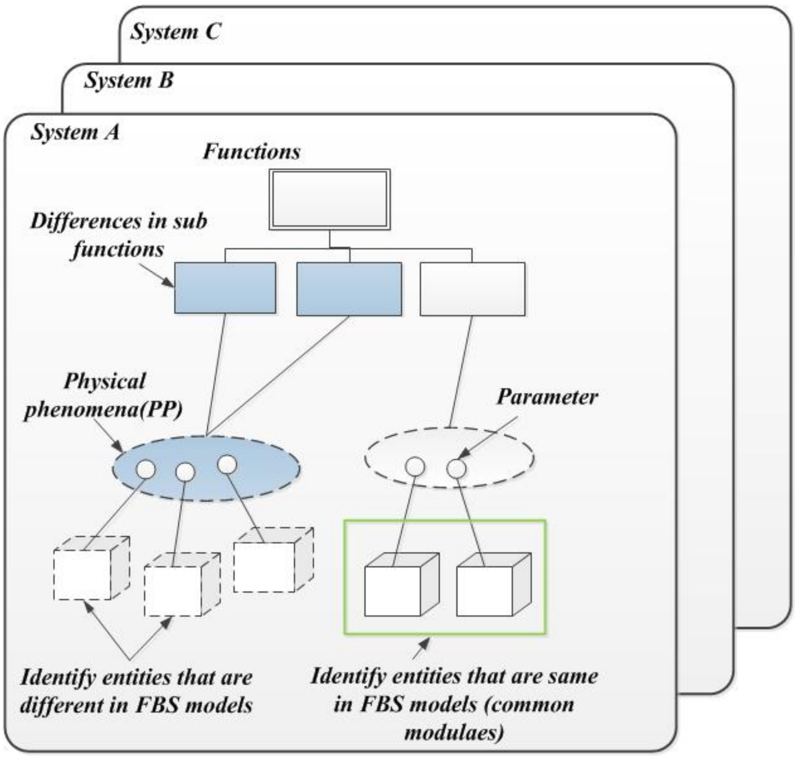


Fig.12. Analysis of the multiple products regarding functions and their physical realization. Colours are used to highlight their differences at functional and physical levels.

Once the FBS models of the VC robots are formed, colours are used to highlight their differences, as shown in Fig. 5. These differences are evident in functions (i.e. *to collect dust and debris* and *to navigate itself*), in subsystems and components (e.g. room positioning system and camera) and in their environment (e.g. magnetic boundary markers and virtual walls). The functional decomposition reveals that the top-level functions of the three systems are the same:

- To collect dust and debris
- To navigate itself
- To generate motion
- To release clean air

As the above functions are further decomposed into sub-functions, differences are realized in two of the functions: *to navigate itself* and *to collect dust and debris.* The functions *to release clean air* and *to generate motion* are not decomposed further into behaviours and states, because their physical realization is the same in all systems. The function *to navigate itself* includes different behaviours that are realized by each robot depending on customer requirements and their respective capabilities, such as to facilitate efficient navigation. Similarly, all the robots have differences in the function *to collect dust*. The functional differences of the three robots are summarized in Table 4.

Table. 4. Differences in functions of three robots (systems).The differences in functions are useful to identify the differences at structures levels.

Function	Sub-functions		Robot A	Robot B	Robot C
To navigate itself	To sense dust		X		
	To sense debris		X		X
	To follow walls		X		X
	Room coverage	Back and forth: line by line		X	X
		Room mapping		X	X
		Random path	X		
	To confine to a particular area	To sense virtual wall	X		X
		To sense boundary markers		X	
To collect dust and debris	To sweep dust (one side)		X		
	To sweep dust (both sides)				X

Function modelling support to identify the entities, for instance in the above table (the result of FBS models of the systems), the sub function : to confine to particular area, robot A and robot C sensing virtual wall using optical sensors (entities),while robot B sensing boundary markers using magnetic sensors (entities) for the intended purpose. Similarly objects, walls, dust and debris can be linked to entities using the PP in between.

The differences at the physical level can also be realized; however, this can be identified after the system architectures are formed.

Step 4. Product Architecture Development in SA-CAD

In this step, each product is modelled by hierarchical physical decompositions at various levels (i.e. functional, behavioural and structural) in SA-CAD. With this modelling the architecture of the system is formed as a network of parameters that are related to various entities or building blocks realized by the physical phenomena.

Step 4.1. Construction of Building Blocks for Architecture Modelling

In this sub step, the mapping from conceptual relations to entities in the FBS models is used to identify a mapping from conceptual relations to the customer requirements and also to acquire the design parameters of the products and their dependency. The designer extracts parts of these mappings to develop building blocks by the hierarchical decomposition of each product at various levels: functional, behavioural and structural.

For instance, in the VC example the building blocks in the suction system are *vacuum motor*, *fan* and *duct*. Their *PP* is rotation, air suction and airflow. Relations between the vacuum motor and the fan are angular velocity and torque, respectively, as shown in Fig. 13.

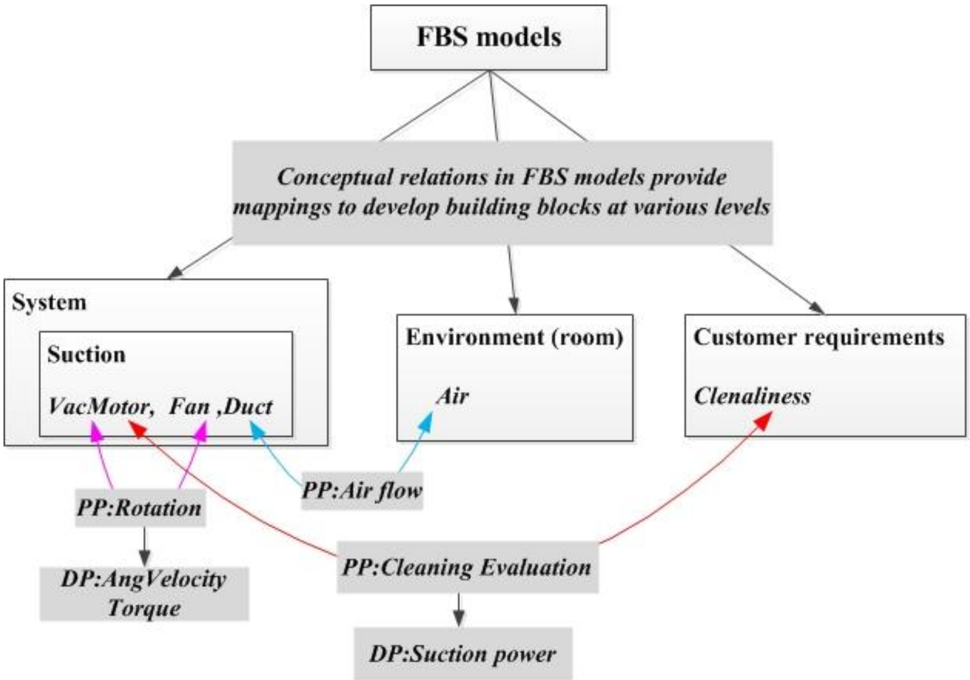


Fig. 13. Example of building blocks and their parameters for network modelling. The building blocks are related with relevant attributes.

In the following section, the design concepts and knowledge base development is described before architecture modelling.

A conceptual design process based on FBS modelling is supported by the knowledge intensive engineering framework (KIEF). It is defined as a framework for integrating design modelling systems that has embedded knowledge about domain theories [Yoshioka, 2004]. Several concepts such as physical phenomena, physical features, attributes of entities and physical laws are described in the knowledge intensive engineering framework. This study uses the concepts of *function, behaviour, state, entity, attributes, physical phenomena* and *physical law* as defined in the knowledge intensive engineering framework and in FBS representation as follows:

Entity: An entity represents an atomic physical object and its purpose is to describe the abstract concrete relationship among concepts. In this study, an entity is used as a building block or module. The entity concept in the KIEF is defined with its 'name' and 'supers'. In this framework, 'supers' represent a higher level category preceding the name of concepts. The following is an example of an entity:

Name: Fan (?e)
Supers: MechanicalParts (?e)

Relation: A relation represents a relationship among entities to characterize a static structure. The following is an example of a relation:
Name: Contact (? contact)
Supers: Relation (? contact)
HasRelations: HasRelation(?contact,?fan,?vacmotor), Has-Relation(?contact,?vacmotor,?fan)

Attribute: An attribute is a concept attached to an entity and takes a value to indicate the state of the entity. It is defined with its names, supers and statements. In addition to the two definitions (i.e. names and supers), 'statements' describe additional information such as the dimension of the attribute and its definitional relations with other attributes. The following is an example of an attribute:

Name: AngularVelocity (?a)
Supers: Attribute (?a)
Statements: DifferentialOf (?a, AngDisplacement,Time), UnitOf (?a,?,'m/s'), DimensionOf (?a, (Lt), (10-20))

Physical phenomenon: A physical phenomenon indicates physical laws or rules that govern behaviours. In many cases, a physical phenomenon represents the relationship among attributes of entities, and 'attributes' define the related entities and attributes of the defined phenomenon. For a complex physical phenomenon that is difficult to represent as a relationship among attributes, it is defined by using only 'entities' information. 'Statements' describe the relationships among entities, attributes and the physical phenomenon. In statements, the 'OccurTo' predicate describes the relationship between the phenomenon and related entities. The 'HasAttribute' predicate describes the relationship among related attributes and related entities. The following is an example of a physical phenomenon:

Name: Rotation (?p)
Supers: Motion (?r)
Attributes: Torque (? t), AngularVelocity (?angvel)
Entities: Mass (? fan), Mass (? vacmotor)
PhysicalLaws: SecondLawOfNewtonLaws (? f,?m,?acc)
Statements: OccurTo(?p,?fan,?vacmotor), HasAttribute(?t,?vacmotor), HasAttribute(?angvel,?fan), HasAttribute (? angvel,?vacmotor)

Physical law: A physical law illustrates a simple relationship among attributes. 'Name' and 'attributes' define the name and related attributes, respectively, of the defined physical law. 'Expression' defines the relationship among attributes by using a mathematical equation.

Name: SecondLawOfNewtonsLaw
Attributes: f_Force, m_Mass, a_Acceleration
Expression: Sigma (f) = m * a

Behaviour: Behaviour is defined by sequential changes of states of a physical structure over time. For example, in VC robot the behaviour of a fan depends on the torque generated by the vacuum motor.

State: States are the different modes of a physical system or entity. Changes in these modes are the underlying causes of behaviours. In the VC robot example, 'rotation' is physical phenomenon between a vacuum motor and a fan, and it is caused by the 'torque generation' of the motor; rotation depends on the state transition of 'torque' of the vacuum motor.

Step 4.2. Knowledge Base Development in SA-CAD

In SA-CAD, the parameter network modeller supports the development of the parameter network of a product, which represents the parameters and their relations. In this tool, a knowledge base of the systems is developed by writing the physical phenomenon, attributes, entities and their relationships. The FBS models support to develop the building blocks of the subsystems that are further used to develop the knowledge base in the tool.

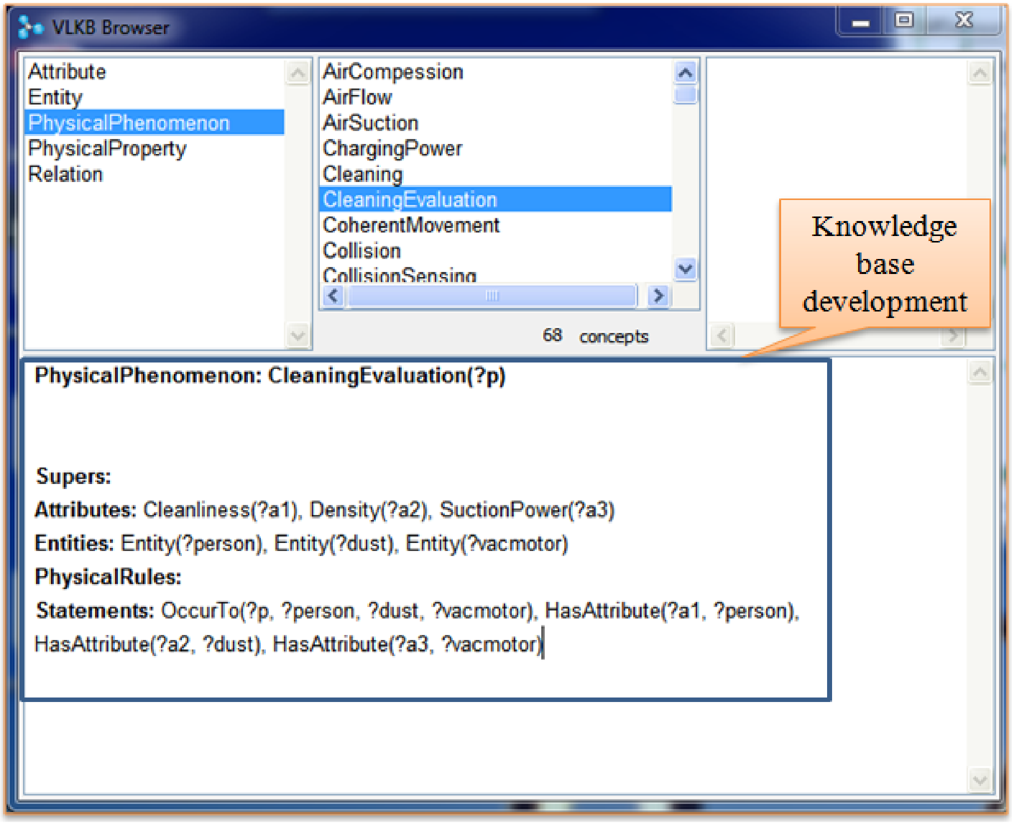


Fig.14.An example of Knowledge base development in SA-CAD. A knowledge base of the concepts such as physical phenomenon (i.e. cleaning evaluation), attributes, entities and their relationships is written in SA-CAD.

Referring to VC robot example (Fig.14.), cleaning evaluation is the PP, and cleanliness is the customer perception with regard to the ability of the system to clean floors, density and suction power are the other attributes that are entered for dust and vacuum motor respectively. Similarly, Hall Effect (i.e. to generate voltage difference across a conductor) is another concept; the physical phenomenon of Hall Effect is developed by entering the respective attributes, entities and the relations between the entities.

Metamodel Development in SA-CAD

In SA-CAD, a metamodel represents a design object as a network of concepts, and these concepts are written and stored in the concept base of the tool. A metamodel is built as a network of these instantiated physical concepts. Important feature of this modelling representation is, once the model is formed, any modification of the metamodel of the system is possible by either changing the existing concepts or new concepts can be added later on.

For instance in VC example, an initial view of the metamodel is shown in Fig.15, this represents the concepts related to a function (i.e. cleanliness), along with physical phenomenon, attributes, entities and their relations are defined and written in Sa-CAD.

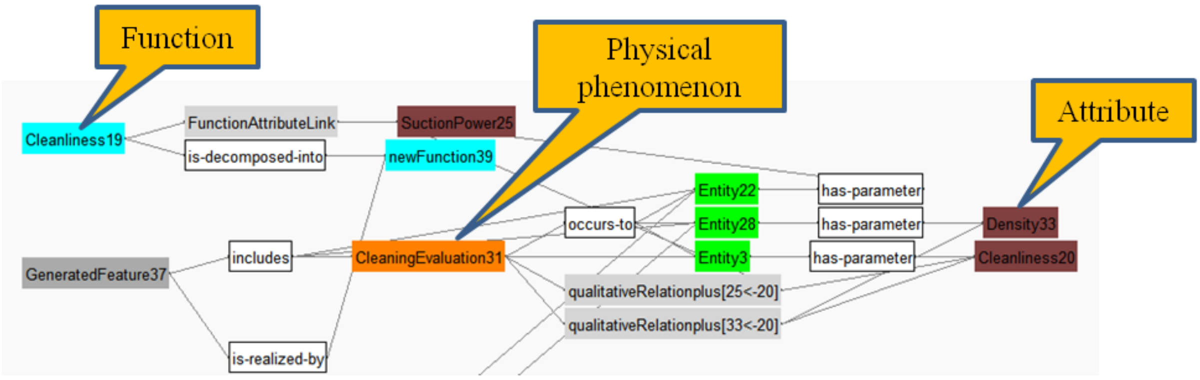


Fig.15. An example of the metamodel in SA-CAD.

Step 4.3. System Modelling in SA-CAD

With the building blocks and knowledge base, each product is modelled by hierarchical physical decompositions at various levels (i.e. functional, behavioural and structural) in SA-CAD. The interface of the tool is used to select the physical phenomena (*PP*), the related parameters and the decomposition candidates to develop the architecture by selecting the system entities and their relations.

In the hierarchical decomposition of VC robots in SA-CAD (Fig 16), the following steps are performed:

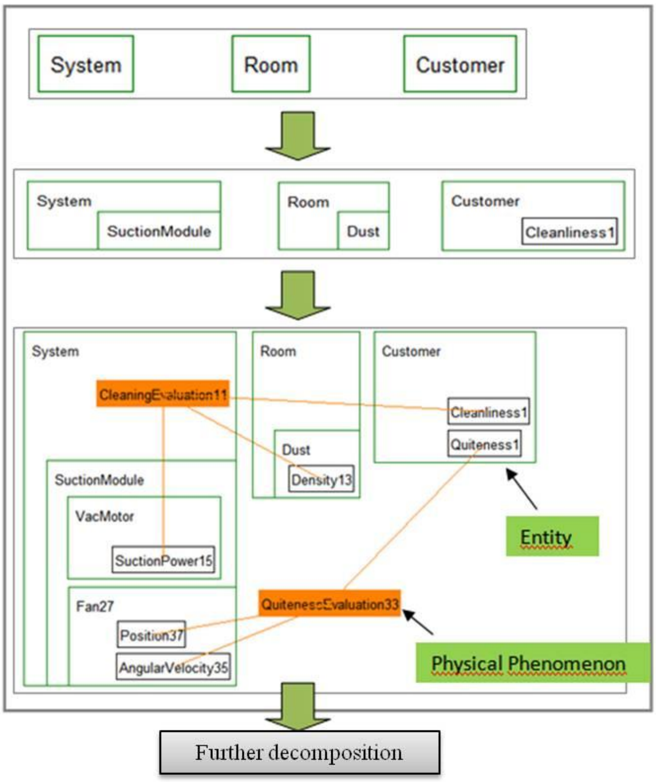


Fig.16. Hierarchical system decomposition process of VC robot.

- First, the decomposition candidates are identified. In this example they are system, room and customer.
- Next, relevant entities are selected that has already been entered in the knowledge base, such as suction module, dust etc. (Fig. 16).

- Finally, the SA-CAD decomposition interface is used to relate the entities, their design parameters and physical phenomena. For instance, the system architect defines *quietness* (as required by customer) as a function of the angular velocity of the fan and the position of the fan in a fan-duct design. Here, entities (person and robot fan) and the design parameters (quietness, position and angular velocity) are related parameters. Then the architect relates the target parameter (i.e. quietness) and physical phenomena with the system elements (or entities) and executes the decomposition in Sa-CAD as shown in Fig. 17.

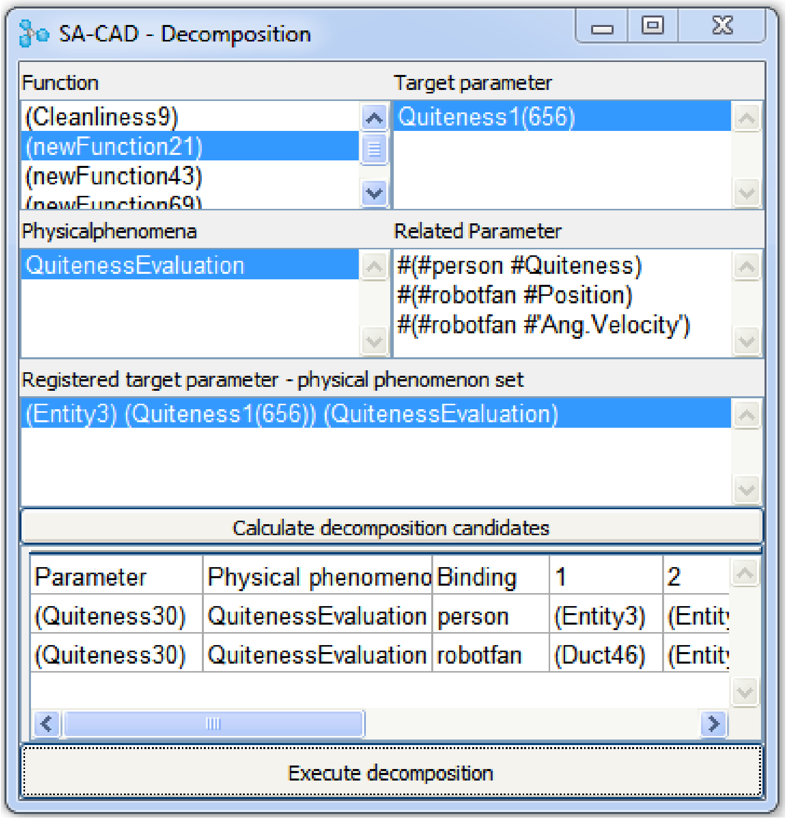


Fig. 17. Decomposition interface in SA-CAD. This interface is used to develop the parameter network in SA-CAD. Entity 3 represent the system element, quietness is a function and quietness evaluation is the physical phenomenon defined in the decomposition interface.

Referring to the VC robot example, the designer identifies essential customer needs (i.e. *cleanliness*, *quietness*, *saving energy and convenience in charging*) and connects them to entities in the environment (i.e. room) and system. For instance, *cleanliness* is the customer perception with regard to the ability of the system to clean floors. Its physical phenomenon (*PP)* can be *cleaning evaluation* that is linked to entities such as vac motor and dust. The respective design parameters are suction power in the vacuum motor and the collection of dust or debris by the system. Others functions are also linked in this manner. The system architect defines *quietness* as a function of the angular velocity of the fan and the position of the fan in a fan-duct design. Similarly, as most of the energy is consumed by the system motors, *saving energy* is associated with the power consumed by the motors. Greater energy is consumed in three entities, the vacuum motor, the main brush motor and the wheel actuators. Hence, energy consumption is related to these three entities within the system. In addition, another function *convenience in terms of charging* is relevant to entities such as the charging base and receiver of the system. In all three systems, the physical phenomena and their attributes relevant to customer needs are the same. Further decomposition relevant to customer needs, such as the size, small weight and shape of the system, can also be introduced in the process.

System Architectures as a Parameter Network

The architecture of the system is formed as a network of parameters that are related to various entities or building blocks realized by the physical phenomena.

In the VC example, in the suction module, three entities, the vacuum motor, fan and duct, are formed by the parameters using the *PP*, such as *rotation*, *air suction* and *air flow*, and they are related to sub-functions, such as *accelerating the air*, *upright bypass* and *transferring air and dust*. The system entities are related to each other through parameters; for instance, the fan and the vacuum motor represent a *PP*, i.e. *rotation*, with the parameters being angular velocity and power. These parameters are the interfaces between the entities, and they link the system to the room environment. In the suction module, vacuum generated by the fan because of the change in pressure forces the air to flow inside the system. That is why the system architect uses *air suction* as *PP* to connect the pressure of the fan to the force of the air.

By using this method, multiple system architectures are developed (Fig. 18) to facilitate product platforms and product family modelling. The commonality of the systems can be analyzed in the parameter network to find common building blocks.

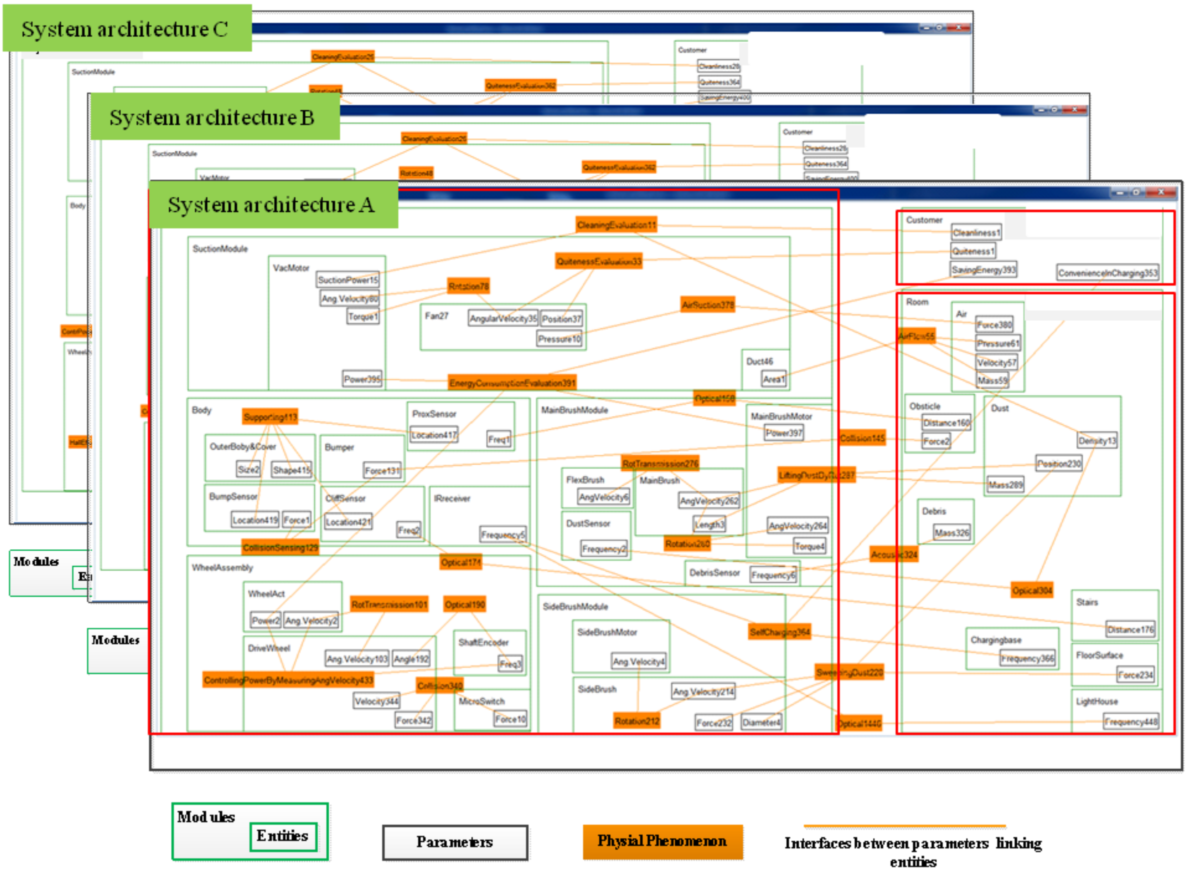


Fig. 18. Architectures of the VC robots in a parameter network relating physical phenomena, entities and parameters or attributes.

The parameter network can also be used to identify the differences in the three VC robot architectures. The differences in their entities and respective physical phenomena (are shown in Table 5) that can be used to develop product families. Further, these parameters and entities can also be displayed in matrix form.

Table 5. Differences in the three architecture at behaviour and structural levels. Numbers in each system represent the quantity of respective entities.

Entities	Physical phenomena	System A	System B	System C
Side brush	Rotation, sweeping dust	1	0	2
Side brush actuator	Rotation	1	0	2
Mechanical switch	Collision sensing	2	4	3
Light house	Optical	1	0	0
Proximity sensor	Optical	1	1	3
Debris sensor	Acoustic	1	0	1
Wheel drop sensor	Collision	3	2	0
Cliff sensors	Optical	6	2	3
Shaft encoder	Optical/Hall effect/Coriolis effect	2	2	2
Magnetic sensors	Magnetic induction	0	2	0
Boundary markers	Magnetic induction	0	1	0
LASER sensor	Optical mapping	0	1	0
IR sensors	Optical	0	0	3

Step 4.4. Display Entities and Parameters in Design Structure Matrix

This sub step represents the relationships between entities and attributes of the system. They describe the design concept developed at the conceptual design stage. The tool helps designers construct matrix representations (Fig. 19) for the analysis of system architecture with matrix-based methods (e.g. DSM), and such analysis can be used for tasks such as architecture development and process organization.

Fig.19 based on VC example; show the correspondences between the entities and the parameter relations that link the behaviours of the systems with their respective structures.

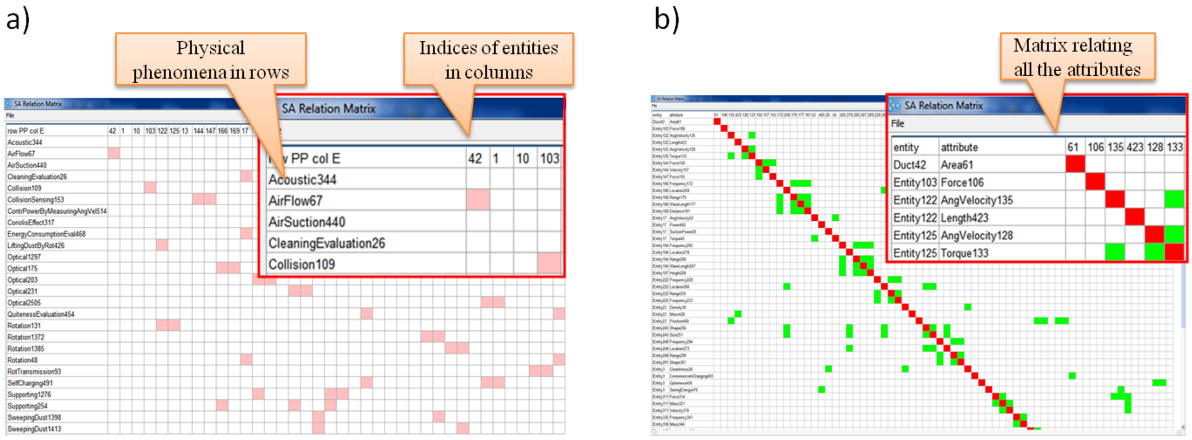


Fig. 19. Correspondences between entities and parameter relations in the VC robot example. These relations are derived from physical phenomena, and they illustrate the differences in these systems at behavioural levels. b) Matrix representation of relations among attributes.

<u>Step 5. Identify Similarities and Differences in Entities</u>

In this step, the commonality of the three systems is identified manually in the parameter network to determine common building blocks or entities. Once the size of the system, the number and location of entities are defined that depend on the requirements of different designs; this differentiation in the architectures can be used for developing product families.

For instance, in the parameter network (architectures) of the three VC robots (Fig. 18), the entities and the associated physical phenomena in the suction system and the wheel assembly are approximately the same. In the body module, building blocks such as bumpers, bump sensors, proximity sensors and cliff sensors are common entities in all three systems. Their physical phenomena and design parameters are also the same. The only difference is the number of these sensors and their location. For instance, System A uses six, System B uses two and System C uses three cliff sensors (Table 6). Similarly, in the main brush module, the main brush and main brush motor are common entities in all three systems. The side brush module is common only to System A and System C.

Table.6. Differences in Entities of three VC robots (systems).

		System A	System B	System C
1	Main brush module	Two counter-rotating brushes	Single rotating brush	Single rotating brush
		Dust sensor		
		Debris sensor		Debris sensor
2	Side brush	Single side brush		Two side brushes
		Single motor		Two motors
3	Body	IR receiver		IR receiver
				Virtual wall sensor
		One proximity sensor	One proximity sensor	Three proximity sensors
				Camera
			Magnetic sensor	
4	Room mapping system		IR Laser sensor	
			RPS motor	
			IR receiver	

The differentiation in the architecture can be used for developing product families. This differentiation is evident in functions such as *to collect dust and debris* and *to navigate itself*, and their respective entities as shown in Table 6. Because of system decomposition into building blocks and the differentiation in architectures, designers can identify and develop product families even in the conceptual phase of multidisciplinary mechatronic systems.

Step 6. Product Platform Development

The proposed method could identify several levels of classifications regarding the level of commonality observed in existing systems, such as behavioural (physical phenomena) level and structural levels (type of components (entities) used, and their numbers and locations). Commonality and differences identified based on such classifications have been used for the development of product platforms. Product platforms can offer the following benefits when applied successfully: companies can develop differentiated products by sharing components across a product platform, they can reduce development time and cost and system complexity and they can acquire the ability to upgrade and redesign products. To utilize the advantages of platform development, standard entities and differentiation entities must be balanced inside a modular architecture. The aim must be to maximize the use of common modules in the architecture while maintaining their distinctiveness.

In the VC example, two platforms are developed that comprise common entities based on the round design and D-shaped design of the robots (Fig 20). These platforms are further extended to product families, where the derivative products can be developed from the three architectures (Table 7).

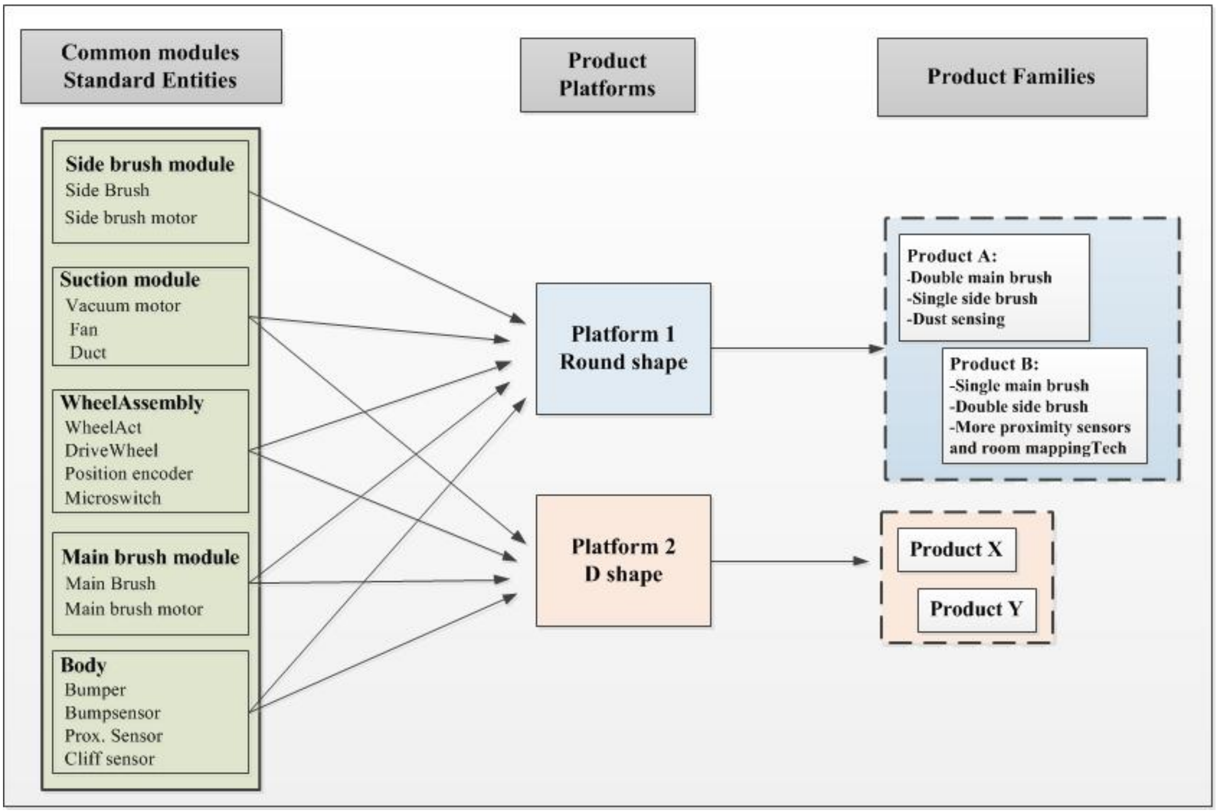


Fig.20. Product platforms based on common modules of the three systems architectures.

Table. 7. Product families (based on two platforms) can be developed using the differentiation in the product architectures (Refer to table 4 and 5).

Differentiation in entities in the three architectures	Platform 1		Platform 2	
	Product A	Product B	Product X	Product Y
Two counter-rotating main brushes	■			
Single rotating main brush		■	■	■
Single side brush	■			
Two side brushes		■		
Dust sensor	■			
Debris sensor	■	■		
Virtual wall sensor		■		
Single proximity sensor	■		■	■
Three proximity sensors		■		
IR receiver	■	■		
Camera		■		
Single magnetic sensor			■	
Two magnetic sensors				■
Room positioning system			■	■

In the VC example, only two derivative products are shown for each platform (Fig. 20). However, it is possible to have more derivative products based on:

- Number and type of sensors, for instance the number of cliff sensors can be varied,
- Number of actuators; depend on the requirement and type of platform, as in D- platform the number of actuators are less than round platform
- Type of vacuum (i.e. direct or bypass)
- Way of lifting dust and debris.

In short, one of the basic requirements for the development of product platforms is using a maximum number of standard components, which (along with minimal architecture changes) allows the development of differentiated products. The objective is to find components that can be re-used between products in a way that provides flexibility to respond to various market needs.

5.1.2. Design Optimization and Advantages of the Proposed Method

From a design perspective, the shape of the robot is important for two reasons: (1) to be able to reach corners and (2) to be able to escape when stuck in a narrow area. From the design perspective, a square-shaped robot can manoeuvre and clean in corners, but the drawback is that the robot is unable to escape from narrow areas. A circular-shaped robot can move in narrow areas, but for it to clean corners and near walls, it requires side brushes. For this reason, side brushes are used in System A and System C. The designer can either increase the diameter of the side brush or extend the side brush outwards from the device to promote corner cleaning. These design aspects can be implemented through system decomposition using function-behaviour-structure modelling, as demonstrated in the VC architectures.

In the modular structure of the VC robot, the selection and location of sensors are important considerations for enhancing performance. In the wheel assembly, all building blocks and their design parameters are the same; the only difference is the type of sensor used. For example, to sense wheel rotation, System A uses the optical principle, System B uses the Hall Effect and System C uses the Coriolis Effect as physical phenomena. The decision about sensor selection can be based on the merits and cost of each type of sensor and the opinion of domain experts. For autonomous cleaning, the robot must be able to sense dust and debris and confine itself to clean that area first, before moving to other places. In this situation, the decision about sensor location is influenced by how effectively the system responds to the amount of dust. One possibility is to respond directly to the amount of dust (according to the air passage) by fixing sensors behind the main brush. For achieving better performance, dust and debris sensors must be a part of the system, as is the case of architecture A.

The proposed method illustrates how the system architect uses FBS modelling to identify the parameters of entities, i.e. building blocks, to develop product architecture. Multiple system architectures were developed in SA-CAD to explore the possibility of developing product platforms and modelling product families that can be used for next generation products. Commonality instances in the three architectures (on the basis of the common modules or entities) were identified to develop product platforms, because these platforms may offer benefits when applied successfully. These benefits are that companies can develop differentiated products by sharing components across a product platform, reduce development time and system complexity and acquire the ability to upgrade and redesign products. Furthermore, the process of system modelling enables system designers to establish appropriate component-level specifications to communicate with domain experts. As a result, system modelling provides a solution to the challenge of cooperation and communication among design engineers in different domains.

In the design of mechatronic systems, the building blocks in mechanical design can be machine elements and machine components; those in control design are block diagrams (that represent sensors, actuators and controllers); finally, those in software design can be functions and subroutines. From the perspective of control design, building blocks of sensors and actuators are developed and the designer must identify the measured and control parameters. In the VC robot example, a position encoder, which is based on Hall Effect phenomenon and whose qualitative parameters is voltage, magnetic field and current, is linked to the drive wheel to measure its angular velocity. These measured signals are used to control the actuator power for speed control. The designer uses the conceptual relations in the FBS models and develops a knowledge base in SA-CAD to perform network modelling. In this case study, the building blocks are based only on qualitative relations; however, quantitative relations based on differential equations can also be developed in SA-CAD. The designer may use external tools to analyse these relationships. For example, the information flow in a VC robot can be analysed by examining the relations between the parameters of the sensors and actuators (i.e. block diagrams in control design); for structure analysis and dynamics analysis, CAD tools and the finite element method, respectively, may be employed.

The observed advantages of the proposed method are:

- Transforming customer requirements into system level specifications.
- Management of complexity through the hierarchical decomposition of the system, and the development of modules and their interfaces using a computational tool.
- Consistency of system descriptions at different levels of the hierarchy; for example, the decisions at the lower level should be traced to higher level abstract concepts, such as functions.
- Capability to redesign and develop multiple system architectures to facilitate product platforms and product family modelling.

5.1.3. Conclusion

In the early stage of mechatronics systems design, designers and engineers have the crucial task of designing and optimizing modules and their interfaces by analyzing the system architecture of existing products of the same type. Computational support is crucial for the effective execution of this task. The proposed method is used for supporting the task using FBS modelling and SA-CAD. The proposed method has been validated and its applicability was demonstrated using a case study, in which the method was applied to a comparative analysis of architectures of three autonomous VC robots.

This method presented a comparative analysis of design concepts of mechatronic systems as performed with the SA-CAD tool, which supports system decomposition and modularization considering the dependencies among the parameters of subsystems. Multiple system architectures are developed to identify product platforms and model product families for next generation products. The commonalities of the three considered systems are analyzed in the parameter network to determine common building blocks. The proposed method identified several levels of classifications regarding the level of commonality observed in existing systems.

5.2. Multidisciplinary Product Decomposition and Structure Analysis

The design structure matrix (DSM) is a network modelling tool to represent the components of a system and their interactions, therefore highlighting the systems architecture. The complexity in multi domain products requires decomposing them into subsystem and components, to guide the design requirements and to identify the solution space for functional improvements. This study implement component based design structure matrix in order to address the issues of multi-domain system decomposition, interface management and to assess the degree of modularity in multi-domain systems.

In order to model the component based matrix to represent the architecture of a system for modelling and analysis, the overall approach (Fig. 21) is represented in the following steps.

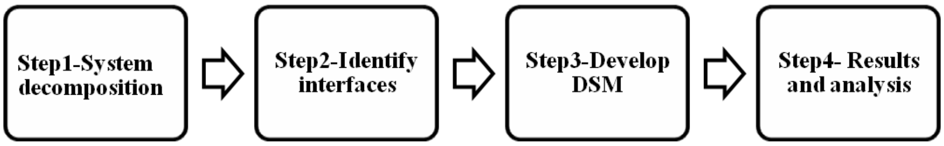


Fig. 21. Overall approach using DSM modelling for system decomposition and structure analysis.

<u>Step 1. System Decomposition</u>

At first the system is decomposed into subsystems and further into components at several hierarchical levels (Fig. 22).

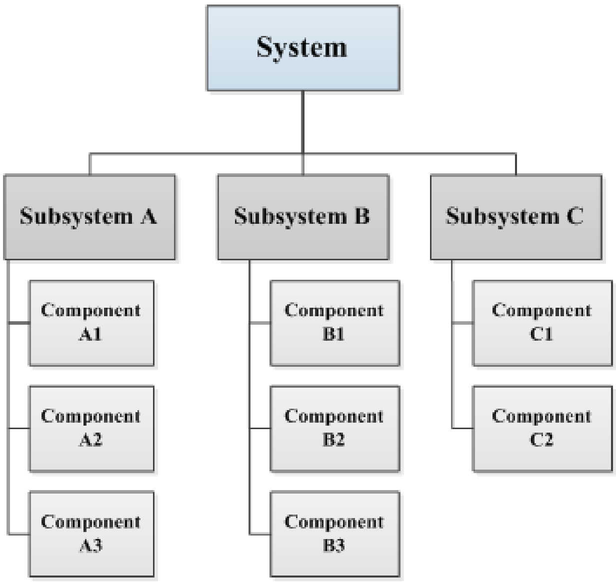


Fig. 22. System decomposition into subsystems and further into components.

In this study, a Laser Jet printer is used as an example for system decomposition and structure analysis. For this example, data about interfaces and physical structure are collected from product manuals, product videos and physical observation of the product. The decomposition of the printer system results into eight subsystems and thirty eight functional components. Subsystems are formatter system, control system, LASER/Scanner system, image formation system, paper feed system, paper delivery system, fuser system and printer driving system. Further decomposition of the subsystems into components is given in the appendix.

Step 2. Identify Interfaces

Second, the interfaces among the subsystems are identified. The following types of interfaces are identified in the printer system,

- Physical connections
- Material (e.g. toner, paper)
- Energy flows (e.g. mechanical rotary, electrical, thermal, chemical etc.)
- Information flows (e.g., image data, sensor signals, and actuator commands)

These interfaces are further explained in the following table.

Table. 8. Interfaces identified in the printer system. The reasons for using the interfaces between the functional components are also described.

Interface type	Description in relation to the printer system
Physical (P)	Physical or spatial interfaces indicate that physical adjacency is needed for alignment, orientation, serviceability, assembly and weight. Referring to the printer example, scanning mirror and focusing lens are in physical contact with scanning motor, when scanning mirror is rotated by motor, LASER beam reflects off the mirror, through a set of focusing lenses that is directed on photosensitive drum. A spatial connection between scanning motor and mirror is established in order to reflect beam on photosensitive drum. The alignment and orientation of the drum and charging roller is a necessary feature to create a uniform negative potential on the drum surface that is necessary for the image development and its subsequent transfer to paper. Thus a physical interface between charging roller and photosensitive drum is identified.
Material interface (M)	Material interfaces indicate a functional requirement related to transferring mass flows such as toner and paper. In the printer example, the developing cylinder must be able to attract toner and the toner must obtain negative surface charge as the developing cylinder is connected to power supply. Thus developing cylinder depends on power supply to be able to attract toner, while the toner must be attracted by this process. This results in a symmetrical dependency.
Energy flow (E)	Energy flow indicates a functional requirement related to transferring mechanical energy, heat energy, vibration energy, electrical energy and noise etc. In printer example, for instance, the variation in the print density depend on the DC bias given to the developing cylinder, that causes more or less toner to be attracted to the developing cylinder, hence developing cylinder and power supply is related by (electrical) energy . Similarly, motor and drive assembly are related by power transmission due to mechanical energy. Heat transfer from heater to cooling fan is kind of (thermal) energy interface. Although energy interfaces such as chemical, vibrations are also present in this kind of systems, however they are not considered in this example.
Information flow (I)	Information interface indicates a functional requirement related to transferring sensor signals or controls, image data and actuator commands. For instance information about LASER beam is send to central processing unit (CPU) by the beam detect (BD) sensor; these two are related by information interface. Similarly, information about image formation is transferred from control panel to formatter CPU is highlighted by information relation. When the power switch of the printer is turned on and the printer enters in the standby mode, the CPU outputs the signals to drive the loads such as Laser Diode, motors, and solenoids, based on the print commands and the image data input from the external device. CPU and the loads are connected by information (image data and actuator commands) dependency.

Step 3. Develop Design Structure Matrix

In this step, the design structure matrix is formed using CAM tool (Cambridge advanced modeller). All the elements of the system are placed along rows and columns in a matrix display format. A square matrix representing the elements in a system (the shaded cells along the diagonal) and their interactions (the off-diagonal marks) is formed. There are two possibilities to read the matrix i.e. across an element's row to see its inputs and down its columns to see its outputs although the opposite convention. The transpose of the matrix, can also be used. For instance element D receiving inputs from elements B and C and providing an output to element B, is illustrated in Fig.23.

	A	B	C	D
A			X	
B				X
C	X			
D		X	X	

Fig. 23. Example of a matrix representing the elements of a system in rows and columns. In this matrix four elements of a system and their relationships are given.

To model product architecture, the DSM elements are product components and their interactions are the interfaces between the components. In structure analysis, DSM elements are called as nodes and their interactions as edges of a system. Using the CAM tool the composite DSM (comprising multiple interfaces) of printer model is presented in Fig.24.

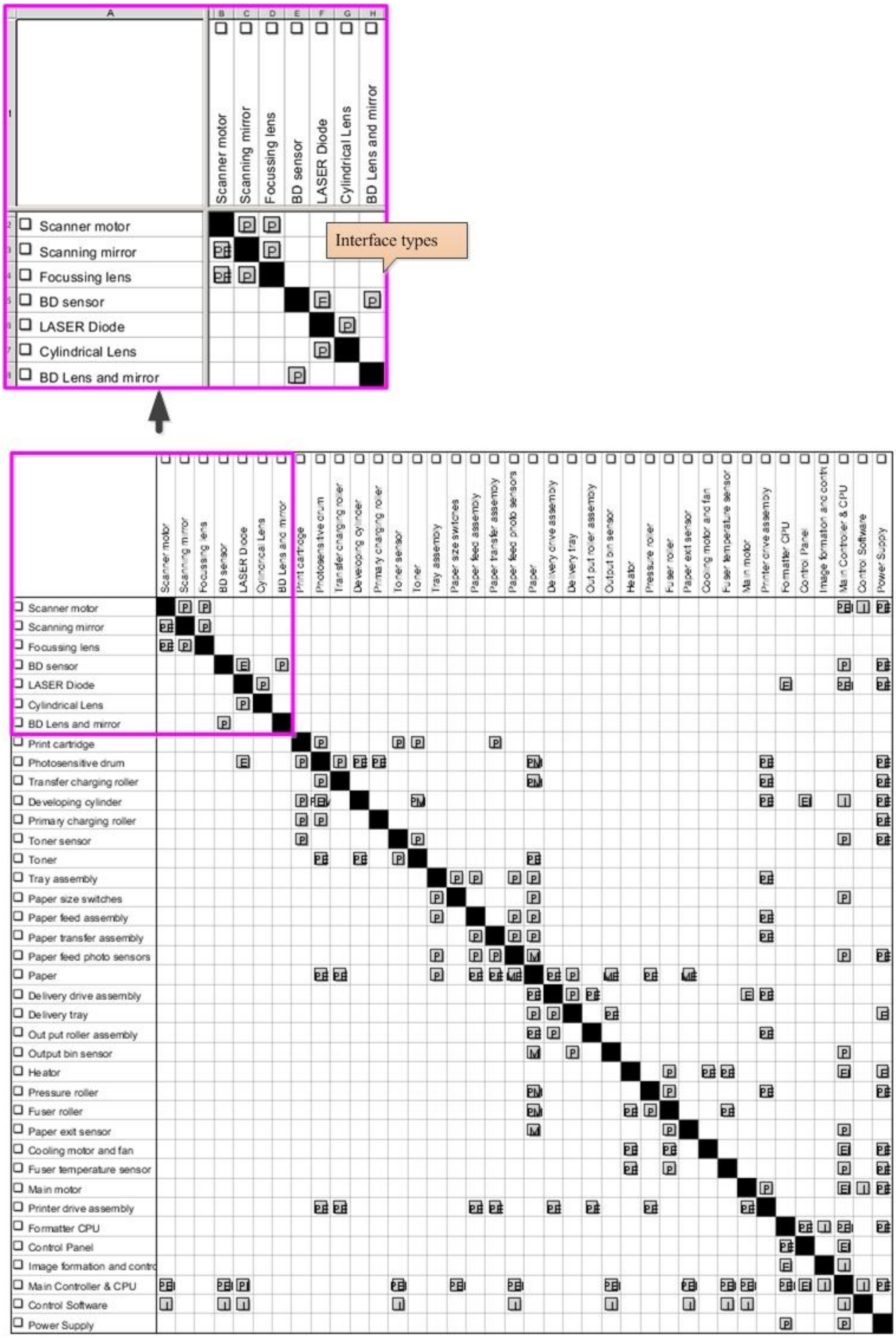


Fig.24. The composite DSM (comprising multiple interfaces) of printer model representing elements and their relationships. Elements edges close to diagonal can be clustered into modules. Interfaces of control and information elements are spread in the matrix.

Step 4. Results and Analysis

Once the DSM model is developed, the analysis of the system can be performed with the following sub steps: partitioning, clustering and structure analysis.

Step 4.1. Partitioning of Composite Design Structure Matrix

In general, partitioning can provide information about the existence of feedback loops and can determine the strongly connected parts implied in a product structure. Groups of nodes can also be identified that are suitable for modular design. However, partitioning is not able to provide information about feedback loops in specific nodes.

Referring to printer example, the result after partitioning is (Fig 25) that the interfaces of the main controller, power supply and printer drive assembly are more distributed than the other elements.

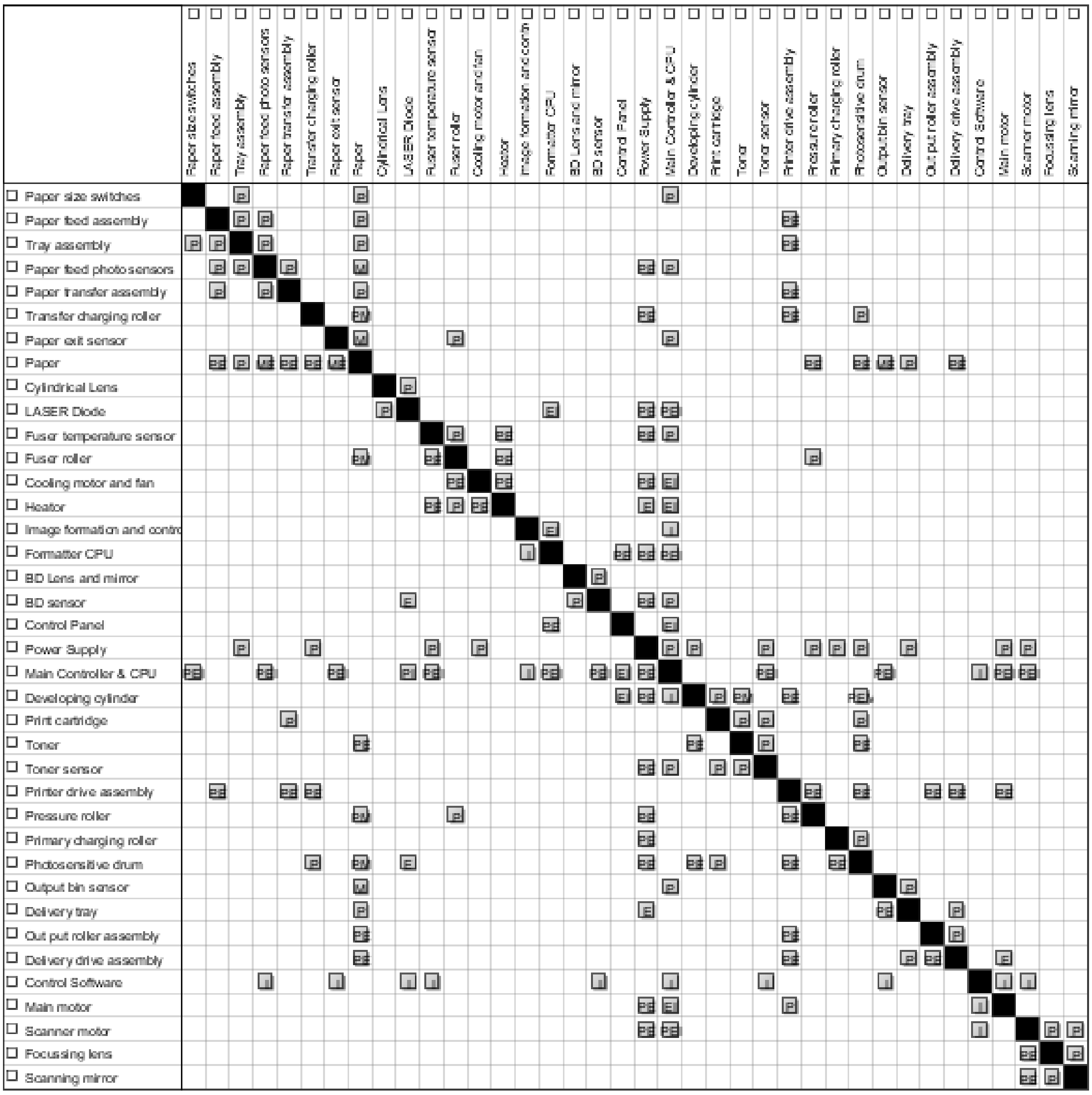


Fig.25. Partitioning of the composite DSM (comprising multiple interfaces), to bring the elements as close to diagonal that can be used as possible module candidates.

The remaining elements are relatively close to the diagonal that can be considered as possible module candidates. The reason for not accumulating near the diagonal is due to many interfaces shared by some of the elements in a system.

Step 4.2. Clustering of Physical and Composite Design Structure Matrix

The designer can draw useful insights from the DSM architecture after clustering the elements in DSM. Clustering is a set of elements grouped together because of certain relationships, also can be defined as module or a subsystem.

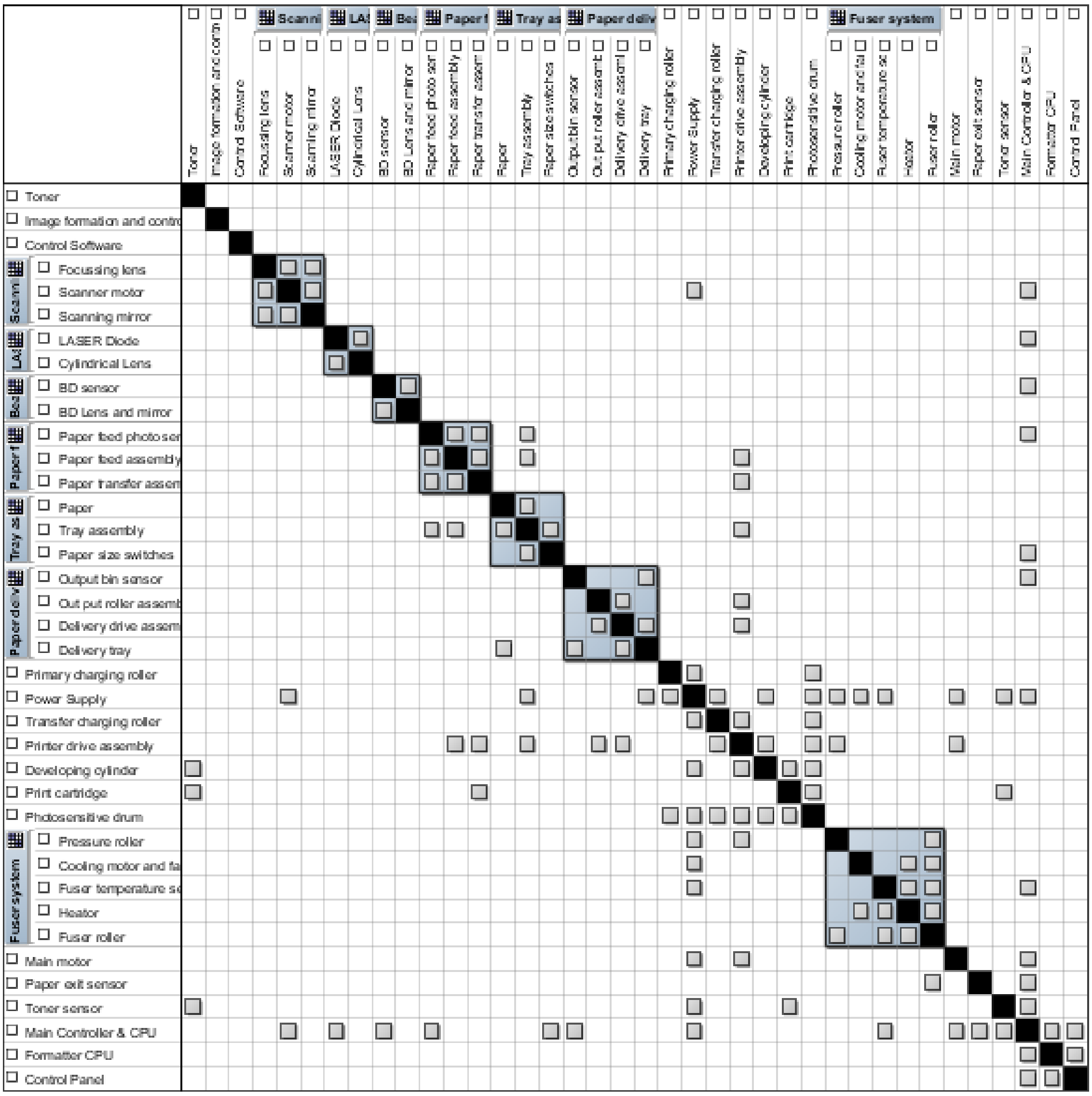


Fig. 26. Clustering of the physical DSM (formed by only physical interfaces) to identify modules. Seven of the subsystems are modular.

By applying clustering algorithm to the physical DSM of the printer example, results into seven modules (in physical DSM only physical interfaces are used) as shown in Fig. 26. However, by applying the algorithm to composite DSM, the result is different (Fig 27), where only four of the subsystems are identified as somehow modular, since more interfaces among components in each subsystem are used. These modular subsystems are scanning system, paper feed system, paper delivery and fuser system. The composite DSM also show the remaining subsystems as more distributed. These subsystems are printer control system, main motor and printer drive assembly that are more functionally distributed across the printer system or in other words their structure is more integrative than modular one.

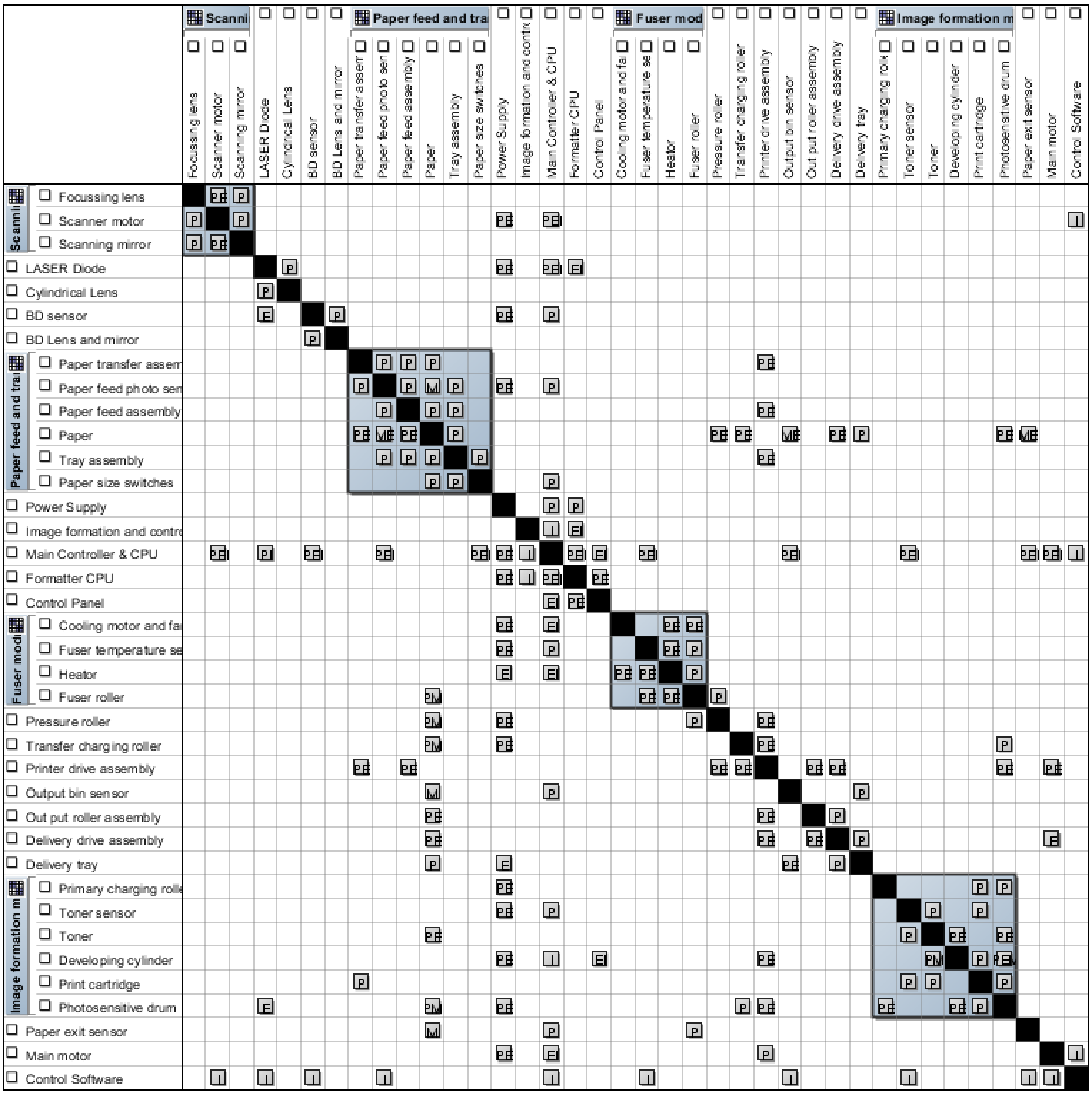


Fig. 27. Application of clustering algorithm to composite DSM to form modules. Four of the subsystems (i.e. scanning system, paper feed system, paper delivery and fuser system) are modular and the rest of the subsystems are more integrative.

Step 4.3. Structure Analysis and Results

Structure analysis is mainly relevant to the overall connectivity in elements, number of clusters formed and their relationships.

Table. 9. Structure analysis result of the printer system

Structure analysis results	
Metric	Value
Number of sets	4.0
Number of unconnected nodes	0.0
Number of edges per node	4.52
Number of edges	172
Number of nodes(discounting duplicates)	38
Number of nodes	38
Number of edges that cross a set border	60
Relational density (Non-zero fraction)	0.122
Singular-value modularity index (SMI)	0.255

The structure analysis result from the printer example is shown in Table 9. This table shows the total number of clusters formed, number of nodes, edges and their connections, non-zero fraction (NZF) as well as singular valve modularity index (SMI) in the structure. The NZF, is the fraction of non zero entries without diagonal values. The singular value modularity index evaluates the overall connection scheme between the components as described by [Katja et al, 2007].

In the composite DSM in Fig. 27, there are in total four clusters, where all the nodes are connected with some kind of interface. According to the non zero fraction, the density of the system is 0.122 or 12%. In other words, only 172 of the 1406 off-diagonal cells are occupied in the system. While complex, with as many as 38 elements and higher number of interfaces, the density is only 12%. The singular value modularity index (SMI), in printer case is closer to zero, that indicates an integral system, even though the algorithm formed some modules. One reason can be the number of edges formed by elements i.e. power supply, paper and main controller CPU are more distributed in the design structure matrix.

5.2.2 Discussions and Conclusions

Structure analysis of the system architecture, is an important aspect in modelling products that represent nodes, edges and modules. A high degree of connectivity in a structure can make the system analysis difficult. The quantity of feedback loops may increase drastically as connectivity of elements in a structure becomes higher. This indicates integral products architectures and results in more connected product that require more efforts to redesign. From the singular valve modularity index and Ulrich definition of product architecture, the printer architecture is close to integrative, though some modules are formed after clustering. Structural optimization is useful when a fundamental system structure has to be redesigned in order to form product platforms to create variety. This supports the creation of specific system variants for product customization. It can be implied that through structure optimization using DSM methods, a product platform can be developed from a single product that can be used for customization in the product. Structure optimization involve application of various approaches such as tearing and structure Pareto analysis etc.

After applying the clustering algorithm to the printer example, it can be observed that the DSM formed by only physical interfaces is different than the composite DSM. In case of physical interface DSM, the number of interfaces is significantly less and not spread like composite DSM, hence more modules are indicated after applying the algorithm. However, in the composite DSM, more interfaces are there, as elements related to information and control is more spread in the structure as compare to those related to mechanical elements.

According to Ulrich's definition of modular architecture, there is a one to one mapping between functions and components, and modular structures must have a smaller function to component ratio than integral products. In the printer example, on average more than one interface is present with each component, it indicates that the functions performed by each component in the architecture is distributed. Though the function to component ratio is not calculated in this case, but the distribution of the interfaces is much higher, that must be reduced to increase the degree of modularity in the architecture.

The attainability in product architecture is measured to assess the change in any node and its impact on other nodes. Attainability of a node is the ratio of the number of nodes it reaches to the maximum number of nodes it could reach theoretically. In the printer example, it is observed that the attainability in the nodes such as CPU control, printer drive assembly and power supply is relatively more than the remaining nodes of the system. A high valve of attainability means that many other nodes can be affected by any change in the node that is considered for any change. In general, the attainability in the nodes must be reduced in order to be able to make structure changes or redesign the architecture.

The development of product architecture based on design parameters and their interfaces is a useful approach for product upgrading and mass customization. For instance, for better performance increasing the copying speed of the printer can be achieved by changing design parameters such as speed of a motor or its size. Once the modular structure is in place along with relevant interfaces, the designer can decide either to replace the component (with a high speed motor) or using a controllable component (variable resistance in this case), that also need change in the control software, for instance in multidisciplinary mechatronic products. This upgrading may not change to a large extent the physical configuration of the system. In the printer example, though main motor is not placed in any module, any change may influence the interfaces with subsystems and components such as printer drive assembly, main CPU and control software. Furthermore, the design parameters can be changed to create variety in the product, such as changing speed and size of the motor. These changes must have an effect on the overall performance of the product.

The optimal solution in case of mechatronic systems could be a high performance product, with a few modules that can be used for commonality and flexible design for customization. However, the *degree of modularity* in mechatronic products varies and cannot be generalized due to performance requirements (i.e. power consumption, weight, size, speed etc), product structure and market demands etc.

5.3. Mechatronic Module Development

A basic mechatronic module uses several disciplines of mechatronics (e.g. mechanics, automatic control techniques, electrical design etc.). Such a mechatronic module is composed of domain-specific components. Basic mechatronic modules represent a mechatronic sub-system at the lowest hierarchical level of a mechatronic system and are indivisible within the set of mechatronic sub-systems or this mechatronic module can itself be a mechatronic system [Hehenberger, 2008, 2009]. In a mechatronic system, the subsystems and components of the systems should not be developed independently without addressing the system integration, subsystem interaction and the intended operation of the overall system. Such an approach makes a mechatronic design more optimal than a conventional design.

In this section two examples are introduced to illustrate the system design steps of electromechanical systems (also mentioned in V-model), primarily related to 1) modelling and simulations and 2) modification and investigations of physical prototype of an actuator.

5.3.1 Modelling and Simulation Using Bond Graph Approach

The bond graph is a graphical approach to modelling and simulation of multi-domain dynamic systems, in which component energy ports are connected by bonds that specify the transfer of energy between system components. Power, the rate of energy transport between components, is the universal currency of physical system [Gawthrop, 2007]. In bond graph, the physical system is built with power bonds which represent the power distribution amongst the individual elements, while the control part follows signal flows. This approach is suitable to present mechatronic system in a single model. Moreover, the bond graph model can be transformed into a mathematical model, based on differential equations that represent the behaviour of dynamic system. This modelling scheme is used to support:

- Integration of multi-domain systems
- Flow of energy and interaction of the functional elements
- Facilitate behavior analysis in a software tool
- Automatic code generation for controller design

In Bond Graph (BG) , the energy transfer between the components of the system is denoted by half arrow, representing the direction of the power. Power consists of two variables i.e effort *e* and flow *f*.

P(t)= e(t) . f(t)

There are two other variables called energy variables describing a dynamic system i,e *momentum p* and the *displacement q*, are the time integrals of effort and flow, respectively.

$$p(t) = \int e(t) \, . \, dt$$

$$q(t) = \int f(t) \, . \, dt$$

Then the energy flow can be described as

$$E(t) = \int p(t) \, . \, dq = \int q(t) \, . \, dp$$

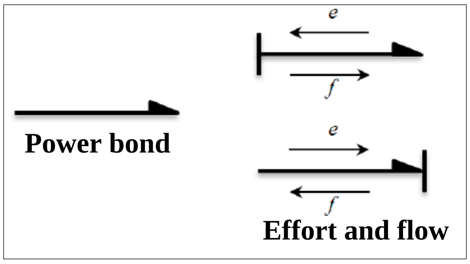


Fig.28. Basic equations, sign convention and notation of effort and flow in Bond Graph [adapted from Behbahani, 2007]

One of the key aspects in the process of creating a physical device from scratch is the creation of computer based model. Multi-domain systems (combined mechanical, electrical, pneumatic, hydraulic and thermal systems) can be modelled using a common notation, which is especially important in the design of mechatronic systems. This modelling is crucial in mechatronic design, to represent the process (i.e. flow of energy in physical part) and designing a controller (the information part). Multi-domain systems such as internal combustion engine involving thermo-mechanical and hydro mechanical models can be developed with this approach. Similarly, many electrical and electromechanical systems contain magnetic circuits and devices, such as motor design, solenoid and transformer, can be modelled with bond graph approach.

A specific example (example 1) is related to modelling and simulation of dish antenna system using the bond graph (BG) approach. Basic elements in the antenna drive system consist of a DC motor, gears and pedestal. The functional components of the drive system are connected by bonds, along with the effort and flow variables. For instance, in case of motor and shaft element the effort is torque and flow variable is angular velocity.

Causality is the most significant concept embedded in the Bond Graph approach. Causality is used to define which energy variables are input variables and which are output variables with respect to system elements. It is represented by causal stroke, placed perpendicular to the bond at one of its ends. The causal stroke indicates the direction of the effort and flow. The direction of the causal stroke is independent of the power direction as shown in Fig. 28.

In the dish antenna example (Fig 29-2), the causality analysis are performed to identify the input and output variables and to facilitate the development of state space equations in the following steps:

- Source “Se” specifies the effort in bond one, in 1-junction one bond specifies the flow, and that is bond 2. The causal implications apply to bond 3 and 4.
- At the next 1-junction, bond 6 is flow, the causality implications are, all other bonds 5, 7, 8 are efforts.
- At the 0-junction, only one bond specifies effort, bond 9 is effort and bond 10 becomes flow.
- At the last 1-junction, bond 12 is flow; bond 11 and 13 becomes efforts.
- This procedure also fulfils the requirements of integral causality to storage elements I (induction) and C (shaft stiffness).

Using the antenna example, the whole process of modelling is summarized in a systematic way from design integration to simulation of the system. With these steps, development process is simplified and structured, integration of multi-domain systems and automatic code generation for controller design is possible.

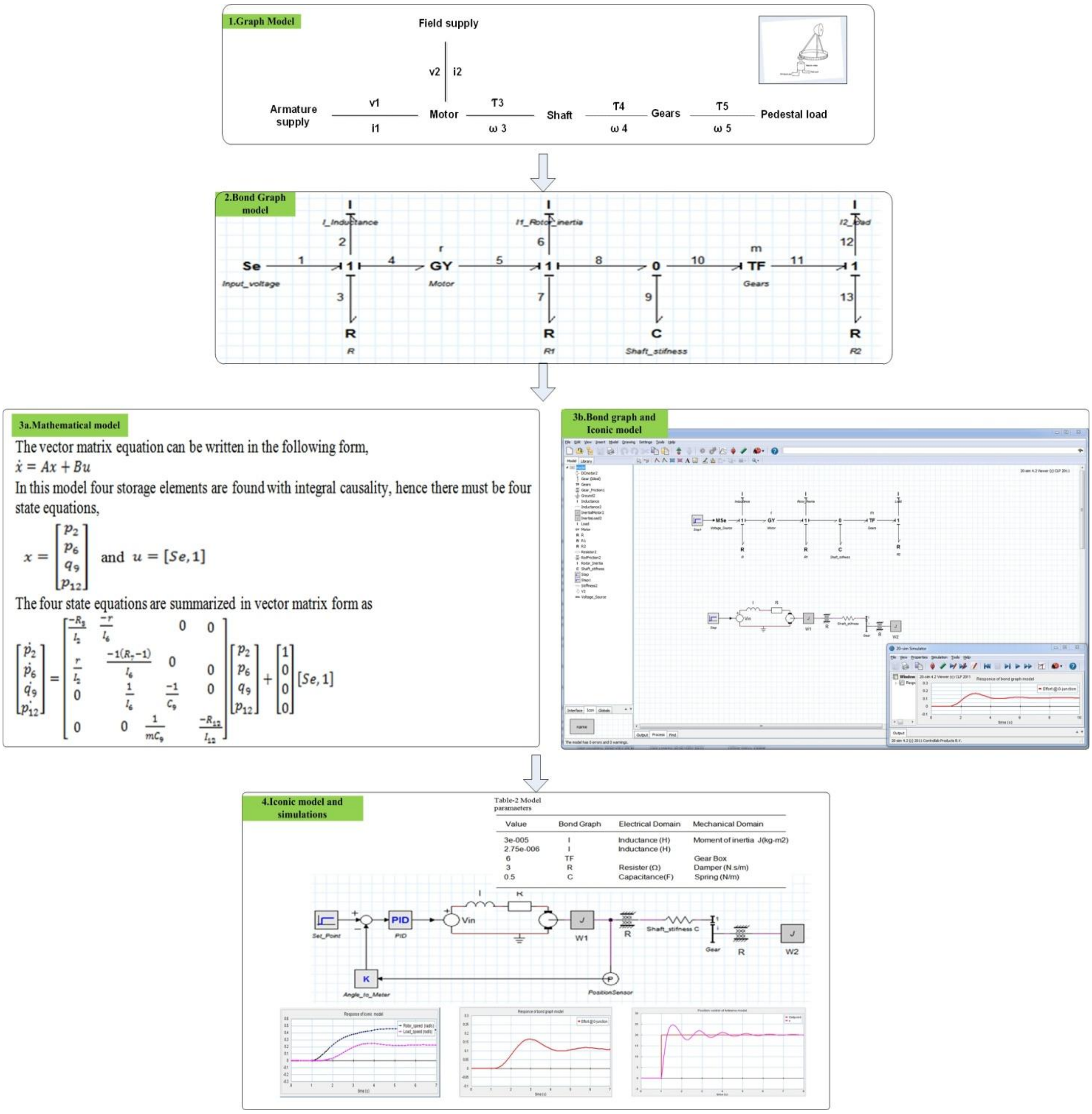


Fig.29. Modelling and simulation steps of a dish antenna system using Bond Graph approach.

From the Bond Graph model, automatic code generation for controller design can be used for customization purposes. After the desired response through (BG models and C code generation) it is also possible to develop a number of controllers with varying parameters of the individual elements representing the physical system. These controllers are basically software that can be used for upgrading and customization in products or for delayed differentiation at customer site.

5.3.2 Modelling and Investigation of Physical Prototype of an Actuator

This study is about investigation of electromechanical actuator by changing the design parameters at simulated pressure conditions for single cylinder engines. The design parameters are optimized to get the desired results. Apart from design, changes to the standard actuator the effects of spring rate, armature lift and exhaust gas forces on valve are discussed. The actuator is simulated in a lab and got experimental results with the support of control desk environment and Matlab.

The fixed valve motion by camshaft engines compromises the fuel economy, combustion stability and maximum torque performance at different loads. The conventional camshaft is replaced by electromechanical actuator in order to improve the performance of a combustion engine with a flexible scheme in valve timing at all engine operating conditions. After multi-valve technology became standard in engine design, Variable Valve timing becomes the next step to improve engine output. With electromechanical valve train (EMVT) systems valve timings are completely independent from crankshaft position and with flexible valve timings, cylinder air charge and residual gas can be optimized. By controlling the intake valve events the throttled operation is eliminated in the gasoline engine and by doing so reduce the pumping loss which results high fuel efficiency [Rokni, 2010; Seethaler, 2009].

In this example (example 2) main part of the system is an electromechanical actuator, which operates as a free oscillation system with electromagnets holding the valves in both final positions. The actuator consists of lower magnetic coil for opening the valve and an upper magnet for closing the valve. Actuator and valve spring push on armature and valve stem through spring retainers. At mid position the armature is centred between lower and upper magnets (Fig. 30).

At start, a voltage is applied to one of the electromagnets to move the armature from middle position to the fully open position. A holding current is then maintained to hold the armature in place against the spring force. The mechanical spring force and the magnetic force determine the actuator and valve operation. At valve closing, the armature moves to the upper magnet and a holding current is applied to hold the armature at closing magnet against the actuator spring force.

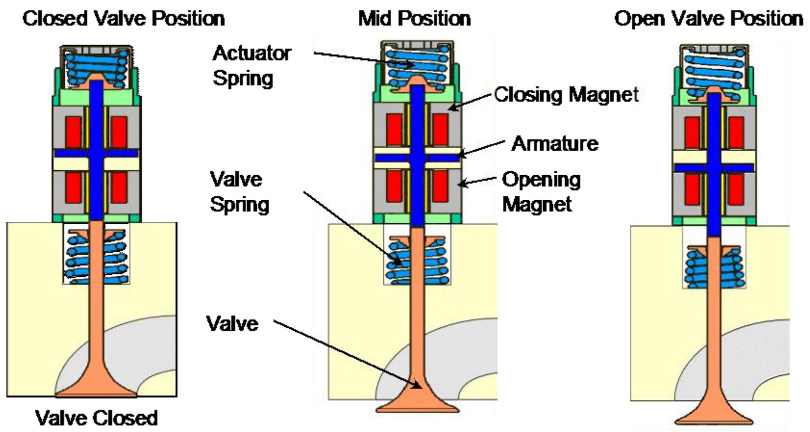


Fig. 30. Valve position by the action of magnetic and spring forces [FEV]

A bench top experiment set up has been developed. The d-SPACE (experiment) software provides all the functions to control and monitor the experiments. Control boards are used for the investigations, they keep the computational power and the input-output (I/O) channels are connected to the actuator to facilitate the data acquisition. Two Pulse width modulators (PWM) drivers, which are controlled by the d-SPACE software to regulate the power supply to the two magnets. The set up has sensors to measure the chamber pressure, the armature velocity/position, and the current of the magnets. Inductive sensor is used to measure the valve lift and installed on top of the actuator. A piezoresistive sensor is used to measure the pressure of the exhaust gases, installed in the combustion chamber. Disassembled view of an actuator is shown in Fig 31.

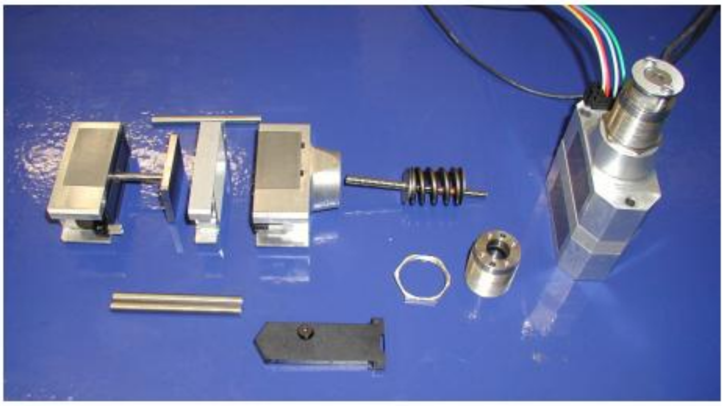


Fig. 31. Disassembled view of an actuator parts.

At a reduced Lift and at high speed, stiff springs needed to open the valve against the gas forces due to which some design changes implemented to the standard actuator. Springs of reduced height is used so additional valve and actuator spring spacers were designed to accommodate the springs in their respective positions. Another design change was to reduce the volume of the test rig (combustion chamber) compatible to two stroke engine.

The standard actuator for passenger car applications is able to work up to 6000 rpm the target for the scaled down actuator is set as 8000 rpm. Therefore, in the scaled down actuator higher spring rates (i.e. 75 N/mm to 120 N/mm) are used by reducing the armature lift from 8mm to 6.5 mm.

Table. 10. Boundary conditions of the scaled down actuator in comparison to the standard actuator. With reduced armature lift and using high spring rates the transition time is reduced to achieve a high speed.

	Boundary conditions	**Standard actuator**	**Scaled down actuator**
1	Oscillation time intake/exhaust	6.12 ms	4.8 ms
2	Speed	6000 rpm	8000 rpm
3	Transition time	2.9 ms	2.3 ms
4	Cylinder pressure at exhaust valve opening	0-7 bar	0-7 bar
5	Cylindrical spring with average spring constant	2x75 N/mm	2x120 N/mm
6	Valve lift	8 mm	6.5 mm
7	Operating voltage	42-55 V	42-55 V

Experimental results at valve opening and closing are carried through a test rig in the same way as real engine with the following parameters,
The real engine parameters are,

- Combustion chamber volume : 500 cm^2
- Two exhaust valves at each cylinder
- Exhaust gas temperature is around 950 C°.

The test rig situation is:

- Combustion chamber volume: to be calculated
- One exhaust valve on cylinder
- Gas temperature at about ambient (293 C°)

The compression volume in the test rig is calculated by the following formula,

$T_{engine} = 1223$ K, $T_{compression} = 293$ K

With $a = \sqrt{KRT}$, that results into

$$\frac{a_{engine}}{a_{compression}} = \sqrt{\frac{T_{engine}}{T_{compression}}} = \sqrt{\frac{1223}{293}} = 2.043$$

$$\frac{V_{engine}}{V_{compression}} = \frac{2.A_{valve}a_{engine}}{A_{valve}a_{compression}} = 4.087$$

With a cylinder of $V_{cylinder} = 200\ cm^3$ results $V_c = 200/4.807 = 50\ cm^3$

For this actuator the optimum pressure chamber volume is calculated as 50.25 cm^3.

The experimental results include swing out curves, velocity trace at valve opening and closing, lift curves, transition times and current trace at a pressure from 1 to7 bar at 100 rpm. These results are obtained through oscilloscope and the data is processed in Matlab.

Transition times for valve opening increases as the pressure on the valve increases, while the transition times for valve closing event is almost same due to the fact that at valve closing the pressure has already disappeared (Table 11).

Table. 11. Transition times at valve opening and closing event from 1 bar to 7.5 bar pressure.

Pressure [bar]	Transition time opening [ms]	Transition time closing [ms]
1	2.5	2.1
2	2.8	2.1
3	3.1	2.1
4	3.2	2.2
5	3.6	2.2
6	3.5	2.2
7	3.8	2.2
7.5	3.7	2.2

Armature lift curve as it moves from upper magnet to lower magnet or the valve opening event. Fig.32, demonstrates that, the instant the armature starts to lift; the holding current comes to zero and the catching current starts to rise till the armature reaches its maximum lift. Catching current is more than holding current in order to overcome the friction losses.

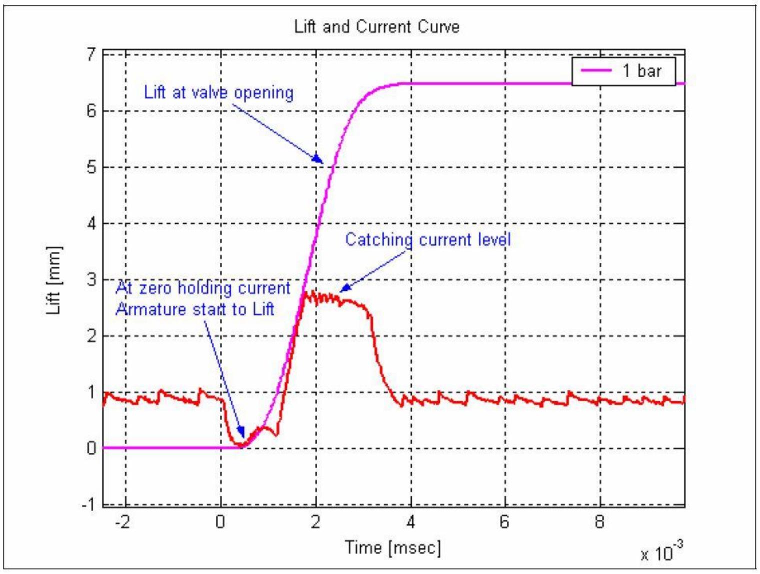


Fig.32. Starting event of the lift curve. At zero holding current armature start to lift. The total lift of the valve is 6.5 mm as desired.

More catching current is needed as the pressure on the valve increases as shown in Fig.33. At the start of the valve lift, holding current is almost same for all pressures because the upper magnet is at holding phase always working against the upper spring force and not against the pressure, the upper magnet holding current is independent to the pressure, more catching current is needed as the force on the valve increases while opening.

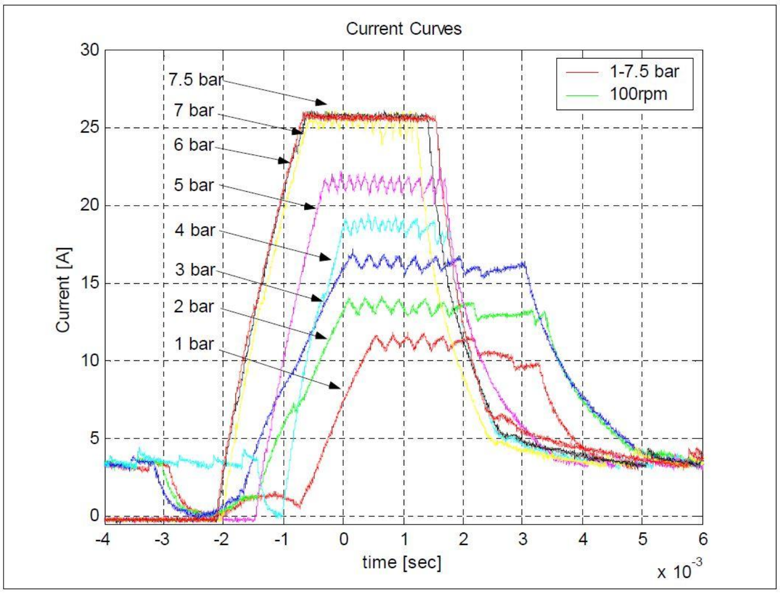


Fig. 33. Current curves at a pressure from 1.0 up to 7.5 bar.

Maximum velocity of the armature reduces as pressure increases. At 1 bar pressure the maximum velocity is 3900 mm/sec while at 7.5 bar pressure the maximum velocity reduces to nearly 2000 mm/sec as shown in Fig.34. A speed reduction of approximately 48% is observed. At the valve opening ,exhaust gases pressure present in combustion chamber exerts force on the valve, the amount of this force increases as the exhaust gas pressure increases eventually resulting in a reduction of armature velocity.

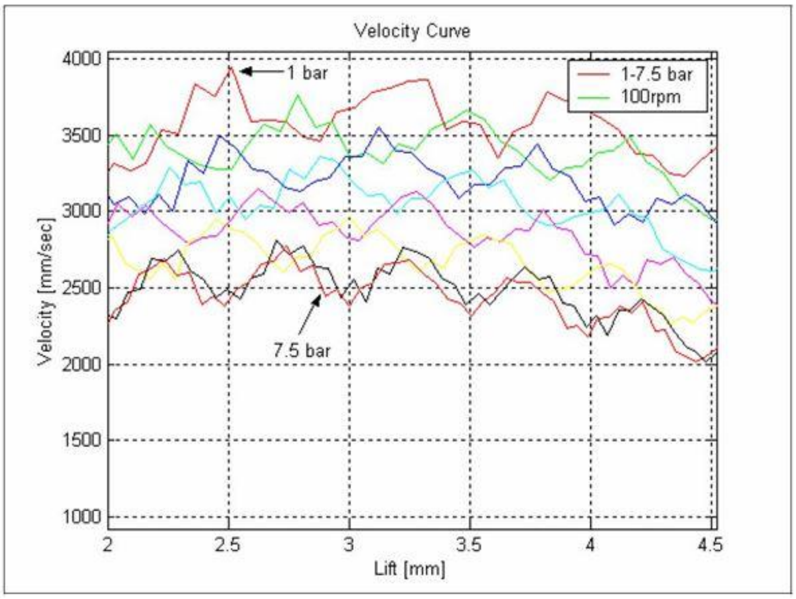


Fig.34. Velocity trace at a pressure from 1.0 up to 7.5 bar.

Higher pressure reduces the armature speed. The kinetic energy ($1/2\ mv^2$) of the system is more at the centre, due to it the system will move faster at the centre as compare to the ends, the reduction in velocity would be more at the centre as compare to the ends as pressure increases, as shown in Fig.35.

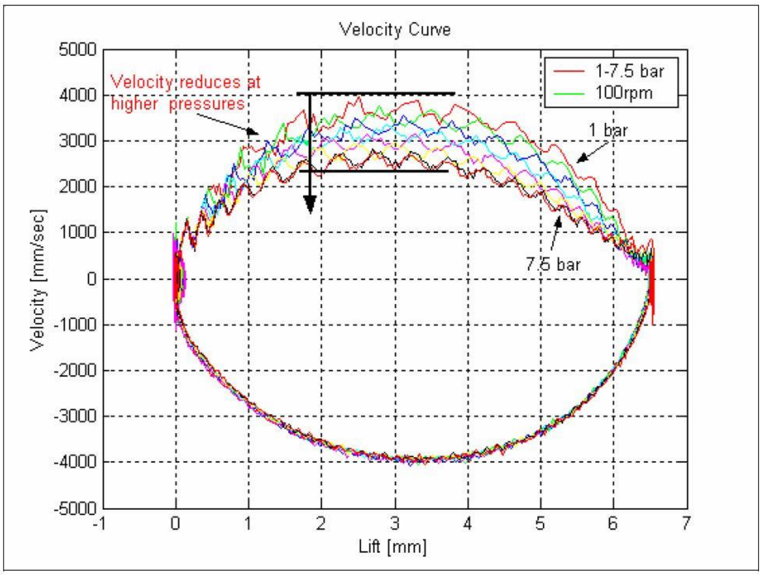


Fig.35. Velocity trace art a pressure from 1.0 to 7.5 bar. Higher kinetic energy at the centre. Visible reduction in speed is observed at valve opening event due to varying pressures. But at valve closing event in the other half of the curve, there is no variation in speeds as the pressure disappears.

Swings out curves are shown from 1 to 7.5 bar pressure in Fig.36. Higher pressures forces on the valve causes the armature to settle down (at the centre) quickly as compare to low ones, the settling time at 7.5 bar is 0.071ms while at 1 bar the settling time is 0.15 ms. Armature lift reduces at higher pressures.

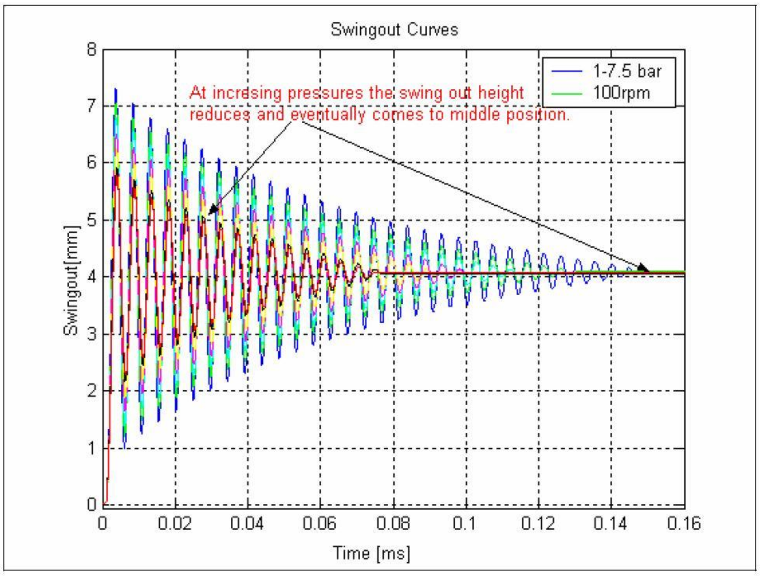


Fig.36. Swing out curves from 1to7.5 bar pressure.

The eddy current generates as the flux varies due to the change in the air gap, depend on the frequency, amplitude of the current and the permeability of the core material, at high pressures more eddy current produces and cause more energy consumption. In addition to that, energy is also lost due to hysteresis, copper losses in the magnetic core and mechanical friction etc.

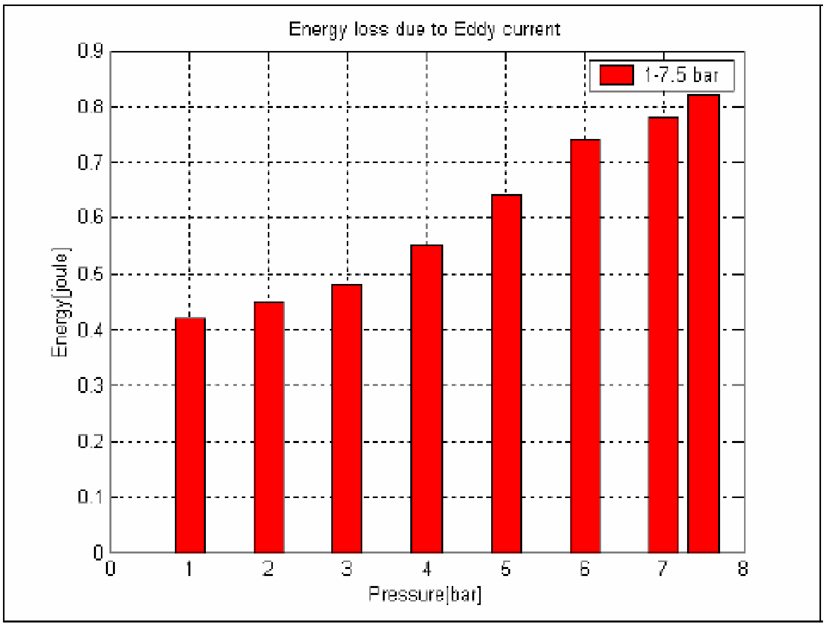


Fig.37.Energy loss due to eddy current.

5.3.3 Discussions and Conclusions

These results demonstrate that a fully variable valve train actuator is designed for two stroke engine applications. Since a stronger spring rates (240 N/mm, light springs) is used in the scaled down actuator, the oscillation time is considerably reduced and a high speed for the actuator is reached. Experimental results at valve opening and closing are carried through a test rig in the same way as real engine with a reduced chamber volume of 50 cm^3, which is able to operate up to 8000 rpm engine speed, and is investigated on a test rig having a lift of 6.5 mm.

Higher end stop forces are needed when higher spring rates are used; it means more magnetic forces will be needed against this spring rate at end positions due to which more current is needed, results in higher energy consumption. Magnetic force can be maximized by increasing the current at higher spring rates, but there is a limitation, a saturation point will reach beyond which the magnetic induction will not increase appreciably by giving more current. Furthermore, speed and friction of moving parts will increase that also results into high energy consumption.

As a result of induction, eddy currents are built up in the magnetic core called eddies, they move to flow in closed paths within the magnetic material and depend on the frequency, amplitude of the current and the permeability of the core material. It also generates as the flux varies due to the change in the air gap (armature lift). This leads to heat losses and to a delay of the built up and decrease of magnetic field. The reduction of these losses is carried out through the suitable material selection and an assembly of thin insulated sheet metal which must be oriented in a direction parallel to the flow of magnetic flux.

One issue with increasing the armature lift is that more catching current is required. The greater the amount of current applied, the stronger the magnetic field in the component. But a point is reached that an additional increase in the current will produce very little increase in the magnetic flux; the material has reached a point of saturation. Therefore scaling of the actuator must be in permissible limits to get the desired results.

By changing the boundary conditions (design parameters) for a scaled down actuator, the magnetic force is sufficient to hold the armature at end positions against the spring forces, and also able to open the valve against the gas force. Hence, a scaled down actuator is designed to reach the desired results as demonstrated from the simulation results. After the initial results, further investigations can be to control the bouncing effect, for soft landing of the actuator to reduce noise and energy losses. Moreover, further scaling of the actuator can be possible by changing the design parameters related to armature lift (with the same magnetic core and end stop forces), to observe the effect on transition times, saturation effects, changes in moving mass and valve diameters etc.

6. TESTING AND MODIFYING THE MODULARIZATION METHOD

This chapter presents the results and contributions related to testing and modification of the modularization method. An industrial case is used as an illustrative example for this purpose.

In this study, the modular function deployment method (MFD) from Ericsson [1999] is applied to an industrial mechatronic product. The objective is to test this method on a mechatronic product and to propose modifications to the original method. The testing and analysis consists of the identification of design requirements derived from customer needs, mapping of the product functions onto product structure, and, in the process, identifying solution space development to facilitate customization. This is followed by the identification and development of modules, based on reasons related to product life cycle issues such as design variety, manufacturing, quality, upgrading and recycling. Finally, evaluation of these modules is performed using interface management. However, as was stated in the discussion of methods, quantitative methods are not used to find overall costs and design for assembly (DFX). The main steps in the modular function deployment method are shown in Fig.38.

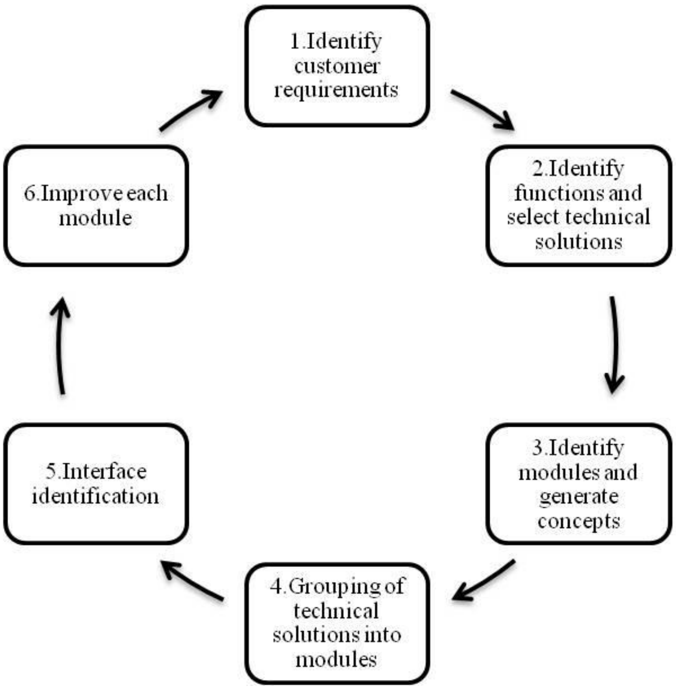


Fig.38. Steps in modular function deployment (MFD) method [adapted from Ericsson , 1999].

In order to perform the analysis, the CARMEN platform is used as an illustrative example. This platform is a prototype modular robotic test bench for assembly functions. This platform reduces the development time and increases the number of applications that can be automated using robots. The CARMEN system is a more efficient and flexible assembly automation that uses robots. The main components of the system are the handling system, control box, vision system, frame, pallets and cable kit. In this study, the modularization is performed in three phases namely decomposition, modularization and evaluation.

6.1. Decomposition Phase

In the decomposition phase, the specifications derived from user needs are transformed into product specifications by defining (or decomposing) subsystems and components. This is the conceptual design process, where the overall product function, its most important sub functions and their interaction lead to a functional structure. Subsequently, technical solutions are identified for each function, and the lower sub-functions are mapped onto technical solutions. This serves two purposes: the user requirements can be transformed into product specifications, and the solution space can be identified with multiple technical solutions for product customization.

Customer requirements can be relevant to

- Functional performance,
- Physical requirements (e.g. size, weight, and how it fits together internally and with other systems),
- Reliability,
- Life cycle concerns,
- Resource concerns and
- Manufacturing requirements.

It is also important to find product attributes that are measurable and derive from customer needs. Product attributes can be size, weight, speed, precision and accuracy, as well as the number of assembly functions and modules.

Referring to Fig.38, the product decomposition is performed in the following steps:

Step 1. Analyse Requirements and Product Attributes Using the Quality Function Deployment Matrix

Initially, the product properties are derived from the customer's requirements. Once the company-specific needs and market needs have been identified, the next step is the use of the quality function deployment (QFD) matrix to analyse the customer requirements and the product attributes. This analysis helps to determine what product properties are essential and can be used to fulfil the respective customer needs.

Referring to the CARMEN example (Table 12), for instance, requirements like flexibility in use (variety) have a higher score and an indication of future trends. Controller software can be easily developed, maintained and upgraded, which may be the reason for the high score indicating its importance for the designers. Another higher weight-age is for modularization that is related to flexibility in use and easy maintenance (due to standardized modules).

Table. 12 . Quality function deployment matrix for CARMEN platform. This matrix shows which product properties are essential and can be adjusted to fulfil the respective customer needs. The relations are graded with a point system of 9 (strong relation), 3(medium relation) and 1(weak relation).The grades are multiplied with the customer demand weight and then added vertically. Future probable trends for the customer requirements are shown with arrows.

Customer Requirements \ **Product Attributes**	**Weight**	**Trend**	**Size**	**Material**	**Controller Software**	**Degree of variation**	**Guidance range**	**Weight**	**Arm Speed**	**No. of skills assembly**	**Degree of Accuracy**	**Precision**	**Modularization**	**Feeding speed**	**Total**
Higher skills to perform assembly	**5**	↑								9					**9**
Low price	**5**	↑	1	9	3			1							**13**
High Reliability (precision/high quality)	**5**	→									1	9			**10**
High accuracy	**5**	→									9				**9**
Flexibility in use	**4**	↑	9		3	9							9		**30**
Facilitate part guidance	**4**	→					9								**9**
Facilitate component feeding	**4**	→	1						3					9	**10**
Fast arm positioning	**3**	→						3	9						**12**
Easy upgrading	**3**	→			9										**9**
Easy maintenance	**2**	→			3								9		**12**
Total			**45**	**45**	**60**	**36**	**36**	**10**	**39**	**45**	**50**	**45**	**54**	**36**	

Step 2. Identify Functions and Select Technical Solutions.

In this step, the main functions of the product are identified and the technical solutions are selected. Then the main functions are decomposed into sub-functions until no further decomposition is possible. This method is modified by introducing the concept of organs because solution space development (SSD) can be supported by identifying variants in organs. The concept of organs is taken from the theory of domains (refer to chapter 2). However, in this sub-step, only variants are introduced and the identification of interfaces is postponed until module evaluation. Finally, the technical solutions are derived from the organs by further decomposition into assembly and part domains. The most suitable technical solutions are selected with regard to customer requirements and to issues related to cost and resource constraints etc.

In the CARMEN example, the primary function of this platform is to perform product assembly. The main assembly function is decomposed into three sub-functions: the control, support and performance of assembly functions. This help the designers to see that there are three main systems in this platform. The technical solutions are derived from the organs by further decomposition into assembly and part domains. For instance, the main subsystems in the articulated manipulator are the robot arm, tool changer, mounting adapter and grippers. The most suitable gripper in this application is selected along with the provision of variants. Similarly, the technical solution in the control (function) is mapped to the main controller, robot controller, and vision system. The decomposition of this platform is represented in Function-organ-part domains as shown in Fig. 39.

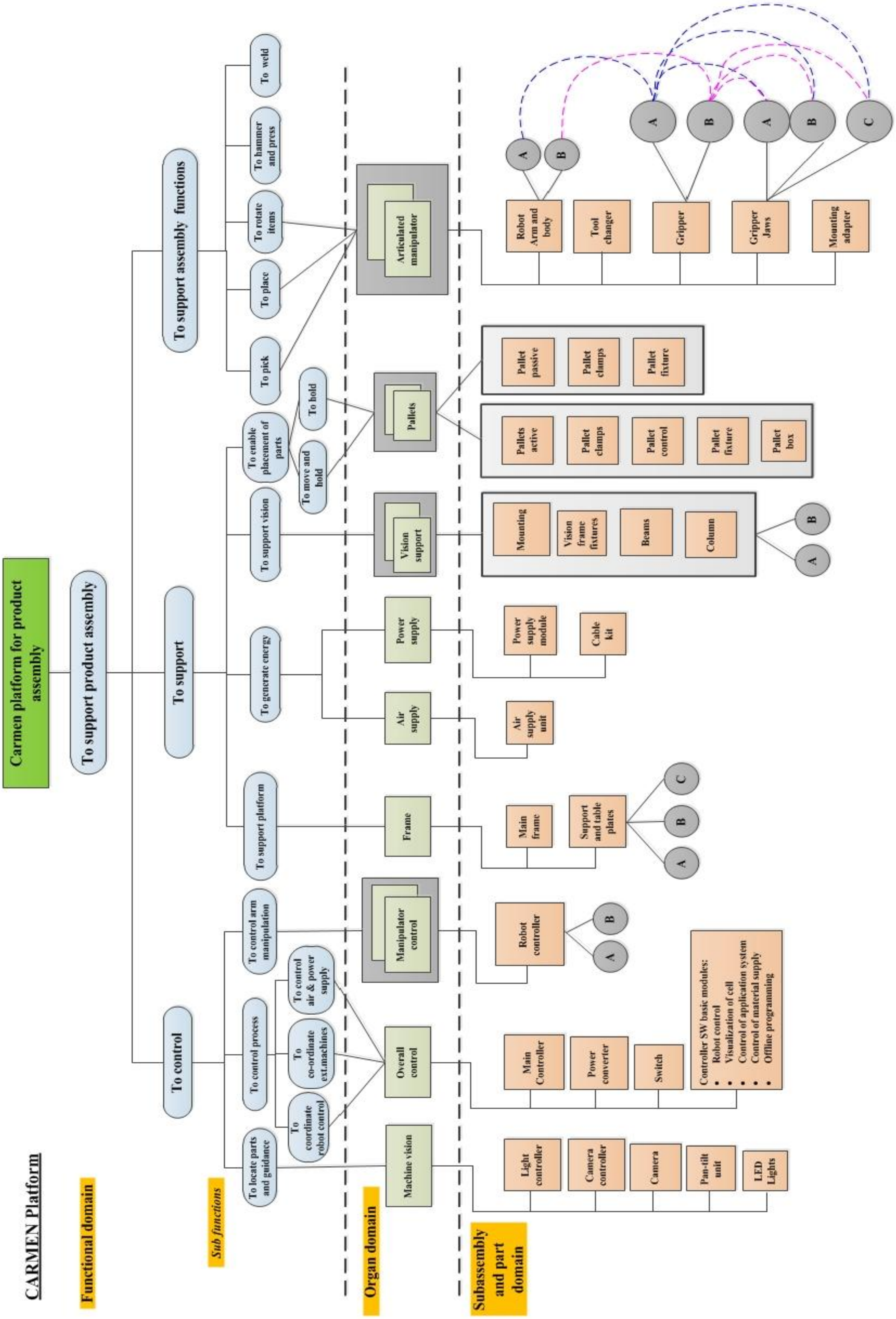


Fig. 39. Decomposition of CARMEN platform into Functions-Organs-Part domain. The three main functions are decomposed into sub functions connected to organs. Organs are connected further to subassembly/part domains.

6.2. Module Identification and the Integration of Technical Solutions

In this phase, each technical solution is analysed by the module drivers to form potential modules. This analysis is performed in a module interaction matrix (MIM). The results of the decomposition phase are essential in supporting the formation of modules when using the module drivers to evaluate the technical solutions.

The module drivers represent various reasons for modularization. The first module driver is *carryover:* a specific function will carry over to different products and no technology changes are expected. The next two, *technology push and planned product changes*, take both unexpected and expected changes into account. One is influenced by external factors such as technology evolution, while the other is internal and company related, such as decisions to develop and change the parts of the product. Technology push might enable an update of the module without upgrading the entire product. *Different specification* enables product variation, and *styling* considers how the modularity choice will affect the appearance of the product. *Common unit* involves parts that are identical in all products and used in several. *Process and/or organization* refers to the fact that parts of the product that require the same production process can be combined into a module that might improve the efficiency of the production process. The possibility of *separate testing* of each module might improve the quality due to a reduction of feedback times. *Supplier availability, service and maintenance* are related to the organizational effects of modularization. *Upgrading* allows redesign and future additions to the product. *Recycling*, the last modularity driver, considers issues related to the 'afterlife' of the product, such as sustainability and environmental concerns.

<u>Step 3. Develop a Module Interaction Matrix to Identify Modules</u>

In the module interaction matrix, each technical solution is assessed with respective module drivers on a scale (9, 3 and 1) according to the importance of its reasons for becoming a module. According to this method, highly weighted, many and unique module drivers, indicate that the technical solution under consideration is likely to form a module or the basis for module. On the other hand, the few and low weighted module drivers, indicate that the technical solution under consideration can be group together with other technical solutions. The integration of the technical solutions should be performed if there is a match in the module driver pattern or there are no contradictions.

The module driver scores in the module interaction matrix for the CARMEN platform are shown in Fig.40. It can be seen in this matrix that the main controller is a strong candidate for a module as it has many module drivers with a higher module score. The main controller gets a higher score because

- It is a common unit in the product family
- It uses the same production process
- It offers separate testing, supplier availability and the possibility of future upgrading.

On the other hand, technical solutions whose module drivers are low weighted and few can easily be grouped together with other technical solutions to form modules. For instance technical solutions such as mounting adapter and table plates can be encapsulated with other modules.

The module interaction matrix (Fig.40) shows large totals for common unit, supplier availability, carryover, process/organization and different specifications. This suggests a mature product (or subassemblies) as well as the availability of variants. Higher scores for module drivers such as *common unit* and *carryover* influence the overall cost of the product. A lower cost can be observed for this example. There is no score for styling and planned design changes because product appearance is not a customer requirement as observed in the QFD matrix. Drivers such as planned design changes are not considered for this product because it is in the initial phase of development. Strong supplier availability signifies that assemblies and parts for this product are available, which will substantially improve the quality of the final product.

The following technical solutions having the highest scores (vertical summation) in the matrix can be the prospective modules such as Camera, LED light, Main controller, Robot controller, Controller software, Main frame, Power supply unit, Air supply unit, Cable kit, Pallets, Pan-tilt-unit, Tool changer, Robot arm and body, Grippers and Grippers jaws.

Module drivers \ Technical Solutions		Camera	Camera controller	LED light	Light controller	Main controller	Robot controller	Power converter	Switch	Controller SW	Support & table plates	Main frame	Vision frame fixtures	Beams	Mounting	Column	Power supply module	Cable kit	Air supply unit	Pallet control	Pallet clamps	Pallet box	Pallets	Pallet fixture	Pan-tilt unit	Gripper jaws	Tool changer	Robot arm and body	Gripper	Mounting adapter	Total
		Vision system				Control system					Support system															Handling system					
Design	Carryover											●	●	●	●	●	◎	◎	●		●		●	●	●		●			●	114
	Technical push (external)	◎				◎	◎	◎	◎																			●			24
	Planned design changes (internal)																														0
Variance	Different specification						●				●	○			○		◎						●	●		●		●	●		68
	Styling																														0
Manufacturing	Common unit	◎		◎		●	●	●	●	●		●					●	●	●	●	●		●		●	●	●	●	●	●	168
	Process/ organization					●	●	◎	◎		●	●	●	●	●	●		◎		●	◎	◎	●	◎							108
Quality	Seperate testability	●	●	●	●	●												◎							●	●			●		75
Purchase	Supplier availability	●	●	●	●	●	●	◎	◎	●							●	◎							●	●	●	●	●	○	127
After -sales	Service/ maintenance	◎		●		●											◎		●						◎	◎	◎		○		43
	Upgrading					○	●			●																					19
	Recycling																●	◎													12
	Total	27	18	30	18	49	48	18	18	27	18	28	18	18	19	18	36	24	27	18	21	3	36	21	39	39	30	36	37	19	

● = Strong module driver (9 points) , ◎= Medium module driver (3 points), ○= Weak module driver (1 point).

Fig.40. In the module indication matrix for CARMEN platform, each technical solution from the decomposition phase is assessed against the module drivers. This platform represents the four main systems (i.e. control, vision, support and handling system) from which the technical solutions having the highest scores are considered as prospective module candidates.

Step 4.The Grouping of Technical Solutions into Modules

After module indication, the next step is combining the remaining technical solutions with module candidates.

In the handling system of the CARMEN example, the gripper and gripper jaws are suitable to form a gripper module because there is no conflict in the module drivers: they are the same. However, a possible difference is that for each gripper there may be different kinds of jaws to facilitate the grasp of different shapes. Another module can combine beams, columns, vision frame fixtures and mounting to form a vision frame module, because the most important module drivers for all of them are carryover and process /organization. Similarly, in the support system, supports and table plates are integrated into the main frame to form a base module to provide three variants to customers. The main control software is considered as a separate module in order to facilitate upgrading of the system without significant change to the main controller hardware, even though the control software has the same module driver profile as the main controller. This is also relevant to interfaces with other machines (though external to CARMEN platform) such as conveyers and feeders, and to other cells in the manufacturing system.

In this example, the grouping of technical solutions produces fourteen modules (in this platform, further decomposition of technical solutions will substantially increase the number of subsystems, assemblies and especially parts, such that the number of parts cannot be easily estimated). Hence further decomposition is not performed.

From the module profile (Fig.41), it can be observed that the most weighted module drivers are relevant to internal organization issues (carryover, common unit, supplier available and process/organization), while the customer-related drivers (different specification, styling) receive lower scores. One reason is that industrial products do not need to be stylish, whereas issues relevant to production and cost are central concerns. This gives engineers a good insight into what issues are important enough to be considered during the conceptual phase of product development.

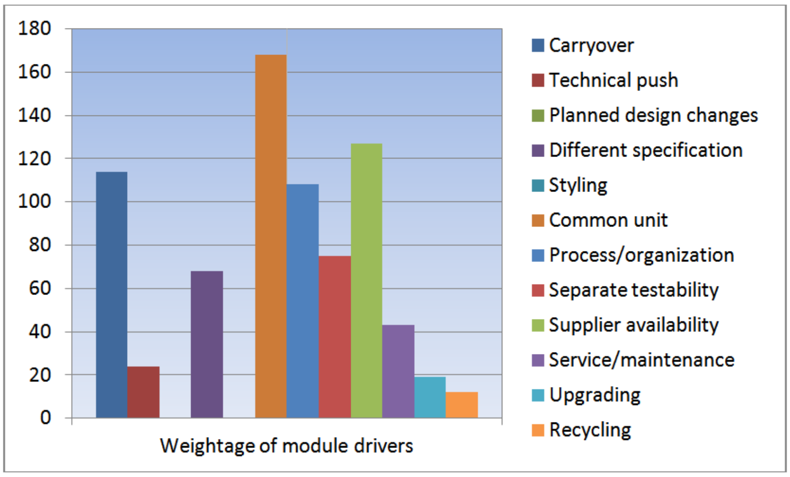


Fig.41. The module driver profile for the CARMEN platform shows large totals for common unit, supplier availability, carryover, process/organization, different specifications and separate testability. This points towards a mature product (or subassemblies) with availability of variants as well as good quality due to separate testability. However, the module profile for technology evolution and upgrading receives relatively small scores while planned design changes and styling receive no scores.

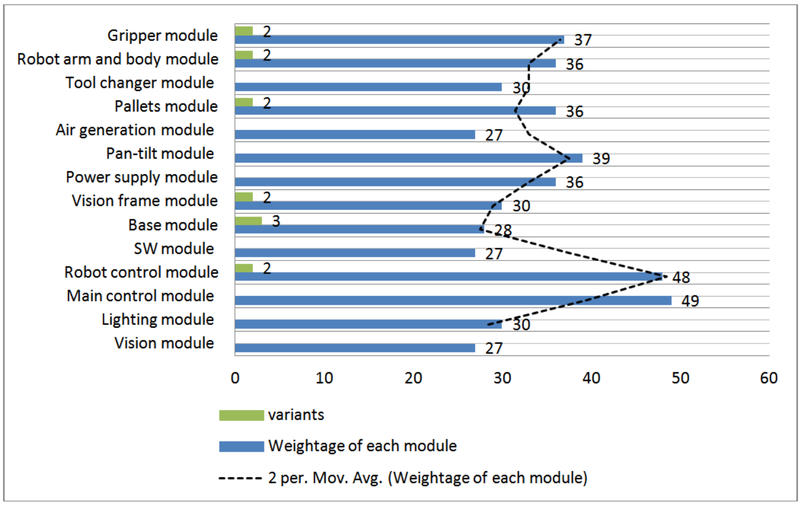


Fig.42. Fourteen modules are identified in CARMEN platform. The technical solutions having the highest scores are considered as possible modules. The number of variants is also shown.

Table. 13. The final fourteen modules chosen from the CARMEN platform that represents module candidates, strongest drivers, variants and technical solutions.

Systems (decomposition)	Modules	Strongest drivers	Number of variants (*SSD)	Technical solutions (integration)
Vision system	Vision module	Different specification Supplier availability		Camera Camera controller
	Lighting module	Different specification Service & maintenance		LED lights Lights controller
	Pan-tilt module	Carryover Separate testability		Pan-tilt unit
Control system	Main control module	Common unit Upgrading		Main controller Power converter Switch
	Robot control module	Common unit Different specification	2 variants	Robot controller Robot control SW
	SW module	Common unit Upgrading		Main controller software
Support system	Base module	Different specification Process/organization	3 variants	Main frame Supports Table plates
	Vision frame module	Carryover Process /organization	2 variants	Beams Columns Vision frame Fixtures and mounting
	Power supply module	Common unit Supplier availability		Power supply Cable kit
	Air generation module	Common unit Service & maintenance		Air supply unit (Manifold, Dryer, valve, compressor)
	Pallets module	Different specification	2 variants (active,passive)	Pallets (vision type) Pallets control & fixtures Pallet clamps
Handling system (articulated manipulator)	Robot arm and body module	Technical push Common unit Supplier availability	2 variants *Kuka* *UR*	Robot arm Mounting adapter
	Tool changer module	Carryover		Tool changer
	Gripper module	Different specification Supplier availability	2 variants (3 jaws to each gripper)	Grippers Gripper jaws

*(*SSD: solution space development)*

6.3. Evaluation of the Modules

Evaluation of the system includes interface design, cost targets, planned development and the description of variants.

Step 5. Interface Identification

After module identification, the next step is to identify interfaces between modules. In the interface diagram, product structure is identified along with modules and the interfaces between them. Hence the identification of the interfaces is the essential part of the evaluation. Various interfaces can be identified between the modules, including

- Physical connections
- Material (e.g. air transfer)
- Energy flows (e.g. mechanical rotary, electrical, thermal etc.)
- Information flows (e.g. sensor signals, actuator commands, control commands)

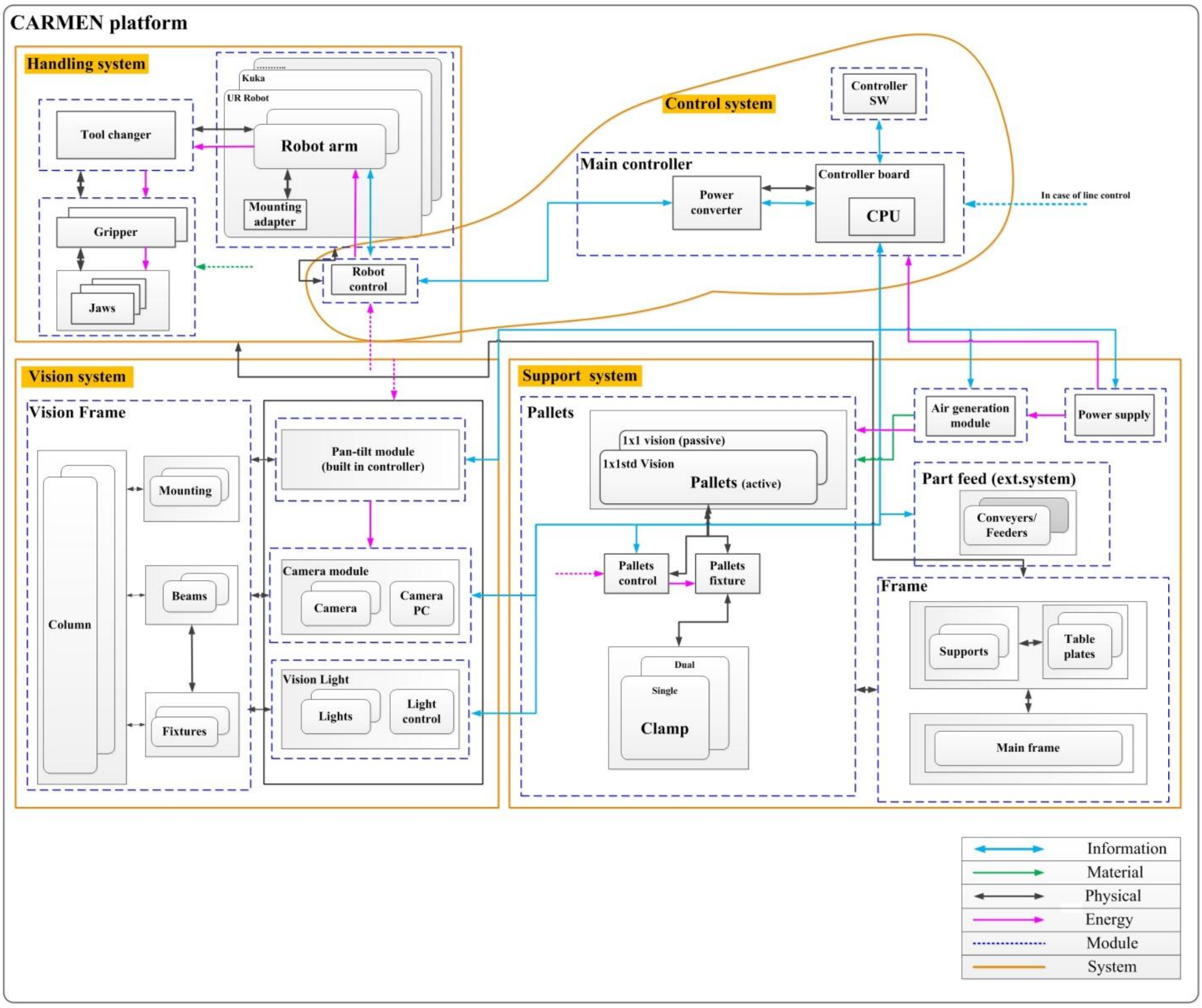


Fig. 43. Technical solution and their encapsulation into modules, variants and interface design.

In the interface diagram of the CARMEN platform (Fig.43), the product structure is composed of four systems: vision, support, handling and control systems. The modular product structure assumes that the design changes and number of variants should not be spread within the whole system, but be confined to just a few modules. In this product in the support system, variants are introduced only in the base, vision and pallets modules. For instance, two variants are introduced in pallets, such as active and passive. Active pallets not only facilitate holding the fixture but also help to move it, while passive pallets only hold the fixture. Pallets modules have an interface with the main control, power supply, air generation and main frame.

Step 6. Improve Each Module

In this step, the improvement that is performed in each module relates to specifications, design for assembly and design for manufacturing. The specifications of each module can be technical information, cost targets, description of variants etc. The module interaction matrix gives important information about what is important for each individual module. Further evaluation is mainly concerned with overall costs and design for assembly. The metrics and rules to find the quantitative relations are not discussed here, but they are related to development costs, lead time in development, product costs, quality, variant flexibility, service, and upgrading. This is an iterative process that uses the three main steps to optimize the modules in a product.

In this example, the specifications related to the gripper module can be that the technical solutions are gripper and gripper jaws, that the strongest drivers are different specifications (i.e. description of variants), and supplier availability (i.e. related to cost), that it has a physical and energy interface with tool changer, etc. Further evaluations related to cost and design for assembly is not discussed.

6.4. Conclusion

In the early stage of mechatronic systems design, designers and engineers have the important task of designing and optimizing modules and their interfaces by analysing the system architecture of existing products. In this study, an existing modularization method was analyzed and its applicability was demonstrated using an example with the following modifications:

- In the modular function deployment method, there is no discussion of the level of product decomposition (i.e. functions and technical solutions). In the original method, functions are decomposed only into sub functions, and then technical solutions are identified, but there is no description of the level of function decomposition. However, in this example, hierarchical functional decomposition was performed until no more decomposition was possible.

- The original method is modified with the introduction of organ domain to support and identify variants in the organs for solution space development. Here, these organs basically act as function carriers. At a very abstract level, the organ structure also supports the interfaces among them, as described in the theory of domains. However, the interfaces between organs are not identified at the organ level due to the further decomposition into subassembly and part domains.

In this method, product modularization is performed in three main steps. Initially, the system decomposition process is used, which is the key concept of the modular function deployment method. In this process, product properties are first derived from customer requirements. Then quality function deployment (QFD) analysis is used to determine what product properties are essential for a design that can fulfil the customer requirements. Second, the product’s main functions are decomposed into sub-functions until no further decomposition is possible, and they are then linked to organs. The technical solutions are derived from the organs by further decomposition into assembly and part domains. In modularization, each technical solution is analyzed with respect to the module drivers to form potential modules in a module interaction matrix (MIM). In this matrix, each technical solution is assessed against respective module drivers according to the importance of its reasons for becoming a module. Finally, in the evaluation step, module interfaces are identified. All these steps are used in an iterative process to optimize the individual modules.

7. CONCLUSION

This chapter is divided into five sections. The first section contains the conclusion originating from hypothesis 1 and the associated research questions, and, in relation to system architecting, it discusses the proposed method, as well as the analysis and results. The second section presents conclusions and product analysis originating from hypothesis 3 and the related research questions; here, the emphasis is on design structure matrix modelling in relation to product decomposition and structure analysis. The third section is about conclusions from the modelling and simulation of mechatronic systems. The fourth section contains conclusions and modifications regarding a modularization method also originating from hypothesis 3. The last section is about the limitations and future perspectives of the research topics.

7.1. System Architecting

A method was proposed in the system design of mechatronic systems to develop system architectures for next generation products. The proposed method has been validated and its applicability demonstrated using a case study in which the method was applied to a comparative analysis of the architectures of three autonomous vacuum cleaning robots.

The system decomposition process, which is the key concept of the proposed method, was divided into four processes. First, customer requirements were translated into system requirements. Second, system-level specifications were transformed into subsystems and components. Third, the physical and logical configurations of subsystems and components that realized the desired functions and behaviours were defined. Fourth, the system performance was analysed, for example in terms of functionality. In these processes, the intended interactions between the components and the technologies from different mechatronic domains were identified. The interactions were described using the design parameters that link the building blocks (which correspond to specific components and technologies) through physical phenomena. In addition, the decomposition process identified commonality instances (in entities and in physical phenomenon) in the product architectures, which can be used for the development of product platforms, and the differentiation in these architectures, which can be used for product family modelling.

In this method, a comparative analysis of the design concepts of mechatronic systems was performed with the SA-CAD tool, which supports system decomposition and modularization based on the dependencies among the parameters of subsystems. Multiple system architectures were developed to introduce product platforms and model product families. The commonality of the three considered systems was analysed in the parameter network to determine common building blocks. The proposed method could identify several levels of classifications regarding the level of commonality observed in existing systems, such as the behavioural level (physical phenomena) and the structural level (type of components, i.e. entities, used and their numbers and locations). In the case study with the vacuum cleaning robots, it was observed that the suction system and the wheel assembly were common in all three analysed autonomous robots regarding physical structure and behaviours (physical phenomenon). In the body module, all robots exhibited common behaviour because of common components such as bumpers, bump sensors, proximity sensors and cliff sensors. The robots differed slightly in terms of the number of common components and their location.

The commonality instances and differences identified by such classifications were used for the development of product platforms. In this case study, two platforms were developed, based on the round design and D-shaped design of the robots. These platforms were further extended to model product families, where the derivative products were developed on the basis of the variation of the existing architectures. Variation was identified mainly in terms of physical entities such as the main brush, side brush and the number and type of sensors. Using the proposed method, designers would be able to identify the maximum number of standard components required and implement minimal architectural changes. The proposed method would enable the development of several products in a product family and the identification of components that are shared by products in a product family to provide designers with the flexibility needed to deal with drastically changing market needs.

7.2. Product Decomposition and Structure Analysis

This study has implemented a component-based design structure matrix in order to address the issues of multi-domain system decomposition, interface management, and structure analysis and to assess the degree of modularity in these systems.

From the result of the structural analysis of the printer system case, it can be observed that the element relationships are more spread in the matrix and there are more connections per element. From the singular valve modularity index, the printer structure is integrative with few modules identified in the composite design structure matrix. According to the Ulrich and Eppinger definition, one to one mapping from function to structure cannot be applied in this case. In the printer example, more than one interface is present with each component on average, which indicates that the functions performed by each component in the architecture are distributed. The distribution of the relationships is much higher, and this must be reduced to increase the degree of modularity in the architecture. For instance, any change in the printer motor (for instance, to increase print speed) may influence the interfaces with subsystems and components such as the printer drive assembly, the main CPU and the control software.

Attainability in product architecture is measured to assess the change in any node and its impact on other nodes. A high value of attainability means that many other nodes can be affected by any change in the node in consideration. In the printer example, the attainability in nodes such as CPU control and printer drive assembly is higher than in the other nodes of the system.

The optimal solution in the case of mechatronic systems could be a high performance product with a few modules that can be used for commonality, flexible design and customization. However, the degree of modularity in mechatronic products varies and cannot be generalized due to factors such as performance requirements (power consumption, weight, size, speed etc), product structure and market demands.

7.3. Mechatronic Module Development - System Modelling and Simulation

In this study, two examples have been used to illustrate the modelling and simulations of electromechanical systems. In the first example, Bond Graph modelling is applied to the design and simulation of mechatronic systems. This modelling is important to represent the process (i.e. flow of energy in the physical part) and design of a controller (the information part) in mechatronic systems. Using the antenna example, the modelling process is summarized in a systematic way from design integration to simulation of the system. With this modelling, the development process is simplified and structured, the integration of multi-domain systems is obtained and automatic code generation for controller design becomes possible.

In the second example, the modelling and simulation of a physical prototype of an actuator are conducted. A fully variable valve train actuator is designed for two stroke engine applications. Since a stronger spring rate (240 N/mm, light springs) is used in the scaled down actuator, the oscillation time is considerably reduced and a high speed is reached for the actuator. Experimental results at valve opening and closing are carried through a test rig in the same way as for a real engine with a reduced chamber volume of 50 cm^3, which is able to operate at up to 8000 rpm engine speed, and is investigated on a test rig with a lift of 6.5 mm.

7.4. Testing and Modification of Modular Function Deployment Method

In the modular function deployment research approach, product modularization is divided into three phases. Initially, system decomposition is performed as the key concept of the modular function deployment method. In the decomposition phase, the product properties are first derived from the customer requirements. Then quality function deployment analysis is performed to determine which product properties are essential for a design that can be used to satisfy the customer requirements. Second, the product main functions are decomposed into sub-functions until no further decomposition is possible. After the functional decomposition, the concept of organs is introduced to indicate variants. The technical solutions are derived from the organs by further decomposition into assembly and part domains. In the modularization phase, each technical solution is analyzed with respect to the module drivers to form potential modules in a module interaction matrix. In this matrix, each technical solution is assessed with respective module drivers according to the importance of its reasons for becoming a module. Finally, in the evaluation phase, module interfaces are identified. All these steps are applied in an iterative process to optimize the individual modules.

Two modifications are applied to this method**.** First, there is no discussion in the original method of the level of product decomposition (functions and technical solutions). In the original method, functions are decomposed only into sub-functions, and then technical solutions are identified, but there is no description of the level of function decomposition. However, in this research, hierarchical functional decomposition is performed until no more functional decomposition is possible. This decomposition supports the mapping of functions onto structures with more details. Second, organ domains are introduced between functions and technical solutions to support and identify variants in the organs for solution space development. Here, these organs basically act as function carriers.

7.5. Limitations and Future Perspective

In this research, the following issues, limitations and perspectives have been observed and merit further research in the system design phase of multi-domain systems.

7.5.1. System Architecting

During the study, the following limitations and perspectives for further research have been observed regarding the comparative analysis method as well as the technical capability of the implementation used to present the method.

First, in the case study, three existing products were used for the comparative analysis. It is better to include a larger number of products within the scope of comparative analysis to obtain a wider variety of unique building blocks for the development of design concepts.

Second, as the case study has shown, the number of nodes such as entities, parameters, and their relations becomes very large along with the decomposition of the function. Thus, visualization might be useful to classify these nodes at various hierarchical levels. In particular, automated procedures to highlight (and hide) nodes that are relevant to a specific function node are useful to clarify the interest of the designer in terms of the state of function decomposition.

Third, this method supports designers in the development of design concepts by identifying the design parameters characterizing each design concept and common modules in existing systems that have been subjected to comparative analysis. However, the method cannot support the quantitative analysis of the performance of design concepts on the basis of numerical simulations. The integration of quantitative and qualitative information is an important topic for further research.

7.5.2. Modelling with the Design Structure Matrix

The design structure matrix can be used for system decomposition and establishing relationships in the subassemblies and components of the existing system for product up-gradation. However, this method cannot be used as a complete solution for product customization to generate solution space development. The following limitations have been observed by applying this method to a multi-domain system.

First, the design structure matrix and domain mapping matrix do support system decomposition and modelling relationships between elements from different mechatronic domains, but they do not support the establishment of relationships from function to form at multiple levels in a single model. Second, the design structure matrix approach does not support the generation of variety in products that can be used for product family modelling. However, the method establishes relationships among system elements to represent product architecture and might guide designers to perform structural optimizations that can be useful for the creation of specific system variants for product customization.

7.5.3. Modular Function Deployment Method

The following limitations have been observed in applying this method to an industrial case. First, the method includes a step in which interface design can identify the assembly structure, but it lacks detailed interfaces, especially in the case of multi-domain products. Second, the method is suitable for module-based product life cycle issues, but it does not support the technical information (such as the design parameters) necessary for up-gradation and the optimization of modules. Third, this method does not specify how to generate and identify variants in the decomposition phase.

In the modular function deployment method, the design structure matrix can be introduced at the interface step to identify the detailed relationships between subassemblies and components of the system to evaluate each module for further improvement.

Bibliography

Albers, A. Braun, E. Sadowski, D.C. Wynm, D. Wyatt, P. Clarkson, System architecture modelling in a software tool based on the contract and channel approach (C&C-A), J. Mech. Design 133(10) (2011).

Andreasen,M.M (2005).Functional Reasoning in Mechatronic Design, Lecture notes. Lecture notes edn., Lyngby.

Andersson, Gunnar. (1994). Criticism and the history of science: Kuhn's, Lakatos's and Feyerabend's criticisms of critical rationalism. Leiden, Holland: E.J. Brill.

Arbnor and Bjerke, (2009), Methodology for creating business knowledge, Second Edition, Thousand Oaks, London: Sage.

Boucher, M., Houlihan, D. (2008) 'System design: new product development for Mechatronics', Aberdeen Group; Boston (MA, USA).

Behbahani.S, De Silva.W, (2007) "Mechatronic modelling and Design", Mechatronic systems, Devices, Design, Operation and Monitoring, CRC Press, 17-1-17-26.

Burr, J., 1990, "A theoretical approach to mechatronic design", PhD thesis, Technical university of Denmark, Denmark.

Berman B., (2002), should your firm adopt a mass customization strategy? Business Horizons, 45(4), 51– 60.

Borches,P.D., and Boneema, G.M., (2010). A3 architecture overviews: Focusing architecture knowledge to support evolution of complex systems, 20th annual INCOSE symposium (IS2010).

Boneema, G.M., (2011). Insight innovation, and big picture in systems design, System Engineering, 14(3), pp 223-238.

Cabrera, A.A. Alvarez , M.J. Foeken , O.A. Tekin, K. Woestenenk , M.S. Erden , B. De Schutter ,M.J.L. van Tooren , R. Babuška , F.J.A.M. van Houten , T. Tomiyama , (2010) ' Towards automation of control software: A review of challenges in mechatronic design', Mechatronics, vol. 20, no.8, (2010) 876–886.

C. da Cunhaa, B. Agardb and A. Kusiak, (2010), Selection of modules for mass customization, International Journal of Production Research Vol. 48, No. 5, 1 March 2010, 1439–1454.

Craig. (2009). 'Mechatronic system design', ASME Newsletter 2009.

C.Y. Baldwin, K.B. Clark, Modularity in the design of complex engineering systems, In: Understanding Complex Systems, Springer Berlin, Heidelberg, 2006, pp. 175–205.

Coughlan, P. and D. Coghlan. 2002. Action Research for Operations Management. International Journal of Operations and Production Management, 22 (2): 220-240.

Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches* (3rd ed.). Thousand Oaks, California: Sage Publications.

Crotty, M. (1998). The foundations of social research: Meaning and perspective in the research process. London: Sage.

Davis, S.M. 1989, "From "Future perfect": Mass Customizing". Strategy & Leadership, vol.17 (2), 16-21.

Dahmus, J.B., Gonzalez-Zugasti, J.P., and Otto, K.N., 2001. Modular product architecture. Design Studies, 22, 409–424.

Danilovic. M, Tyson R. Browning. (2007). Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of Project Management 25 (2007) 300–314.

De Silva, C.W. (2007) 'Mechatronic Systems', Technology needs for mechatronic systems, CRC Press, 2007.

De Silva,C.W.(2005). Mechatronics: An integrated approach. CRC Press, Boca Raton, 2005.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T., Monostori, L., Complexity in engineering design and manufacturing. CIRP Annals - Manufacturing Technology 61 (2012) 793–814.

Erden, M.S, Komoto, H., van Beek, T.J., D'Amelio, V., Echavarria, E. and Tomiyama, T. (2008). "A review of function modeling: Approaches and applications". *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 22: 147–169.

Eppinger Steven D, Browning Tysen .Engineering systems: Design structure matrix methods and applications. MIT press, 2012.

Ericsson, A. and Erixon, G., 1999. *Controlling design variants: Modular product platforms*. ASME Press. New York, NY.

Eisenhardt, K.M. (1989) "Building theories from case study research." Academy of Management Review, 14(4), pp.532-550.

Fixson, S. K. *The multiple faces of modularity – a literature analysis of product concept for assembled hardware products*. Technical report. 03-05 Industrial & Operations engineering. University of Michigan, Ann Arbor, MI. 2003.

Fabrizio, S., Piller, F., & de Holan, P. M. (2009). Cracking the Code of Mass Customization. MIT Sloan Management review, 50(3), 70 - 79.

G. Huang, S. Bin, G. Halevi, Product platform identification and development for mass customization, Annals of the CIRP 52(1) (2003) 117–120.

Gawthrop, P. and Bevan, G. (2007), "Bond-graph modelling".*IEEE Control Systems Magazine*, 27:24, 45. doi:10.1109/MCS.2007.338279.

Gillham,B, (2000). Case Study Research Methods. Contium Publishers, UK. (2000), p-1.

Gummesson, E. (2000). Qualitative Methods in Management Research (2nd ed.) London, UK: SAGE.

Ghauri, P., & Grønhaug, K. (2005). *Reserach Methods in Business Studies: A practical Guide* (3rd ed.). Harlow: Prentice Hall.

Guba, E.G. (1990). The alternative paradigm dialog. In E.G Guba (Ed), the paradigm dialog (pp17-30.Newbury park, a: Sage.

Hubka V, Eder WE (1988) Theory of Technical Systems: A Total Concept Theory for Engineering Design. Springer Verlag.

Hansen CT, Andreasen MM (2002) Two Approaches to Synthesis Based on the Domain Theory. In: Chakrabarti A (ed).Engineering Design Synthesis. Springer-Verlag, London.

Hvam, L., Mortensen, N.H. & Riis, J. (2008), Product customization, Springer Verlag , Edition 1, 17-40 (Book)

Hsiao, S.W. and Liu, E., 2005. A structural component-based approach for designing product family. Computers in Industry, 56, 13–28.

Isermann, R.: Mechatronic Systems Fundamentals, Springer Publishing Group, Berlin, Germany, 2002.

Isermann, R. (2008) 'Mechatronic systems-Innovative products with embedded control', *Control Engineering Practice, vol.*16, p.14–29.

Isermann, R. (2008) 'Mechatronic Systems Fundamentals', *Meccanica*, Vol. 43, Issue 4, 2008, p.459-460.

J. Jiao, M. Tseng, Fundamentals of product architecture, Integrated Manufacturing Systems 11(7) (2000) 469–483.

J. H. Mikkola and O. Gassmann, "Managing modularity of product architectures: Toward and integrated theory," *IEEE Trans. Eng. Manage.*, vol. 50, no. 2, pp. 204–218, May 2003.

Joergensen, K. A. (2000). A selection of system concepts. *Special Report Department for Production, Aalborg University*, 2000.

Katja Hölttä, Eun Suk Suh, and Olivier de Weck. (2005). Tradeoff between modularity and performance for Engineered systems and products. International conference on engineering design. ICED 05 Melbourne, august 15 – 18,2005.

Karnopp, D., Margolis, D, and Rosenberg, R., "System dynamics: modeling and simulation of mechatronics systems", John Wiley & Sons, 2006.

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*, 1st. ed., Chicago: Univ. of Chicago Pr., 1962, p. 168.

Komoto.H , Tomiyama.T, A system architecting tool for mechatronic systems design. Annals of the CIRP 59(1) (2010) 171–174.

K. Forsberg, H. Mooz, The relationship of systems engineering to the project cycle, Engineering Management Journal 4(3) (1992) 36–43.

Kreimeyer, M; Braun, S; Gürtler, M; Lindemann, U. (2008).Relating two Domains via a Third – An Approach to Overcome Ambiguous Attributions using Multiple Domain Matrices .In: ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE.

K.T. Ulrich, S.D. Eppinger, Product Design and Development, McGraw-Hill, New York, 2008.

Karnopp, D., Margolis, D, and Rosenberg, R.,(2006) "System dynamics: modeling and simulation of mechatronics systems", John Wiley & Sons.

Karnopp, D., Margolis, D, and Rosenberg, R.,(2006) “System dynamics: modeling and simulation of mechatronics systems”, John Wiley & Sons.

Lincoln, Y.S., & Guba, E.G. (2000), Paradigmatic controverseries, contradictions, and emerging confluences. In Y.S. Lincoln & E. G. Guba (Eds), Handbook of qualitative research (pp.163-188). Thousand Oaks, CA: Sage.

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: Proceedings, IEEE/ASME international conference on mechatronic and embedded systems and applications.2008, pp. 393–398.

M.J. Van Wie, Designing product architecture: A systematic method, Ph.D. Thesis, University of Texas, Austin, 2002.

Modelica. A unified object oriented language for physical systems modelling, language specification. Version 3.2. Modelica association 2010.

Martin, M.V. and Ishii, K., 2002. Design for variety: developing standardized and modularized product family architectures. Research in Engineering Design, 13, 213–235.

M. H. Meyer and A. P. Lehnerd: The Power of Product Platforms. The Free Press, New York (1997).

Mikkola, Juliana, H. (2007) Management of Product Architecture Modularity for Mass Customization: Modelling and Theoretical Considerations. IEEE transactions on engineering management, vol. 54, no. 1, February 2007.

Mouton,J. (1996). Understanding Social Research. Van Schaik Publishers, 1996. 272 p.

Mertens, D .M. (1998).Research methods in education and psychology: Integrating diversity with quantitative and qualitative approaches. Thousand Oaks, CA: Sage.

Neuman, W.L. (2000).Social research methods: Qualitative and quantitative approaches (4th ed.).Boston: Allyn & Bacon.

Nagel, R.L., Stone, R.B., Hutcheson, R.S., McAdamas, D.A., and Donndelinger, J.A., 2008, “ Function design framework (FDF): Integrated process and function modelling for complex systems” ASME 2008 International design engineering technical conferences and computers and information in engineering conference, IDETC/CIE 2008, pp 273-283.

Nielsen. K, Brunoe, T.D, Joergensen K.A, Taps, S. B. (2014). Mass customization measurements matrics. Proceedings of the 7th world conference on mass customization, personalization and co-creation (MCPC 2014). Aalborg, Denmark.

O’Grady Peter. *The age of modularity*. Adams and Steele. 1999. ISBN 0-9670289-0-6.

Pahl,G, Beitz,W. Engineering design: A Systematic Approach, Springer, London, 1996.

P. Hehenberger; Zeman, K.(2008) : “The role of hierarchical design models in the mechatronic product development process”, TMCE International Symposium Series on Tools and Methods of Competitive Engineering, Izmir, Turkey, 2008.

P. Hehenberger, M. Follmer, R. Geirhofer, K. Zeman. (2013) Model based system design of annealing simulators, Mechatronics. Volume, April 2013, Pages 247–256.

P. Frederiksson and L. E. Gadde, “Flexibility and rigidity in customization and build-to-order production,” Ind. Marketing Manage., vol. 34, pp. 695–705, 2005.

P. Hehenberger , F. Poltschak , K. Zeman , W. Amrhein. (2010). Hierarchical design models in the mechatronic product development process of synchronous machines. Mechatronics. Mechatronics 20 (2010) 864–875.

P. Hehenberger, Application of mechatronic CAD in the product development process, Computer-Aided Design & Applications 6(2) (2009) 269–279. DOI: 10.3722/cadaps.2009.269-279.

Pine, B.J. (1993), Mass customization: The new frontier in business competition, Harvard Business school Press, Boston, MA.

Popper, KR (1974). Conjectures and refutations: The growth of scientific knowledge. London: Routledge and Kegan Paul.

Popper, K R (1959). *The logic of scientific discovery*. London: Hutchinson.

Popper, K R (1992). In search of a better world: Lectures and essays from thirty years.London: Routledge.

Rehman, F.U. and Yan, X-T. (2011) ‘Application of context knowledge in supporting conceptual design decision-making’, Int. J. Product Development, Vol. 13, No. 1, pp.47–66.

R.B. Stone, K.L. Wood, R.H. Crawford, A heuristic method for identifying modules for product architectures. Design Studies 21(1) (2000) 5–31.

R.B. Stone et al., A customer needs motivated conceptual design methodology for product portfolio planning. Journal of Engineering Design.19(6) (2008) 489–514.

R. I. v. Hoek and H. R. Commandeur, “Reconfiguring logistics systems through postponement strategies,” J. Bus. Logistics, vol. 19, no. 1, pp.33–54, 1998.

Rokni, H., Seethaler, R. J., and Milani, A., “Improved Mechanical Design for a Fully Variable Electromechanical Valve Actuation System for Internal Combustion Engines,” Proceedings of ASME, IMECE, Paper No. IMECE2010-38711, 2010.

Steffen N, J. (2013). Developing modular manufacturing system architectures: The foundation to volume benefits and manufacturing system. Changeability and responsiveness, PhD thesis, Aalborg University Denmark.

20-Sim, the dynamic modeling and simulation package for iconic diagram,bond graph, block diagram and equation models. Controllab products BV. 2010.

Simon, H. A. The architecture of complexity. Proceedings of the American Philosophical Society. 106. pp. 467-482, Dec 1962.

Simpson, T.W., Maier, J., and Mistree, F., 2001a. Product family design: method and application. Research in Engineering Design, 12, 2–22.

Suh ,N.P., (2001), Axiomatic design: Advances and applications, Oxford university press, New York.

Suh, N.P (1990). The principle of design. New York: Oxford university press. http://www.axiomaticdesign.com/

Stone, R. and Wood, K., 2000. Development of a functional basis for design. Journal of Mechanical Design, 122 (4), 359–370.

Seethaler, R., and Duerr, K., “Electromagnetic Valve Actuation with Two Configurations,” Proceedings of ASME IDETC/CIE, Vol. **7**, pp. 391–398, 2009.

Schroeder-Heister, P. (2004).On the notion of assumption in logical systems. In R.Bluhm & C. Nimtz (Eds.), Selected Papers Contributed to the Sections of GAP5, Fifth International Congress of the Society for Analytical Philosophy, Bielefeld, 22-26 September 2003. Paderborn: mentis, 27–48.

Simulink® is a block diagram environment for multidomain simulation and Model-Based Design by Mathworks.

Steward DV. The design structure system: a method for managing the design of complex systems. IEEE Trans Eng Manage 1981; 28:71–4.

Torry-Smith, J.M, Qamar.A, Achiche. S, Wikander.J, Mortensen.N.H, During.C. (2013). Challenges in designing mechatronic systems. Journal of Mechanical Design. Vol.135/011005-1.

Tomiyama,T, V. D’Amelio, J. Urbanic, W. El Maraghy, Complexity of multidisciplinary design, Annals of the CIRP 56(1) (2007) 185–188.

T.U. Pimmler, S.D. Eppinger, Integration analysis of product decompositions, In: Proceedings of ASME Design Theory and Methodology Conference 68 (1994) 343–351.

Tomizuka,M.: Mechatronics: From the 20th to 21st century. 1. IFAC conference on mechatronic systems, Vol I, 2000, S. 1/10.

Ulrich , K. and Tung, K., 1991, Fundamentals of product modularity. Proceedings of the 1991 ASME Design Engineering Technical Conferences on Design/Manufacture Integration, Miami, FL.

Ulrich, K. and Eppinger, S. D., 1995, Product Design and Development (New York: McGraw-Hill).

Wyne, D.C.,Wyatt, D.F.,Nair,S.M.T. and Clarkson, P.J. “An introduction to the Cambridge Advanced Modeller”. Proceedings of MMEP 2010, Cambridge,UK, 2010.

Walcher, D., & Piller, F. (2012). The customization 500: A global benchmark study of online B to C mass customization (1st ed.) www.mc-500.com.2012.

Umeda, Y., Ishii, M., Yoshioka, M., & Tomiyama, T.(1996). Supporting conceptual design based on the function–behaviour–state modeller. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10(4)*, 275–288.

Van beek T.J, and Tomiyama, T, (2009), “Integrating conventional system views with function behaviour state modelling” , 19th CIRP conference, competitive design, pp 65-73.

Van Beek T, Erden MS, Tomiyama T (2010) Modular Design of Mechatronic Systems with Function Modeling. Mechatronics 20:850–863.

VDI 2206. Design handbook 2206, Design methodology for mechatronic systems, VDI Publishing Group, Düsseldorf, 2004.

www.FEV.com

www.mathworks.se/products/matlab/

W. Dieterle, Mechatronics systems: Automotive applications and modern design methodologies. Annual Reviews in Control 29 (2005) 273–277.

Yoshioka M, Umeda Y, Takeda H, Shimomura Y, Nomaguchi Y, Tomiyama T(2004). Physical Concept Ontology for the Knowledge Intensive Engineering Framework. Advanced Engineering Informatics 18:95–113.

Yin, Robert K. (2009). Case Study Research. Design and Methods. Sage Publications, Thousand Oaks, 4th ed. 2009, pp. 240.

Zhuo Liu, Yoke San Wong and Kim Seng Lee.(2010). Modularity analysis and commonality design: a framework for the top-down platform and product family design. International Journal of Production Research. Vol. 48, No. 12, 15 June 2010, 3657–3680.

Appendix

Printer systems
Formatter system
- Formatter CPU
- Control panel
- Image formation and control software

Control system
- Fuser heater control IC
- Controller Microprocessor
- Control software
- Power supply circuit
- Regulator IC

LASER/Scanner system
- Scanner motor
- Scanning mirror
- Focusing lens
- BD sensor
- LASER unit
- BD lens and mirror

Image formation system
Print cartridge
- Photosensitive drum
- Transfer charging roller
- Developing cylinder
- Primary charging roller
- Toner sensor

Paper feed system
- Tray assembly= (tray assembly+ tray pick up assembly)
- Paper size switches
- Paper feed assembly= Feed rollers and clutches+ pick up roller+ separation roller+ solenoids
- Paper transfer assembly= pre transfer roller + transfer solenoids
- Paper feed sensors = Photo sensors/media detection sensors

Paper delivery system
- Delivery tray
- Output rollers assembly/unit
- Output bin sensor

Fuser system
- Heater
- Pressure roller
- Fuser roller
- Paper exit sensor
- Cooling motor and fan

Fuser temperature sensor
Printer driving system
- Main motor
- Printer drive assembly = (Transmission Gears and Belts)

Optional:
Duplexing unit (printing on both sides of paper)
- Microprocessor
- Duplexing unit Solenoids

- Duplex feed motor
- Reversing motor
- Pickup paper detection sensor
- Reversed paper sensor
- Face up sensor

Envelope feeder system

- Envelope microprocessor
- Envelope detection sensor
- Roller assembly
- Envelop multiple feed sensors

System Design of Mechatronic Products

Models and Methods to Utilize Mass Customization

Part II Papers

List of Papers

Paper 1
Innovations in Mechatronic Products and Mass Customization. Habib, Tufail; Nielsen Kjeld; Jørgensen, K Asbjørn. Proceedings of MCPC 2011 international conference on Mass Customization and open innovation (16^{th}-19^{th} Nov, 2011) Berkeley, San Francisco, USA.

Paper 2
Comparative Analysis of Design Concepts of Mechatronics Systems with a CAD Tool for System Architecting. Habib, Tufail; KOMOTO, Hitoshi. Published in Mechatronics Journal 2014. DOI: 10.1016/j.mechatronics.2014.03.003.

Paper 3
Design Models in the Development of Mechatronic Systems: Virtual Prototyping and Mechatronic module development process. Habib, Tufail; Thomas Ditlev; Nielsen Kjeld; Jørgensen, K Asbjørn. Submitted to International Journal of Design Engineering, 2014 (under review).

Paper 4
Multidisciplinary Product Decomposition and Analysis Based on Design Structure Matrix (DSM) Modelling. Habib, Tufail. Proceedings of the 7^{th} world conference on Mass Customization, Personalization and Co-Creation (February 4^{th} - 7^{th}, 2014) Aalborg, Denmark.

Paper 5
Modelling and Investigation of Electromechanical Valve Train Actuator at Simulated Pressure Conditions. Habib, Tufail. Proceedings of the Nord design 2012 conference (August 22^{nd}-24^{th}, 2012), Aalborg, Denmark. Pp.208-216.

Paper 1

Innovations in Mechatronic Products and Mass Customization.

Habib, Tufail ; Nielsen, Kjeld ; Joergensen, Kaj Asbjørn.
Proceedings of MCPC 2011 international conference on Mass Customization and open innovation (16^{th}-19^{th} Nov, 2011). ISBN 978-1-471-63023-1.Berkeley, San Francisco, USA.

Innovations in Mechatronic Products and Mass Customization

Tufail Habib, Kjeld Nielsen, Kaj A Joergensen
Department of Mechanical and Manufacturing Engineering, Aalborg University, Denmark

Abstract

Mass Customization (MC) has been recognized as a successful strategy in the design and development of products tailored to customer needs. Global competition demands new products with added functionalities, as in the case of mechatronic products. These products are becoming more and more important as a product type and new inventions have resulted in drastic changes in design and development of mechatronic products. Conventional mechanical systems are enhanced by mechatronic systems.

In this paper, the particular structure and properties of mechatronic products compared to conventional mechanical systems is presented. The successful strategy of mass customization has contributed to innovations in the development of mechatronic products and the paper presents an overview of typical changes regarding functionalities from mechanical products towards mechatronic products.

1. Introduction

Current market situation require new product development, to fulfill rapidly changing demands and continuous improvement in existing technology. Time reduction to introduce new products to the market with high quality and low cost is the requirements for the competition.

Mass customization (MC) is often seen as a solution to the demands of the market. MC relates to the ability to provide customized products or services through flexible processes in high volumes and at reasonably low cost. The concept has emerged in the late 1980s and may be viewed as a natural follow-up to processes that have become increasingly flexible and optimized regarding quality and costs. The concept was introduced by (Davis, 1989) and followed by (Pine, 1993) and (Pine et al., 1993). Much more research has followed to identify which success criteria's to choose for MC, to justify the development of MC, how to implement, what to benefit by using MC strategy and how different types of companies implemented it successfully (Gilmore and Pine, 1997), (Jiao, J et al., 1998), (Ahlstrom.P et al., 1999), (Silveira et al., 2001), (Hvam, 2007) (Salvador. F et al., 2009).

Description of mechatronics structure in relation to MC has potential to improve design and development of mechatronic products. The demand for new products with more features has led to innovations in mechatronic products. The driving force behind these innovations is the functional improvement in the products.

As the development focus shifted more from traditional mechanical systems towards electronic components, the number of sensors, actuators, switches, control systems, cables and electrical connections also increases. The issues of size, weight and energy consumption arises. Similarly, the manufacturing of stylish, cost competitive and content rich products are the other challenges.

2. Transition from Mechanical to Mechatronics

Mechanical systems and products have used some form of electrical engineering principles and devices have been developed and used since the early part of the 20th century. The systems included typewriter, aircraft, automobiles and tool machines. These electromechanical systems were not mechatronic systems, because they did not employ the integrated approach. By mid fifties, some form of control were used in mechanical systems such as electronic control lifts. In the seventies, rapid advances in digital computers, and communication as a result of integrated circuits (IC) and micro computer technologies. Engineers and scientists felt the need for an integrated multidisciplinary approach to design, hence a mechatronic approach (De Silva, 2008). This term was first used by Yasakawa electric in Japan. This trend was visible in the development of products and processes like machine tools, industrial robots and automation processes. After eighties, this multidisciplinary approach matured in the shape of mechatronic systems and products.

Further, developments in electronics and information technologies resulted into shift of functions from mechanics to electronics, followed by the addition of extended functions (Isermann, 2009). Additionally, systems are developed with certain intelligent or autonomous functions from the artificial intelligence research.

These developments have improvements in the traditional designs and also initiates the development of innovative mechatronic systems. The technological evolution has an impact on the system structure and functions as summarized in the following table.

Table 1: The differences and the transition from mechanical to mechatronic systems (Bradley, 1994).

Mechanical system characteristics	**Mechatronic system characteristics**
Bulky system	compact
Complex mechanisms	simplified mechanism
Non adjustable movement cycle	programmable movement
Constant speed drive	variable speed drive
Mechanical synchronization	electronic synchronization
Rigid heavy structures	lighter structures
Accuracy determined by tolerance of mechanism	accuracy achieved by feedback
manual controls	automatic and programmable controls

Mechatronics is the science of systems (machines, equipment, products, etc.), which integrates mechanical and electronic components, including computers and software (Nielsen et al., 2009). This definition implies the integrated approach being used in mechatronic systems.

Mechatronic design deals with the integrated and optimal design of a mechanical system and its embedded control system. This definition implies that the mechanical system is enhanced with electronic components in order to achieve a better performance, a more flexible system, or just reduce the cost of the system (Ameregon, 2002).

All these transitions towards mechatronics are used by companies to customize their products, with new features and added functionalities in their products.

3. Key Elements of Mechatronic systems and their functions

The elements of the mechatronic systems are the base technologies comprising from mechanical, electrical and software engineering. The integration of these elements results into a mechatronic system, with the desired functions. The most basic functions of each element are described in the following text:

Mechanical elements refer to mechanical structure, mechanisms, thermo-fluid, and hydraulics aspects in a mechatronic system. These elements may include static and dynamic characteristics and interacts with its environment. These elements require physical power to produce some kind of motion, force, and heat.

Electromechanical elements are sensors and actuators. All processes and systems base on close loop control requires sensors, providing information on the process, as input for the control circuit. A variety of physical variables can be measured using sensors such as light, temperature, position, speed, acceleration, force, pressure, sound, direction, touch and humidity etc. The physical process is modified and influence by the actuators. Actuators in a mechatronic system transform electrical inputs into mechanical outputs as position, angle, speed and forces. For this purpose, actuators like motors, relay, solenoid, valve, cylinders, speaker, light emitting diode (LED), shape memory alloy, electromagnet and pumps are used.

The ***electrical-electronic*** elements are used to interface electro-mechanical parts such as sensors and actuators to the control interface-computing hardware elements. Electrical components are resistor (R), capacitor (C), inductor (L), transformer, and the circuits composed of these components while processing analog signals. Electronic elements can be refer to analog/digital electronics, transistors, thyristors, opto-isolators, operational amplifiers, power electronics, and signal conditioning.

Control computing hardware implements a control algorithm, which uses sensor measurements, to compute control actions to be applied by the actuator. Control interface hardware allows analog/digital interfacing. It is a communication of sensor signal to the control computer and communication of control signal from the control computer to the actuator. It is mainly comprised by microprocessors, microcontrollers, signal conditioning and data control elements.

4. Mechatronic products

Mechatronics is, “Synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes” (Harshama and Tomizuka, 1996). This definition implies that the *combined effect* (the intended purpose) in industrial products and processes is achieved through the integration of electronics, mechanical engineering and information processing.

For the design and production of high performance machines and consumer products the structure of mechatronic system and the interaction between components is of utmost importance. Mostly, in modern mechatronic products the physical system and the computer (digital signal processing) are interfaced by sensors and actuators along with a possible link to other systems and human machine interface.

Mechatronic products play a major role in e.g. medical technology, industrial automation, aerospace, energy, automotive, civil engineering, modern office environment and in household applications. The nature of mechatronic products increases the complexity of final goods that are often composed of multiple mechatronic products as well as conventional products.

Mechatronic products comprising of hardware and software components as shown in figure 1.

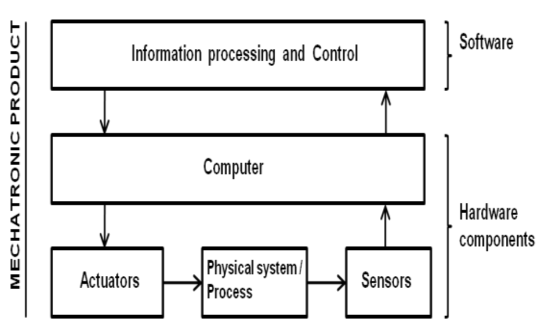


Fig. 1: Relationships between primary components in mechatronic product

4.1. Functions versus Structure of Mechatronic products

New inventions regarding electronics and information technology has resulted in a drastic change in design and development of mechatronic products. A large range of products have shifted from pure mechanical products to mechatronic products with electro mechanic and electric parts (Bishop, 2002; Chen 2009). Frequently, formerly mechanical functions are replaced by electronically controlled functions, resulting in simple mechanical structures and increased functionality. This is actually the evolutionary process in mechatronic products, which is basically the mixture of components (hardware) with an increased number of software driven functions (Isermann, 2009).

Mechatronics is the result of applying information technology to physical systems. According to Bradley, the level of abstraction is also highly different between the technical domains. Mechanical engineering deals with more physical properties of the design whereas software engineering concerns more abstract properties. Bradley illustrates this relation by putting electronics between the two, as indicated in figure 2.

A mechatronic system consists of a mechanical part that has to perform certain motions and an electronic part that adds intelligence to the system. In the mechanical part of the system power plays a major role; while in the electronic part of the system, information processing is the main issue. (Ameregon, 2002).

A simplified model of the mechatronic product consists of mechanic, electro mechanic, electronic and software modules (see figure 3). By this way, the structural and functional views are introduced. In the traditional understanding, structural issues of mechatronic products are entirely related to the hardware modules, while functional issues are related to both hardware and software. Furthermore, it is very characteristic for mechatronic products that functional issues are strongly related to the software module (Nielsen, 2009).

The functional view of the product can be as software module, which along with hardware modules is responsible for the overall control and capabilities of the product. This control can be digital, feedback and fault diagnosis. It can also enhance the adaptive and learning abilities of mechatronic products.

Mechanical Engineering	Spatial relationships Motion in three dimensions Forces Structures	**Physical**
Electronics	Signal processing Information transfer Communications	↕
Software	Algorithms Manipulation of data Logic	**Abstract**

Fig. 2: Levels of abstraction for mechatronics technologies (Bradley, 1997)

In a nutshell, the structural issues is related to physical part of the product, while the functional issues is more at abstract level, but somehow present also in the structural level as well.

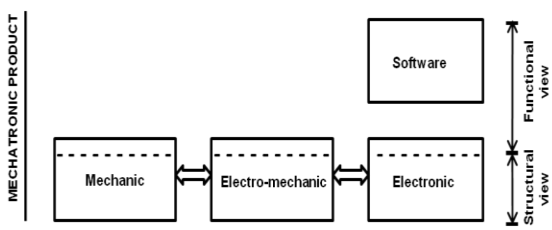


Fig.3: Simplified model of mechatronic products with indication of structural and functional views

5. Innovative products and MC

The potential benefits of mechatronics come from the innovative capabilities of the technologies and the functional and spatial integration of the technologies. The innovation potentials of the technologies are accelerated by the changing demands of the customers, which imply that MC has a role in these innovations.

All innovations in mechatronics are used by companies to customize their products, with functions suitable to customer requirements and needs. One aspect of these developments is ***Human machine interaction***, which have been enhanced with the development of electronics and information technologies (e.g. panels, touch screens, text messages, voice control, remote control) and interactions have become more versatile and user-friendly.

Another, aspect of innovations in mechatronics is ***adoptive customization***. It is possible for the users to modify or change the functions of a product. In the industry the modern drives are programmed via control panel. Software have given products increased capabilities with less hardware, for example a drive, controlling a conveyer belt can be programmed in different ways, such as starting, stopping at different intervals and advancing a certain distance. Similarly, the drives used in the ventilation system can be programmed to maintain constant air pressure. In case of robots, making modifications or changing settings through programming for desired positions and actions. In CNC machines, a program code from the user is used to implement machining processes like milling, shaping and lathe operations.

The innovations in embedded system led to the ***design flexibility*** in mechatronic products. Embedded Software can be designed with a number of parameters, which can easily be assigned different values and thereby be used for customization. These characteristics have also a great impact on manufacturing postponement because software, or even better, software parameters, can be changed late in the supply chain, ultimately at the customer site (Kaj, 2009).

With the improved VLSI techniques, complete SoC (system on chip) is possible. It has all the ingredients of embedded system with ***configurable platforms***. These systems provide high performance and low energy consumption. Due to increasing demand for more functions and features, the number of applications of embedded systems is growing, ultimately reducing the cost of mechatronic products.

Multiple features are integrated into a single device with mechatronic technologies. A printer may not only be able to print but may also enable faxing and scanning.

Successful implementation of ***Process flexibility*** in industry is through advanced manufacturing technologies. Manufacturing processes are influenced by the mechatronic technologies. That is evident in the form of CNC, CAM, and automation through robotic technologies.

6. Future trends

Future trends in the field of mechatronics are influenced by market, design approaches and technology evolution as summarized below,

Market oriented: new product generations will be smaller, cheaper and provide added functionality.

- Human machine interface
- Communication systems (internet, remote access, telemonitoring and telediagnosis)
- More customized products for specific needs and wishes.

Design Requirements: More electronics and software than mechanical and hydraulic systems. An integrated approach for the development and implementation of innovative mechatronic systems.

- Moors Law, which states that "The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years".
- Intelligent systems with learning behavior and decision making.
- Software will dominate functions and quality.

Technology Evolution: Micro electromechanical (MEMS) and nano electro mechanical (NEMS) based sensors and actuators. With more use of silicon technologies, size will be reduced and it has an impact on cost as well.

7. Conclusion

Due to the developments in electronics and information technologies, a large range of products have shifted from pure mechanical to mechatronic products. These developments enhance the capabilities of the mechatronic products with respect to performance, achieve flexibility and have an impact on costs.

This paper gives an overview of the key elements of mechatronic systems and their functions. Presenting, a model of the functional and structural view of mechatronic products. Such that, structural issues of mechatronic products are entirely related to the physical part, while functional issues are related to both hardware and software.

Furthermore, innovations in mechatronics are used to customize the products, with functions suitable to customer requirements and needs. From future perspective, more electronics and information systems are visualized in mechatronic products.

References

Ahlstrom, P., Westbrook, P. (1999). Implications of Mass Customization for Operations Management: An exploratory survey, International Journal of Operations and Production Management, 19(3): 262- 274.

Ameregon, J.V. (2002). The role of control in mechatronics, chapter 21, The mechatronics handbook-2 Volume set, Edited by Robert H Bisshop.CRC press 2002.

Bradley, D. A. (1997). The what, why and how of mechatronics. IEEE Journal of engineering science and education, 6(2): 81-88.

Bishop, R. H., Ramasubramanian, M. K.(2002). What is mechatronics? In: R. H. Bishop (ed.) The Mechatronics Handbook, CRC Press, p 1272.

Bradley, D. A., Dawson,D.,Burd,N.C and Loader,A.J. (1994). Mechatronics: Electronics in products and proceses, Chapter 21: Mechanical systems and design, p 398.

Chen K., Bankston J., Panchal J. H and Schaefer, D. (2009). A Framework for Integrated Design of Mechatronic Systems. In: Collaborative Design and planning for Digital Manufacturing, Springer London, ISBN 978-1-84882-286-3, 37-70.

Davis, S. (1989). From future perfect: Mass Customizing, Planning Review.

De Silva, C.W. (2008), Mechatronic systems, Technology needs for Mechatronic systems, Edited by Clarance, W De Silva, CRC Press, 1-11.

Gilmore J., Pine J. (1997). The Four Faces of Mass Customization, Harvard Business Review, 75 (1).

Hvam L., Mortensen N.H and Riis J. (2007). Product Customization, Springer Verlag, Lyngby, 2(17).

Harshama,F., Tomizuka,M and T. Fukuda.(1996). Mechatronics-what is it, why, and how? -and editorial, IEEE/ASME Trans. on Mechatronics, 1(1): 1-4.

Isermann R. (2009). Mechatronic Systems: A short introduction, Springer Handbook of Automation, 317-331.

Jiao J., Tseng M. M., Duffy, V. G and Lin, F.(1998). Product Family Modeling for Mass Customization, Computers and Industrial Engineering, 35:495–198.

Jørgensen, K .A. (2010). Customization and Modeling Applied to Electronic and Mechatronic Products. Nord design 2010, Gothenburg, Sweden, 24-27.

Nielsen K., Jorgensen K.A and Petersen T.D. (2009). Mechatronics and Mass Customization, MCPC Proceedings, Helsinki, Section 13.

Pine, B. J. (1993). Mass Customization the New Frontier in Business Competition, Harvard Business School Press, Boston Massachusetts.

Silveira G. D., Borenstein D and Fogliatto F. S. (2001). Mass customization: Literature review and research directions. Int. Journal of production Economics, 72: 1–13.

Salvador F., De Holan P. M., Piller F. (2009). Cracking the Code of Mass Customization, MIT Sloan Management Review, 50(3).

Paper 2

Comparative Analysis of Design Concepts of Mechatronics Systems with a CAD Tool for System Architecting.

Habib, Tufail; KOMOTO, Hitoshi. Published in Mechatronics Journal 2014. DOI: 10.1016/j.mechatronics.2014.03.003.

ARTICLE IN PRESS

Mechatronics xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting

Tufail Habib [a,*], Hitoshi Komoto [b,1]

[a] Department of Mechanical and Manufacturing Engineering, Aalborg University, Fibigerstraede 16, Aalborg, DK-9220, Denmark
[b] Advanced Manufacturing Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Namiki 1-2, Tsukuba, Ibaraki 305-8564, Japan

ARTICLE INFO

Article history:
Received 9 May 2013
Revised 12 March 2014
Accepted 23 March 2014
Available online xxxx

Keywords:
System architecting
Conceptual design
Modular design

ABSTRACT

In mechatronics systems design, designers need to deal with complexity derived from the integration of subsystems with various engineering disciplines. In particular, while developing product architecture for the next generation systematically, the present generation systems in the market should be reviewed in terms of their functional overview as well as module structure. This paper proposes a method for developing product architecture by a comparative analysis of the functional overview as well as physical decomposition of existing mechatronics systems. The method employs function–behaviour–state modelling and a computer-aided design (CAD) system for system architecting (SA-CAD) as a modelling scheme and modelling environment, respectively. The paper describes the application of the proposed method in the development of product architecture of autonomous vacuum cleaning robots.

1. Introduction

Mechatronics systems such as industrial robots, hybrid vehicles, modern computer numerical control machines, medical instruments and communication and satellite systems have been developed through functional and spatial integration of subsystems with various engineering disciplines. The complexity of these systems has increased owing to such integration, because no common language has yet been established for describing such integrated product models. However, such a language is crucial to enable designers and engineers to transfer design information among the domains as derived from various engineering disciplines. In the early design stage of mechatronic systems, methods and tools for systematic support are crucial. With the aim of managing complexity and providing systematic support to designers and engineers in the early design stage, the industry and academia have been focusing on modularity and configurability. Modularity is the use of modules to develop product variants, while configurability is how the composition of these modules can fulfill certain design requirements [1], which is relevant primarily to the development of product architecture.

Conceptual design is one of the important phases in the development of mechatronic systems, in which the product's overall functions, important sub-functions and their interactions are determined. Principle solutions, along with a structure of realisable modules and their interactions and interfaces, are determined to achieve such a functional description and facilitate the systematic design and development of mechatronic systems [4]. Conceptual design determines the principle of a solution, wherein the product's main properties—functions, behaviours, performance, cost and weight—are determined and fixed [2,6]. It is the main process for identifying the overall structure of the product through function decomposition, following the divide and conquer principle [7] and search for physical realisation. Consequently, suitable solutions are generated through the combination of basic building blocks corresponding to decomposed functions [5,8,9]. According to the V-model of product development from a systems engineering perspective [10–12], conceptual design is also referred to as system architecting and is considered part of the system decomposition phase, and it is described as the process of translating requirements into system requirements and transforming system-level specifications into subsystems and components [10]. In the process of conceptual design, designers and architects define the system decomposition and interfaces among the subsystems. They also perform an overview of the desired system behaviour (search for possible physical configurations of the subsystems to realise the behaviour) and evaluate the overall system performances, such as its functionality and costs [13]. In performing

* Corresponding author. Tel.: +45 29 16 78 80.
E-mail addresses: habib@m-tech.aau.dk (T. Habib), h.komoto@aist.go.jp (H. Komoto).
[1] Tel.: +81 (0) 298614139.

http://dx.doi.org/10.1016/j.mechatronics.2014.03.003
0957-4158/

these tasks, the designers have to deal with complexity derived from the interactions and constraints among the subsystems in multi-domain systems [14,26].

In the literature, various research groups have proposed approaches and tools for supporting the embodiment of function structures into forms and for developing product architecture. For example, Pahl and Beitz [2] described that in the development of modular systems, the physical structure comprising the assemblies and components used as building blocks and the relationships among these assemblies and components must be reflected in the function structure. Baldwin and Clark [3] focused on critical modules in the design of complex systems. Jiao and Tseng [15] described product family architecture (PFA) by applying functional–behavioural–structural-view modelling to mechanical products. A well-developed PFA can provide a generic architecture to capture and utilize commonality, within which each new product instantiates and facilitates future designs in a common product line structure. Stone et al. [16,17] proposed a heuristic method for identifying the functional modules of a product. In this method, energy and signal flows were analysed and heuristic rules were developed to identify functional modules for the development of modular product architecture. Ulrich, Pimmler and van Wie [19,33,18] used block diagrams with functional aspects to model system architectures. Further, Albers et al. [20] applied the contact and channel approach to mechatronic products with the support of a software tool to help designers understand and communicate the complex dependencies between function and form and generate system architecture through the function and part database. Sosa, Kreimeyer and Danilovic [28–30] described the decomposition of a product into subsystems, components and functions and the relationships among them by applying design structure matrix (DSM) methods to represent product architecture. However, despite all the apparent advantages of these approaches, they also have some limitations, especially in relation to mechatronic product development; these are explained below.

Some of the approaches proposed in the literature on product architecture are applicable only to mechanical systems [33,15,19] and not to mechatronic systems. Other methods such as the DSM and domain mapping matrix support system decomposition and modelling of relationships between elements from different mechatronic domains, but they do not support the modelling of relationships from function to form at multiple levels in a single model. Methods such as the PFA [15] cannot be implemented in software tools for dealing with the complexity derived from the interactions and constraints in multidisciplinary mechatronic systems. Commercial tools such as Modelica [31] and 20-Sim [32] do not support hierarchical system decomposition, especially the functional decomposition before physical realisation of the product. In the case of functional modelling, functional thinking supports part of the design activities but such methods do not provide sufficient information on issues such as how system elements contribute to system properties. Even though some established approaches can be applied to mechatronic products to deal with complex interactions between function and form [16–18,34], these approaches and methods do not provide sufficient knowledge and tools to support the various tasks of system architecting. The method proposed in this paper addresses these limitations by supporting the tasks of system decomposition, interface management and identification of system properties; it uses a CAD system for system architecting and has the ability to redesign and develop multiple product architectures.

The objective of the paper is to establish a method for supporting the development of product architecture for the next generation (a) by effectively utilizing the design knowledge of the current generation and (b) by computationally supporting the development process. To demonstrate and validate the proposed method, having a product modelling scheme and computational modelling environment are crucial. In particular, the modelling scheme should support domain-independent modelling description and the computational modelling environment should be compatible with the modelling scheme. As an example of the available combinations of modelling schemes and computational environments, the study employs the function–behaviour–state (FBS) [21] modelling scheme and a CAD system for system architecting (SA-CAD) [25]. This paper explains the suitability of this scheme and computational environment to the execution of the intended tasks in this study in comparison with related work. Further, the proposed method is validated by its application to the development of product architecture for autonomous vacuum cleaning (VC) robots through a comparative analysis. As explained in detail later in the paper, several autonomous VC robots in the market are modelled from the functional, behavioural and structural perspectives, and they are subsequently used for the development of a design guideline for the product architecture for the next generation from these perspectives. The observed advantages of the proposed method are as follows:

- Management of complexity through the hierarchical decomposition of the system, and development of modules and their interfaces using a computational tool.
- Consistency of system descriptions at different levels of hierarchy; for example, the decisions at the lower level should be traced to higher level abstract concepts, such as functions.
- Capability to redesign and develop multiple system architectures to facilitate product platform development and product family modelling.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the FBS modelling scheme. Section 3 describes the proposed method schematically and presents an overview of the overall approach. Section 4 validates the proposed method by its application to the development of product architecture of autonomous VC robots as a case study. This section also illustrates how to develop FBS models of these systems and explains the similarities and differences of these systems in terms of their functional and structural levels. Building blocks are constructed to perform network modelling of each product by hierarchical physical decompositions for developing the product architecture in SA-CAD. Section 4 also presents a comparison of product architectures and display of entities and parameters in the DSM. Furthermore, this section presents a commonality and differentiation analysis of the multiple architectures that support the development of product platforms. Section 5 discusses the optimal module structure and design optimization in relation to the considered case study. Section 6 provides limitations of the proposed method. Finally, Section 7 summarizes and concludes the paper.

2. FBS modelling

Before explaining the proposed method, this section briefly introduces FBS modelling [21], which is used as a representation scheme for product models in the proposed method. FBS modelling is based on three main concepts: function, behaviour and state. It defines a function as 'a description of behaviour abstracted by humans through recognition of the behaviour in order to utilize it' [35]. It represents a function as an association of two concepts: symbol of human intention represented in the form of *to do something* and behaviour that can exhibit the function. The information about the symbol along with behaviour is essential for supporting design and its results. Fig. 1 shows the relationship among function, behaviour and state. A function–behaviour (F–B) relation

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

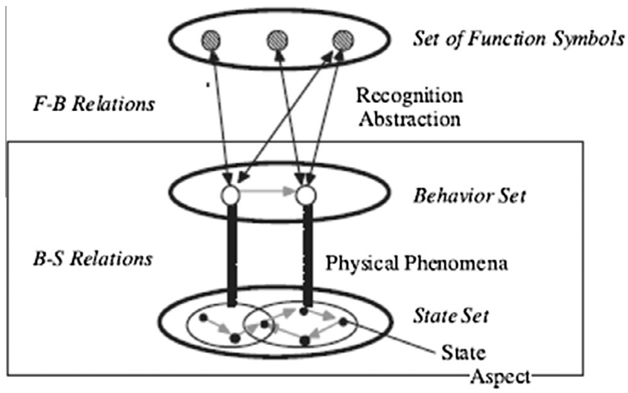


Fig. 1. Relationship among function, behaviour and state (adapted from Umeda [21]).

associates a function symbol with a set of behaviours. Behaviour is defined as a transition of states over time, and a behaviour–state (B–S) relation relates behaviour to states by using physical knowledge. Here, a state is represented by entities, their attributes and their structure, which also represents a physical realisation of the system [21].

2.1. Application of FBS modelling to mechatronic systems

FBS modelling is a domain-independent modelling method that can be used to represent mechatronic systems in one model. In this modelling, a function is a concept at the abstract level that is independent of any domains and can be applicable to both hardware and software, as well as to purely mechanical and electronically controlled sensor actuator systems. In this representation, behavioural description of the product is necessary to gather information about its structure and combine it with the information on the involved phenomenon, with the aim of determining how the product works and the relations between its parameters.

The advantage of FBS modelling over other modelling techniques is that it associates the functional descriptions with the states representing physical structure via a behavioural level. In this way, development of the functional model of the system occurs in parallel with consideration of the real physical environment. This modelling can be implemented in a computational tool that supports system architecting tasks (e.g. system decomposition and complexity management) in complex mechatronic systems.

System architecting using FBS modelling facilitates the physical realisation of the system in the form of building blocks. These building blocks represent the qualitative and quantitative relations between parameters. In this study, the analysis of the building blocks is based only on qualitative relations, because the aim here is to develop product platforms. With further analysis, quantitative relations between these building blocks can also be derived from detailed parameter relations (e.g. differential equations). The use of methods and tools before the detailed design phase is required to support product platforms, multidisciplinary concurrent design and complexity management in mechatronic systems.

3. Proposed method

The proposed method supports the development of product platforms (or product families) based on the analysis of mechatronics systems of the same kind by FBS modelling. Fig. 2 shows a brief schematic overview of the proposed method. The designer

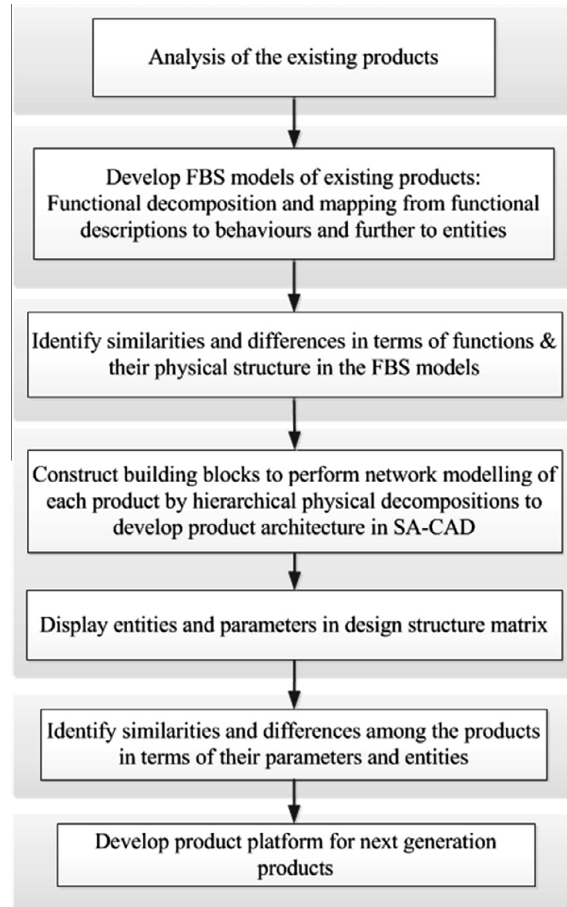


Fig. 2. Schematic description of steps in the proposed method.

first clarifies the functions of existing products of the same kind in the market and then analyses how these functions are physically realised. The designer subsequently analyses the differences among the products in terms of their functions and physical realisation. The FBS models of the products are developed to perform the comparison. Using these FBS models, the designer can explicitly identify the similarities and differences in the products in terms of the functions and their decomposition. Furthermore, a mapping from the functions to their realisations in terms of physical phenomena and entities is identified (Sections 4.1 and 4.2). The conceptual relations between the physical phenomena and the entities given in these FBS models provide designers with a mapping from the conceptual relations to the customer requirements and also with parameters used for design of the products and their dependency. The designer extracts parts of these mappings and constructs building blocks at different (functional, behavioural and structural) levels. With these building blocks, the same products are modelled by hierarchical physical decompositions and compared in terms of the dependency among the parameters of the products (Sections 4.3 and 4.4). Finally, the designer identifies the similarities and differences among the products in terms of their parameters and dependencies, and this information is subsequently used for the development of a product platform for the next generation (Section 4.5).

The entire process of system decomposition, implementation in SA-CAD and development of product platforms is further illustrated in Fig. 3. All these steps are explained in a case study with VC robots as an example.

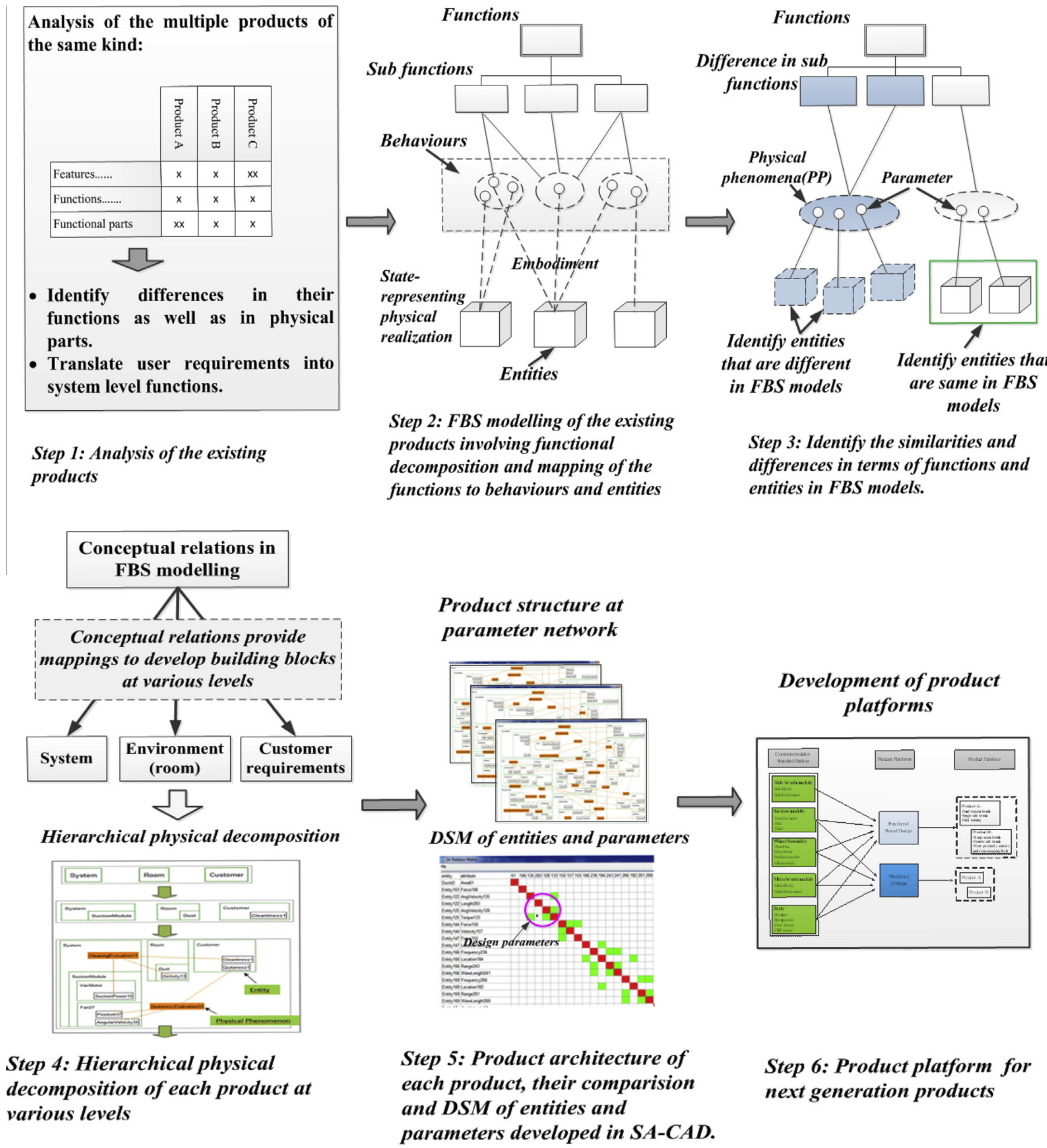


Fig. 3. Overall approach using analysis, FBS modelling, parameter network in SA-CAD and the development of product platforms.

4. Application of proposed method in a case study

This case study considers an example of system decomposition for autonomous VC robots by using the proposed method. This example illustrates how the designer performs FBS modelling of existing products to develop product architecture for autonomous VC robots and uses it to develop product platforms based on the parameters of the product and their dependency.

4.1. Analysis of existing products

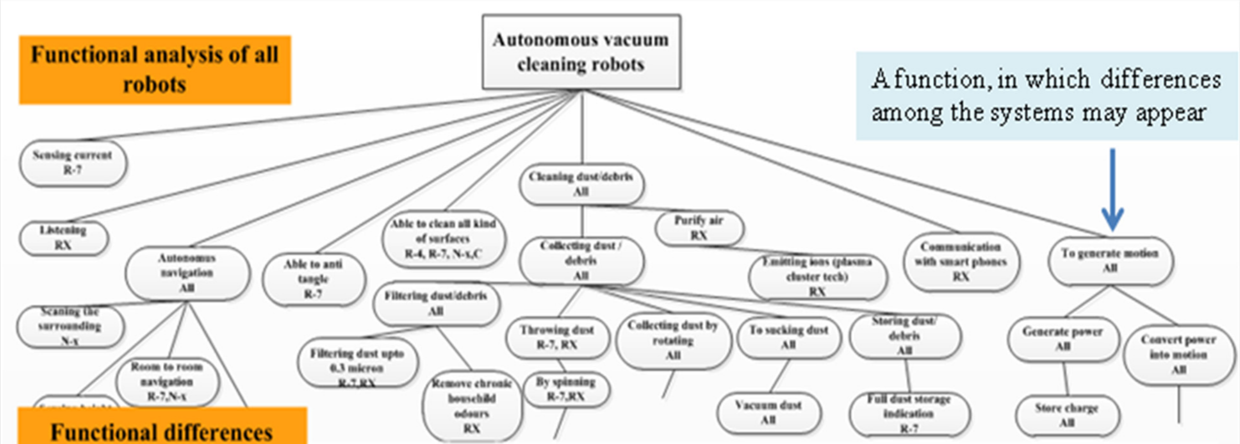
The designer first identifies the functions of existing products of the same kind in the market and analyses how these functions are physically realised. The designer then analyses the differences among the existing products in terms of the functions and their physical realisation, as shown in Fig. 4.

4.2. FBS modelling of VC robots

The next step in the proposed method is the development of FBS models of multiple VC robots. To perform system decomposition and compare the systems, the FBS models are developed with the following system architecting tasks:

- The user requirements are translated into system-level specifications.
- The main function is decomposed into sub-functions until no further functions can be realised.
- The system models are developed by embodying the functions into behaviours and structures, identifying the physical phenomena and identifying their relation with the entities of the system.

First, customer requirements are translated into system-level specifications. Then, the main functions are identified and



Functional differences

Functional parts of the products	System A	System B	System C
Storing debris/dust	X	X	X
Side wall cleaning	X		X
Vacuum generation	X	X	X
Filtering dust/debris	X	X	X
Collecting dust	X	X	X
Spot dust/debris	X		
Stay at dirty spots	X		X
	X		X

Differences at physical level

	Differences at the physical level the three systems :		
	System A	**System B**	**System C**
1	2 main brushes	1 main brush	1 main brush
2	1 bumper	1 bumper	No bumper
3	Use IR light	Use LASER positioning system	Use ceiling shape recognition
4	1 side brush	No side brush	2 side brushes
5	Follow random path	Follow straight path	Clean in straight (ext square) path
6	6 cliff sensors	2 cliff sensors	3 cliff sensors
7	Round shape	Square front	Round shape
8	....	....	....
9	....	....	....

Major differences of functionalities among the various VC robots:

- Side wall cleaning- throwing dust debris under the robot.
- Spot dust and debris
- Stay at dirty spots or spot cleaning mode.
- Room confinement and room to room navigation.
- Ability to anti tangle.
- Cleaning specific area by placing boundary markers
- LASER scanning
- Filtering very minute (in microns) dust particles.
- Way of collecting dust

Fig. 4. Analysis of multiple VC robots (product) to identify the differences in their functions and their physical structure.

decomposed into sub-functions until no further functions can be realised. In the case of VC robots, *cleanliness* is the main function that is decomposed into sub-functions, such as *to collect dust and debris, to release clean air* and *to move itself* (autonomous cleaning), as shown in Fig. 5.

After the functional decomposition, in the second phase, the system architect or designer develops the system by embodying the functions into behaviours and structures. At the behaviour level, the physical phenomena and the related parameters are identified. At the lower level of decomposition, all sub-functions must be realised by structural elements, called entities or building blocks. These entities are related to each other by parameters. The parameter network represents the architecture of a system. The designer uses this process to decompose the system specifications into subsystems and components until all the functions of the system have been realised.

As shown in Fig. 5, functions are linked to physical phenomena (*PP*) at the behaviour level and further linked to relevant entities at the state level. FBS models of the remaining two systems are developed in the same manner, and they are shown in Figs. 6 and 7.

4.2.1. Highlighting of differences in functions and entities of FBS models

Colours are used to highlight the differences in the three FBS models, as shown in Figs. 5–7. These differences are evident in functions (i.e. *to collect dust and debris* and *to navigate itself*), in subsystems and components (e.g. room positioning system and camera) and in their environment (e.g. magnetic boundary markers and virtual walls). The number of sensors and actuators also determine the performance of these systems. The differences in the three models help to clarify how these systems can be redesigned.

The functional decomposition reveals that the top-level functions of the three systems are the same:

- To collect dust and debris.
- To navigate itself.
- To generate motion.
- To release clean air.

As the above functions are further decomposed into sub-functions, differences are realised in two of the functions: *to navigate itself* and *to collect dust and debris.* The functions *to release clean air* and *to generate motion* are not decomposed further into behaviours and states, because their physical realisation is the same in all systems. The function *to navigate itself* includes different behaviours that are realised by each robot depending on customer requirements and their respective capabilities, such as to facilitate efficient navigation. Similarly, all the robots have differences in the function *to collect dust*. The functional differences of the three robots are summarized in Table 1.

The functions *to generate motion* and *to release clean air* are not considered for this analysis, because as mentioned above, all three robots have the same functional and structural considerations.

4.3. Construction of building blocks in each product for architecture modelling

In this step, the mapping from conceptual relations (i.e. physical phenomena) to entities in the FBS models enables designers to identify a mapping from conceptual relations to the customer requirements and also to acquire the design parameters of the products and their dependency. The designer extracts parts of these mappings to develop building blocks by the hierarchical decomposition of each product at various levels: functional, behavioural and structural. For instance, the building blocks in the suction system are *vacuum motor*, *fan* and *duct.* Their *PP* are rotation, air suction and air flow. Strong vacuum generated by a powerful motor is thus linked to *cleanliness* as one of the customer requirements, and their *PP* is *cleaning evaluation* and the design parameter is *suction power.* Relations between the vacuum motor and the fan are angular velocity and torque, respectively, as shown in Fig. 8.

The reason for clustering building blocks such as *vacuum motor*, *fan* and *duct* in the same module is that they are physically adjacent in the system. However, there may be exceptions to such clustering, as some entities cannot be clustered into modules because of the existence of numerous external relationships; such entities are therefore considered as separate entities. An example of such an entity is a microcontroller.

In the design of mechatronic products, the building blocks in mechanical design can be machine elements and machine components; those in control design are block diagrams (that represent sensors, actuators and controllers); finally, those in software design can be functions and subroutines. From the perspective of control design, building blocks of sensors and actuators are developed and the designer must identify the measured and control parameters. In the VC robot example, position encoder which is based on Hall effect phenomenon and whose qualitative parameters are voltage, magnetic field and current, is linked to the drive wheel to measure its angular velocity. These measured signals are used to control the actuator power for speed control. The designer uses the conceptual relations in the FBS models and develops a knowledge base in SA-CAD to perform network modelling. In this case study, the building blocks are based only on qualitative relations; however, quantitative relations based on differential equations can also be developed in SA-CAD. The designer may use external tools to analyse these relationships. For example, the information flow in a VC robot can be analysed by examining the relations between the parameters of the sensors and actuators (i.e. block diagrams in control design); for structure analysis and dynamics analysis, CAD tools and the finite element method, respectively, may be employed.

4.4. Product architecture development in SA-CAD

4.4.1. Concepts in knowledge intensive engineering frameworks

A conceptual design process based on FBS modelling is supported by the knowledge intensive engineering framework (KIEF). It is defined as a framework for integrating design modelling systems that has embedded knowledge about domain theories [22,23]. Several concepts such as physical phenomena, physical features, attributes of entities and physical laws are described in the KIEF. This study uses the concepts of *function*, *behaviour*, *state*, *entity*, *attributes*, *physical phenomena* and *physical law* as defined in the KIEF [23] and in FBS representation [21] as follows:

Entity: An entity represents an atomic physical object and its purpose is to describe the abstract concrete relationship among concepts. In this study, an entity is used as a building block or module. The entity concept in the KIEF is defined with its ‘name’ and ‘supers’. The following is an example of an entity:

Name	Fan(?e)
Supers	MechanicalParts(?e)

Relation: A relation represents a relationship among entities to characterize a static structure. The following is an example of a relation:

Name	Contact (?contact)
Supers	Relation (?contact)

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

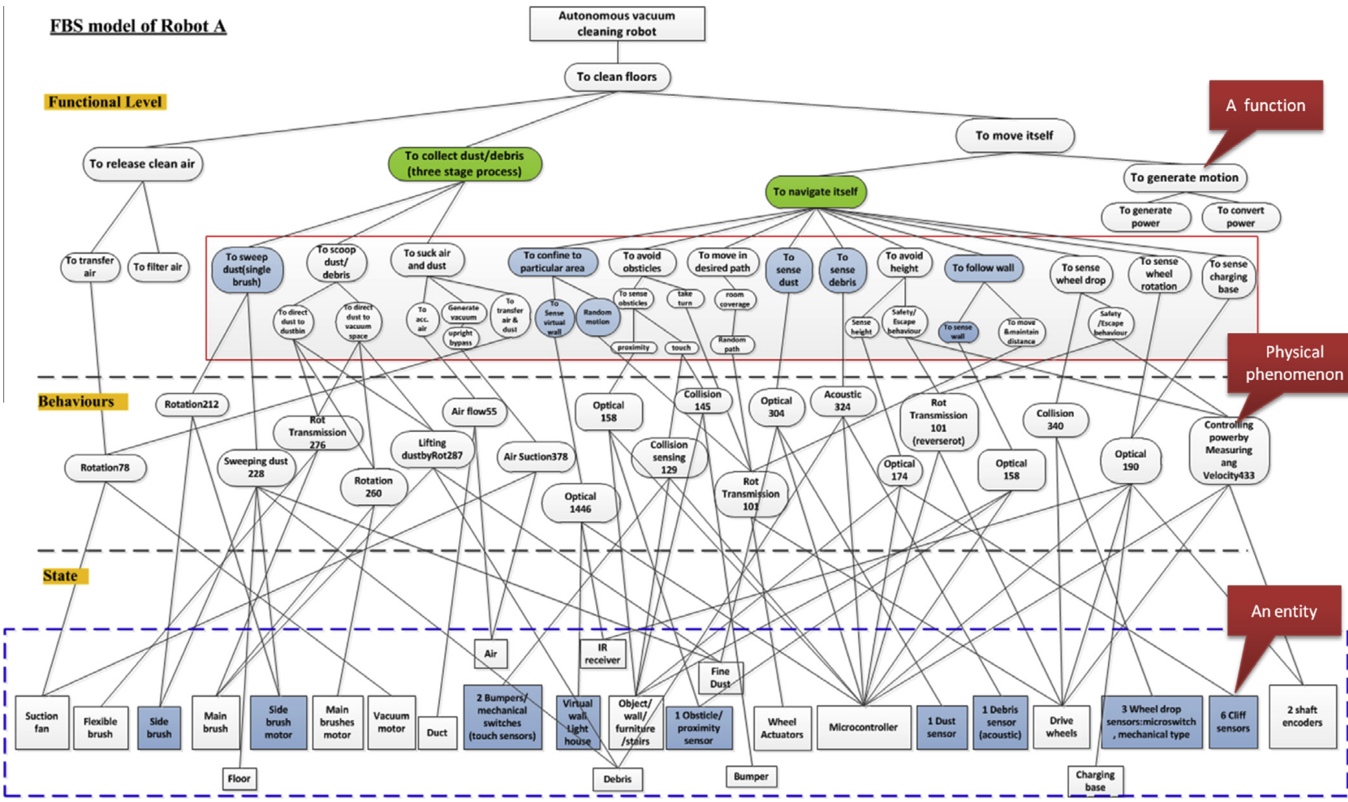


Fig. 5. FBS model of robot System A. Coloured boxes represents the differences in functions and entities in the FBS models.

HasRelations	Has Relation (?contact, ?fan, ?vacmotor), Has-Relation (?contact, ?vacmotor, ?fan)

Attribute: An attribute is a concept attached to an entity and takes a value to indicate the state of the entity. It is defined with its names, supers and statements. In addition to the two definitions (i.e. names and supers), 'statements' describe additional information such as the dimension of the attribute and its definitional relations with other attributes. The following is an example of an attribute:

Name	AngularVelocity (?a)
Supers	Attribute (?a)
Statements	DifferentialOf (?a, AngDisplacement, Time), UnitOf (?a, ?, 'm/s'), DimensionOf (?a, (Lt), (10–20))

Physical phenomenon: A physical phenomenon indicates physical laws or rules that govern behaviours. In many cases, a physical phenomenon represents the relationship among attributes of entities, and 'attributes' define the related entities and attributes of the defined phenomenon. For a complex physical phenomenon that is difficult to represent as a relationship among attributes, it is defined by using only 'entities' information. 'Statements' describe the relationships among entities, attributes and the physical phenomenon. In statements, the 'OccurTo' predicate describes the relationship between the phenomenon and related entities. The 'HasAttribute' predicate describes the relationship among related attributes and related entities. The following is an example of a physical phenomenon:

Name	Rotation (?p)
Supers	Motion (?r)
Attributes	Torque (?t), AngularVelocity (?angvel)
Entities	Mass (?fan), Mass (?vacmotor)
Physical Laws	SecondLawOfNewtonLaws (?f, ?m, ?acc)
Statements	OccurTo (?p, ?fan, ?vacmotor), HasAttribute (?t, ?vacmotor), HasAttribute(?angvel, ?fan), HasAttribute (?angvel, ?vacmotor)

Physical law: A physical law illustrates a simple relationship among attributes. 'Name' and 'attributes' define the name and related attributes, respectively, of the defined physical law. 'Expression' defines the relationship among attributes by using a mathematical equation.

Name	SecondLawOfNewtonsLaw
Attributes	f_Force, m_Mass, a_Acceleration
Expression	Sigma (f) = m * a

Behaviour: Behaviour is defined by sequential changes of states of a physical structure over time. For example, the behaviour of a fan depends on the torque generated by the vacuum motor.

State: States are the different modes of a physical system or entity. Changes in these modes are the underlying causes of behaviours. In the VC robot example, 'rotation' is physical phenomenon between a vacuum motor and a fan, and it is caused by the 'torque generation' of the motor; rotation depends on the state transition of 'torque' of the vacuum motor.

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

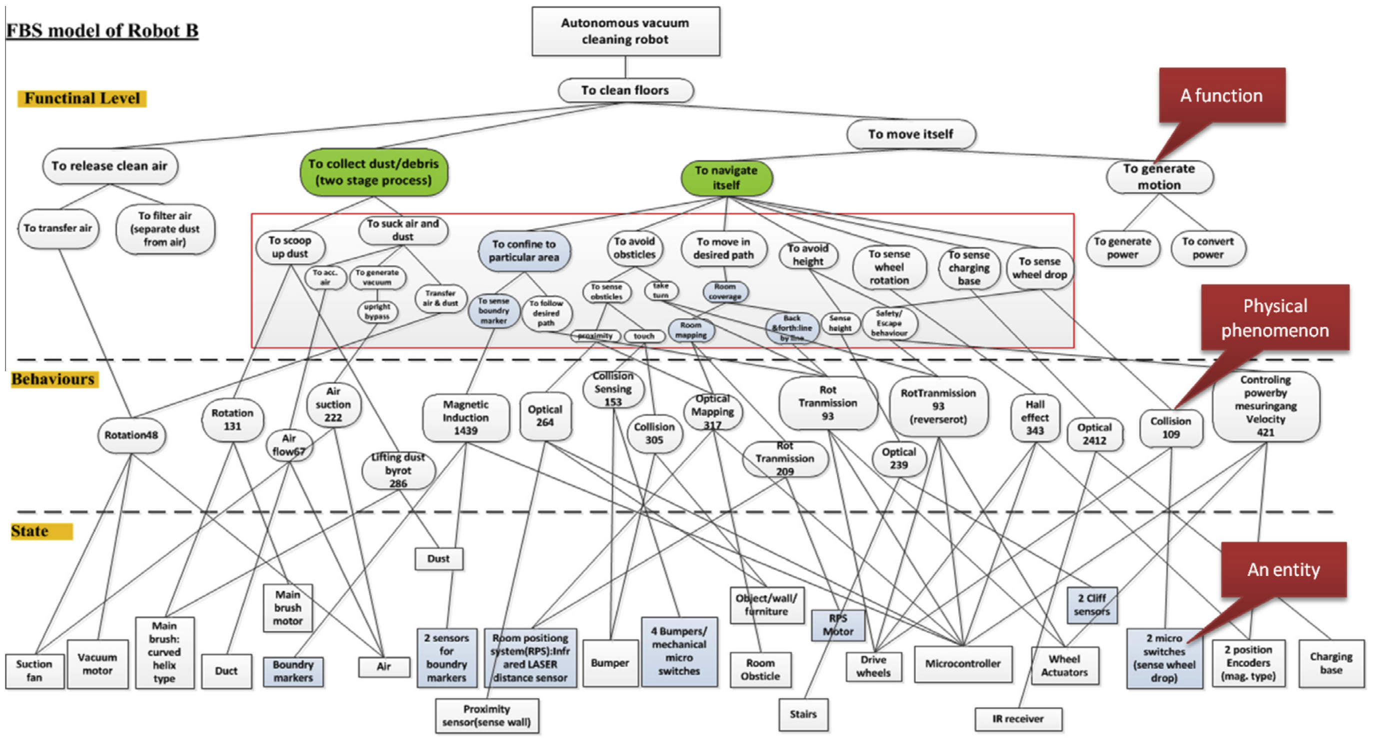


Fig. 6. FBS model of Robot B. The coloured boxes represents the differences in functions and entities in the FBS models.

4.4.2. System architecting CAD tool (SA-CAD)

The design of mechatronic systems requires not only integration of technologies but also cooperation between design teams, especially in the early phases of design. To achieve integration, a holistic approach is required for the design of mechatronic systems, which considers interactions and interrelations among design domains; tools are needed to support such an approach to meet the abovementioned requirements [24]. To support the

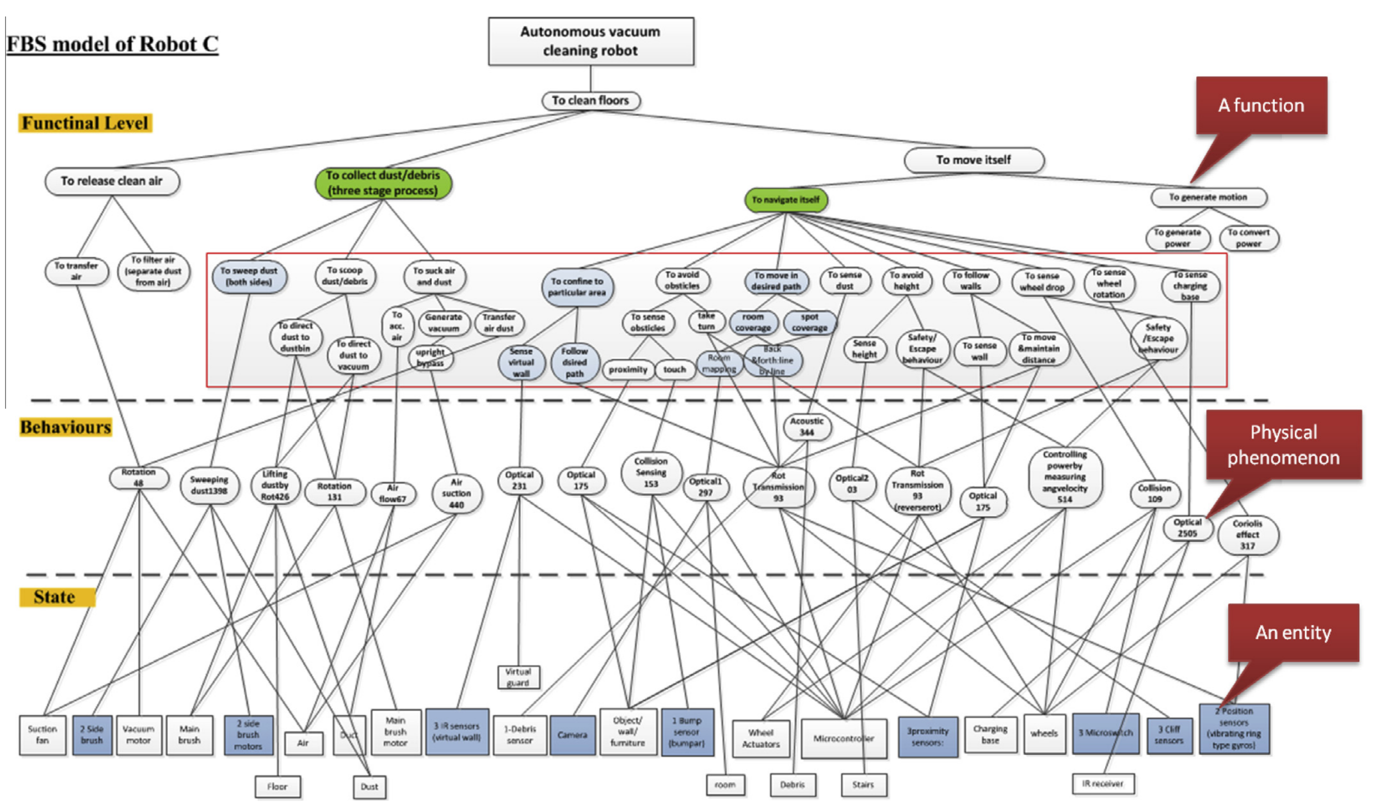


Fig. 7. FBS model of Robot C. The coloured boxes represents the differences in functions and entities in the FBS models.

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

system decomposition tasks for mechatronic systems (such as system architecting, interaction and interrelations among design domains) and to support configuration tasks, a CAD tool (SA-CAD) is used. The architecture of this tool consists of a parameter network modeller, an FBS modeller, a geometric modeller and a process modeller, as shown in Fig. 9. It is integrated in a physical ontology-based design support system called the KIEF.

In SA-CAD, the model of a product consists of its *metamodel* (which is the conceptual network of a product in terms of its function, behaviour and structure) and *parameter network* (Fig. 9). In the parameter network, the behaviour and structure are described using a set of physical phenomena, entities and attributes.

4.4.3. Knowledge base development in SA-CAD

In SA-CAD, the parameter network modeller supports the development of the parameter network of a product, which represents the parameters and their relations. In this tool, a knowledge base of the respective systems (i.e. VC robots) is developed by entering the physical phenomenon and the attributes of the system. For instance, in Fig. 10, the *PP* of *Hall effect* is developed by entering the respective attributes, entities and the relations between the entities.

Table 1
Differences in functions of three robots (systems).

Function	Sub-functions		Robot A	Robot B	Robot C
To navigate itself	To sense dust		×		
	To sense debris		×		×
	To follow walls		×		×
	Room coverage	Back and forth: line by line		×	×
		Room mapping		×	×
		Random path	×		
	To confine to a particular area	To sense virtual wall	×		×
		To sense boundary markers		×	
To collect dust and debris	To sweep dust (one side)		×		
	To sweep dust (both sides)				×

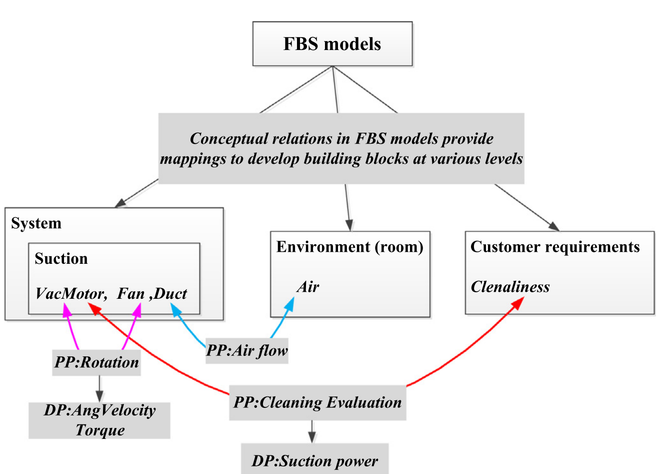


Fig. 8. Example of building blocks and their parameters for network modelling.

4.4.4. Development of product architecture in SA-CAD

With the building blocks and knowledge base, each product is modelled by hierarchical physical decompositions at various levels (i.e. functional, behavioural and structural) in SA-CAD (Fig. 11a). The interface of the tool is used to select the physical phenomena (*PP*), the related parameters and the decomposition candidates to develop the architecture by selecting the system entities and their relations (Fig. 11b).

The designer identifies essential customer needs (i.e. *cleanliness*, *quietness*, *saving energy and convenience in charging*) in VC robots, and connects them to entities in the environment (i.e. room) and system. For instance, *cleanliness* is the customer perception with regard to the ability of the system to clean floors. Its *PP* can be *cleaning evaluation* that is linked to entities such as vac motor and dust. The respective design parameters are suction power in the vacuum motor and the collection of dust or debris by the system. Others functions are also linked in this manner. The system architect defines *quietness* as a function of the angular velocity of the fan and the position of the fan in a fan-duct design. Similarly, as most of the energy is consumed by the system motors, *saving energy* is associated with the power consumed by the motors. Greater energy is consumed in three entities—the vacuum motor, the main brush motor and the wheel actuators; hence, energy consumption is related to these three entities within the system. In addition, another function *convenience in terms of charging* is relevant to entities such as the charging base and receiver of the system. In all three systems, the physical phenomena and their attributes relevant to customer needs are the same. Further decomposition relevant to customer needs, such as the size, small weight and shape of the system, can also be introduced in the process.

The architecture of the system is formed as a network of parameters that are related to various entities or building blocks realised by the physical phenomena (Fig. 12). For instance, in the suction module, three entities—the vacuum motor, fan and duct—are formed by the parameters using the *PP*, such as *rotation*, *air suction* and *air flow*, and they are related to sub-functions, such as *accelerating the air*, *upright bypass* and *transferring air and dust*. The system entities are related to each other through parameters; for instance, the fan and the vacuum motor represent a *PP*, i.e. *rotation*, with the parameters being angular velocity and power. These parameters are the interfaces between the entities, and they link the system to the room environment. In the suction module, vacuum generated by the fan because of the change in pressure forces the air to flow inside the system. That is why the system architect uses *air suction* as *PP* to connect the pressure of the fan to the force of the air.

Figs. 13a, 15a and 17a show the correspondences between the entities and the parameter relations that link the behaviours of the systems with their respective structures. They describe the design concept developed at the conceptual design stage. Figs. 13b, 15b and 17b show relations among the parameters of entities. The tool helps designers construct such matrix representations for the analysis of system architecture with matrix-based methods (e.g. DSM), and such analysis can be used for tasks such as architecture development and process organization.

Further decomposition of the robot system involves an interface with the environment. Various entities are connected to the system; for instance, the *PP* of *lifting dust by rotation* means that the main brush is rotating to lift the dust from a certain position, and the amount of dust also depends on the length of the brush. Sensor–actuator combinations and their relations are relevant from the viewpoint of control design. For example, in the wheel assembly module, the shaft encoder measures the angular velocity of the wheel or wheel actuator; hence, it controls the desired power of the wheel

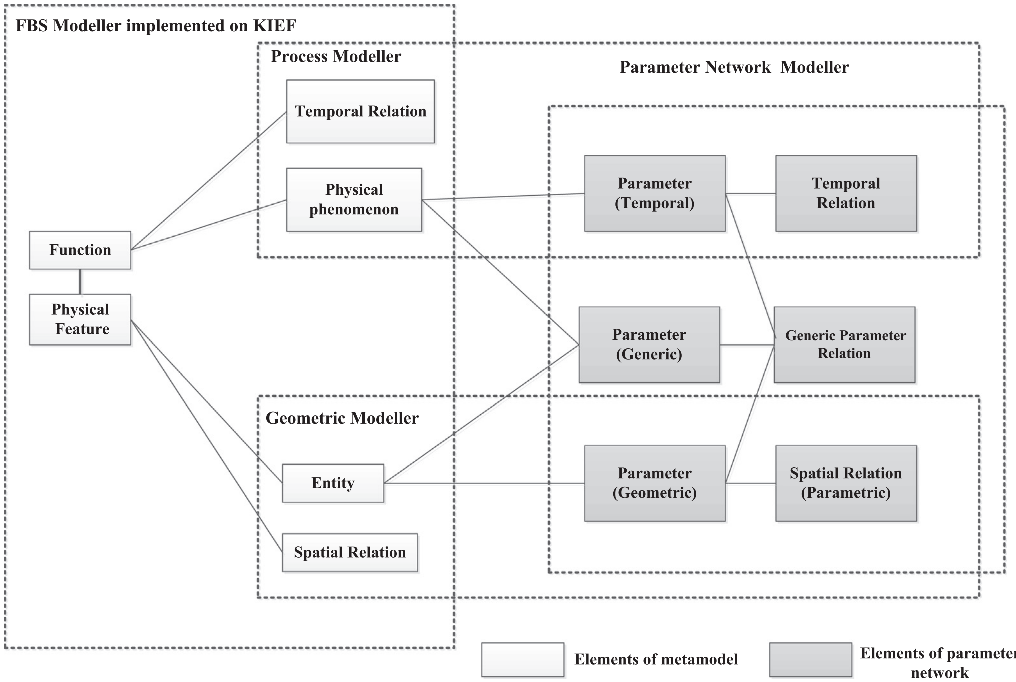


Fig. 9. Architecture of SA-CAD and details of each modeller for model development. These modellers are the parameter network modeller, an FBS modeller, a geometric modeller and a process modeller [25].

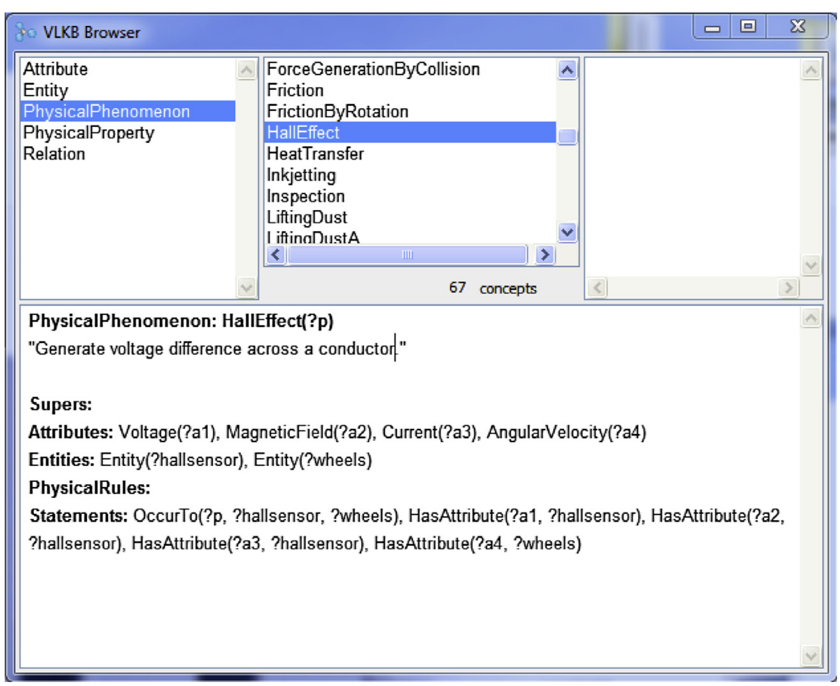


Fig. 10. Example of knowledge base development in SA-CAD.

actuator. The system architect defines the sensor–actuator combinations and their relations through parameters.

The parameter network represents the architecture of a system (Figs. 12, 14 and 16). Matrices relating physical phenomena, entities and attributes of the respective systems are shown in Figs. 13, 15 and 17, respectively.

The designer uses this process to decompose the system specifications into subsystems and components until all the functions of the system have been realised.

4.5. Development of product platforms for next generation products

4.5.1. Commonality and differentiation analysis of multiproduct system architectures

The parameter network is essentially the product architecture that is to be analysed for a set of modules, using which product variants can be created. Common modules or standard entities constitute product platforms, while the differentiation in their structures can be used for developing product families. The

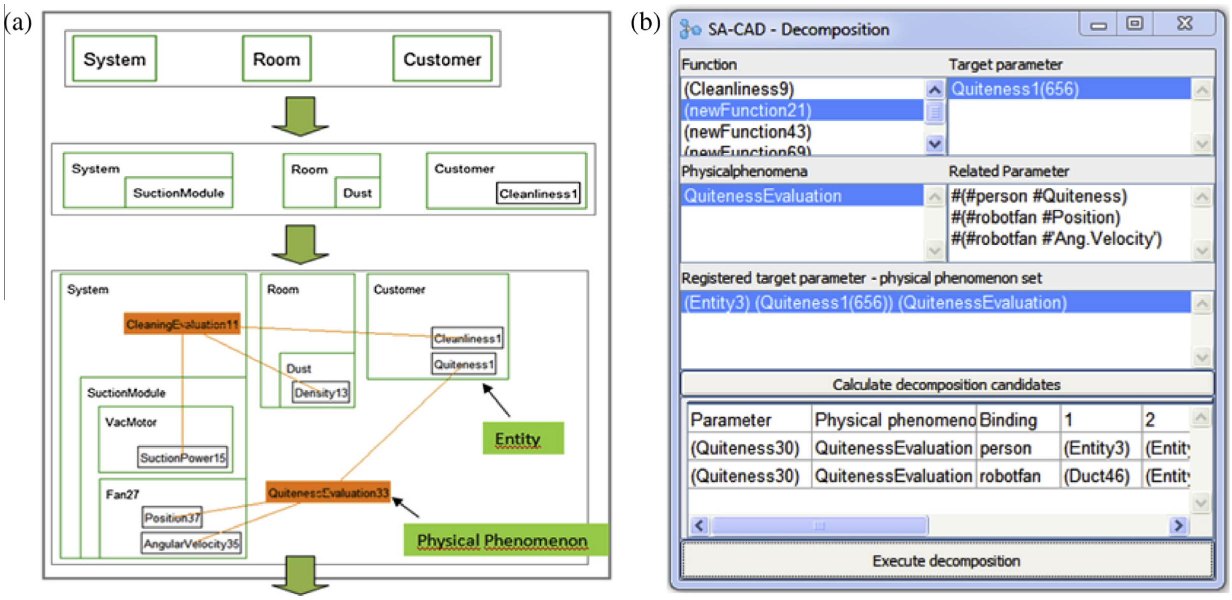


Fig. 11. (a) Hierarchical system decomposition process. (b) Decomposition interface in SA-CAD.

System A

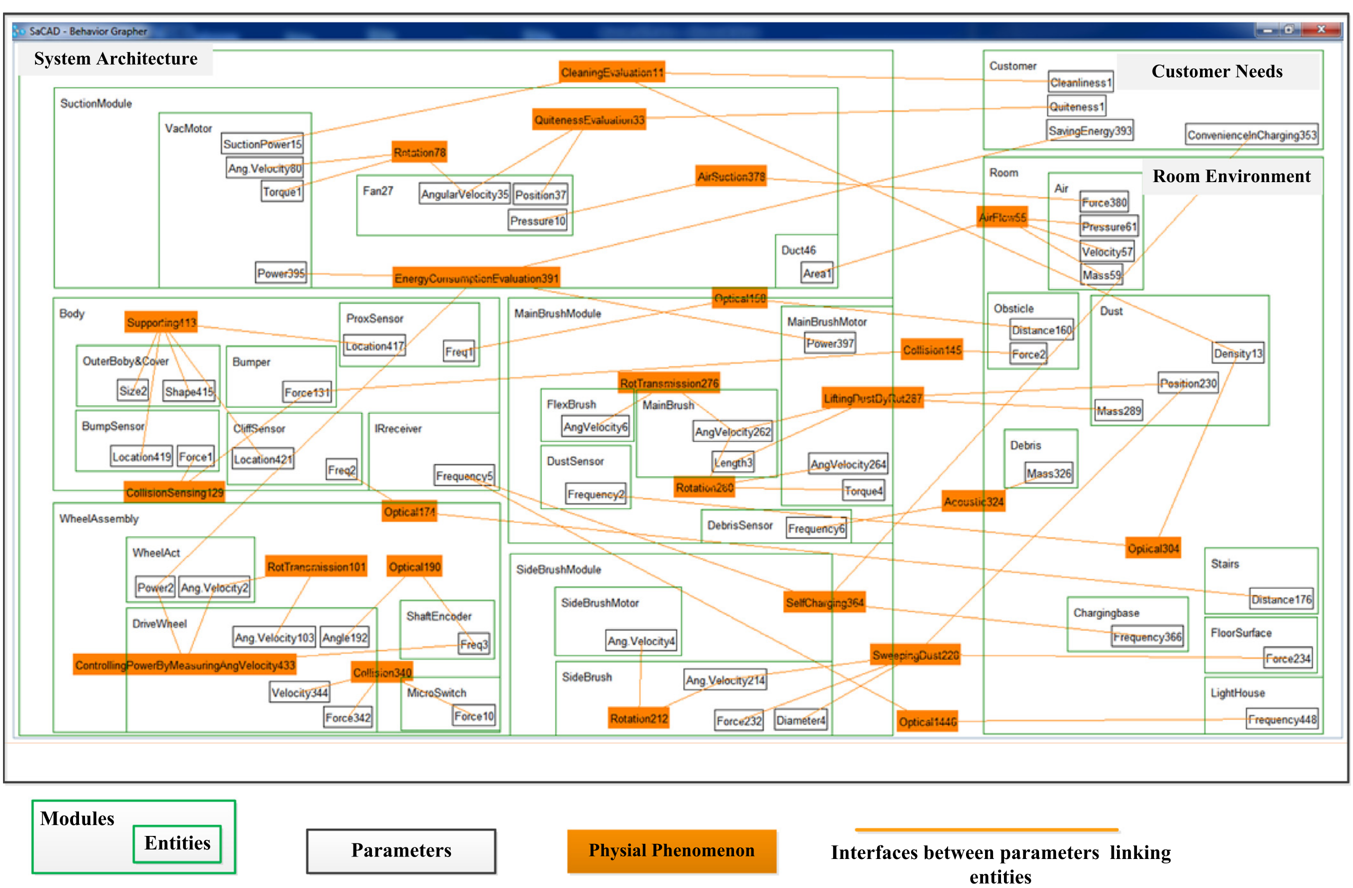


Fig. 12. Parameter network of 'System A' relating physical phenomena, entities and parameters or attributes.

designer can analyse systems to determine an optimal module structure to minimize life cycle costs and maximize common parts in a multiproduct family and modular interactions to attain high reliability. Furthermore, the system architect can redesign the architecture by introducing new modules or changing the existing structure, which is supported by SA-CAD.

SA-CAD is capable of performing system decomposition in the case of large networks (or highly detailed designs).

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

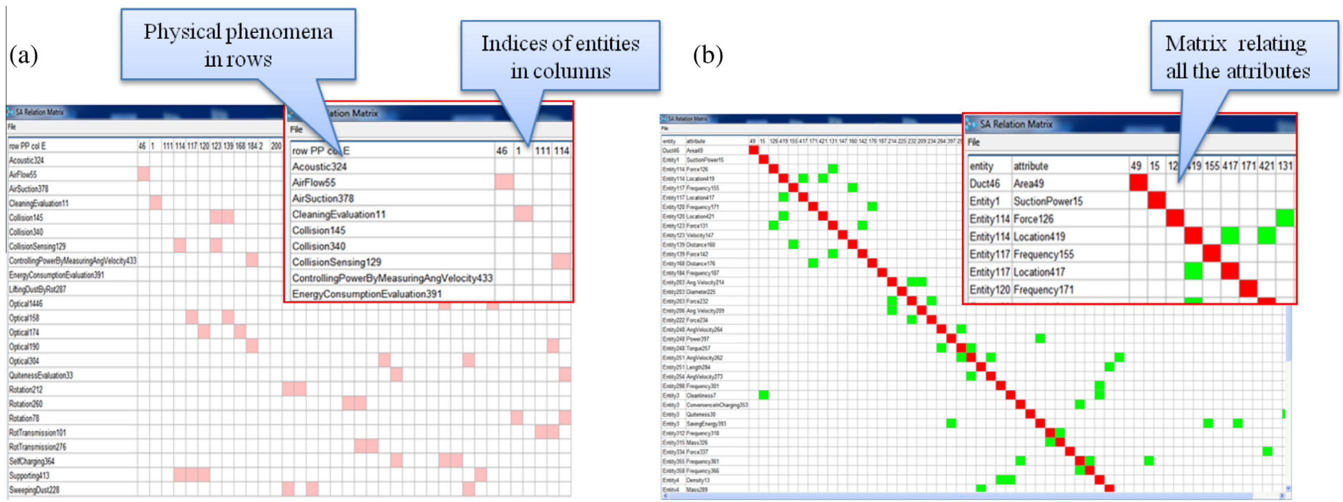


Fig. 13. (a) Correspondences between entities and parameter relations. These relations are derived from physical phenomena, and they illustrate the differences in these systems at behavioural levels. (b) Matrix representation of relations among parameters.

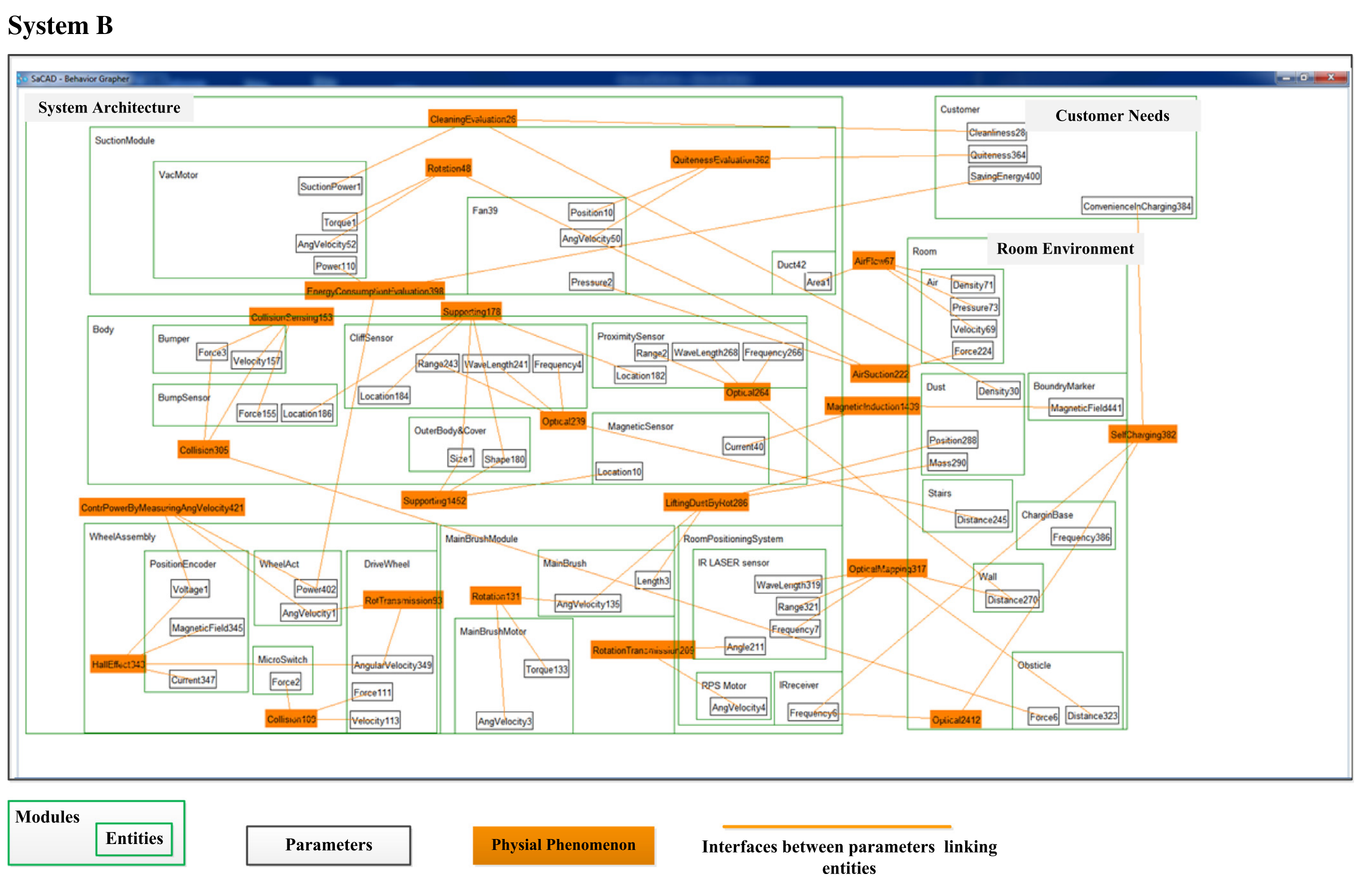


Fig. 14. Parameter network of 'System B' relating physical phenomena, entities and attributes.

However, in highly detailed designs, in the case of a single decomposition, when more entities are linked to more parameters, decomposition candidates becomes very large in the decomposition interface (Fig. 11b), and occasionally, more time is required to select the relevant entities for network modelling. In the case of detailed design with more product variants and components, the focus should be on the identification of those building blocks (and their physical phenomena) that can be used for differentiation in their architectures. After the initial analysis in the proposed method, it becomes relatively easy to decide which functions do not need further decomposition to develop building blocks. For example, in VC robots, functions such as *to generate motion* and *to release clean air* are not considered for further decomposition to simplify the architecture, because all three systems have the same functional and structural considerations.

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

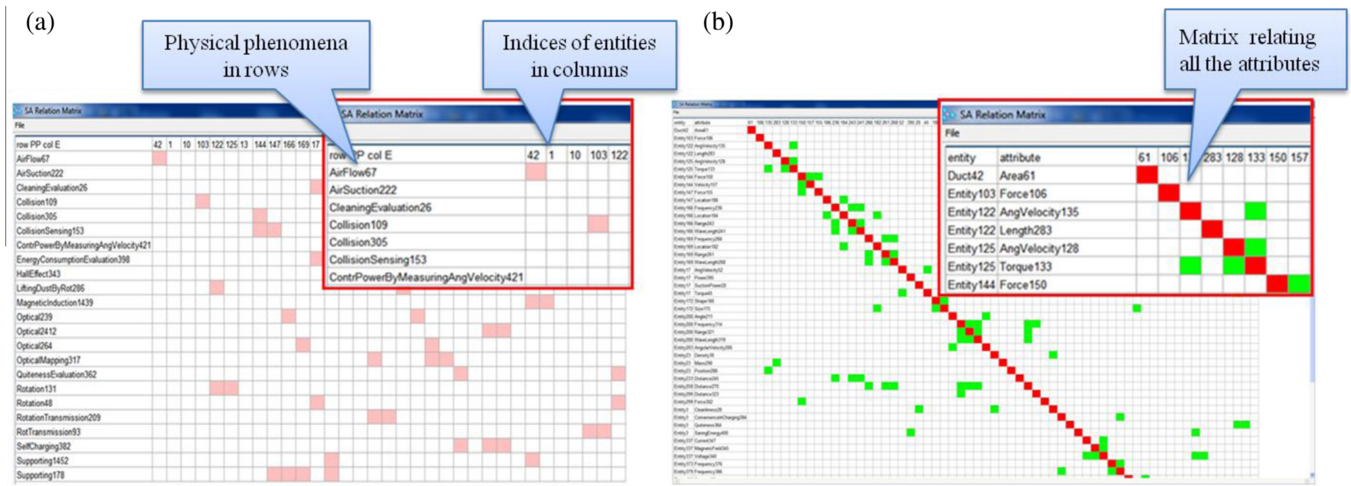


Fig. 15. (a) Correspondences between entities (*E*) and parameter relations. These relations are derived from physical phenomena, which illustrate the differences in these systems at behavioural levels. (b) Matrix representation of relations among parameters.

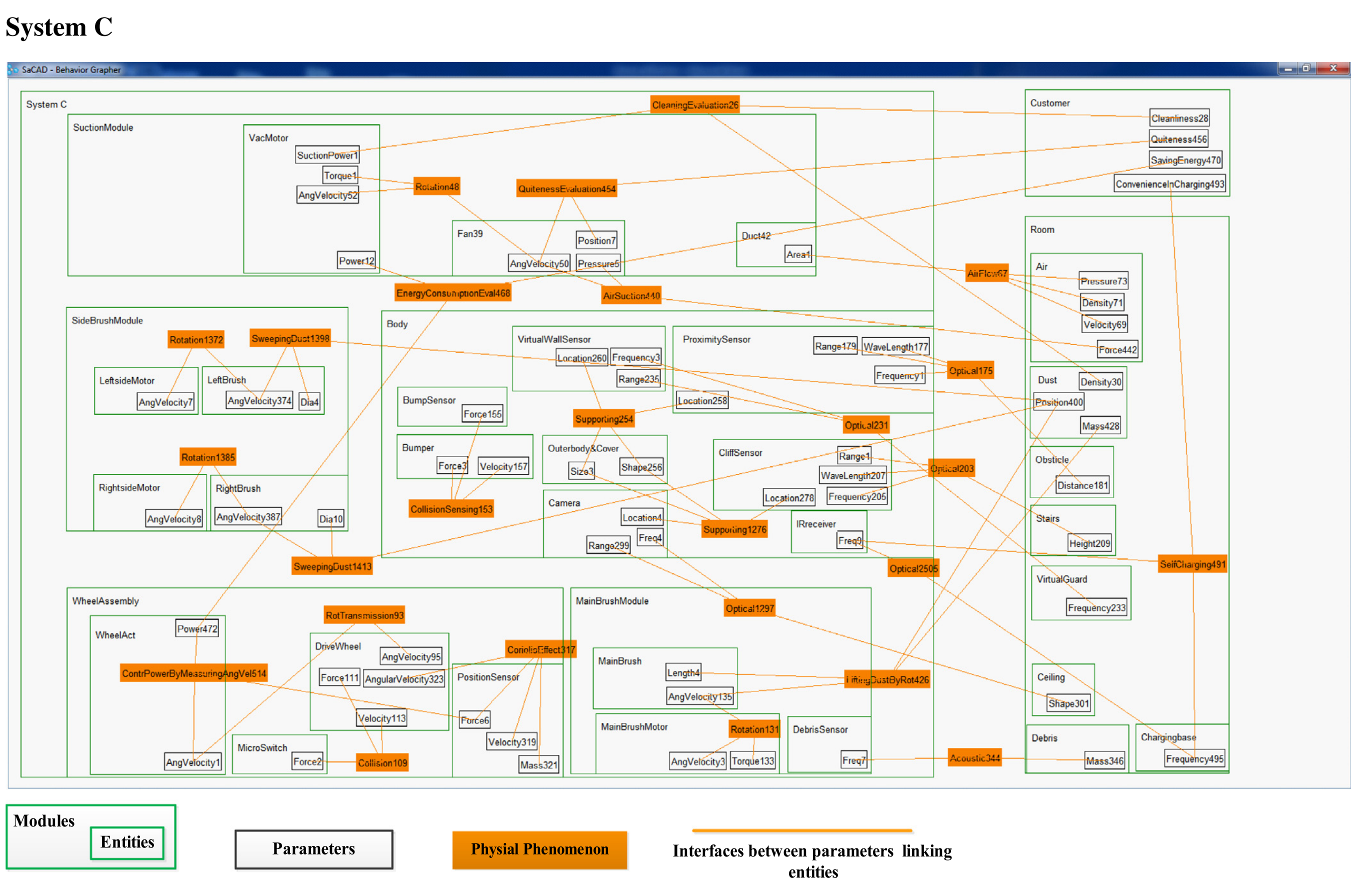


Fig. 16. Parameter network of 'System C' relating physical phenomena, entities and attributes.

4.5.2. Identification of commonality instances in architectures of the three systems

The commonality of the three systems is identified manually in the parameter network to determine common building blocks or entities. In the parameter network of the three robots, the building blocks and the associated physical phenomena in the suction system and the wheel assembly are approximately the same. In the body module, building blocks such as bumpers, bump sensors, proximity sensors and cliff sensors are common entities in all three systems. Their physical phenomena and design parameters are also the same. The only difference is the number of these sensors and their location. For instance, System A uses six, System B uses two and System C uses three cliff sensors. Once the designer defines the size of the system, their number and location are based on the requirements of their different designs; this can be a differentiation in their architectures and can be used for developing product families. Similarly, in the main brush module, the main brush and main brush motor are common entities in all three systems. The side brush module is common only to Systems A and C.

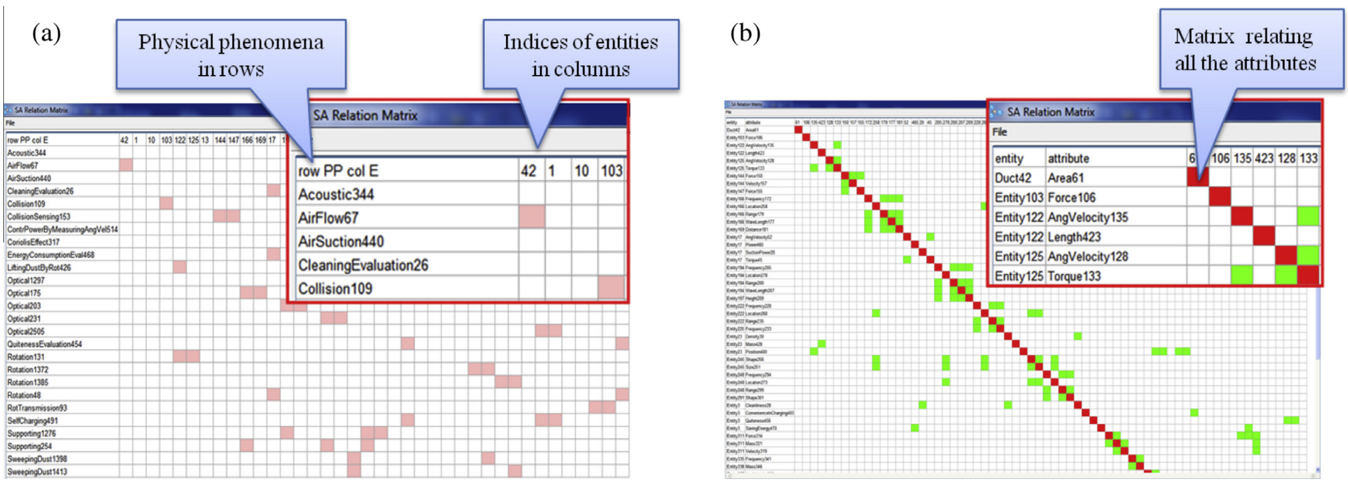


Fig. 17. (a) Correspondences between entities (*E*) and parameter relations. These relations are derived from physical phenomena, which illustrate the differences in these systems at behavioural levels. (b) Matrix representation of relations among parameters.

Table 2
Differentiation in architecture of the three systems.

		System A	System B	System C
1	Main brush module	Two counter-rotating brushes	Single rotating brush	Single rotating brush
		Dust sensor		
		Debris sensor		Debris sensor
2	Side brush	Single side brush		Two side brushes
		Single motor		Two motors
3	Body	IR receiver		IR receiver
				Virtual wall sensor
		One proximity sensor	One proximity sensor	Three proximity sensors
				Camera
			Magnetic sensor	
4	Room mapping system		IR laser sensor	
			RPS motor	
			IR receiver	

4.5.3. Differentiation in architecture of the three systems

The differentiation in the architectures of the three systems in terms of entities is summarized in Table 2.

The system architect observes the functions and their physical realisation. For implementing functions such as *scoop up dust*, rotating brushes are used. System A uses two counter-rotating brushes (instead of one) to do this job effectively. However, from a design perspective, the addition of an extra entity and related transmission elements results in a trade-off between efficient lifting and energy consumption, and the designer must consider such issues while performing the system architecting tasks.

All three architectures are different in terms of implementation of functions such as *spot cleaning* and *to confine to a particular area*. For instance, Systems A and C use infrared (IR) sensors and light houses to implement these functions. The designer can decide on the number of IR sensors and their locations for achieving better performance. For *spot cleaning*, System C uses magnetic sensors for sensing magnetic markers that confine the system to a particular area. This is one of the differences between these systems. Sensing dust and debris is an important feature of autonomous VC robots; System A uses optical sensors for sensing dust and acoustic sensors for locating debris, while Systems B and C use no such sensors, which is a potential differentiation in the architecture of the proposed systems.

The differentiation in the architecture can be used for developing product families. This differentiation is evident in functions such as *to collect dust and debris* and *to navigate itself*, as well as in the building blocks represented in Table 2. Because of system decomposition into building blocks and the differentiation in architectures, designers can identify and develop product families even in the conceptual phase of multidisciplinary mechatronic systems.

4.5.4. Product platform development

Commonality instances in the three architectures, on the basis of common modules or entities, may lead to the development of product platforms. Product platforms can offer the following benefits when applied successfully: companies can develop differentiated products by sharing components across a product platform, they can reduce development time and cost and system complexity and they can acquire the ability to upgrade and redesign products. To utilize the advantages of platform development, standard entities and differentiation entities must be balanced inside a modular architecture. The aim must be to maximize the use of common modules in the architecture while maintaining their distinctiveness, as shown in Fig. 18.

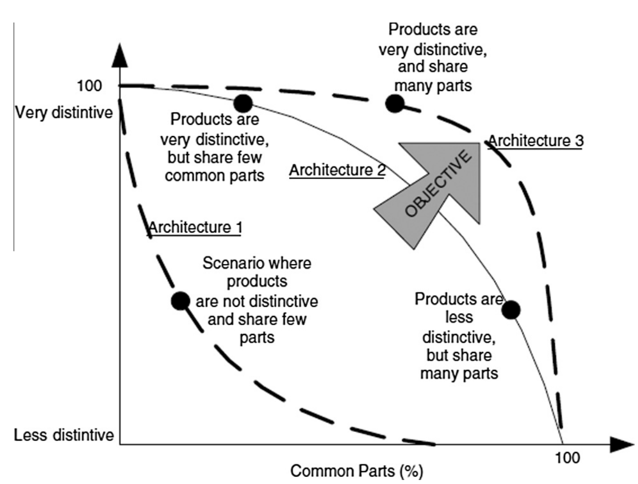


Fig. 18. Trade-off between distinctiveness and commonality, which depends on the architecture characteristics [27].

Please cite this article in press as: Habib T, Komoto H. Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics (2014), http://dx.doi.org/10.1016/j.mechatronics.2014.03.003

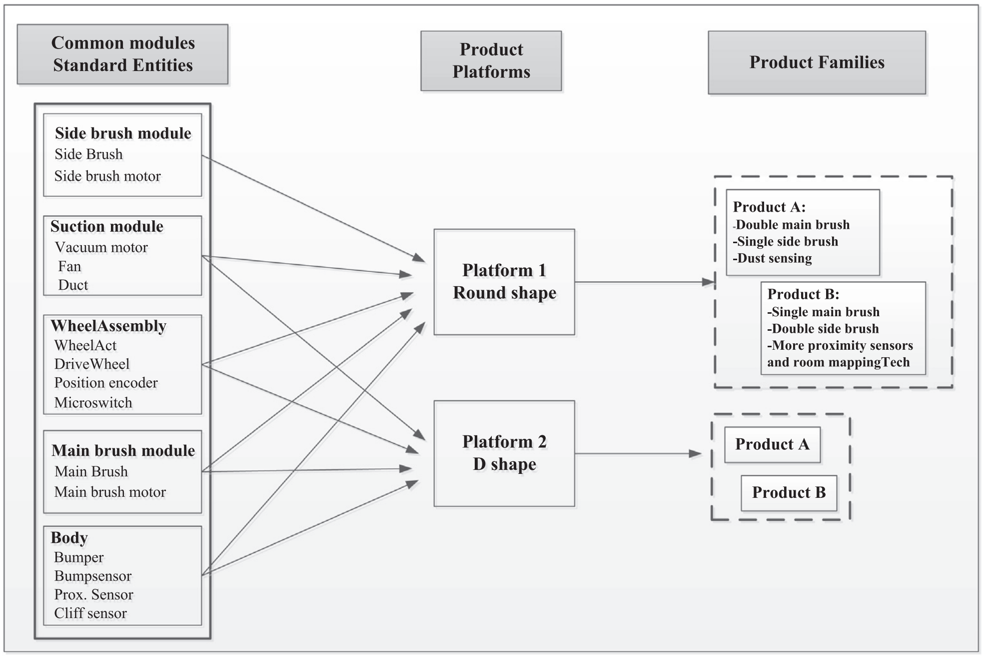


Fig. 19. Product platforms based on common modules of the three architectures.

In the proposed case study, two platforms are developed that comprise common entities based on the round design and D-shaped design of the robots (Fig. 19). These platforms are further extended to product families, where the derivative products can be developed from the three architectures (Fig. 20). For instance, the number and type of sensors and entities relevant to the function, i.e. *to collect dust and debris*, can be the differentiating factors in the three systems, and they also determine the cost of the products.

In short, one of the basic requirements for the development of product platforms is using a maximum number of standard components, which (along with minimal architecture changes) allows the development of differentiated products. The objective is to find components that can be re-used between products in a way that provides flexibility to respond to changing market needs.

5. Optimal module structure and design optimization

On the basis of system decomposition, the system architects and domain experts are able to make the following decisions:

- Selection of the optimal module structure that enables the minimization of life cycle costs.
- Redesigning of the system based on physical phenomena.

Once the FBS model of the system has been developed, it can be redesigned for functional improvements. One way to redesign is to add and remove functions in the FBS model. For instance, System A can be redesigned by adding a function *to sweep dust* along with its corresponding physical phenomena (rotation, sweeping dust) and its physical realisation in the form of entities (side brush and side brush motor) at the state level, as shown in Fig. 5. Sensing of fine dust particles is an added functionality that is part of System A, but not used in System B. Another way to redesign is to change the values of design parameters such that the desired performance can be achieved. For instance, one way to increase the sweeping area is to replace components with more suitable ones (e.g. replacing a brush with one having a large diameter). The design parameters are linked to the respective entities in the parameter network.

In the suction module, the structure and behaviours described using building blocks related by parameter relations are the same in all architectures. For instance, the motor power and air pressure for generating air suction are one of the important characteristics governing the performance of the VC robot. All VC robots operate on air flowing from the opening in the cleaning head; passing through the duct, dustbin and filter system; and then flowing out of the exhaust port. The vacuum motor consists of electrical components attached to a fan. When the fan spins, a partial vacuum is created, and the pressure inside the VC robot drops below the ambient air pressure in the room. Because the air pressure is higher outside the VC robot than inside it, air rushes through the VC robot. It is the force of this *airflow* across a surface (duct) that helps the robot pick up the dirt and move it to the dustbin. Therefore, the stronger the *airflow*, the better is the cleaning ability of the vacuum cleaner. For this reason, the designer uses *air suction* as a *PP* related to the pressure generated by the fan to force ambient air through the robot. The fan's design parameters are further related to the vacuum motor by using *rotation* as a *PP*. For this reason, the vacuum motor is one of the main components of a VC robot. After all, the more powerful the motor, the greater is the pressure

Differentiation in entities in the three architectures	Platform 1		Platform 2	
	Product A	Product B	Product A	Product B
Two counter-rotating main brushes	(shaded)			
Single rotating main brush		(shaded)	(shaded)	(shaded)
Single side brush	(shaded)			
Two side brushes		(shaded)		
Dust sensor	(shaded)			
Debris sensor	(shaded)	(shaded)		
Virtual wall sensor		(shaded)		
Single proximity sensor	(shaded)		(shaded)	(shaded)
Three proximity sensors		(shaded)		
IR receiver	(shaded)	(shaded)		
Camera		(shaded)		
Single magnetic sensor			(shaded)	
Two magnetic sensors				(shaded)
Room positioning system			(shaded)	(shaded)

Fig. 20. Development of product families based on two platforms using the differentiation in the three architectures.

differential, and correspondingly, the higher are the suction and airflow. It is for this reason that the specifications concerning *cleaning ability* relate to the suction power of the vacuum motor and the density of the collected dust. Moreover, the designer can further decompose the system with the physical phenomena and their parameters (such as the type of fan, its location (bypass or direct), the duct design (based on geometric and thermodynamic aspects), air filtration (filter size and location, resistance to airflow and the storage capacity of the dustbin) and all these design aspects are potential candidates for further decomposition.

In a modular structure, the selection and location of sensors are important considerations for enhancing performance. In the wheel assembly, all building blocks and their design parameters are the same; the only difference is the type of sensor used. For example, to sense wheel rotation, System A uses the optical principle, System B uses the Hall effect and System C uses the Coriolis effect as physical phenomena. The decision about sensor selection can be based on the merits and cost of each type of sensor and the opinion of domain experts. For autonomous cleaning, the robot must be able to sense dust and debris and confine itself to clean that area first, before moving to other places. In this situation, the decision about sensor location is influenced by how effectively the system responds to the amount of dust. One possibility is to respond directly to the amount of dust (according to the air passage) by fixing sensors behind the main brush. For achieving better performance, dust and debris sensors must be a part of the system, as is the case of architecture A.

From a design perspective, the shape of the robot is important for two reasons: (1) to be able to reach corners and (2) to be able to escape when stuck in a narrow area. From the design perspective, a square-shaped robot can manoeuvre and clean in corners, but the drawback is that the robot is unable to escape from narrow areas. A circular-shaped robot can move in narrow areas, but for it to clean corners and near walls, it requires side brushes. For this reason, side brushes are used in Systems A and C. The designer can either increase the diameter of the side brush or extend the side brush outwards from the device to promote corner cleaning. These design aspects can be implemented through system decomposition using FBS modelling, as demonstrated in the three architectures.

The proposed method illustrates how the system architect uses FBS modelling to identify the parameters of entities, i.e. building blocks, to develop product architecture. Multiple system architectures were developed in SA-CAD to explore the possibility of developing product platforms and modelling product families that can be used for next generation products. Commonality instances in the three architectures (on the basis of the common modules or entities) were identified to develop product platforms, because these platforms may offer benefits when applied successfully. These benefits are that companies can develop differentiated products by sharing components across a product platform, reduce development time and system complexity and acquire the ability to upgrade and redesign products. Furthermore, the process of system modelling enables system designers to establish appropriate component-level specifications to communicate with domain experts. As a result, system modelling provides a solution to the challenge of cooperation and communication among design engineers in different domains.

6. Limitations and opportunities for further work

During the study, several limitations have been observed regarding the comparative analysis method as well as the technical capability of the employed implementation used to present the method (i.e. SA-CAD).

First, in the case study, three existing products have been used for the comparative analysis. It is desired to include a larger number of products within the scope of comparative analysis. By doing so, it is expected to obtain a wider variety of unique building blocks for the development of design concepts.

Second, as the case study has shown, the number of nodes such as entities, parameters, and their relations becomes very large along with decomposition of the function. Thus, visualization supports to classify these nodes with various hierarchical levels might be useful. In particular, automated procedures to highlight (and hide) nodes relevant to a specific function node are useful to clarify the interest of the designer in terms of the state of function decomposition.

Third, this method supports the designers to develop design concepts by identifying the design parameters that characterize each design concept and common modules in existing systems under comparative analysis. However, the method cannot support quantitative analysis of the performance of design concepts based on numerical simulations. Integration of quantitative and qualitative information is the major topic of further research.

7. Summary and conclusions

In the early stage of mechatronics systems design, designers and engineers have the crucial task of designing and optimizing modules and their interfaces by analysing the system architecture of existing products of the same type. Computational support is crucial for the effective execution of this task. This paper proposed a method for supporting this task using FBS modelling and SA-CAD. The proposed method was validated and its applicability was demonstrated using a case study, in which the method was applied to a comparative analysis of architectures of three autonomous VC robots.

In this paper, the system decomposition process, which is the key concept of the proposed method, was divided into four processes. First, customer requirements were translated into system requirements. Second, system-level specifications were transformed into subsystems and components. Third, the physical and logical configurations of subsystems and components that realised the desired functions and behaviours were defined. Fourth, the system performance, for example, in terms of functionality, was analysed. In these processes, the intended interactions between the components and the technologies from different mechatronic domains were identified. The interactions were described using the design parameters that link the building blocks (which correspond to specific components and technologies) through physical phenomena. In addition, the decomposition process identified commonality instances in the product architectures, which can be used for the development of product platforms, and the differentiation in these architectures, which can be used for product family modelling.

This paper presented a comparative analysis of design concepts of mechatronics systems as performed with the SA-CAD tool, which supports system decomposition and modularization considering the dependencies among the parameters of subsystems. Multiple system architectures were developed to develop product platforms and model product families. The commonality of the three considered systems was analysed in the parameter network to determine common building blocks. The proposed method could identify several levels of classifications regarding the level of commonality observed in existing systems, such as the behavioural level (physical phenomena) and the structural level (type of components, i.e. entities, used and their numbers and locations). In the case study, it was observed that the suction system and the wheel assembly were common in all three analysed autonomous

VC robots regarding physical structure and behaviour. In the body module, all robots exhibit common behaviour because of common components such as bumpers, bump sensors, proximity sensors and cliff sensors. The robots differ slightly in terms of the number of common components and their location.

The commonality instances and differences identified by such classifications were used for the development of product platforms. In this case study, two platforms were developed, which were based on the round design and D-shaped design of the robots. These platforms were further extended to model product families, where the derivative products were developed on the basis of the variation of the existing architectures. Variation was identified mainly in terms of physical entities such as the main brush, side brush and the number and type of sensors. Using the proposed method, designers can identify the maximum number of standard components required, along with implementing minimal architecture changes. The proposed method would enable the development of several products in a product family and identification of components that are shared by products in a product family to provide designers with flexibility to deal with drastically changing market needs.

Acknowledgments

This research was partially supported by the Otto Monsted Fund and Aalborg University, Denmark. The authors are grateful to Prof. Kaj A. Joergensen and Prof. Shaoping Bhai for their valuable ideas and support in selecting the case study.

References

[1] Hehenberger P, Follmer M, Geirhofer R, Zeman K. Model based system design of annealing simulators. Mechatronics 2013. http://dx.doi.org/10.1016/j.mechatronics.2012.12.001.
[2] Pahl G, Beitz W. Engineering design: a systematic approach. London: Springer; 1996.
[3] Baldwin CY, Clark KB. Modularity in the design of complex engineering systems. Understanding complex systems. Heidelberg: Springer Berlin; 2006. p. 175–205.
[4] Hehenberger P. Application of mechatronic CAD in the product development process. Comput.-Aided Des. Appl. 2009;6(2):269–79. http://dx.doi.org/10.3722/cadaps.2009.269-279.
[5] Isermann R. Mechatronic systems fundamentals. Berlin, Germany: Springer Publishing Group; 2002.
[6] De Silva CW. Mechatronics: an integrated approach. Boca Raton: CRC Press; 2005.
[7] Chmarra MK, Cabrera AAA, van Beek T, D'amelio V, Erden MS, Tomiyama T. Revisiting the divide and conquer strategy to deal with complexity in product design. In: Proceedings, IEEE/ASME international conference on mechatronic and embedded systems and applications, 2008. p. 393–8.
[8] Navinchandra D. Innovative design systems, where are we and where do we go from here, Part I: design by association. Knowl Eng Rev 1992;7(3):183–213.
[9] French MJ. Conceptual design for engineers, design council. 2nd ed. London: Springer Verlag; 1985.
[10] Forsberg K, Mooz H. The relationship of systems engineering to the project cycle. Eng Manage J 1992;4(3):36–43.
[11] VDI 2206. Design handbook 2206. In: Design methodology for mechatronic systems, Düsseldorf: VDI Publishing Group; 2004.
[12] Dieterle W. Mechatronics systems: automotive applications and modern design methodologies. Annu Rev Control 2005;29:273–7.
[13] Komoto H, Tomiyama T. A system architecting tool for mechatronic systems design. Ann CIRP 2010;59(1):171–4.
[14] Tomiyama T, D'Amelio V, Urbanic J, El Maraghy W. Complexity of multidisciplinary design. Ann CIRP 2007;56(1):185–8.
[15] Jiao J, Tseng M. Fundamentals of product architecture. Integrated Manuf Syst 2000;11(7):469–83.
[16] Stone RB, Wood KL, Crawford RH. A heuristic method for identifying modules for product architectures. Des Stud 2000;21(1):5–31.
[17] Stone RB et al. A customer needs motivated conceptual design methodology for product portfolio planning. J Eng Des 2008;19(6):489–514.
[18] Van Wie MJ, Designing product architecture: a systematic method, Ph.D. Thesis, University of Texas, Austin, 2002.
[19] Ulrich KT, Eppinger SD. Product design and development. New York: McGraw-Hill; 2008.
[20] Albers A, Braun A, Sadowski E, Wynm DC, Wyatt D, Clarkson P. System architecture modelling in a software tool based on the contract and channel approach (C&C-A). J Mech Des 2011;133(10).
[21] Umeda Y, Ishii M, Yoshioka M, Tomiyama T. Supporting conceptual design based on the function–behaviour–state modeller. Artif Intell Eng Des Anal Manuf 1996;10(4):275–88.
[22] Ishii M, Sekiya T, Tomiyama T. A very large-scale knowledge base for the knowledge intensive engineering framework. In: Mars NJI, editor. Towards very large knowledge bases. Amsterdam (Oxford, Tokyo, Washington, DC): IOS press; 1995. p. 123–31.
[23] Yoshioka M, Umeda Y, Takeda H, Shimomura Y, Nomaguchi Y, Tomiyama T. Physical concept ontology for the knowledge intensive engineering framework. Adv Eng Inform 2004;18:95–113.
[24] Alvarez Cabrera AA, Foeken MJ, Tekin OA, Woestenenk K, Erden MS, De Schutter B, et al. Towards automation of control software: a review of challenges in mechatronic design. Mechatronics 2010;20:876–86.
[25] Komoto H, Tomiyama T. A framework for computer-aided conceptual design and its application to system architecting of mechatronics products. Comput Aided Des 2012;44:931–46.
[26] ElMaraghy W, ElMaraghy H, Tomiyama T, Monostori L. Complexity in engineering design and manufacturing. CIRP Ann – Manuf Technol 2012;61:793–814.
[27] Robertson D, Ulrich K. Platform product development. Working Paper, The Wharton School, University of Pennsylvania, 1998.
[28] Manuel Sosa, Steven Eppinger, Craig Rowles. Identifying modular and integrative systems and their impact on design team interactions. J Mech Des 2003;125(2):240–52. http://dx.doi.org/10.1115/1.1564074.
[29] Kreimeyer M, Braun S, Gürtler M, Lindemann U. Relating two Domains via a third – an approach to overcome ambiguous attributions using multiple domain matrices. In: ASME 2008 international design engineering technical conferences & computers and information in engineering conference IDETC/CIE, 2008.
[30] Danilovic M, Browning Tyson R. Managing complex product development projects with design structure matrices and domain mapping matrices. Int J Project Manage 2007;25:300–14.
[31] Modelica. A unified object oriented language for physical systems modeling, language specification. Version 3.2. Modelica association, 2010.
[32] 20-Sim, the dynamic modeling and simulation package for iconic diagram, bond graph, block diagram and equation models. Controllab products BV, 2010.
[33] Pimmler TU, Eppinger SD. Integration analysis of product decompositions. Proc ASME Des Theory Methodol Conf 1994;68:343–51.
[34] Huang G, Bin S, Halevi G. Product platform identification and development for mass customization. Ann CIRP 2003;52(1):117–20.
[35] Umeda Y, Takeda H, Tomiyama T, Yoshikawa H. Function, behaviour, and structure. In: Gero JS, editor. Applications of artificial intelligence in engineering V. Southampton/Berlin: Computational Mechanics Publications/Springer–Verlag; 1990. p. 177–93.

Paper 3

Design Models in the Development of Mechatronic Systems: Virtual Prototyping and Mechatronic Module development process.

Habib, Tufail; Thomas Ditlev; Nielsen Kjeld; Jørgensen, K Asbjørn. Submitted to International Journal of Design Engineering, 2014 (under review).

Design Models in the Development of Mechatronic Systems: Virtual Prototyping and Mechatronic Module Development Process

Tufail Habib, Thomas Ditlev Brunoe, Kjeld Nielsen, Kaj A Joergensen
Department of Mechanical and Manufacturing Engineering, Aalborg University, Denmark

Abstract:

Modelling and simulation tools support to reduce the number of physical prototypes in the development of multidomain systems. Virtual prototyping of the multidomain systems is one of the essential steps to decrease the time of the product development. In mechatronic systems, mechanical, electrical and software domains cannot be developed independently from each other at the beginning of the earliest design phases. Integrated, virtual and mathematical models are developed as they are less time consuming and are less expensive than physical prototypes.
This paper explores design process of mechatronic product development with the aid of models. V-model is used as a basic approach in the design process. Virtual model of an electromechanical system is developed by using bond graph method and the response of the system is simulated in a software tool 20 Sim. This development process is summarized in a model from design integration to simulation of the system. Furthermore, the overall design process of the system is illustrated in a model that is based on mechatronic module development. That includes mainly the allocation of requirements to individual domains and illustrating the steps in the design process.

Keywords: Bond Graph modelling, mechatronic module development, design models.

1 Introduction

In view of widespread application of mechatronic systems and the competition to offer customized products at high quality and low cost, there has been considerable attention to the design and development of mechatronic systems. Mechatronics is the synergetic combination of technologies and systems thinking to create improved and enhanced products and systems. But the consequence is that the complexity in the design process of these systems is usually increased due to the intended beneficial interaction between subsystems and technologies from different domains of mechatronics. While designing a mechatronic system, it is possible to design the mechanical system before any of the control system design has been initiated. An apparent drawback of this sequential approach is the possible lack of compatibility between the sub-systems which results in additional efforts and costs to optimally meet the specifications of the integrated system. Design engineers have to balance mechanical, electrical, electronic and software solutions. Without coordination between the different domains, it is difficult to find a suitable or even a better solution (Hehenberger, 2009).

In addition to domain specific engineering such as mechanical, electrical, information technology and user interface, an integrated and concurrent approach is necessary in the design and development of mechatronic systems. Such an approach makes a mechatronic system more optimal than systems formed on conventional design (De Silva, 2007; Isermann, 2008; Tomizuka, 2000). The design and development of mechatronic system requires the integrated approach to deal with the subsystems and sub processes of a multidomain system. The subsystems of a mechatronic system must not be designed or developed in isolation without addressing the issues of system integration, subsystem interactions and matching, and the intended operation of the overall system. Because systems formed by combining independently designed and manufactured components will not have the best match and compatibility between components functions. Therefore, integrated and concurrent approach is required primarily at the initial design phase to develop improved and enhanced systems (De Silva 2007). In this perspective, the use of test models and prototypes represents an important component of the development process.

The V-model describes the generic procedure for designing mechatronic systems (VDI, 2206, 2004) that consist of a system design phase, domain specific design and integration phase. One of the key steps at the system design phase is modelling and simulation of a virtual prototype of the system. Since the building and testing of prototypes is time and cost intensive, there are efforts to minimize as much as possible, number of physical prototypes. Virtual prototypes, i.e. the analysis of the computer models of objects that are in development can effectively support the development of multi-domain systems. Virtual prototyping using Bond Graph (BG) approach (karnopp, 2006; Gawthrop, 2007; Behbahani, 2007) is significantly important in the design of multidomain systems. In addition to modelling and simulation at the system design phase, it is also useful to analyze mechatronic module development illustrating the allocation of requirements to individual domains and presenting the steps in the design process.

Mechatronic systems are composed of multiple domains. The development of these systems requires collaboration of engineers from different fields and integration of different components from various domains are easier by using the concept of modular design. Therefore, the designers and developers of mechatronic systems would benefit from modular design. A recent survey demonstrates that companies developing mechatronic products favour "breaking the product up into specific systems, subsystems, assemblies, components and to allocate requirements to the individual subsystems and components" (Boucher et al., 2008).

In general, the challenges identified by academic research and industry during the design and development of mechatronic products or systems are: lack of tools and methods supporting multi-disciplinary design, as there are not so many specialized tools that support the initial part of the design and able to extend to the subsequent stages, is one of the challenges in the design of mechatronic systems. But there are tools that are more flexible by the use of mathematical models, such as block diagrams or Bond Graphs (Boucher,2008;Wang,2002;Wikander,2001), cooperation and communication among the design engineers due to lack of methods and tools that support system engineering and system architecting activities in order to facilitate the flow and exchange of information between designers (Boucher,2008; Hehenberger,2009; Schoner,2004;Wang,2002), persistence of a sequential design process, is not suitable because of its lack of flexibility, which increases design cost as well as the development time. Instead concurrent approach is recommended over sequential approach to deal with the design of mechatronic systems. (Wang, 2002;Wikander, 2001), simultaneous consideration of designs from different disciplines, it is necessary that the modeling of the physical systems and that of the controller design can be done simultaneously, and the design must be based on real systems approach, no afterthought solutions and adds on are allowed. Simultaneous optimization of all the system components is one of the challenges and a design requirement as well (Boucher, 2008; Craig, 2009; Wikander, 2001), multi-disciplinary modeling (Craig, 2009; Wikander, 2001) early testing and verification (Boucher, 2008; Craig, 2009).

The importance of integration such as, simultaneous design in different domains, data sharing, cooperation among designers, knowledge management is identified as a key element by authors like de silva (2005), Iserman(2008), Schonar (2004), Craig(2009), Wikander et al (2001), Tomizuka(2000), and Wang et al(2002). Similarly, another important aspect is design verification; the four classical verification methods are test, demonstration, inspection and analysis (Martin, 1997). Out of these, the first three require physical prototypes while the latter is based on mathematical representation of the system, also known as model. Developing appropriate models for analysis to verify various aspects of the system including control software represent a challenge Cabrera et al (2010). In practice specific models are developed to perform tests at different stages of design. Due to use of domain specific modelling tools, such models usually correspond to the specific point of view of the system, either electrical or mechanical aspects, or dynamic and sequential behaviour (Jackson, 2006). Assessing these challenges, core issues are identified which influence the problems in the development of mechatronic systems, these issues mainly relate to design integration, design verification and generation of control software.Therefore, further concepts and models are required to assist the design process, such as simultaneous design and integration of the domains, use of software tools for design verification (i.e. design specifications, dynamic requirements and performance measures etc), and the concept of modular design in the development process.

The rest of this paper is organized as follows. Section 2 provides the research methods employed in this paper. Section 3 provides an overview of mechatronic product development process by using V-model as an overall design approach. Section 4 presents modelling and simulations at system design using Bond Graph approach. Analysis of the computer model of antenna system by using a Bond graph approach and simulation of the model in a software tool 20-Sim. Section 5 is based on models using the concept of modularity in the design process. Model of the hierarchical structure of mechatronic system at three levels such as systems, module and component is explained. Furthermore, in this section, the design process of mechatronic module along with the allocation of requirements to individual domains and illustrating the steps in the design process is presented. Section 6 is a discussion about mechatronic module development. Section 7 summarizes and concludes the paper.

2 Methods

Design models are being developed to enrich the system design phase of mechatronic systems, because these models provide useful information and guide the design process. V-model is being used as overall design approach for design and development of mechatronic systems. A method using Bond Graph (BG) is presented and exemplified in a model to integrate the electromechanical domains, analysed the component interaction and tested the model in a simulation tool to get the dynamic response of the system. Bond Graph method is used such that all the elements of the electrical and mechanical domains of system are combined with power bonds which represent the power distribution amongst the individual elements, while the control part follows signal flows. All the design steps are summarized in model abstraction level to structure the design process. Model of the structure of mechatronic system is developed at three levels such as system, module and components. The design process of mechatronic system is analysed through the use of mechatronic module development, allocation of requirements to individual domains and illustrating the steps in the design process. Model of mechatronic module is developed to illustrate the design process that is based on V-model.

3 Mechatronic module development process

Mechatronics offers potential for success in products but at the same time need special requirements mainly at the initial development phase. These requirements are: integration of domains, design verification, generation of control software and reducing the complexity in the process. Since the interaction between mechanical, electrical and information processing of components and functions influences the behaviour and composition of the overall mechatronic system, design models and methods are used at various levels to support the development process.

3.1 The V-model

A guide for the basic procedure is originated from the V-model for software development (STARTS guide, 1989; Brohl, 1995) and adapted to the requirements of mechatronics; it describes the logical sequence of important sub steps in the development of mechatronic systems. The V-model describes the generic procedure for designing mechatronic systems (VDI, 2206, 2004). Therefore, V-model (Fig.1) is being used as an overall design approach for mechatronic product development that consist of a system design phase, decomposition phase, integration and verification /validation phase, is described below,

- At system design the specifications derived from user needs are transformed into component level specifications by defining subsystems and components. This is the conceptual design process, where the determination of the product overall function, of its most important sub functions and their interaction lead to a functional structure. Then modelling and simulation is performed and structural and behaviour models (e.g. physical model, mathematical and numerical models etc) are formed. It is rather important to enrich the system design with these models as they provide useful information and supplement the design process.

- At the detailed design, domain specific components are developed further on the basis of established domain specific development methodologies. Furthermore, domain specific development tools are used for modelling, analysis and evaluation of product properties.
- At the system integration and verification phase all the functions, components and the subsystems are combined, and then verified as well as validated in an iterative process in order to conform to the requirements and system specifications.

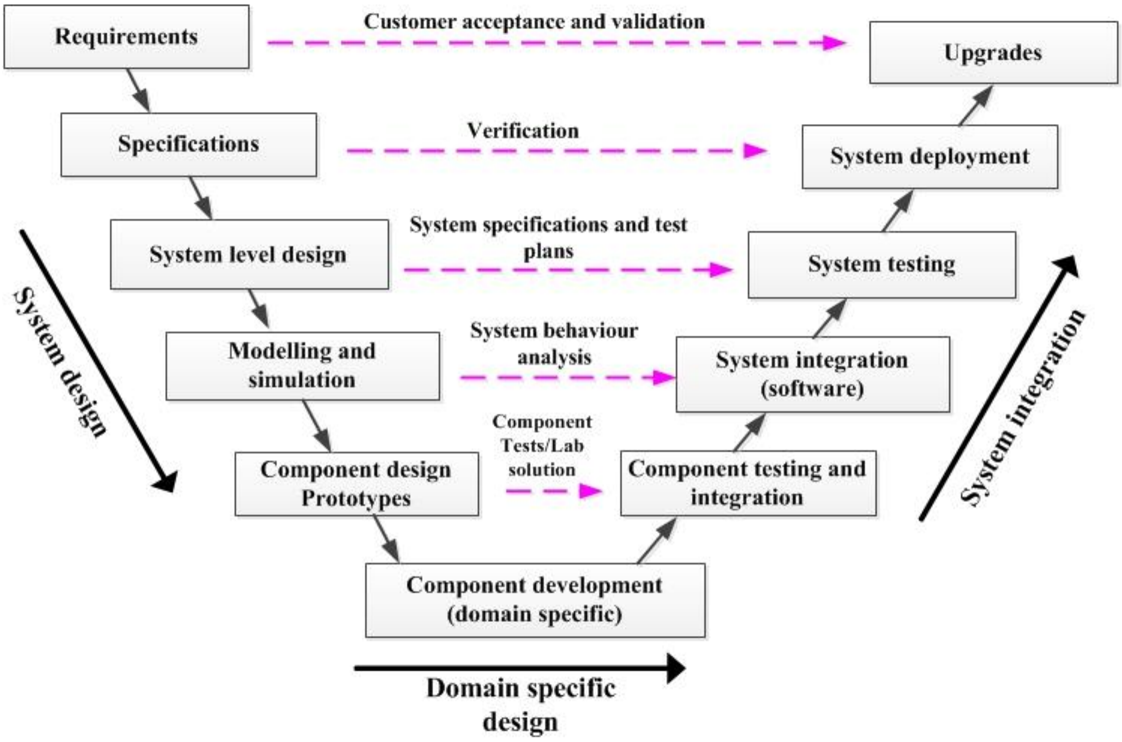


Figure 1 V-model for mechatronic product development (adapted from VDI 2206, 2004).

In the V-model, system design is further emphasized because at this phase, product specifications, the determination of product overall functions and their interactions and principle solutions with a structure of the product is established. With this backdrop, the importance of conceptual design and the link between modeling and design is further discussed.

At system design, the purpose is to define a multi domain solution concept which describes the main structural and functional characteristics of the product. Such that the overall function of the system is partitioned into main sub functions. Their main sub functions are assigned solution concepts and the performance of the function is tested through modelling and analysis.

3.1.1 Conceptual design

Conceptual Design is an important design phase, where problems are identified, functions and specifications are laid out and suitable solutions are generated through the combination of some basic building blocks using some working principles (Navinchandra, 1992; French, 1985). Conceptual Design is the most essential phase of the design process (Roozenburg and Eekels, 1995), because the decisions made here will have a strong impact on all subsequent phases of the design process. A weak concept can never result into an optimum detailed design. In addition, the main functions of the conceptual design are to generate and evaluate broad solutions, given the specification, which provides a suitable starting point for preliminary design and detail design (Rehman, 2011).

3.2 Link between modelling and design

At the beginning of the design process, the desired system does not exist. In this context, a model of the anticipated system can be very useful. In view of the complexity of a design process, particularly when trying for an optimal design, it is useful to incorporate system modeling as a tool for design iteration (Necsulescu, D., 2002).

Modelling and design is to be performed in an iterative manner. In the beginning, by having some information about the system (e.g., desired functions, performance specifications, past experience and knowledge of related systems) and using the design objectives, it will be possible to develop a model of adequate (low to moderate) detail and complexity. By analyzing and carrying out computer simulations of the model, it will be possible to produce useful information that will guide the design process (e.g., generation of a preliminary design). In this way, design decisions towards next design steps can be made and the model can be refined using the available (improved) design. This iterative link between modelling and design can be used in the design process of mechatronic systems (De Silva, 2007) that is schematically shown in figure 2.

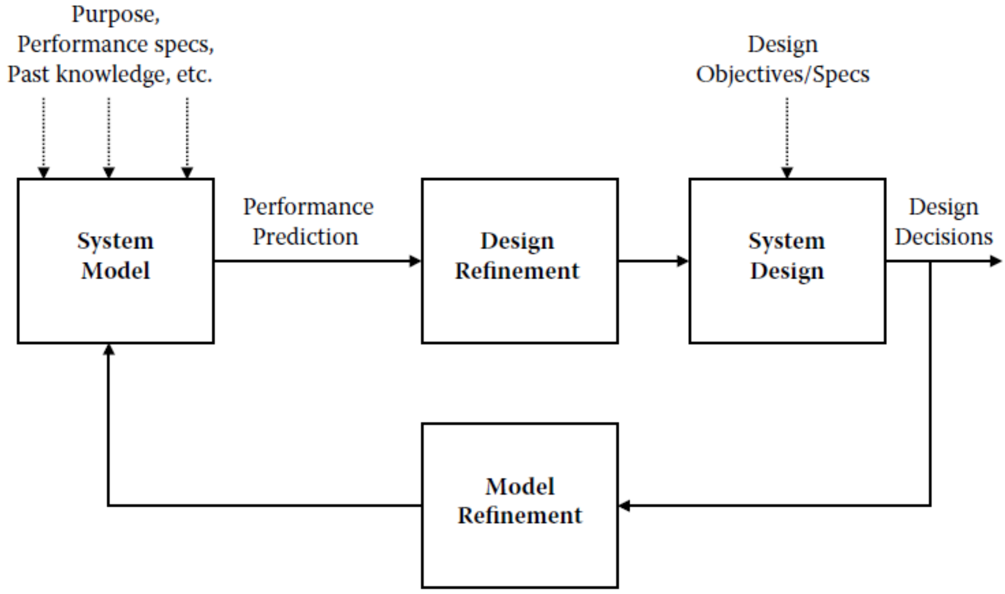


Figure 2 Link between modeling and design (de-Silva, 2007).

One of the essential part at system design phase is modeling and analysis of the of the mechatronic systems by using software tools.

4 Modelling and simulations at system design

Computer aided software tools are being used for virtual prototyping of the multidomain systems. Modelling and simulation tools in mechatronic design are one of the most important steps to decrease the time of the product development (Hehenberger, 2009). Multi domain systems cannot be developed independently from each other at the beginning of the earliest design phases due to compatibility and matching issues arises among subsystems. Integrated, virtual and mathematical models are developed as they are less time consuming and are less expensive than physical prototypes. Moreover, modelling and simulation tools help to reduce the number of physical prototypes.

Modelling and simulation is useful at the system design phase in order to generate information for the preliminary design of the system. Virtual prototypes, i.e. the analysis of the computer models of objects that are in development, can effectively support the development of multi-domain systems. Multi domain systems (combined mechanical, electrical, pneumatic, hydraulic and thermal systems etc) can be modelled using a common notation such as Bond Graph which is significantly important in the design of mechatronic systems. In Bond Graph, the physical system is built with power bonds which represent the power distribution amongst the individual elements, while the control part follows signal flows. With the BG modelling, state equations representing the behaviour of the system can be derived and can be easily simulated in software tools. Complexity is reduced through BG, with the integration and interaction of all the functional elements, flow of energy in the elements and generating the overall response of the system that is illustrated in the following example.

4.1 Illustrative Example:

This example explains the development of virtual model of the antenna system by using BG method. The basic model of the drive system consists of a DC motor, gears and pedestal. The functional components of the drive system are connected by bonds, along with the effort and flow variables, for instance in case of motor the effort variable is torque and the flow variable is angular velocity as shown in figure 3.

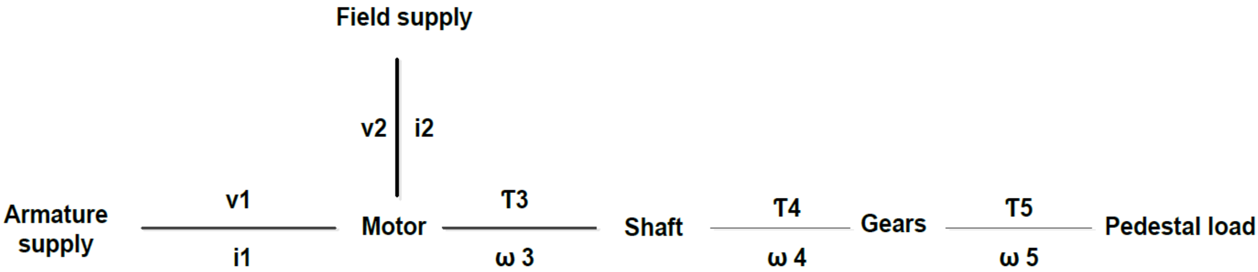


Figure 3 Word bond graph for the Antenna system.

The bond graph model (Fig 4) consists of an electrical and a mechanical part. “Se” is represented by an effort source in the form of input voltage. The current is common to the armature resistance “R” and to the inductance “L”. The inductance is represented by an “I- element”, the resistance by an “R-element”. Both of the elements are attached to 1 junction of the bond graph. The electromagnetic action is represented by an electric motor shown by the “GY-element”, the rotor inertia and friction are modelled by the “R” and “I”-elements respectively, attached to the right 1 junction of the GY (motor). This 1 junction produces angular velocity when a voltage is applied. The “C- element” represents the shaft stiffness and is attached to the “0-junction”. Gears are used to represent the reduction in torque and speed. The gear train is modelled as transformer and the ratio of the number of teeth on each wheel becomes the transformer modulus “m” and modelled as “TF-element”. Antenna load and friction are represented by “I” and “R” elements in 1 junction right to the “TF” element.

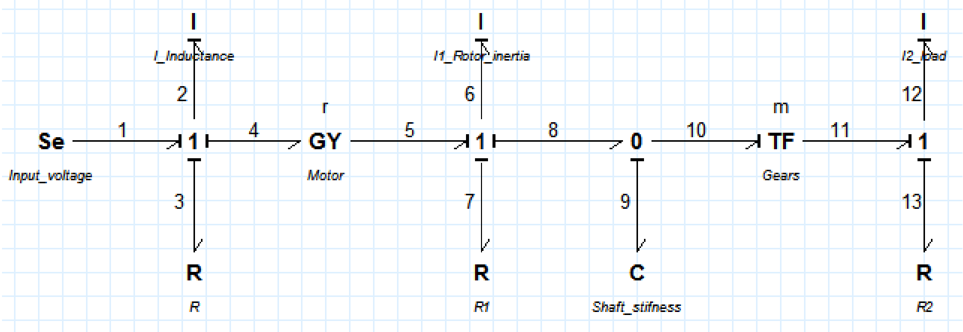


Figure 4 Bond Graph model for the antenna system

In a typical mechatronic system, the dynamic behaviour of the process (or plant) has to be controlled to achieve a desired response. Main /subsystems components of electromechanical system which are interconnected through the flow of either information or power are shown in figure 5. The high power energy transfers are shown by half arrows, whereas the information transfers are shown by full arrows.

Electromechanical and mathematical models are generated such that they can also be used for the generation of embedded software.

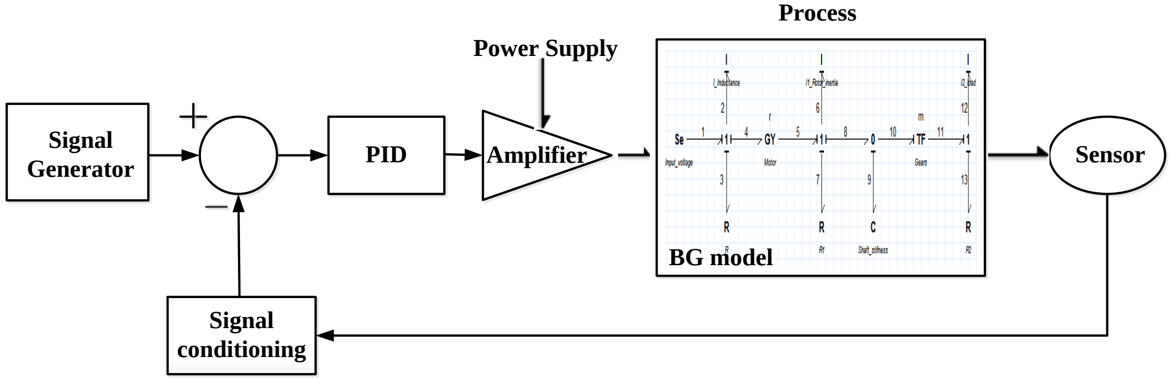


Figure 5 Schematic model of antenna pedestal system, with information and energy flows.

A complex system can be abstracted (i.e. reduce its complexity by emphasizing essential characteristics) from various aspects, for instance, by domain (e.g. the hydraulic system of an airplane -mechanical domain), by flow (e.g. the autopilot control system – signal flow), by function (e.g. the propulsion system – propulsion function) (Vargas, 2003).The procedure for modelling may vary for different domains of mechatronics. In the domain of software technology, for instance, the interrelationships between requirements for the system and the subsystems may be structured and represented in a functional description (Bertram, 1997). In order to describe the design process of the antenna system, different models of the system are formed on various abstraction levels (Figure 6).

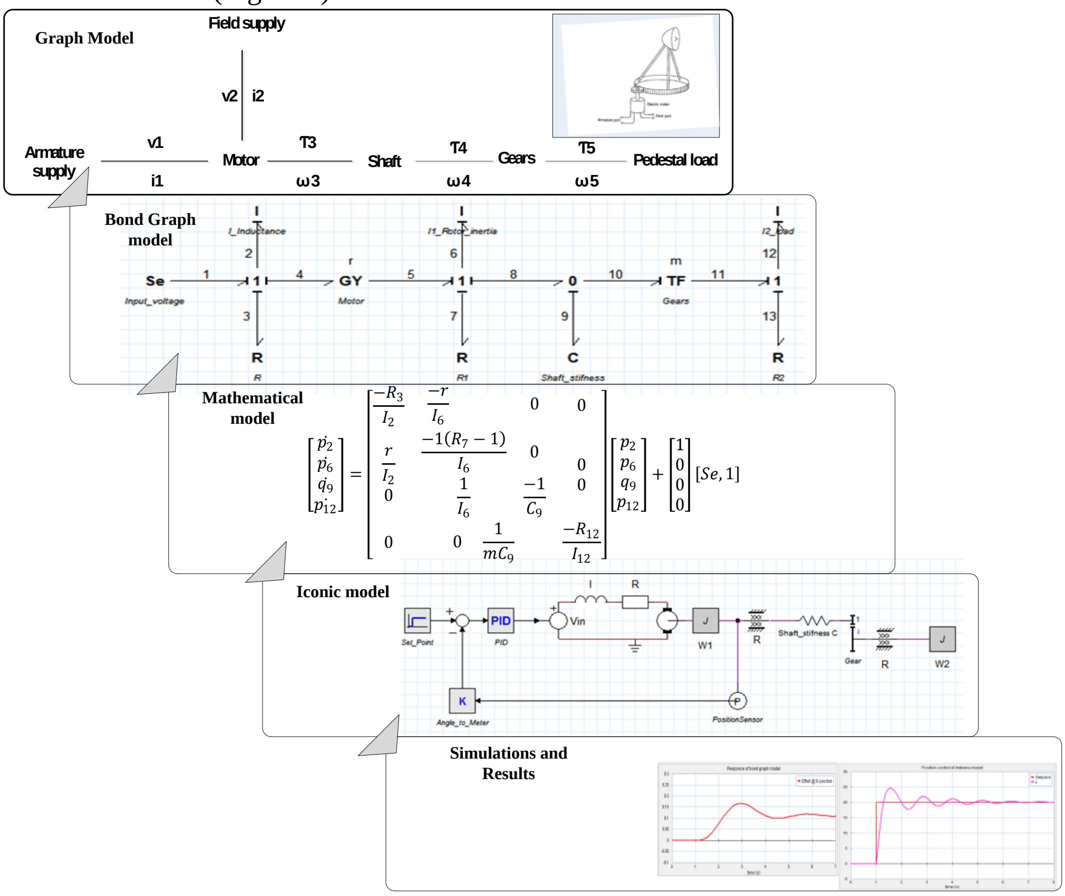


Figure 6 Model abstraction levels to describe the design process of antenna drive system.

Companies may benefit by using the virtual prototypes with the support of modelling and simulation tools to analyze the product characteristics at system level and avoid the physical prototypes to accelerate the product development time and thus reduce the costs. After the desired response through (BG models, C code generation and real time implementation) it is also possible to generate a number of controllers with varying parameters of the individual elements representing the physical system. These controllers are basically software that can be used for customization in products or for delayed differentiation at customer site.

4.2 Simulations results using 20-Sim

The Bond Graph is a "system engineering tool" where the overall structure of the system is established and for that, need not to be domain expert, as in the case of antenna system the initial model and its response is simulated in 20-Sim. The response of the system such as the effort at zero junction is obtained as shown in figure 8. Similarly, a PID controller with a set point is being used for the position control of the motor as shown in figure 8.

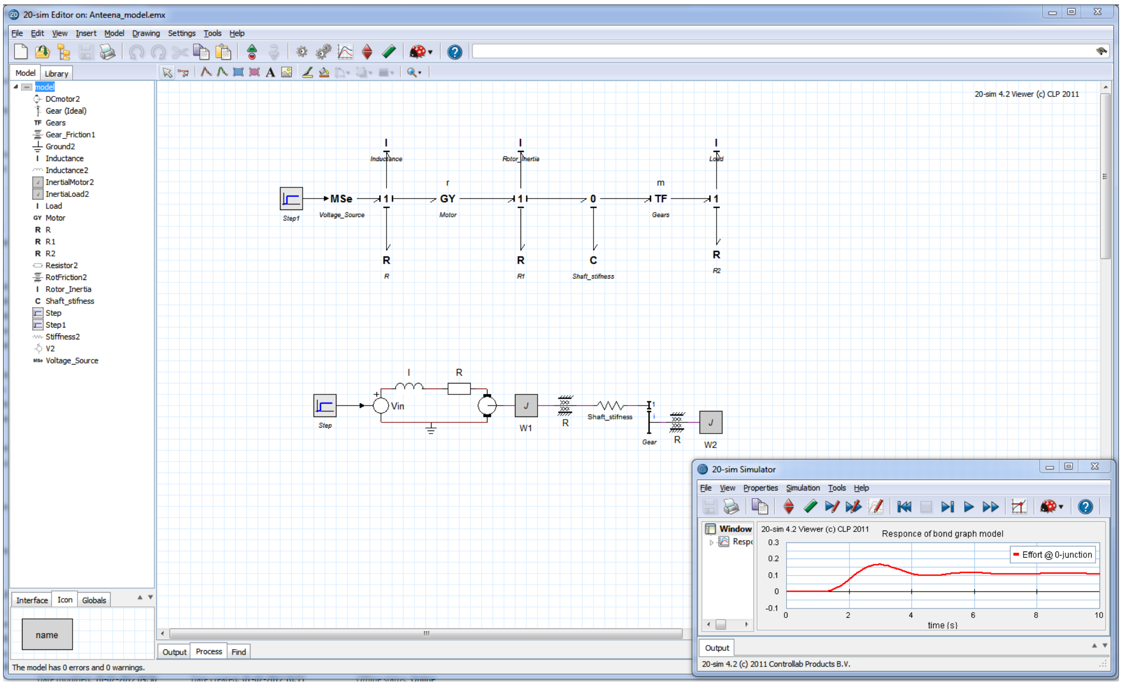


Figure 7 Response of the bond graph model of the system

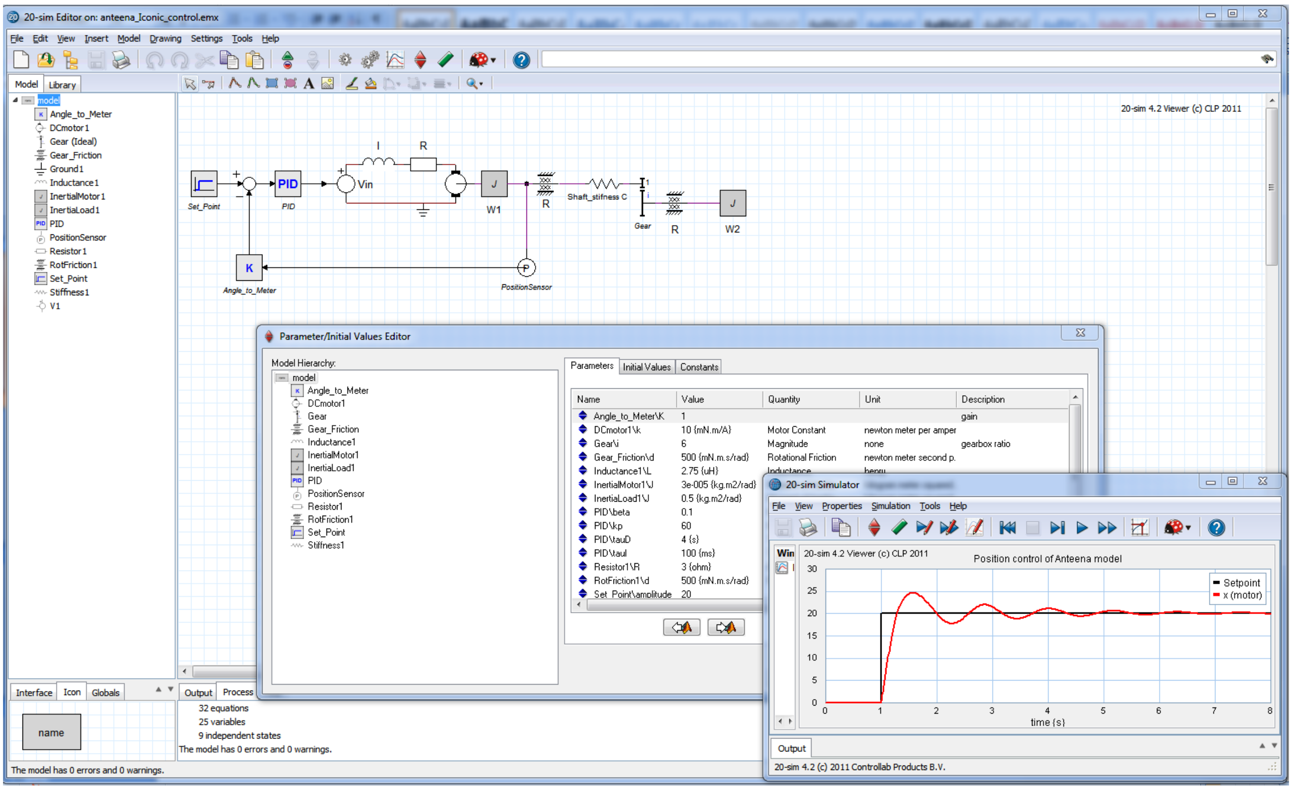


Figure 8 Set point and position control of the motor.

In short, modelling and simulation of the multidomain system is performed to get the simultaneous design and integration of domains and also to support the earliest design phase. For this purpose BG method is used, such that the integration of electrical and mechanical components is connected by bonds that specify the transfer of energy between system components. These bonds represent effort and flow variables in the system. Moreover, the bond graph model is transformed into a mathematical model, based on differential equations that represent the behaviour of dynamic system. After constructing bond graph of the antenna system, the response of the model is simulated in a software tool 20-Sim. Complexity is reduced through Bond Graph, with the integration and interaction of all the functional elements, flow of energy in the elements and generating the overall response of the system. The whole process is summarized in a model in a systematic way from design integration to simulation of the system.

5 Models of Mechatronic module design

In mechatronic product development, one of the forms of integration is modular integration, where the overall system is made of modules of defined functionality and standardized dimensions in various size. Each module may not be necessarily linked to a single function. These modules that are included in modular system can be flexibly combined and make it possible to obtain functional variety (VDI 2206, 2004).

A basic mechatronic module (MM) uses several disciplines of mechatronics (e.g. mechanics, automatic control techniques, electrical design etc.). In such a mechatronic module exclusively domain-specific components are merged. That means a basic mechatronic module can be decomposed only into domain-specific (non mechatronic) components (i.e. mechanics, electronics and control software etc), but not into other mechatronic modules or mechatronic system components (Fig 9). Basic mechatronic modules therefore represent a mechatronic sub-system at the lowest hierarchical level of a mechatronic system and are indivisible within the set of mechatronic sub-systems or this mechatronic module can be a mechatronic system itself. With the mechatronic pillar design model (Hehenberger, 2008, 2009) all couplings between the several mechatronic disciplines (domain pillars) can be described in a data platform, that represent design models and simulations.

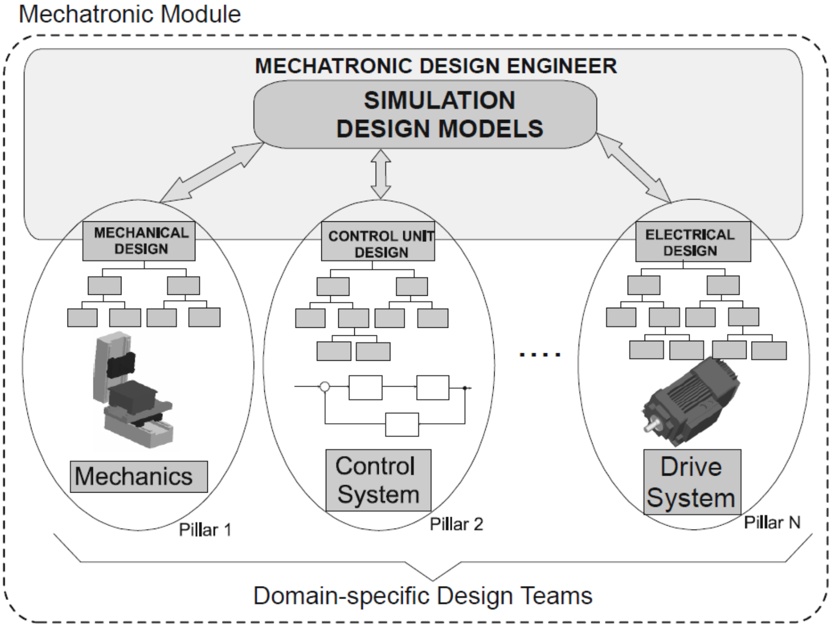


Figure 9 Mechatronic pillar design model (Hehenberger, 2009)

5.1 Model of mechatronic system at three levels

The structure of mechatronic systems generally consists of various mechatronic modules comprising system elements and components which are combined to form a group and perform certain functions (Fig 10). If a number of mechatronic modules are combined then a system of higher order is created with further functions (e.g learning process and adaptation, communication network etc), where further sensor and information processing is required along with additional tasks from the basic mechatronic modules.

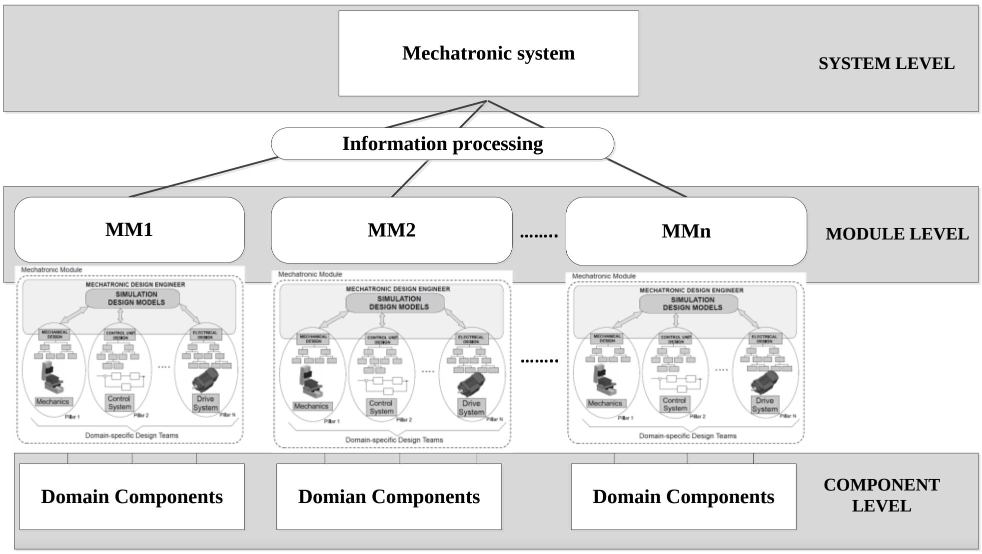


Figure 10 Model of the structure of mechatronic systems at three levels.

Generally, basic structure of mechatronic system comprises sensors, actuators and information processing and is connected by means of material, energy and information flows. There may also be a connection with environment that may be in the form of user interface, communication systems and the sensors itself. While basic system within mechatronic system is generally any (mechanical, electromechanical, pneumatic, hydraulic and thermal etc) systems or a combination of them.

5.2 Process of basic mechatronic module development

At the system design phase, the specifications derived from user needs are transformed into component level specifications by defining subsystems and components. In the process, determination of the product overall function, of its most important sub functions and their interaction lead to a functional structure. Then modelling and simulation is performed and structural and behaviour models (e.g. physical model, mathematical and numerical models etc) are formed. It is rather important to enrich the system design with these models as they provide useful information and supplement the design process. For instance, the BG model of the antenna system provide information about interaction between electrical and mechanical elements, flow of energy between elements, facilities simulation for the overall response and control of the system.

After the behaviour analysis through simulations, a mechatronic module is developed further into domain specific design. In the example of antenna system, domains of mechanics, control unit and electrical system are developed in detail at multiple levels, for example mechanic part include materials, geometry, dynamics, gear system design, fixing drive unit with antenna and manufacturing details etc (Fig 11).

At system integration, a number of basic mechatronic modules are combined and a system of higher level is created, where further sensor and information processing is required along with additional tasks from the basic mechatronic modules.

Model of the mechatronic module associated with antenna system, illustrate the following information

- V-model steps such as system design, domain specific design and system integration
- Modelling and simulation at system design
- Details of domain specific design

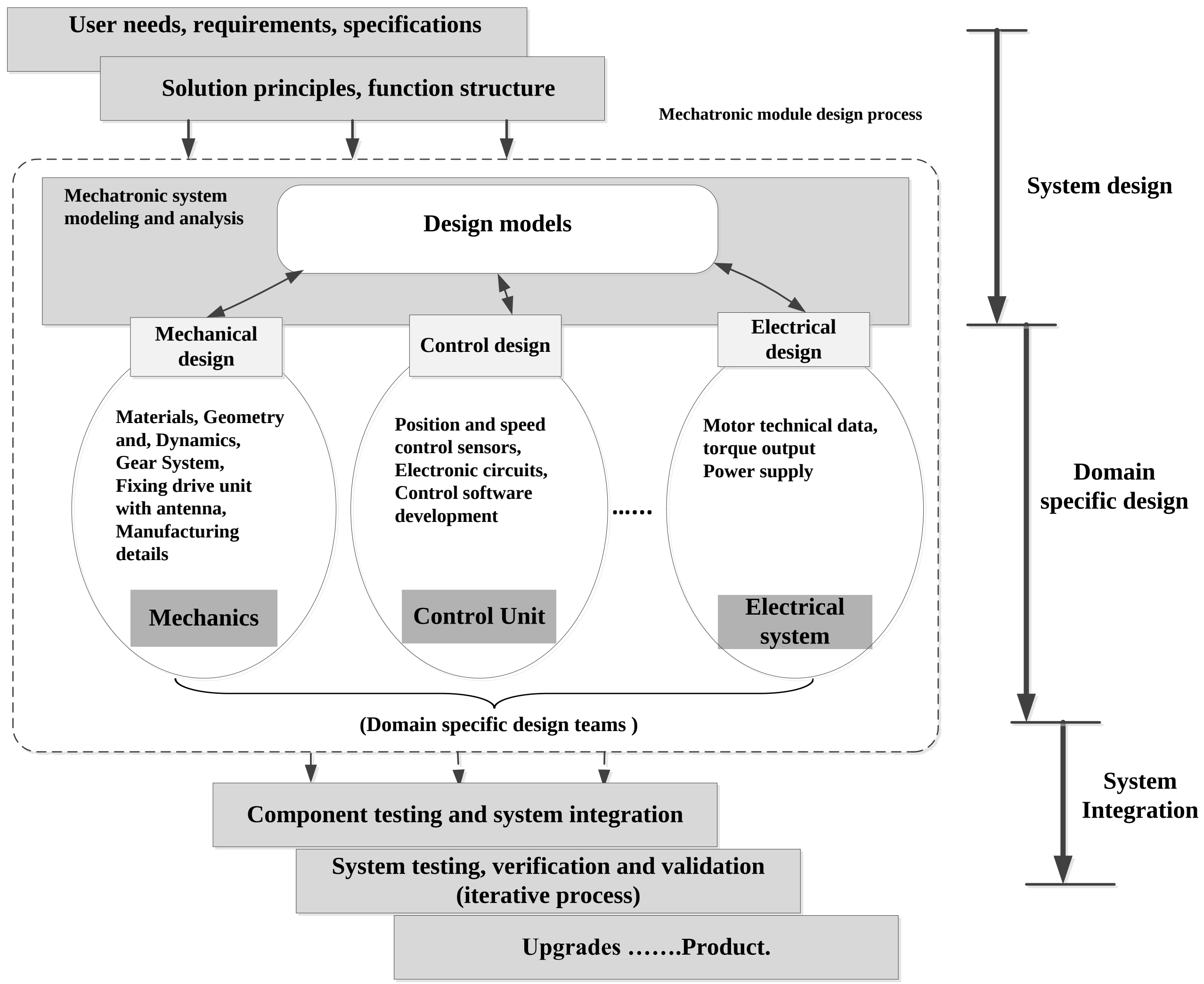


Figure 11 Model of mechatronic module design process associated with antenna system.

In short, the development of multidomain systems requires collaboration of engineers from different fields and integration of different components from various domains are easier by using the concept of mechatronic module design. When the system is developed into modules, therefore, the designers and developers of mechatronic systems would benefit from this approach. Breaking the product up into specific systems, modules and components and to allocate requirements to the individual subsystems and components make it easier and manageable for companies developing mechatronic products.

6 Discussions

The V-model is used as an overall design approach that describes the logical steps in the development of mechatronic systems. At system design phase, one of the key issues in the process of creating a physical device from scratch is the creation of a computer based model and along with its control system. In BG, a model of the physical system is built with power bonds, which represent the power distribution amongst the individual elements, while the control part follows signal flows, as shown in the example of antenna system. More complex models such as internal combustion engine involving thermo-mechanical and hydro mechanical models can be developed. Many electrical and electromechanical systems contain magnetic circuits and devices, such as motor design, solenoid and transformer, can be modelled through this method. The advantages of the BG method is that it is equation based, and thereby, multiple systems can be represented by the same set of equations and hence modelled using a common notation. Causality concept is used to define the input and output energy variables. Similarly, bond graph obeys the laws of physics and the models can easily be simulated in software tools.

The complexity inherited in mechatronic systems is primarily due to interactions and integration issues, is handled through BG method and implemented in software tool (20-Sim). Initially, through BG the interrelationships between elements of the system and the subsystems are structured and represented in a functional description. Then, mathematical and iconic models are developed for the system response through simulations. In the model of antenna system complexity is reduced with the interaction of all the functional elements, flow of energy in the elements and generating the dynamic response of the system with feedback control. From the model, C code can be generated and used for the design of the controller and also be used for customization purposes. With logical steps through model abstraction level, development process is simplified and structured.

The mechatronic modules comprising the domains of mechanics, control unit and electrical systems are developed of defined functionality and standardized dimensions in various sizes. The modules included in a modular system can be flexibly combined and make it possible to obtain functional variety and also to support mass customization. Because the concept of modularity in design process is one of the pre requisites for successful mass customization strategy, such that to create customized products and to reduce complexity and costs. In order to implement mass customization and modular development in mechatronic products, two aspects are vital:

- First, the potential benefits of mechatronics come from the innovative capabilities of the technologies and their functional and structural integration. Functionality in mechatronic products is enhanced, as mechanical functions are replaced by more electronics and software functions, examples are CD players, digital cameras, CNC machines, robots etc. Furthermore, the principles of mechatronics can be utilized to achieve efficient product customization, since customization can also be implemented in software rather than physical components enabling postponement of differentiation point and generally reducing the variety of components ultimately reducing the complexity due to number of components.
- Second, the design process must be based on the development of modules. Modules at defined functionality and standardized dimensions can be combined to obtain functional variety. Mechatronic module composed of several disciplines of mechatronics can be decomposed only into domain specific components. An example of mechatronic module development is illustrated in figure 10, where the respective domains are developed of defined functionality and can be developed with standardized dimensions in various sizes.
- With these two aspects and logical steps adapted in model abstraction level in (Fig 6), companies may benefit and be able to develop customized products much faster with reduced costs.

Further issues related to mechatronic system design such as support of the design of control software, exchange of design models and data, cooperation and communication among the design engineers is not discussed in this paper. Future work will be related to system architecting at conceptual design of mechatronic products by using another computational tool, which employs system modelling. Function modelling with computational tool should support the system modularization using the dependencies among system parameters. Further, the process of system modelling must enable system designers, to form appropriate component level specifications, to communicate with the domain experts.

7 Conclusions

In this paper, modelling of the mechatronic system development process is presented, and two key aspects of the development such as virtual prototyping and mechatronic module development are researched. A virtual model is constructed in Bond Graph to integrate the electromechanical domains, analyse the component interaction and testing in a simulation tool to get the dynamic response of the system. Bond graph method is used such that all the elements of the electrical and mechanical domains of system are combined with power bonds which represent the power distribution amongst the individual elements, while the control part follows signal flows. All the design steps are summarized in model abstraction level to structure the design process. Virtual prototypes support the design process to analyze the product characteristics at system level and avoid the physical prototypes to accelerate the product development time and thus reduce the costs. Model of mechatronic system that represents the hierarchical structure at three levels such as system, module and components is developed. The design process of mechatronic system is supported through the use of mechatronic module development, allocation of requirements to individual domains and illustrating the steps in the design process, that are based on V-model.

References

Behbahani.S, De Silva.W,(2007) "Mechatronic modeling and Design", Mechatronic systems, Devices, Design, Operation and Monitoring, CRC Press, 17-1-17-26.

Brohl, A.P. (1995) 'The V-Model: standard for software development', 2 Ed Munich: Oldenburg.

Boucher, M., Houlihan, D. (2008) 'System design: new product development for mechatronics', Aberdeen Group; Boston (MA, USA).

Bertram, T., Schroder, W. Dominke, P., Volkart A. (1997) 'CARTRONIC-an organizing concept for the control and regulation systems in automotive coatings, system engineering in automotive development', VDI Report 1374, Düsseldorf: VDI Verlag (in German).

Cabrera, A.A. Alvarez , M.J. Foeken , O.A. Tekin, K. Woestenenk , M.S. Erden , B. De Schutter , M.J.L. van Tooren , R. Babuška , F.J.A.M. van Houten , T. Tomiyama , (2010) ' Towards automation of control software: A review of challenges in mechatronic design', Mechatronics 20 (2010) 876–886.

Craig,K.(2009)'Mechatronic system design', ASME Newsletter 2009. http://files.asmeorg/asmeorg/NewsPublicPolicy/Newsletters/METoday/Articles/17845.pdf>

De Silva, C.W., Mechatronics-An Integrated Approach, Taylor & Francis/CRC Press, Boca Raton, FL, 2005.

De Silva, C.W. (2007) 'Mechatronic Systems', Technology needs for mechatronic systems, CRC Press, 2007. DOI: 10.1201/ 9780849307768.ch1

French, M.J. (1985) Conceptual Design for Engineers, Design Council, 2nd ed., Springer Verlag, London.

Gawthrop, P. and Bevan, G. "Bond-graph modeling".*IEEE Control Systems Magazine*, 2007. 27:24, 45. doi:10.1109/MCS.2007.338279.

Hehenberger, P. (2009) 'Application of mechatronic CAD in the product development process', *Computer-Aided Design & Applications*, Vol. 6(2), p.269-279. DOI: 10.3722/cadaps.2009.269-279

Hehenberger, P.; Zeman, K.: "The role of hierarchical design models in the mechatronic product development process", TMCE International Symposium Series on Tools and Methods of Competitive Engineering, Izmir, Turkey, 2008.

Isermann, R. (2008) 'Mechatronic Systems Fundamentals', *Meccanica*, Vol. 43, Issue 4, 2008, p.459-460. 10.1007/s11012-008-9147-1.

Isermann, R. (2008) 'Mechatronic systems-Innovative products with embedded control', *Control Engineering Practice, vol.*16, p.14–29. http://dx.doi.org/10.1016/j.conengprac.2007.03.010

Jackson CK. Simulation driven design benchmark report. Boston (MA,USA): Aberdeen Group; 2006.

Karnopp, D., Margolis, D, and Rosenberg, R.,(2006) "System dynamics: modeling and simulation of mechatronics systems", John Wiley & Sons.

Martin JN. Systems engineering guidebook – a process for developing systems and products. Boca Raton (FL, USA): CRC Press; 1997.

Necsulescu, D., *Mechatronics,* Prentice Hall, Upper Saddle River, NJ, 2002.

Navinchandra, D. (1992) 'Innovative design systems, where are we and where do we go from here, Part I: design by association', Knowledge Engineering Review, Vol. 7, No. 3, pp.183–213.

Roozenburg, N.F.M. and Eekels, J. (1995) 'Product Design: Fundamentals and Methods', John Wiley & Sons: West Sussex, England.

Rehman, F.U. and Yan, X-T. (2011) 'Application of context knowledge in supporting conceptual design decision-making', Int. J. Product Development, Vol. 13, No. 1, pp.47–66.

Schöner, H.P. (2004) 'Automotive mechatronics', *Control Engineering Practice,* Vol. 12, p.1343–51. http://dx.doi.org/10.1016/j.conengprac.2003.10.004

STARTS GUIDE (1989). 'A guide to methods and software tools for the construction of large real-time systems'. Manchester: National Computing Centre.

The Association of Engineers (VDI). (2004) 'Design methodology for mechatronic systems'. VDI 2206, VDI Guidelines, Beuth Verlag, 2004 (in German).

Tomizuka, M. (2000) 'Mechatronics: From the 20th to 21st century, *In Proceedings of the IFAC Conference on Mechatronic Systems*, Darmstadt, Germany, September 18-20,2000. http://dx.doi.org/10.1016/S0967-0661(02)00016-3

Vargas-Hernandez, N., Shah, J., Lacroix, Z. (2003) 'Development of a Computer Aided Conceptual Design Tool for Complex Electromechanical Systems', Computational Synthesis: From Basic Building Blocks to High Level Functionality, Papers from the 2003 AAAI Symposium Technical Report, SS-03-02, 2003, 255–261. (www.aaai.org).

Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A. (2002) 'Collaborative conceptual design – state of the art and future trends', *Comput-Aid Des,* Vol.34,2002,p.981–96. http://dx.doi.org/10.1016/S0010-4485(01)00157-9

Wikander, J., Törngren, M., Hanson, M. (2001) 'The science and education of mechatronics engineering', *IEEE Robot Autom Mag* , Vol.8(2),p.20–6. http://dx.doi.org/10.1109/100.932753.

Paper 4

Multidisciplinary Product Decomposition and Analysis Based on Design Structure Matrix (DSM) Modelling.

Habib, Tufail. Proceedings of the 7th world conference on Mass Customization, Personalization and Co-Creation (February 4th -7th, 2014) Aalborg, Denmark. ISBN: 978-3-319-04270-1

Multidisciplinary product decomposition and analysis based on design structure matrix (DSM) modelling

Tufail Habib

PhD Student, Aalborg University, Department of Mechanical and Manufacturing Engineering, Aalborg, Denmark

habib@m-tech.aau.dk

Abstract. DSM modelling in complex system design support to define physical and logical configuration of subsystems, components and their relationships. This modelling includes product decomposition, identification of interfaces and structure analysis to increase the architectural understanding of the system. Since product architecture has broad implications in relation to product life cycle issues. In this paper, mechatronic product is decomposed into subsystems and components and then DSM model is developed to examine the extent of modularity in the system and to manage multiple interactions across subsystems and components. For this purpose, Cambridge advanced modeler (CAM) software tool is used to develop the system matrix. The analysis of the product (printer) architecture include clustering, partitioning as well as structure analysis of the system. The DSM analysis is helpful to support decisions about product redesign and modularization.

Keywords: Design structure matrix, complexity, interfaces, mechatronic products.

1 Introduction

Mechatronic products such as hybrid vehicles, industrial robots, medical instruments and printers have been developed through the functional and spatial integration of subsystems with various engineering disciplines to fulfill the market needs. For this purpose various approaches, models and analysis tools are used to represent and understand the architecture of complex mechatronic systems. Since, the decisions about product architecture are relevant to the overall function of the product and, which has broad implications related to product performance, product change, product variety, component standardization, and manufacturability [1].

According to Ulrich and Eppinger, "The architecture of a product is the scheme by which the functional elements of the product are arranged into physical chunks and by which the chunks interact" [4]. Product architecture is thought about in terms of its modules and decomposing a system into independent parts or modules that can be treated as logical units [5]. It is the process of rearranging known parts into new architectures, and it revolves around redesigning the interfaces of key components to make them more modular in order to achieve a higher level of system performance in one or various dimensions [6], that is also relevant to mass customization (MC) in products. The aim of developing and using modules in product architecture is partly to make it possible to create customized products for the market and partly to reduce the number of variants which have to be dealt with internally in the company, and thus to reduce complexity and cost [14].

Complexity is involved in the design and development of mechatronic systems due to number of subsystems, components their interactions and other aspects. According to Weber, complexity is an attribute of a system and can be divided into various aspects such as numerical, relational, variational, disciplinary and organization complexity [7]. These aspects can be the number of subsystems, components their relations and variants as important characteristics of complexity. The degree of complexity is also relevant to the number of disciplines and the distribution of work [8]. In the context of mechatronic systems, designers and engineers need to deal with various aspects of complexity. Market requirements are also attributed to complexity due to customization that requires number of variants in products. Interaction of disciplines and distribution of work is also an issue especially for multidisciplinary products. In order to address these issues, various approaches and analysis tools are used; one of such tools is DSM to model complex products.

The DSM is a network modeling tool to represent the components of a system and their interactions, therefore highlighting the systems architecture [13]. DSM first introduced by Steward [3] followed by many authors in different fields with a range of applications to product and organization domains. Various organizations and industry such as BMW, Audi, Hilti, NASA, Boeing, General motors, Intel , Kodak, Mozilla, Timken and BP etc used it for various issues relevant to product, organizational and process architecture modeling. In the domain of product development, the component based DSM could be combined with the task and team DSMs to include the modularization in the rest of the design process planning using multi domain DSM [10]. The method leaves more business oriented factors and product functionality up to the designer's judgment after first simplifying the architecture by decomposition and interface management. Therefore, it is important to raise the following research question:

How the functional elements in a product can be decomposed into components by identifying their interrelationships to assess the degree of modularity as there is always a tradeoff between modules and market requirements on one hand, and functionality and performance on the other in mechatronic products/systems?

The complexity in multi domain products requires decomposing them into subsystem and components, to guide the design requirements and to identify the solution space for functional improvements. This work implement component based DSM (using printer as an example) in order to address the issue of system decomposition and interface management. The outcome of this paper is product decomposition to increase the architectural understanding of the multi domain system, to examine the degree of modularity in the system, and manage interactions across subsystems and components.

2 Methodology

In this methodology through system decomposition, complex products can be decomposed into subsystems, components and functions. A modeling tool such as DSM is used in a software tool to represent the system elements and their interactions in order to generate the system architecture. In this example, the architecture is used to identify modules in the system, manage interactions across subsystems and components and structure analysis results are presented.

Fig.1 represents an overall approach in this paper that is based on [13]. After system decomposition, the relationships between system components are identified. For the printer case, data about interfaces and physical structure are collected from product manuals, product videos and physical observation of the product. In the next step, all the elements of the system are placed along rows and columns in a matrix display format. For this purpose, Cambridge advanced modeler (CAM) software tool [2] is used to develop the system matrix. Finally, the analysis of the system architecture (DSM form) is performed (e,g. clustering, partitioning and displaying the elements in a network diagram). This DSM can be further extended to multiple domain matrixes (MDM) for analyzing issues related to process and organization, however, this paper is limited to product architecture DSM.

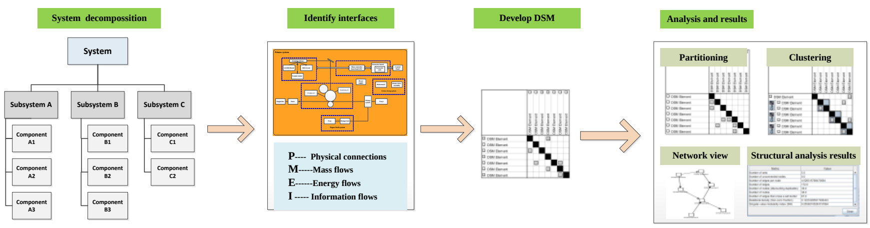


Fig. 1. Overall approach for product decomposition and analysis using DSM modeling (adapted from Eppinger and Browning [13]).

3 System decomposition and identification of interfaces

3.1 System decomposition

In general, main subsystems of the printers are image formation system, paper feed and delivery system, scanning system, formatter and control system and fuser system. In the printer example, other functional elements such as duplexing unit and envelops feeder systems are considered as optional systems to simplify the DSM model.

3.2 Identify interfaces between system components

The following types of interfaces are identified in the printer system:

- Physical connections
- Material (e.g. toner, paper)
- Energy flows (e.g. mechanical rotary, electrical, thermal, chemical etc.)
- Information flows (e.g., image data, sensor signals, and actuator commands)

Spatial interfaces indicate that physical adjacency is needed for alignment, orientation, serviceability, assembly and weight. For example, scanning mirror and focusing lens are in physical contact with scanning motor, when scanning mirror is rotated by motor, LASER beam reflects off the mirror, through a set of focusing lenses that is directed on photosensitive drum. A spatial connection between scanning motor and mirror is established in order to reflect beam on photosensitive drum. The alignment and orientation of the drum and charging roller is a necessary feature to create a uniform negative potential on the drum surface that is necessary for the image development and its subsequent transfer to paper. Thus a physical interface between charging roller and photosensitive drum is identified.

During the fusing process, the toner is fused into the paper by heat and pressure to produce a permanent image. The paper passes between a heated fusing roller and a pressure roller. This melts the Toner and presses it into the paper. The quality of the fusing process depends on heat and pressure produced by fusing roller and pressure roller and their interaction with paper. Thus a spatial interface is created between paper and fusing roller as well as paper and pressure roller.

Material interfaces indicate a functional requirement related to transferring mass flows such as toner and paper. For example, the developing cylinder must be able to attract toner and the toner must obtain negative surface charge as the developing cylinder is connected to power supply. Thus developing cylinder depends on power supply to be able to attract toner, while the toner must be attracted by this process. This results in a symmetrical dependency.

Energy flow indicates a functional requirement related to transferring mechanical energy, heat energy, vibration energy, electrical energy and noise etc. In printer example, for instance, the variation in the print density depend on the DC bias given to the developing cylinder, that causes more or less toner to be attracted to the developing cylinder, hence developing cylinder and power supply is related by (electrical) energy . Similarly, motor and drive assembly are related by power transmission due to mechanical energy. Heat transfer from heater to cooling fan is kind of (thermal) energy interface. Although energy interfaces such as chemical, vibrations are also present in this kind of systems, however they are not considered in this example.

Information interface indicates a functional requirement related to transferring sensor signals or controls, image data and actuator commands. For instance information about LASER beam is send to central processing unit (CPU) by the beam detect (BD) sensor; these two are related by information interface. Similarly, information about image formation is transferred from control panel to formatter CPU is highlighted by information relation. When the power switch of the printer is turned on and the printer enters in the standby mode, the CPU outputs the signals to drive the loads such as Laser Diode, motors, and solenoids, based on the print commands and the image data input from the external device. CPU and the loads are connected by information (image data and actuator commands) dependency.

3.3 Develop design structure matrix (DSM)

The tool that is used to handle relations between items is widely known as the DSM. As shown in Fig. 2, a square matrix representing the elements in a system (the shaded cells along the diagonal) and their interactions (the off-diagonal marks). There are two possibilities to read the matrix. One can reads across an element's row to see its inputs and down its columns to see its outputs although the opposite convention, the transpose of the matrix, is also used. For instance element D receiving inputs from elements B and C and providing an output to element B, as shown in Fig.2.

	A	B	C	D
A			X	
B				X
C	X			
D		X	X	

Fig. 2. DSM showing four elements of a system and their relationships.

To model product architecture, the DSM elements are product components and their interactions are the interfaces between the components. In structure analysis, DSM elements are called as nodes and their interactions as edges of a system.

3.3.1 DSM model

The composite DSM (comprising multiple interfaces) model using CAM (Cambridge advanced modeler) is displayed in figure 3 that shows the decomposition of the printer system into eight subsystems and 38 functional components. Four types of interfaces such as P-M-E-I (physical, material, energy and information) are indicated in the DSM. Eight subsystems and two optional systems are discussed in section 2.

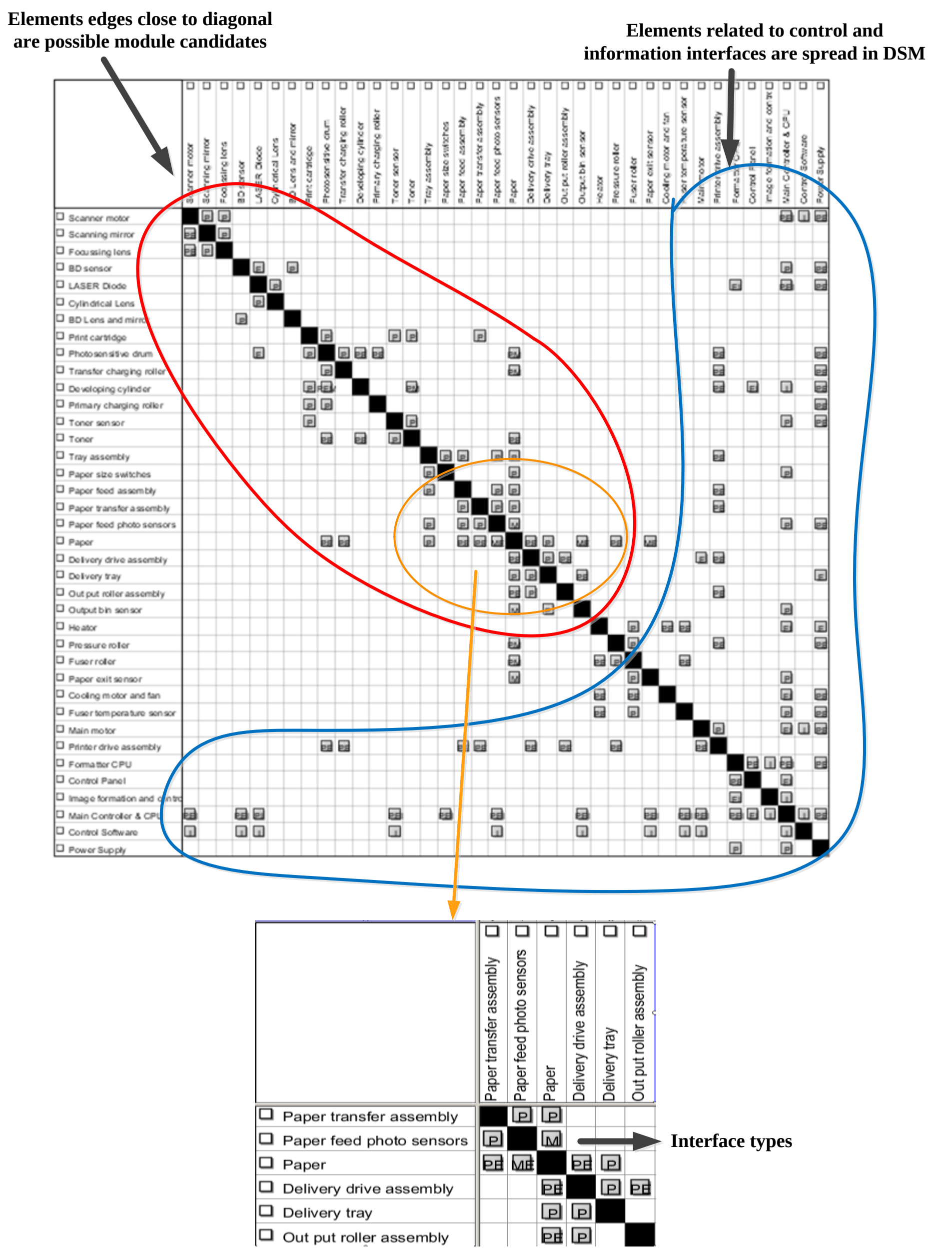


Fig. 3. Printer system (architecture DSM) components and their relationships.

4 Results and analysis

Once the DSM model is developed, the analysis of the system can be performed with the following types:

- Clustering
- Partitioning
- Structure analysis

Initially, a DSM based only on physical interfaces is analyzed by applying clustering algorithm using CAM. The results, as illustrated in Fig.4, identify seven of the subsystems as somehow modular, as having more interfaces among components in each subsystem. These modular subsystems are scanning system, paper feed system, tray assembly, LASER and beam detect system, paper delivery and fuser system. As only physical connections are used there are no interfaces with elements such as image formation and control software and they are placed independently in the model. The DSM also show the remaining subsystems as more spatially distributed. For instance printer control system and printer drive assembly are more functionally distributed across the printer system, or in other words their structure is more integrative than modular one.

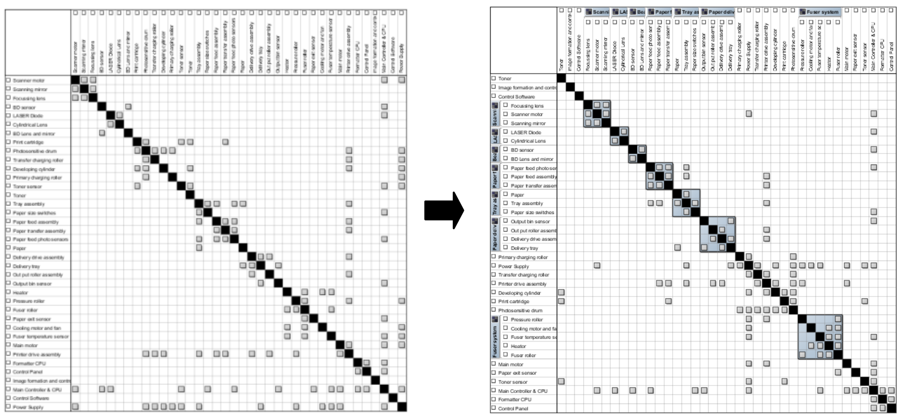


Fig. 4. DSM of physical connections and its modular structure after applying clustering algorithm.

To analyze composite DSM, partitioning algorithm is applied to it. The CAM tool uses loop searching algorithm, which basically tries to accumulate dependencies on one side of the matrix diagonal. If this alignment cannot be realized completely, the partitioning tries to arrange dependencies as close as possible to the diagonal. Partitioning is also used to minimize the size of the feedback loops (A feedback loop consists of two or more nodes of a DSM, which are interlocked sequentially by edges and reciprocally influence each other [8]). Complex structures possess feedback loops that do not allow an alignment of edges at one side of the matrix diagonal. Partitioning then tries to align a minimum of edges below and all edges as close as possible to the diagonal.

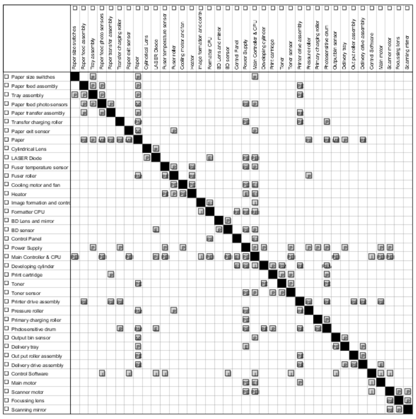


Fig. 5. Partitioning of composite DSM

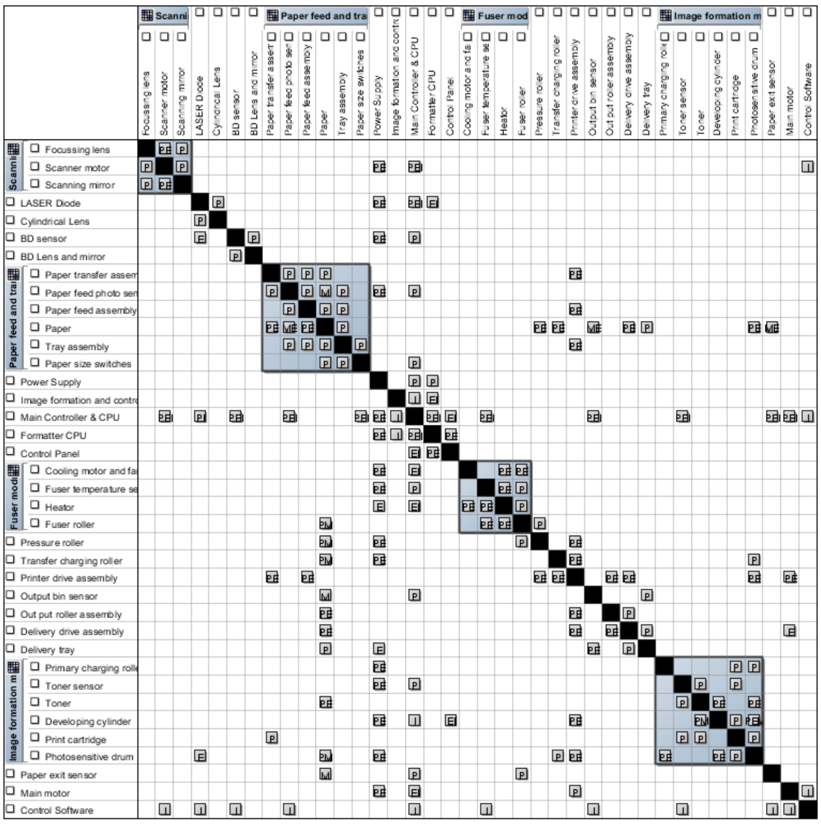


Fig. 6. Clustering of composite DSM

In general, partitioning can provide information about the existence of feedback loops and can determine the strongly connected parts implied in a structure. Groups of nodes can also be identified that are suitable for modular design. However, not able to provide information about feedback loops in specific nodes. In Fig.5, interfaces of the main controller, power supply and printer drive assembly are sparser than the other elements. The remaining elements are relatively close to the diagonal that can be considered as possible module candidates. The reason for not accumulating near the diagonal is due to many interfaces shared by some of the elements in a system.

The designer can draw useful insights from the DSM architecture after clustering the elements in composite DSM. In Fig 6, the DSM model identified four of the subsystems as somehow modular, as more interfaces among components in each subsystem. These modular subsystems are scanning system, paper feed system, paper delivery and fuser system. The DSM also show the remaining subsystems as more distributed. These systems are printer control system, main motor and printer drive assembly that are more functionally distributed across the printer system or in other words their structure is more integrative than modular one.

4.1 Structure representation and analysis

The network view, is used to visualize the dependency information in the detailed printer system, each line represents a connection in the DSM, from this diagram elements with more integrated connections can be visualized, though it is hard to read especially when there are many components and interfaces. In network view in Fig.7, power supply, main controller and printer drive assembly are more connected than other elements.

From the network view, each node represents the components and assemblies in the printer architecture, on average more than one interface is present with each component, it also points towards the functions performed by each component in the architecture is distributed. According to Ulrich's definition of modular architecture having one to one mapping between functions and components, modular structures must have a smaller function to component ratio than integral products. Though the function to component ratio is not calculated in this case, but the distribution of the interfaces is much higher, that must be reduced to increase the degree of modularity in the architecture.

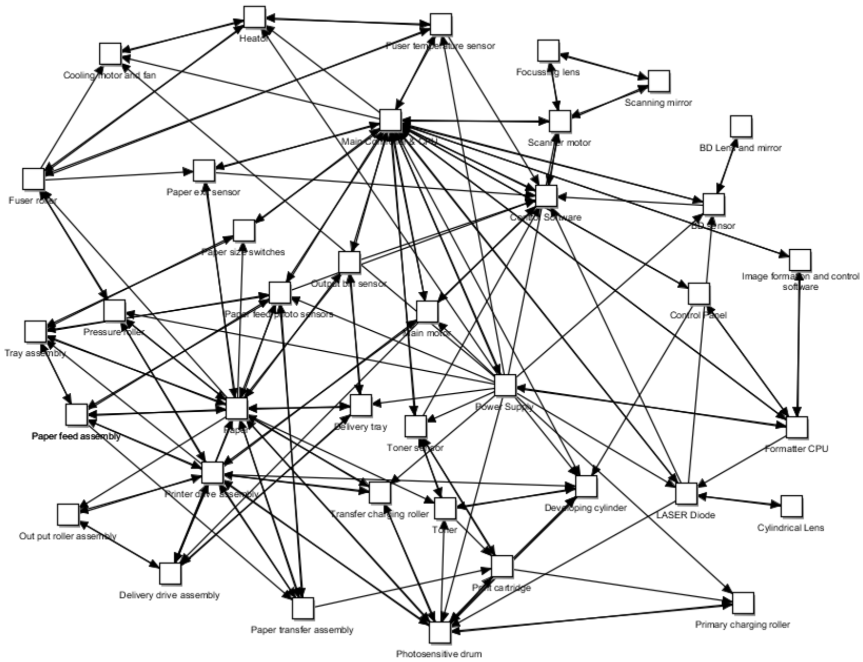


Fig. 7. Structure representation in network view.

The attainability in a structure is measured to assess the change in any node and its impact on other nodes in the structure. Attainability of a node is the ratio of the number of nodes it reaches to the maximum number of nodes it could reach theoretically. In the proposed example, the attainability in the nodes such as CPU control, printer drive assembly, power supply is relatively more than the remaining nodes of the system. A high valve of attainability means that many other nodes can be affected by any change in the node that is considered for any change. In general, the attainability in the nodes must be reduced in order to be able to make structure changes or redesign the architecture. Though, other criteria such as paths and lengths of the nodes are also relevant for proper characterization of the nodes.

Clustering is a measure, or a heuristic, to define the level of connectivity between a group of people or components. The Clustering Coefficient (CC) is the measure of this level of connectivity [9]. In the printer example, scanning mirror is having two edges and both are closely connected, hence its CC is equal to 1.While cylindrical lens is having one edge and no other connections therefore its CC is 0. In Fig.8, CC for most of the nodes is close to centre that point's medium level of connectivity in nodes.

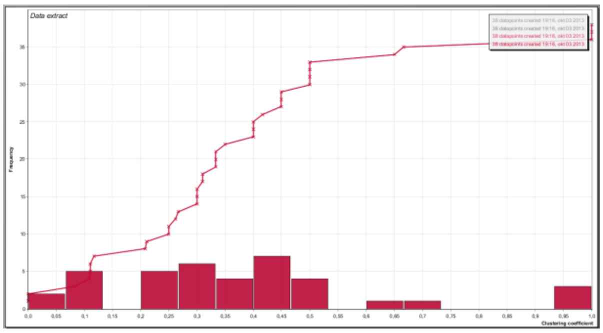


Fig. 8. The level of connectivity in the elements of composite DSM.

Table 1. Structure analysis results of composite DSM

Structural analysis results

Metric	Value
Number of sets	4.0
Number of unconnected nodes	0.0
Number of edges per node	4.526315789473684
Number of edges	172.0
Number of nodes (discounting duplicates)	38.0
Number of nodes	38.0
Number of edges that cross a set border	60.0
Relational density (Non-zero fraction)	0.12233285917496443
Singular-value modularity index (SMI)	0.25563157894736843

Table1 represents the total number of clusters formed, number of nodes, edges and their connections, non-zero fraction (NZF) as well as singular valve modularity index (SMI) in the structure. There are in total four clusters, all the nodes are connected with some kind of interface. The SMI and NZF are introduced by [10,15]. Where NZF, is the fraction of non zero entries without diagonal values, that can be computed as,

$$\text{Non zero fraction} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} DSM_{ij}}{N(N-1)} \qquad (1)$$

Where, N= number of components.

According to the NZF, the density of the system is 0.122 or 12%. In other words only 172 of the 1406 off-diagonal cells are occupied in the system. While complex, with as many as 38 elements and lot of interrelationships, the density is only 12%.

The *singular value modularity index* evaluates the overall connection scheme between the components; however this index does not evaluate that how components are grouped into modules. The SMI measures the decay rate of the singular values in the system [15],

$$\text{SMI} = \frac{1}{N} \arg\ min\, \alpha \sum_{i=1}^{N} \left| \frac{\sigma_i}{\sigma_1} - e^{-[i-1]/\alpha} \right| \qquad (2)$$

Where, σ is the singular values in the matrix, and N is the number of components and α determine the decay rate. This index is theoretically bounded and based on numbers between 0 and 1. According to [10], an SMI closer to 1.0 indicates a higher degree of modularity, where the connectivity information is more distributed. An SMI closer to 0 indicates a more integral system. The SMI, in printer case is closer to zero, that indicates an integral system, even though the algorithm formed some modules. One reason can be the number of edges formed by interfaces related to elements such as power supply, paper, main controller CPU, as these interfaces are more distributed in DSM.

5 Discussion and conclusions

This paper is about a case study analyzing an existing product by applying the component DSM method. The results are not generalizable and primarily dependent on product type. One reason that determines the degree of modularization in a product is dependent on the number of interactions, component connectivity and how spread these interactions are in the matrix.

In this work, decomposition of the mechatronic product is performed to analyze the architecture of the multi domain system. For this purpose, DSM approach is used by decomposing the system into subsystems and components, and in the process establishes the component interfaces. After applying the clustering algorithm, the DSM formed by only physical interfaces is different than the composite DSM. In case of physical interface DSM, the number of interfaces is significantly less and not spread like composite DSM, hence more modules are formed by clustering algorithm. In composite DSM, more interfaces are there, as elements related to information and control are more spread and linked to other elements in the structure as compare to those related to mechanical elements. The matrix formed by partitioning, highlights important aspect related to identification of modules. As partitioning, regroup most of the elements close to the diagonal that can be considered as possible module candidates. Some interfaces are not accumulating near the diagonal is due to many interfaces shared by some of the elements in a system.

Structure analysis of the system architecture, is an important aspect that reperesnt nodes, edges, interfaces and modules. A high degree of connectivity in a structure can make the system analysis difficult. The quantity of feedback loops may increase drastically as connectivity of elements in a structure becomes higher. This indicates integral products architectures and results in more connected product that require more efforts to redesign. From the SMI index and Ulrich definition of product architecture, the printer architecture is close to integrative, though some modules are formed after clustering. Structural optimization is useful when a fundamental system structure has to be redesigned in order to form product platforms to create variety. This supports developers in the creation of specific system variants for product customization [8]. This implies that through structure optimization using DSM methods, a product platform can be developed from a single product that can be used for customization in the product. Structure optimization involve application of various approaches such as tearing and structure Pareto analysis.

The development of product architecture based on design parameters and their interfaces is a useful approach for product upgrading and mass customization. For instance, for better performance increasing the copying speed of the printer can be achieved by changing design parameters such as speed of a motor or its size. Once the modular structure is in place along with relevant interfaces, the designer can decide either to replace the component (with a high speed motor) or using a controllable component (variable resistance in this case), that also involve change in the control software, in case of mechatronic products. This upgrading may not change to a large extent the physical configuration of the system. As shown in the printer example, though main motor is not placed in any module, any change may influence the interfaces with subsystems and components such as printer drive assembly, main CPU and control software. Furthermore, the design parameters can be changed to create variety in the product, such as changing speed and size of the motor. This must have an effect on the overall performance of the product.

One issue related to the design of complex systems is the trade-off between modularity and integrality. What should be the degree of modularity in case of computer controlled mechatronic products? According to Hollta and Whitney [10, 11], integral architecture is driven by product performance (i.e. power consumption, weight, size, speed etc) and cost while modular architecture by business demands such as variety, product change, engineering standards and service requirements. They argue that how total modularity is not always desirable in case of high power mechanical products as opposed to low power signal processor type products.That argument is supported by some high performance systems such as automotive and aerospace vehicles appear to favour highly coupled architectures, where one part fulfills potentially many functions [12]. It means, that products with technical performance constraints (e.g. light weight, tight packaging, power efficiency, speed) tend to have a larger functions-to-component ratio, i.e. they are more integral such as electronic calculator and mobile phones (excluding batteries and cover). But on the other hand products like computers are highly modular as compare to products that (contain computer control mechanical parts) such as printers, car engines even though they are microprocessor based products (or sub products).

The optimal solution in case of mechatronic systems could be a high performance product, with a few modules that can be used for commonality and flexible design for customization. However, that statement cannot be generalized due to various factors and requirements. Though mechatronics is a design process to develop high performance products by the functional and spatial integration of subsystems with various engineering disciplines, as more software and electronics is integrated to mechanical products for improved performance. Apart from these functional improvements, there must be some compromise on performance to satisfy market needs. Therefore, the *degree of modularity* in mechatronic products varies and cannot be generalized due to performance requirements, product structure and market demands etc.

References

[1] Ulrich K., The role of product architecture in the manufacturing firm, Research Policy 24 (1995) 419–441.

[2] Wyne, D.C.,Wyatt, D.F.,Nair,S.M.T. and Clarkson, P.J. "An introduction to the Cambridge Advanced Modeller". Proceedings of MMEP 2010, Cambridge, UK, 2010.

[3] Steward DV. The design structure system: a method for managing the design of complex systems. IEEE Trans Eng Manage 1981;28(3):71–4.

[4] Ulrich K.T., Eppinger S.D., Product Design and Development, McGraw-Hill, New York, 1995.

[5] Pimmler T.U., Eppinger S.D., Integration analysis of product decompositions, In: Proceedings of ASME Design Theory and Methodology Conference 68 (1994) 343–351.

[6] C.Y. Baldwin, K.B. Clark, K.B., Modularity in the design of complex engineering Systems, In: Understanding Complex Systems, Springer Berlin, Heidelberg, 2006, pp. 175–205.

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

[8] Lindemann Udo, Maurer Maik, Braun Thomas. Structural Complexity Management: An Approach for the Field of Product Design. Springer, 2009.

[9] Dahlgren W John. (2007).Developing a Framework for Exploring Clustering Coefficient to Evaluate System Coupling. 12th ICCRTS, "Adapting C2 to the 21st Century".

[10] Katja Hölttä, Eun Suk Suh, and Olivier de Weck. (2005). Tradeoff between modularity and performance for Engineered systems and products. International conference on engineering design. ICED 05 Melbourne, august 15 – 18, 2005.

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

[12] Liebeck R.H., "Design of the Blended Wing Body Subsonic Transport", Journal of Aircraft, Vol. 41, No.1, Jan-Feb 2004, pp. 10-25.

[13] Eppinger Steven D, Browning Tysen. Design structure matrix methods and applications. MIT, 2012.

[14] Hvam L., Mortensen N.H., Riis J., 2007, Product Customization, Springer Verlag, Lyngby, 2(17).

[15] Katja Hölttä-Otto and Olivier de Weck (2007). Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints. *Concurrent Engineering* 2007 15: 113.

Paper 5

Modelling and Investigation of Electromechanical Valve Train Actuator at Simulated Pressure Conditions.

Habib, Tufail. Proceedings of the Nord design 2012 conference (August 22nd-24th, 2012), Aalborg, Denmark. Pp.208-216.

NordDesign 2012
August 22 – 24, 2012
Aalborg, Denmark

Modeling and Investigation of Electromechanical Valve Train Actuator at simulated Pressure conditions.

Tufail Habib
Department of Mechanical and Manufacturing Engineering, Aalborg University
Fibigerstraede 16, DK- 9220 Aalborg
Denmark
habib@m-tech.aau.dk

Abstract

In an electromechanical valve actuated engine, the valves are driven by solenoid-type actuators and cam-shaft is eliminated. Control of each valve provides flexibility in valve timings over all engine conditions and achieves the benefits of variable valve timing(VVT).
This paper is about investigation of Electro-mechanical actuator at simulated pressure conditions for a single cylinder engine. For this purpose, a scaled down actuator with reduced armature lift and high stiffness springs are being used.
Experiments are conducted to measure valve release timings, transition times and contact velocities. Furthermore, discussion about the spring, magnetic, exhausts gas forces and their ability to actuate the system as desired.

Keywords: Electromechanical actuator (EMA), Variable valve timing, electromagnets.

1. Introduction

The fixed valve motion by camshaft engines compromises the fuel economy, combustion stability and maximum torque performance at different loads.The conventional camshaft is replaced by electromechanical actuator in order to improve the performance of a combustion engine with a flexible scheme in valve timing at all engine operating conditions.
Electromechanical actuators are increasingly becoming the actuator of choice in industry, due to their ruggedness, low cost, reduced complexity, relative high force density and ease of control .VVT reduces or eliminates many of the tradeoffs between low and high speed torque, fuel economy, idle quality, and emissions that are currently made with fixed valve timing [1][2].
Several variable valve actuation schemes have recently been studied and reported in the literature. The examples range from more flexible cam-based systems, such as variable or dual cam timing, to totally camless engines for which the valves are independently operated by means of specifically designed valve actuators. Electro-hydraulic actuated systems and electro-magnetic actuated systems are two most common examples of camless actuation technology for achieving variable valve timing [3].
After multi-valve technology became standard in engine design, Variable Valve timing becomes the next step to improve engine output. With electromechanical valve train (EMVT) systems valve timings are fully independent from crankshaft position and with flexible valve timings, cylinder air charge and residual gas can be optimized. By controlling the intake valve

events the throttled operation is eliminated in the gasoline engine and by doing so reduce the pumping loss which results high fuel efficiency [4][5].
Most electromagnetic systems in the literature use a spring system to accelerate and decelerate the valve. Solenoids or motors are used to hold the valves in the end positions and to compensate for friction losses, as well as combustion forces [6].
In EMVT actuator, speed and friction of moving parts results into high energy consumption. Scaled down actuator at higher speed need stronger springs, that require higher currents but the issue of magnetic saturation arises. Similarly at higher pressures especially at valve opening more catching current is required to open the valve against the air pressure.
Prototype electromechanically actuated VVT systems have been proposed by several companies in the automotive industry, the first being proposed by FEV Motorentechnik [7] [8]. Other companies that have worked on this technology include BMW, GM, Renault and Siemens.
This paper is about a scaled down electromechanical actuator designed for motorcycle applications. The actuator is investigated at a reduce armature lift and at higher speed for single cylinder engine with experimental results. Apart from design changes the effects of spring rate, armature lift and exhaust gas forces on valve are discussed.

2. Actuator Model

Main part of the system is an electromechanical actuator, which operates as a free oscillation system with electromagnets holding the valves in both final positions. The actuator consists of lower magnetic coil for opening the valve and an upper magnet for closing the valve. Actuator and valve spring push on armature and valve stem through spring retainers. At mid position the armature is centered between lower and upper magnets.

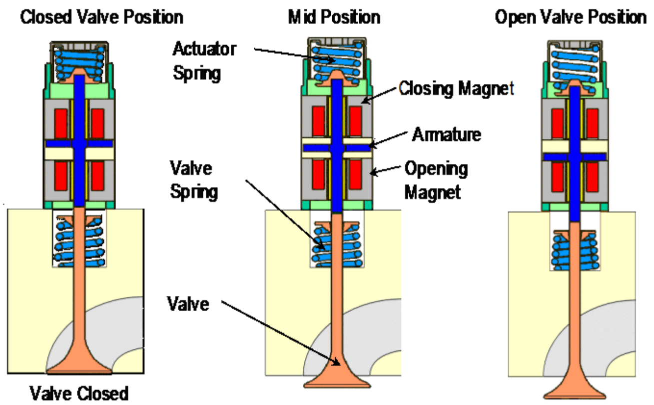


Fig1. Valve position by the action of magnetic and spring forces [9] .

At start, voltage is applied to one of the electromagnets to move the armature from Actuator middle position to the fully open position. A holding current is then maintained to holds the armature in place against the spring force. The mechanical spring force and magnetic force determine the actuator and valve operation.
At valve closing the armature moves to the upper magnet and a holding current is applied to hold the armature at closing magnet against the actuator spring force.
Actuator system comprising electrical, mechanical and magnetic system, Electrical energy is transferred to excite the electromagnets, due to which useful mechanical work is obtained as shown in Fig 2,
Subsystem of EMVT:

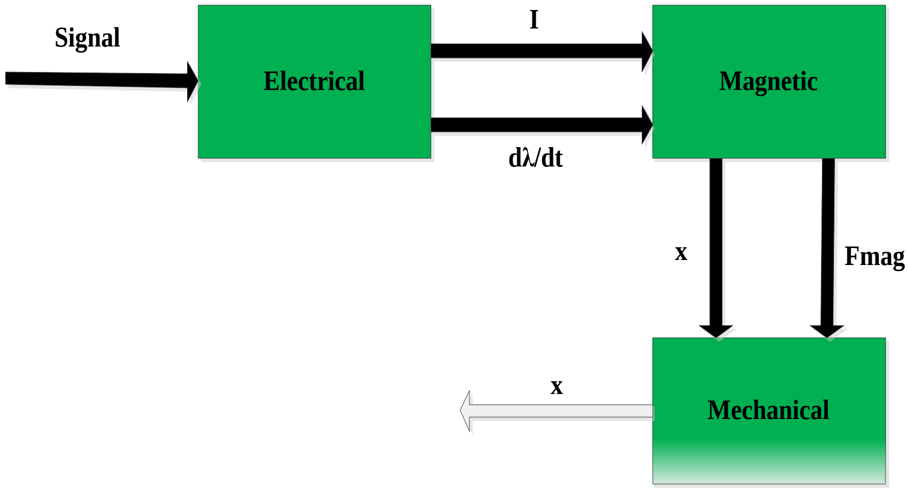


Fig 2. Model of actuator subsystem.

Where, I= current, x = armature position, dλ/dt = change in flux and F_{mag} = magnetic force,

Magnetic energy of the core is

$$W_{mag} = \iiint HdB.V = 4[a.b.d]\frac{B^2}{2\mu_0}. \quad (1)$$

The volume of the magnetic core is shown in Fig 3, while the electrical and magnetic power in the core is

$$P_{mag} = 4ab\frac{B^2}{2\mu_0}d' + 4abd\frac{2BB'}{2\mu_0}, \quad (2)$$

$$P_{el} = V.I = \frac{U - 2n\Phi'}{R} = \frac{U}{R} - \frac{2n}{R}B'a.b, \quad (3)$$

Mechanical power in the system is

$$P_{mech} = F.d' \quad (4)$$

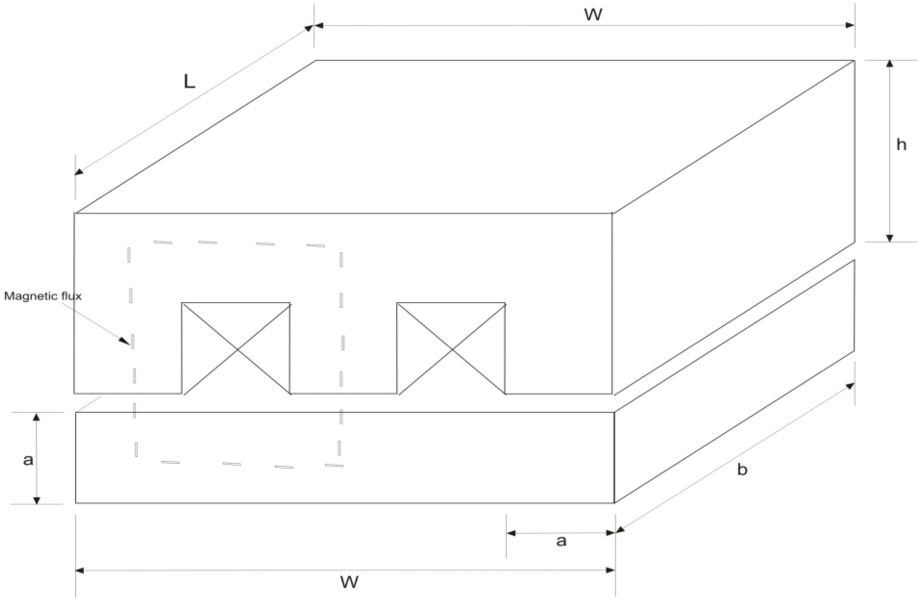


Fig 3. Magnetic core with moving part.

$$P_{el} = V.I - I^2 R = (2n\Phi' + IR).I = 2n\Phi'.I \qquad (5)$$

$$= 2n\Phi' \frac{\Phi}{n} \frac{2d}{\mu_0 A} = 2AB' \frac{2d}{\mu_0} B = 4ABB' \frac{d}{\mu_0}. \qquad (6)$$

The sum of the electrical and magnetic energy will result in mechanical force,

$$P_{mag} + P_{el} = -Fd' \qquad (7)$$

$$P_{mag} = P_{el} + Fd' \qquad (8)$$

$$4ab\frac{B^2}{2\mu_0}d' + 4abd\frac{2BB'}{2\mu_0} = Fd' + 4ABB'\frac{d}{\mu_0}, \text{ a.b = A,} \qquad (9)$$

that will become

$$F_{mag} = 4A\left(\frac{B^2}{2\,\mu_0}\right). \qquad (10)$$

Cylinder pressure

The variation of pressure difference between the cylinder side and on the port side of the valve produces a resultant force on the valve which is transmitted to the actuator. The computational fluid dynamic (CFD) program calculates the pressure difference and the forces at every moment for a circular valve.

$$F_{gas} = \left[\Delta P\left(\frac{\pi dv^2}{4}\right)\right] \qquad (11)$$

Spring force

As by linear law that is representative for the electromagnetic valve train actuator (EMVT) system, the spring force is given by,

$F= c.x$ (12)

3. Methods

A scaled down actuator is investigated for the effect of changing armature lift, spring rates and exhaust force on valve at varying diameters.

Table 1. Boundary conditions of the actuator.

	Standard actuator	**Scaled down actuator**
Oscillation time intake/exhaust	6.12 ms	4.8 ms
Speed	6000 rpm	8000 rpm
Transition time	2.9 ms	2.3 ms
Cylinder pressure at exhaust valve opening	0-7 bar	0-7 bar
Cylindrical spring with average spring constant	2x75 N/mm	2x120 N/mm
Valve lift	8 mm	6.5 mm
Operating voltage	42-55 V	42-55 V

Experimental results at valve opening and closing are carried through a test rig in the same way as real engine with the following parameters,
The test rig situation is:

- Combustion chamber volume: to be calculated
- One exhaust valve on cylinder
- Gas temperature at about ambient (293 C°)

Finding the compression volume:

T_{engine}= 1223 K, $T_{comression}$ = 293 K

With $a = \sqrt{KRT}$ results $\frac{a_{engine}}{a_{compression}} = \sqrt{\frac{T_{engine}}{T_{compression}}} = \sqrt{\frac{1223}{293}} = 2.043$

$$\frac{V_{engine}}{V_{compression}} = \frac{2.A_{valve}a_{engine}}{A_{valve}a_{compression}} = 4.087$$

with a cylinder of $V_{cylinder}$ = 200 cm^3 results $Vc = 200/4.807 = 50\ cm^3$
For this actuator the optimum pressure chamber volume is calculated as 50.25 cm^3.

4. Experimental Results

The results obtained through oscilloscope and the data processed in Matlab [10] for the valve opening phase at 1bar absolute pressure (in the pressure chamber), the armature lift time (from valve close to fully open) is 2.8 ms. As the back pressure increases the lift time also increases due to the fact that valve is pushed against more pressure.

The transition time for the standard actuator with a spring rate of 150 N/mm and moving mass of 142 g is 2.9 ms, while the transition time for a scaled down actuator (used in project) with a spring rate of 240 N/mm and moving mass of 141 g is 2.3mm.
Since a stronger spring rates (240 N/mm, light springs) is used in this project the oscillation time is considerably reduced and a high speed for the actuator is reached.

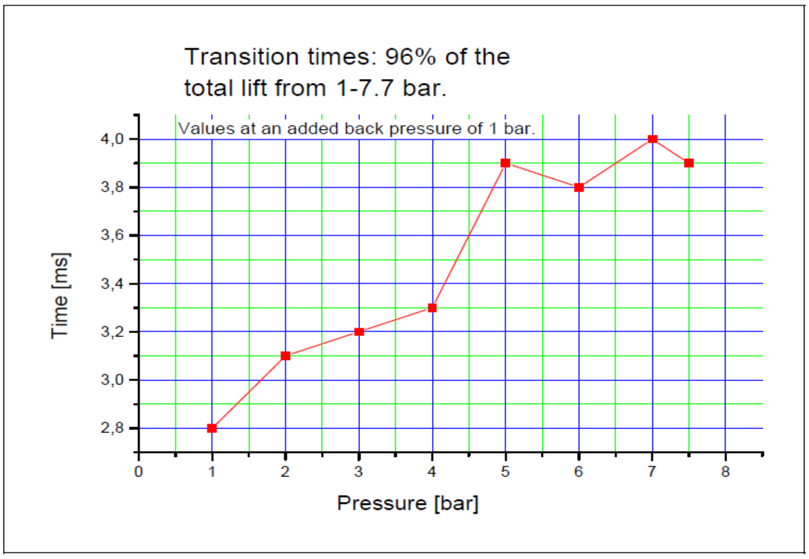


Fig 4. Transition times (2% up to 98% valve lift) from 1.0 bar to 7.5 bar absolute pressure inside the pressure chamber.

Armature lift curve as it moves from upper magnet to lower magnet or the valve opening event. Figure 5 demonstrates that, the instant the armature starts to lift; the holding current comes to zero and the catching current starts to rise till the armature reaches its maximum lift. Catching current is more than holding current in order to overcome the friction losses.

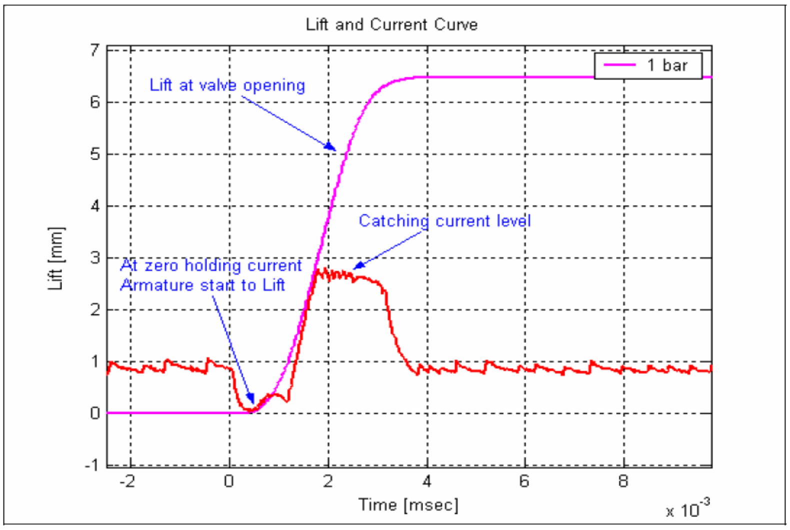


Fig 5. Starting event of the lift curve.

More catching current is needed as the pressure on the valve increases as shown in figure 6. At the start of the valve lift, holding current is almost same for all pressures because the upper magnet is at holding phase always working against the upper spring force and not against the pressure, the upper magnet holding current is independent to the pressure, more catching current is needed as the force on the valve increases while opening.

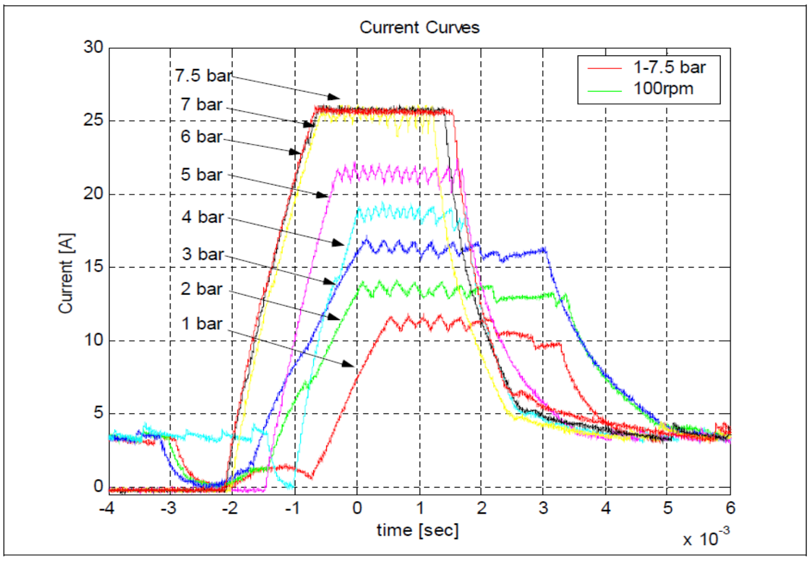


Fig 6. Current curves at a pressure from 1.0 up to 7.5 bar.

Maximum velocity of the armature reduces as pressure increases. At 1 bar pressure the maximum velocity is 3900 mm/sec while at 7.5 bar pressure the maximum velocity reduces to nearly 2000 mm/sec as shown in figure 7. A speed reduction of approximately 48% is observed. At the valve opening ,exhaust gases pressure present in combustion chamber exerts force on the valve, the amount of this force increases as the exhaust gas pressure increases eventually resulting in a reduction of armature velocity.

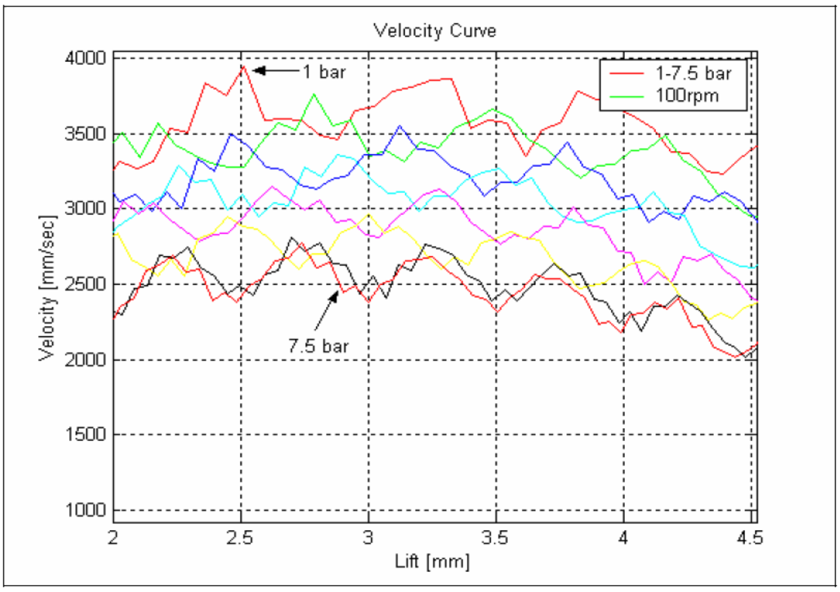


Fig 7. Velocity trace at a pressure from 1.0 up to 7.5 bar.

The kinetic energy (1/2 mv^2) of the system is more at the centre, due to it the system will move faster at centre as compare to the ends, the reduction in velocity would be more at the centre as compare to the ends as pressure increases, as shown in figure 8.

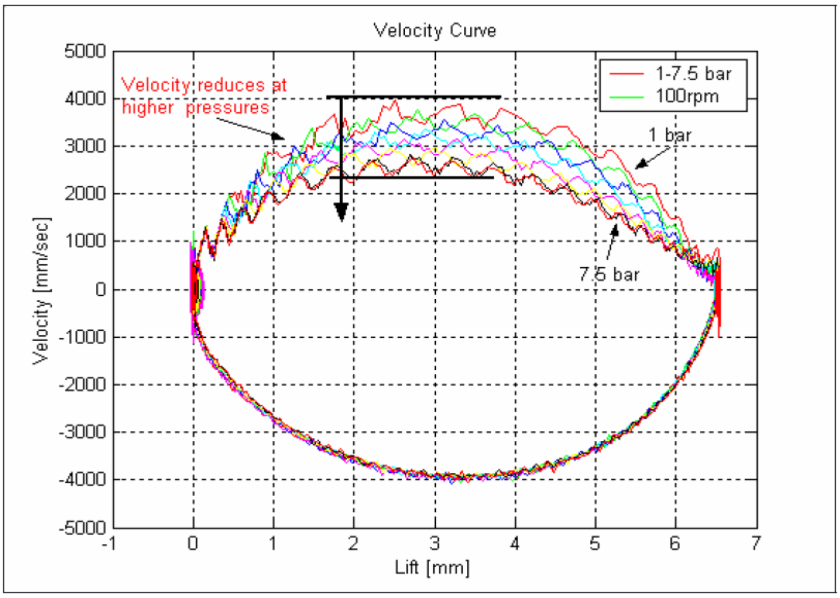


Fig 8. Velocity trace art a pressure from 1.0 to 7.5 bar.

Swings out curves are shown from 1 to 7.5 bar pressure in figure 9. Higher pressures forces on the valve causes the armature to settle down (at the centre) quickly as compare to low ones, the settling time at 7.5 bar is 0.071ms while at 1 bar the settling time is 0.15 ms. Armature lift reduces at higher pressures.

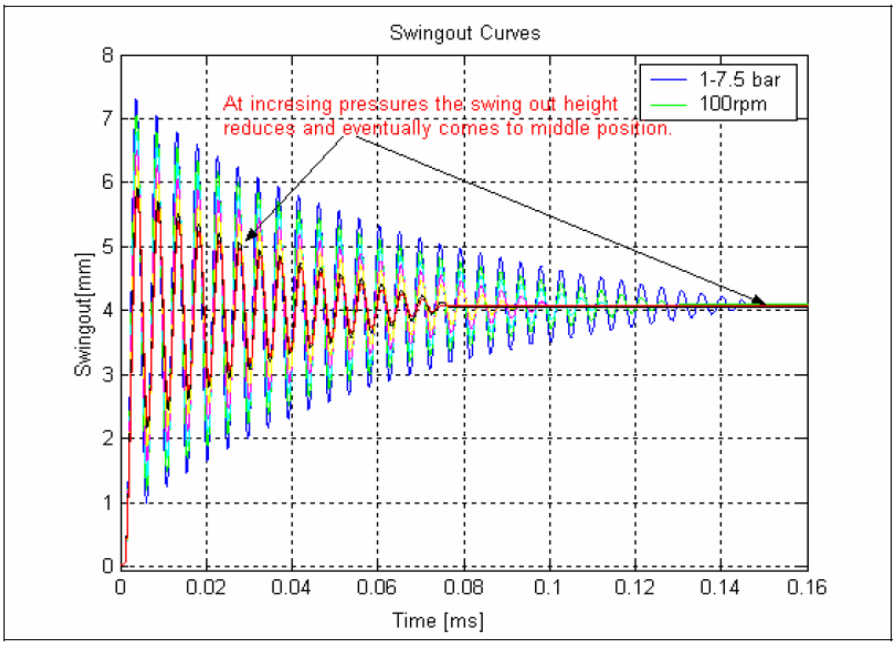


Fig 9. Swing out curves from 1to7.5 bar pressure.

5. Discussion

Higher end stop forces are needed when higher spring rates are used; it means more magnetic forces will be needed against this spring rate at end positions due to which more current is needed, resulting higher energy consumption. Magnetic force can be maximized by increasing the current at higher spring rates, but there is a limitation, a saturation point will reach beyond which the magnetic induction will not increase appreciably by giving more current. Furthermore, speed and friction of moving parts will increase results into high energy consumption.

By increasing the armature lift requires more catching current (to overcome the friction losses) that is also a loss of energy. Another issue with increasing armature lift is the limitation of magnetic induction. The greater the amount of current applied, the stronger the magnetic field in the component. But a point is reached that an additional increase in the current will produce very little increase in the magnetic flux; the material has reached a point of saturation.

As a result of induction eddy currents are built up in the core called eddies, they tend to flow in closed paths within the magnetic material and depend on the frequency, amplitude of the current and the permeability of the core material. It also generates as the flux varies due to the change in the air gap. This leads to heat losses and to a delay of the built up and decrease of magnetic field [11]. The copper losses depend on the resistance of the coil and increase with the square of the current.

The reduction of these losses is carried out through the suitable material selection and an assembly of thin insulated sheet metal which must be oriented in a direction parallel to the flow of magnetic flux.

The forces in the end positions depend on the neutral position, which is the place where the equilibrium of spring forces occurs. The existence of the valve lash produces an increase of the stored potential energy in the close position in comparison with that in open position. When the neutral position is the geometrical centre. The existence of the valve lash causes a large holding force in closed position due to the fact that only the actuator spring force is acting on the armature. The aerodynamic force, work against the armature motion due to the air resistance, its value is negligible as compared to the main forces. The gravity force can be neglected as well for the same reason.

6. Conclusions

A fully variable valve train actuator is designed for motorcycle applications. Actuator is investigated at a reduce armature lift and at higher speed for single cylinder engine. Apart from design changes the effects of spring rate, armature lift and exhaust gas forces on valve are discussed.

Experimental results at valve opening and closing are carried through a test rig in the same way as real engine with a reduced chamber volume of 50 cm^3, which is able to operate up to 8000 rpm engine speed, and is investigated on a test rig having a lift of 6.5 mm. The experimental results include swing out curves, velocity trace at valve opening and closing, lift curves, transition times and current trace at a pressure from 1 to7 bar.

The transition times for opening event increases with higher pressures. The time for closing valve event is smaller than opening valve event and stays constant due to the fact that at valve closing the pressure has already disappeared. There is an appreciable loss of energy due to friction and eddy current. In the swing out curve the armature comes to its mean position as the power is switched off.

The magnetic force is sufficient to hold the armature at end positions against the spring forces, and also able to open the valve against the gas force.

References

[1] Puchalsky, C., Megli, T., Tiller, M., Trask, N., Wang, Y., “Modelica applications for Camless Engine Valve train Development “, 2nd international Modelica Conference, Germany. March 18-19, 2002.

[2] Eyabi, P., Gregory, W., “Modeling and sensorless control of an electromagnetic Valve actuator “, Mechatronics 16 (2006) 159–17.

[3] I. Haskara, V. V. Kokotovic and L. A. Mianzo, “Control of an electro-mechanical valve actuator for a camless engine”, International journal of robust and nonlinear Control; 14: 561–579, (DOI: 10.1002/rnc.903), 2004.

[4] Seethaler, R., and Duerr, K., “Electromagnetic Valve Actuation with Two Configurations,” Proceedings of ASME IDETC/CIE, Vol. **7**, pp. 391–398, 2009.

[5] Rokni, H., Seethaler, R. J., and Milani, A., “Improved Mechanical Design for a Fully Variable Electromechanical Valve Actuation System for Internal Combustion Engines,” Proceedings of ASME, IMECE, Paper No. IMECE2010-38711, 2010.

[6] Zhao, J., and Seethaler, R., “A Fully Flexible Valve Actuation System for Internal Combustion Engines,” IEEE/ASME Transaction on Mechatronics, 16(2), pp. 361–370, 2011.

[7] F. Pischinger *et al.*, “Electromechanical Variable Valve Timing,” *Automotive Engineering International*, 1999.

[8] F. Pischinger *et al.*, “Arrangement for Electromagnetically Operated Actuators,” *US Patent #4,515,343*, 1985.

[9] www.FEV.com

[10] Stephan J. Chapman, ’Matlab Programming for Engineers’ (Thomson Asia Ltd 2002) 2nd Ed, pp.111-112, 2002.

[11] Busch Vishnian and Ilene.J, “Electromechanical sensors and actuators” (Springer verlag New York 1998) pp.134-135, 1998.